

Data Science interview questions

Statistics

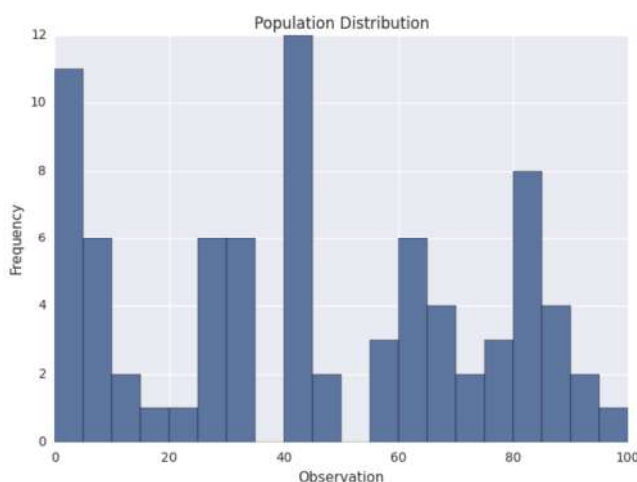
Q1. What is the Central Limit Theorem and why is it important?

<https://spin.atomicobject.com/2015/02/12/central-limit-theorem-intro/>

Suppose that we are interested in estimating the average height among all people. Collecting data for every person in the world is impractical, bordering on impossible. While we can't obtain a height measurement from everyone in the population, we can still sample some people. The question now becomes, what can we say about the average height of the entire population given a single sample.

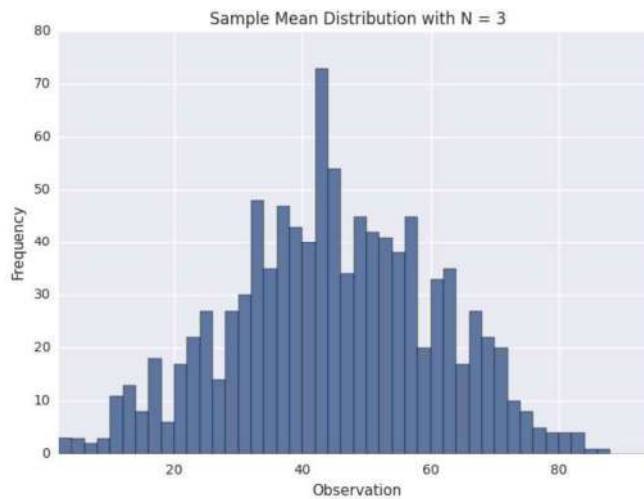
The Central Limit Theorem addresses this question exactly. Formally, it states that if we sample from a population using a sufficiently large sample size, the mean of the samples (also known as the sample population) will be normally distributed (assuming true random sampling), the mean tending to the mean of the population and variance equal to the variance of the population divided by the size of the sampling. What's especially important is that this will be true regardless of the distribution of the original population.

EX:

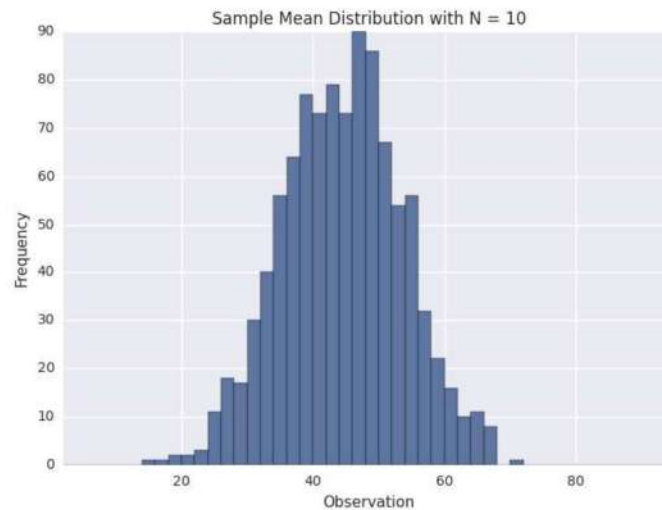


As we can see, the distribution is pretty ugly. It certainly isn't normal, uniform, or any other commonly known distribution. In order to sample from the above distribution, we need to define a sample size, referred to as N . This is the number of observations that we will sample at a time. Suppose that we choose N to be 3. This means that we will sample in groups of 3. So for the above population, we might sample groups such as [5, 20, 41], [60, 17, 82], [8, 13, 61], and so on.

Suppose that we gather 1,000 samples of 3 from the above population. For each sample, we can compute its average. If we do that, we will have 1,000 averages. This set of 1,000 averages is called a sampling distribution, and according to Central Limit Theorem, the sampling distribution will approach a normal distribution as the sample size N used to produce it increases. Here is what our sample distribution looks like for $N = 3$.



As we can see, it certainly looks uni-modal, though not necessarily normal. If we repeat the same process with a larger sample size, we should see the sampling distribution start to become more normal. Let's repeat the same process again with $N = 10$. Here is the sampling distribution for that sample size.



Q2. What is sampling? How many sampling methods do you know?

<https://searchbusinessanalytics.techtarget.com/definition/data-sampling>

<https://nikolanews.com/difference-between-stratified-sampling-cluster-sampling-and-quota-sampling/>

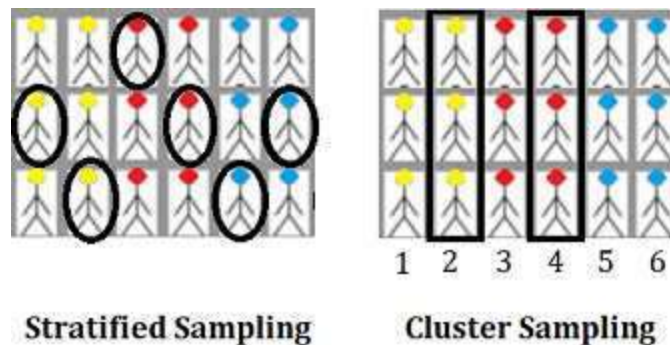
Data sampling is a statistical analysis technique used to select, manipulate and analyze a representative subset of data points to identify patterns and trends in the larger data set being examined. It enables data scientists, predictive modelers and other data analysts to work with a small, manageable amount of data about a statistical population to build and run analytical models more quickly, while still producing accurate findings.

Sampling can be particularly useful with data sets that are too large to efficiently analyze in full – for example, in big data analytics applications or surveys. Identifying and analyzing a representative sample is more efficient and cost-effective than surveying the entirety of the data or population.

An important consideration, though, is the size of the required data sample and the possibility of introducing a sampling error. In some cases, a small sample can reveal the most important information about a data set. In others, using a larger sample can increase the likelihood of accurately representing the data as a whole, even though the increased size of the sample may impede ease of manipulation and interpretation.

There are many different methods for drawing samples from data; the ideal one depends on the data set and situation. Sampling can be based on probability, an approach that uses random numbers that correspond to points in the data set to ensure that there is no correlation between points chosen for the sample. Further variations in probability sampling include:

- Simple random sampling: Software is used to randomly select subjects from the whole population.
- Stratified sampling: Subsets of the data sets or population are created based on a common factor, and samples are randomly collected from each subgroup. A sample is drawn from each strata (using a random sampling method like simple random sampling or systematic sampling).
 - EX: In the image below, let's say you need a sample size of 6. Two members from each group (yellow, red, and blue) are selected randomly. Make sure to sample proportionally: In this simple example, 1/3 of each group (2/6 yellow, 2/6 red and 2/6 blue) has been sampled. If you have one group that's a different size, make sure to adjust your proportions. For example, if you had 9 yellow, 3 red and 3 blue, a 5-item sample would consist of 3/9 yellow (i.e. one third), 1/3 red and 1/3 blue.
- Cluster sampling: The larger data set is divided into subsets (clusters) based on a defined factor, then a random sampling of clusters is analyzed. The sampling unit is the whole cluster; Instead of sampling individuals from within each group, a researcher will study whole clusters.
 - EX: In the image below, the strata are natural groupings by head color (yellow, red, blue). A sample size of 6 is needed, so two of the complete strata are selected randomly (in this example, groups 2 and 4 are chosen).

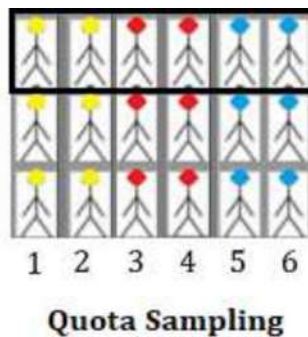


- Multistage sampling: A more complicated form of cluster sampling, this method also involves dividing the larger population into a number of clusters. Second-stage clusters are then broken out based on a secondary factor, and those clusters are then sampled and analyzed. This staging could continue as multiple subsets are identified, clustered and analyzed.
- Systematic sampling: A sample is created by setting an interval at which to extract data from the larger population – for example, selecting every 10th row in a spreadsheet of 200 items to create a sample size of 20 rows to analyze.

Sampling can also be based on non-probability, an approach in which a data sample is determined and extracted based on the judgment of the analyst. As inclusion is determined by the analyst, it can be more difficult to extrapolate whether the sample accurately represents the larger population than when probability sampling is used.

Non-probability data sampling methods include:

- Convenience sampling: Data is collected from an easily accessible and available group.
- Consecutive sampling: Data is collected from every subject that meets the criteria until the predetermined sample size is met.
- Purposive or judgmental sampling: The researcher selects the data to sample based on predefined criteria.
- Quota sampling: The researcher ensures equal representation within the sample for all subgroups in the data set or population (random sampling is not used).



Once generated, a sample can be used for predictive analytics. For example, a retail business might use data sampling to uncover patterns about customer behavior and predictive modeling to create more effective sales strategies.

Q3. What is the difference between type I vs type II error?

<https://www.datasciencecentral.com/profiles/blogs/understanding-type-i-and-type-ii-errors>

Is H_a true? No, H_0 is True (H_a is Negative: TN); Yes, H_0 is False (H_a is Positive: TP).

A type I error occurs when the null hypothesis is true but is rejected. A type II error occurs when the null hypothesis is false but erroneously fails to be rejected.

	No reject H_0	Reject H_0
H_0 is True	TN	FP (I error)
H_0 is False	FN (II error)	TP

Q4. What is linear regression? What do the terms p-value, coefficient, and r-squared value mean? What is the significance of each of these components?

<https://www.springboard.com/blog/linear-regression-in-python-a-tutorial/>
<https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-interpret-regression-analysis-results-p-values-and-coefficients>

Imagine you want to predict the price of a house. That will depend on some factors, called independent variables, such as location, size, year of construction... if we assume there is a linear relationship between these variables and the price (our dependent variable), then our price is predicted by the following function:

$$Y = a + b X$$

The p-value in the table is the minimum α (the significance level) at which the coefficient is relevant. The lower the p-value, the more important is the variable in predicting the price. Usually we set a 5% level, so that we have a 95% confidentiality that our variable is relevant.

The p-value is used as an alternative to rejection points to provide the smallest level of significance at which the null hypothesis would be rejected. A smaller p-value means that there is stronger evidence in favor of the alternative hypothesis.

The coefficient value signifies how much the mean of the dependent variable changes given a one-unit shift in the independent variable while holding other variables in the model constant. This property of holding the other variables constant is crucial because it allows you to assess the effect of each variable in isolation from the others.

R squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

Q5. What are the assumptions required for linear regression?

There are four major assumptions:

- There is a **linear relationship between the dependent variables and the regressors**, meaning the model you are creating actually fits the data,
- The errors or **residuals ($y_i - \hat{y}_i$) of the data are normally distributed and independent from each other**,
- There is **minimal multicollinearity between explanatory variables**, and
- **Homoscedasticity**. This means the variance around the regression line is the same for all values of the predictor variable.

Q6. What is a statistical interaction?

<http://icbseverywhere.com/blog/mini-lessons-tutorials-and-support-pages/statistical-interactions/>

Basically, an interaction is when the effect of one factor (input variable) on the dependent variable (output variable) differs among levels of another factor. When two or more independent variables are involved in a research design, there is more to consider than simply the "main effect" of each of the independent variables (also termed "factors"). That is, the effect of one independent variable on the dependent variable of interest may not be the same at all levels of the other independent variable. Another way to put this is that **the effect of one independent variable may depend on the level of the other independent variable**. In order to find an interaction, you must have a factorial design, in which the two (or more)

independent variables are "crossed" with one another so that there are observations at every combination of levels of the two independent variables. *EX: stress level and practice to memorize words: together they may have a lower performance.*

Q7. What is selection bias?

<https://www.elderresearch.com/blog/selection-bias-in-analytics>

Selection (or 'sampling') bias occurs when the sample data that is gathered and prepared for modeling has characteristics that are not representative of the true, future population of cases the model will see. That is, active selection bias occurs when a subset of the data is systematically (i.e., non-randomly) excluded from analysis.

Q8. What is an example of a data set with a non-Gaussian distribution?

<https://www.quora.com/Most-machine-learning-datasets-are-in-Gaussian-distribution-Where-can-we-find-the-dataset-which-follows-Bernoulli-Poisson-gamma-beta-etc-distribution>

The Gaussian distribution is part of the Exponential family of distributions, but there are a lot more of them, with the same sort of ease of use, in many cases, and if the person doing the machine learning has a solid grounding in statistics, they can be utilized where appropriate.

Binomial: multiple toss of a coin $\text{Bin}(n,p)$: the binomial distribution consists of the probabilities of each of the possible numbers of successes on n trials for independent events that each have a probability of p of occurring.

Bernoulli: $\text{Bin}(1,p) = \text{Be}(p)$

Poisson: $\text{Pois}(\lambda)$

Data Science

Q1. What is Data Science? List the differences between supervised and unsupervised learning.

Data Science is a blend of various tools, algorithms, and machine learning principles with the goal to discover hidden patterns from the raw data. How is this different from what statisticians have been doing for years? The answer lies in the difference between explaining and predicting: **statisticians work a posteriori, explaining the results and designing a plan; data scientists use historical data to make predictions.**

The differences between supervised and unsupervised learning are:

Supervised	Unsupervised
Input data is labelled	Input data is unlabeled
Split in training/validation/test	No split
Used for prediction	Used for analysis
Classification and Regression	Clustering, dimension reduction, and density estimation

Q2. What is Selection Bias?

Selection bias is a kind of error that occurs when the researcher decides what has to be studied. It is associated with research where the selection of participants is not random. Therefore, some conclusions of the study may not be accurate.

The types of selection bias include:

- **Sampling bias:** It is a systematic error due to a non-random sample of a population causing some members of the population to be less likely to be included than others resulting in a biased sample.
- **Time interval:** A trial may be terminated early at an extreme value (often for ethical reasons), but the extreme value is likely to be reached by the variable with the largest variance, even if all variables have a similar mean.
- **Data:** When specific subsets of data are chosen to support a conclusion or rejection of bad data on arbitrary grounds, instead of according to previously stated or generally agreed criteria.
- **Attrition:** Attrition bias is a kind of selection bias caused by attrition (loss of participants) discounting trial subjects/tests that did not run to completion.

Q3. What is bias-variance trade-off?

Bias: Bias is an error introduced in the model due to the oversimplification of the algorithm used (does not fit the data properly). It can lead to under-fitting.

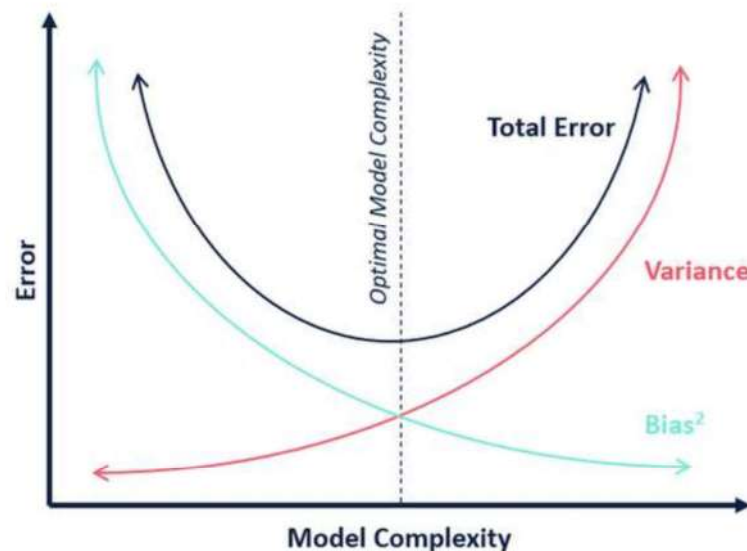
Low bias machine learning algorithms — Decision Trees, k-NN and SVM

High bias machine learning algorithms — Linear Regression, Logistic Regression

Variance: Variance is error introduced in the model due to a too complex algorithm, it performs very well in the training set but poorly in the test set. It can lead to high sensitivity and overfitting.

Possible high variance – polynomial regression

Normally, as you increase the complexity of your model, you will see a reduction in error due to lower bias in the model. However, this only happens until a particular point. As you continue to make your model more complex, you end up over-fitting your model and hence your model will start suffering from high variance.



Bias-Variance trade-off: The goal of any supervised machine learning algorithm is to have low bias and low variance to achieve good prediction performance.

1. The k-nearest neighbor algorithm has low bias and high variance, but the trade-off can be changed by increasing the value of k which increases the number of neighbors that contribute to the prediction and in turn increases the bias of the model.
2. The support vector machine algorithm has low bias and high variance, but the trade-off can be changed by increasing the C parameter that influences the number of violations of the margin allowed in the training data which increases the bias but decreases the variance.
3. The decision tree has low bias and high variance, you can decrease the depth of the tree or use fewer attributes.
4. The linear regression has low variance and high bias, you can increase the number of features or use another regression that better fits the data.

There is no escaping the relationship between bias and variance in machine learning. Increasing the bias will decrease the variance. Increasing the variance will decrease bias.

Q4. What is a confusion matrix?

The confusion matrix is a 2X2 table that contains 4 outputs provided by the binary classifier.

	Predict +	Predict -
Actual +	TP	FN (II error)
Actual -	FP (I error)	TN

A data set used for performance evaluation is called a test data set. It should contain the correct labels and predicted labels. The predicted labels will exactly the same if the performance of a binary classifier is perfect. The predicted labels usually match with part of the observed labels in real-world scenarios.

A binary classifier predicts all data instances of a test data set as either positive or negative. This produces four outcomes: TP, FP, TN, FN. Basic measures derived from the confusion matrix:

$$1. \text{ Error Rate} = \frac{FP+FN}{P+N}$$

$$2. \text{ Accuracy} = \frac{TP+T}{P+N}$$

$$3. \text{ Sensitivity (Recall or True positive rate)} = \frac{TP}{TP+FN} = \frac{TP}{P}$$

$$4. \text{ Specificity (True negative rate)} = \frac{TN}{TN+FP} = \frac{TN}{N}$$

$$5. \text{ Precision (Positive predicted value)} = \frac{TP}{TP+FP}$$

$$6. \text{ F - Score (Harmonic mean of precision and recall)} = \frac{2TP}{(2TP + FP + FN)}$$

Q5. What is the difference between “long” and “wide” format data?

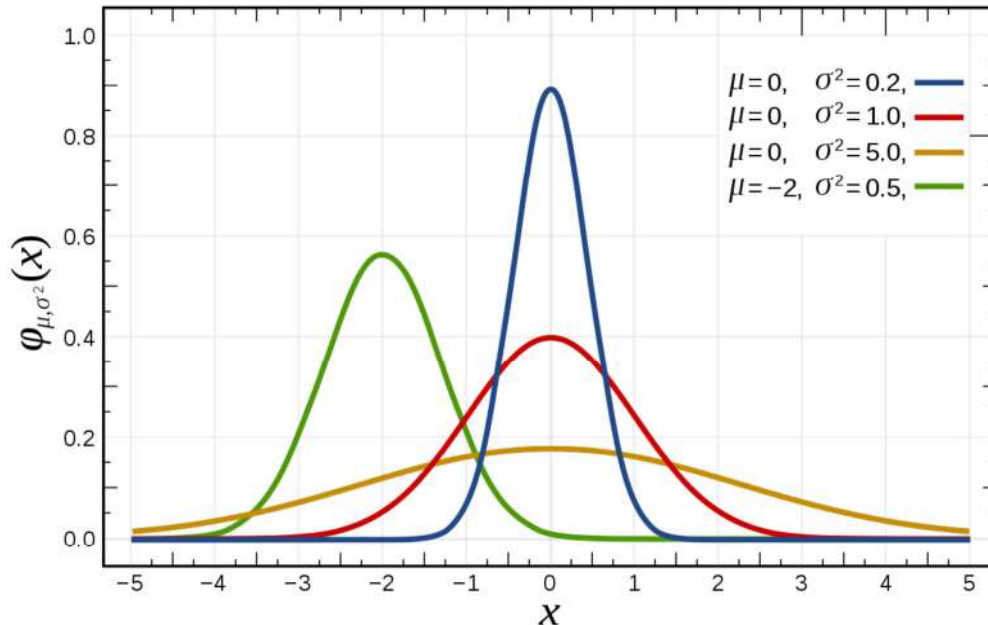
In the wide-format, a subject’s repeated responses will be in a single row, and each response is in a separate column. In the long-format, each row is a one-time point per subject. You can recognize data in wide format by the fact that columns generally represent groups (variables).

X	Y1	Y2	Y3
10	2	3	4
15	0	4	6
20	1	4	5

VarName	X	Value
Y1	10	2
Y2	10	3
Y3	10	4
Y1	15	0
Y2	15	4
Y3	15	6
Y1	20	1
Y2	20	4
Y3	20	5

Q6. What do you understand by the term Normal Distribution?

Data is usually distributed in different ways with a bias to the left or to the right or it can all be jumbled up. However, there are chances that data is distributed around a central value without any bias to the left or right and reaches normal distribution in the form of a bell-shaped curve.



The random variables are distributed in the form of a symmetrical, bell-shaped curve. Properties of Normal Distribution are as follows:

1. Unimodal (Only one mode)
2. Symmetrical (left and right halves are mirror images)
3. Bell-shaped (maximum height (mode) at the mean)
4. Mean, Mode, and Median are all located in the center
5. Asymptotic

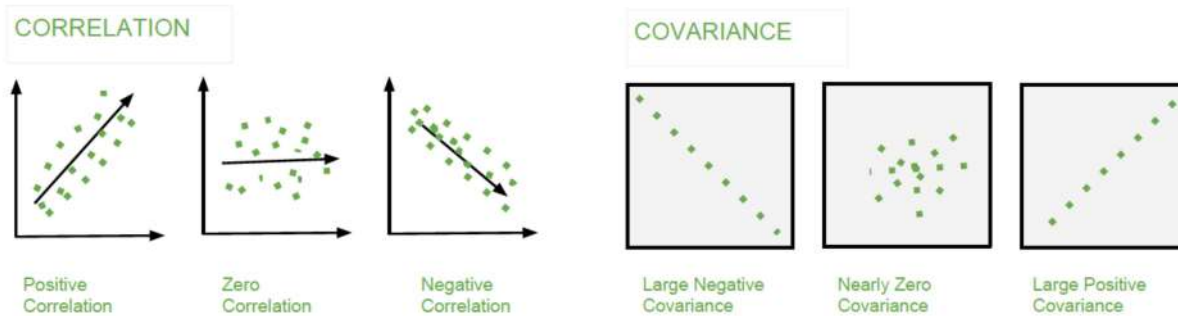
Q7. What is correlation and covariance in statistics?

Correlation is considered or described as the best technique for measuring and also for estimating the quantitative relationship between two variables. Correlation measures how strongly two variables are related. Given two random variables, it is the covariance between both divided by the product of the two standard deviations of the single variables, hence always between -1 and 1.

$$\rho = \frac{Cov(X, Y)}{\sigma(X) \sigma(Y)} \in [-1, 1]$$

Covariance is a measure that indicates the extent to which two random variables change in cycle. It explains the systematic relation between a pair of random variables, wherein changes in one variable reciprocal by a corresponding change in another variable.

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]$$



Q8. What is the difference between Point Estimates and Confidence Interval?

Point Estimation gives us a particular value as an estimate of a population parameter. Method of Moments and Maximum Likelihood estimator methods are used to derive Point Estimators for population parameters.

A confidence interval gives us a range of values which is likely to contain the population parameter. The confidence interval is generally preferred, as it tells us how likely this interval is to contain the population parameter. This likeliness or probability is called Confidence Level or Confidence coefficient and represented by $1 - \alpha$, where α is the level of significance.

Q9. What is the goal of A/B Testing?

It is a hypothesis testing for a randomized experiment with two variables A and B.

The goal of A/B Testing is to identify any changes to the web page to maximize or increase the outcome of interest. A/B testing is a fantastic method for figuring out the best online promotional and marketing strategies for your business. It can be used to test everything from website copy to sales emails to search ads. An example of this could be identifying the click-through rate for a banner ad.

Q10. What is p-value?

When you perform a hypothesis test in statistics, a p-value can help you determine the strength of your results. p-value is the minimum significance level at which you can reject the null hypothesis. The lower the p-value, the more likely you reject the null hypothesis.

Q11. In any 15-minute interval, there is a 20% probability that you will see at least one shooting star. What is the probability that you see at least one shooting star in the period of an hour?

- Probability of not seeing any shooting star in 15 minutes is = $1 - P(\text{Seeing one shooting star}) = 1 - 0.2 = 0.8$
- Probability of not seeing any shooting star in the period of one hour = $(0.8)^4 = 0.4096$

- *Probability of seeing at least one shooting star in the one hour* =
 $1 - P(\text{Not seeing any star}) = 1 - 0.4096 = 0.5904$

Q12. How can you generate a random number between 1 – 7 with only a die?

Any die has six sides from 1-6. There is no way to get seven equal outcomes from a single rolling of a die. If we roll the die twice and consider the event of two rolls, we now have 36 different outcomes. To get our 7 equal outcomes we have to reduce this 36 to a number divisible by 7. We can thus consider only 35 outcomes and exclude the other one. A simple scenario can be to exclude the combination (6,6), i.e., to roll the die again if 6 appears twice. All the remaining combinations from (1,1) till (6,5) can be divided into 7 parts of 5 each. This way all the seven sets of outcomes are equally likely.

Q13. A certain couple tells you that they have two children, at least one of which is a girl. What is the probability that they have two girls?

$$P(\text{Having two girls given one girl}) = \frac{1}{2}$$

Q14. A jar has 1000 coins, of which 999 are fair and 1 is double headed. Pick a coin at random and toss it 10 times. Given that you see 10 heads, what is the probability that the next toss of that coin is also a head?

There are two ways of choosing the coin. One is to pick a fair coin and the other is to pick the one with two heads.

$$\text{Probability of selecting fair coin} = \frac{999}{1000} = 0.999$$

$$\text{Probability of selecting unfair coin} = \frac{1}{1000} = 0.001$$

Selecting 10 heads in a row

$$= \text{Selecting fair coin} * \text{Getting 10 heads} + \text{Selecting unfair coin} \\ = P(A) + P(B)$$

$$P(A) = 0.999 * \left(\frac{1}{2}\right)^{10} = 0.999 * \left(\frac{1}{1024}\right) = 0.000976$$

$$P(B) = 0.001 * 1 = 0.001$$

$$\frac{P(A)}{P(A) + P(B)} = \frac{0.000976}{0.000976 + 0.001} = 0.4939$$

$$\frac{P(B)}{P(A) + P(B)} = \frac{0.001}{0.001976} = 0.5061$$

$$\begin{aligned} \text{Probability of selecting another head} &= \frac{P(A)}{P(A) + P(B)} * 0.5 + \frac{P(B)}{P(A) + P(B)} * 1 = \\ &= 0.4939 * 0.5 + 0.5061 = 0.7531 \end{aligned}$$

Q15. What do you understand by statistical power of sensitivity and how do you calculate it?

Sensitivity is commonly used to validate the accuracy of a classifier (Logistic, SVM, Random Forest etc.).

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

Q16. Why is Re-sampling done?

<https://machinelearningmastery.com/statistical-sampling-and-resampling/>

- Sampling is an active process of gathering observations with the intent of estimating a population variable.
- Resampling is a methodology of economically using a data sample to improve the accuracy and quantify the uncertainty of a population parameter. Resampling methods, in fact, make use of a nested resampling method.

Once we have a data sample, it can be used to estimate the population parameter. The problem is that we only have a single estimate of the population parameter, with little idea of the variability or uncertainty in the estimate. One way to address this is by estimating the population parameter multiple times from our data sample. This is called resampling. Statistical resampling methods are procedures that describe how to economically use available data to estimate a population parameter. The result can be both a more accurate estimate of the parameter (such as taking the mean of the estimates) and a quantification of the uncertainty of the estimate (such as adding a confidence interval).

Resampling methods are very easy to use, requiring little mathematical knowledge. A downside of the methods is that they can be computationally very expensive, requiring tens, hundreds, or even thousands of resamples in order to develop a robust estimate of the population parameter.

The key idea is to resample from the original data — either directly or via a fitted model — to create replicate datasets, from which the variability of the quantiles of interest can be assessed without long-winded and error-prone analytical calculation. Because this approach involves repeating the original data analysis procedure with many replicate sets of data, these are sometimes called computer-intensive methods. Each new subsample from the original data sample is used to estimate the population parameter. The sample of estimated population parameters can then be considered with statistical tools in order to quantify the expected value and variance, providing measures of the uncertainty of the estimate. Statistical sampling methods can be used in the selection of a subsample from the original sample.

A key difference is that process must be repeated multiple times. The problem with this is that there will be some relationship between the samples as observations that will be shared across multiple subsamples. This means that the subsamples and the estimated population parameters are not strictly

identical and independently distributed. This has implications for statistical tests performed on the sample of estimated population parameters downstream, i.e. paired statistical tests may be required.

Two commonly used resampling methods that you may encounter are k-fold cross-validation and the bootstrap.

- Bootstrap. Samples are drawn from the dataset with replacement (allowing the same sample to appear more than once in the sample), where those instances not drawn into the data sample may be used for the test set.
- k-fold Cross-Validation. A dataset is partitioned into k groups, where each group is given the opportunity of being used as a held out test set leaving the remaining groups as the training set. The k-fold cross-validation method specifically lends itself to use in the evaluation of predictive models that are repeatedly trained on one subset of the data and evaluated on a second held-out subset of the data.

Resampling is done in any of these cases:

- Estimating the accuracy of sample statistics by using subsets of accessible data or drawing randomly with replacement from a set of data points
- Substituting labels on data points when performing significance tests
- Validating models by using random subsets (bootstrapping, cross-validation)

Q17. What are the differences between over-fitting and under-fitting?

In statistics and machine learning, one of the most common tasks is to fit a model to a set of training data, so as to be able to make reliable predictions on general untrained data.

In overfitting, a statistical model describes random error or noise instead of the underlying relationship. Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been overfitted, has poor predictive performance, as it overreacts to minor fluctuations in the training data.

Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model too would have poor predictive performance.

Q18. How to combat Overfitting and Underfitting?

To combat overfitting:

1. Add noise
2. Feature selection
3. Increase training set
4. L2 (ridge) or L1 (lasso) regularization; L1 drops weights, L2 no
5. Use cross-validation techniques, such as k folds cross-validation
6. Boosting and bagging
7. Dropout technique

8. Perform early stopping

9. Remove inner layers

To combat underfitting:

1. Add features

2. Increase time of training

Q19. What is regularization? Why is it useful?

Regularization is the process of adding tuning parameter (penalty term) to a model to induce smoothness in order to prevent overfitting. This is most often done by adding a constant multiple to an existing weight vector. This constant is often the L1 (Lasso - $|\alpha|$) or L2 (Ridge - α^2). The model predictions should then minimize the loss function calculated on the regularized training set.

Q20. What Is the Law of Large Numbers?

It is a theorem that describes the result of performing the same experiment a large number of times. This theorem forms the basis of frequency-style thinking. It says that the sample means, the sample variance and the sample standard deviation converge to what they are trying to estimate. According to the law, the average of the results obtained from a large number of trials should be close to the expected value and will tend to become closer to the expected value as more trials are performed.

Q21. What Are Confounding Variables?

In statistics, a confounder is a variable that influences both the dependent variable and independent variable.

If you are researching whether a lack of exercise leads to weight gain:

lack of exercise = independent variable

weight gain = dependent variable

A confounding variable here would be any other variable that affects both of these variables, such as the age of the subject.

Q22. What Are the Types of Biases That Can Occur During Sampling?

a. Selection bias

b. Under coverage bias

c. Survivorship bias

Q23. What is Survivorship Bias?

It is the logical error of focusing aspects that support surviving some process and casually overlooking those that did not work because of their lack of prominence. This can lead to wrong conclusions in numerous different means. For example, during a recession you look just at the survived businesses, noting

that they are performing poorly. However, they perform better than the rest, which is failed, thus being removed from the time series.

Q24. What is Selection Bias? What is under coverage bias?

<https://stattrek.com/survey-research/survey-bias.aspx>

Selection bias occurs when the sample obtained is not representative of the population intended to be analyzed. For instance, you select only Asians to perform a study on the world population height.

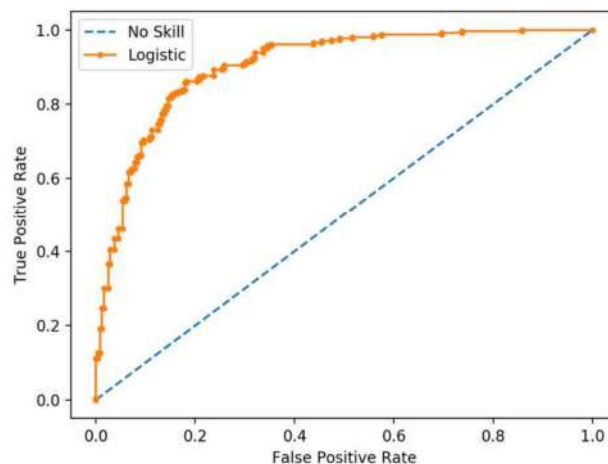
Under coverage bias occurs when some members of the population are inadequately represented in the sample. A classic example of under coverage is the Literary Digest voter survey, which predicted that Alfred Landon would beat Franklin Roosevelt in the 1936 presidential election. The survey sample suffered from under coverage of low-income voters, who tended to be Democrats.

How did this happen? The survey relied on a convenience sample, drawn from telephone directories and car registration lists. In 1936, people who owned cars and telephones tended to be more affluent. Under coverage is often a problem with convenience samples.

Q25. Explain how a ROC curve works?

The ROC curve is a graphical representation of the contrast between true positive rates and false positive rates at various thresholds. It is often used as a proxy for the trade-off between the sensitivity (true positive rate) and false positive rate.

- $TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$
- $TNR = \frac{TN}{TN+FP} = \frac{TN}{N}$
- $FPR = \frac{FP}{TN+FP}$
- $FNR = \frac{FN}{FN+T}$



Q26. What is TF/IDF vectorization?

TF-IDF is short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining.

- $TF = \frac{\# \text{'word' in doc}}{\text{tot \# words in doc}}$
- $IDF = \log\left(\frac{\# \text{ docs with 'word' in it}}{\text{tot docs in collection}}\right)$

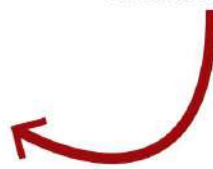
The TF-IDF value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

Q27. Why we generally use Soft-max (or sigmoid) non-linearity function as last operation in-network? Why RELU in an inner layer?

It is because it takes in a vector of real numbers and returns a probability distribution. Its definition is as follows. Let x be a vector of real numbers (positive, negative, whatever, there are no constraints). Then the i -eth component of soft-max(x) is:

$$P(y=j | \theta^{(i)}) = \frac{e^{\theta^{(i)}}}{\sum_{j=0}^k e^{\theta_k^{(i)}}}$$

Softmax function



where $\theta = w_0x_0 + w_1x_1 + \dots + w_kx_k = \sum_{i=0}^k w_i x_i = w^T x$

It should be clear that the output is a probability distribution: each element is non-negative and the sum over all components is 1.

RELU because it avoids the vanishing gradient descent issue.

Data Analysis

Q1. Python or R – Which one would you prefer for text analytics?

We will prefer Python because of the following reasons:

- Python would be the best option because it has **Pandas library** that provides easy to use data structures and high-performance data analysis tools.
- R is more suitable for machine learning than just text analysis.
- **Python performs faster for all types of text analytics.**

Q2. How does data cleaning play a vital role in the analysis?

Data cleaning can help in analysis because:

- Cleaning data from multiple sources helps transform it into a format that data analysts or data scientists can work with.
- Data Cleaning helps increase the accuracy of the model in machine learning.
- It is a cumbersome process because as the number of data sources increases, the time taken to clean the data increases exponentially due to the number of sources and the volume of data generated by these sources.
- It might take up to 80% of the time for just cleaning data making it a critical part of the analysis task.

Q3. Differentiate between univariate, bivariate and multivariate analysis.

Univariate analyses are **descriptive statistical analysis techniques which can be differentiated based on one variable involved at a given point of time.** *For example, the pie charts of sales based on territory involve only one variable and can the analysis can be referred to as univariate analysis.*

The bivariate analysis attempts to understand the difference between two variables at a time as in a scatterplot. *For example, analyzing the volume of sale and spending can be considered as an example of bivariate analysis.*

Multivariate analysis deals with the study of more than two variables to understand the effect of variables on the responses.

Q4. Explain Star Schema.

It is a traditional database schema with a central table. Satellite tables map IDs to physical names or descriptions and can be connected to the central fact table using the ID fields; these tables are known as lookup tables and are principally useful in real-time applications, as they save a lot of memory. Sometimes star schemas involve several layers of summarization to recover information faster.

Q5. What is Cluster Sampling?

Cluster sampling is a technique used when it becomes difficult to study the target population spread across a wide area and simple random sampling cannot be applied. Cluster Sample is a probability sample where each sampling unit is a collection or cluster of elements.

For example, a researcher wants to survey the academic performance of high school students in Japan. He can divide the entire population of Japan into different clusters (cities). Then the researcher selects a number of clusters depending on his research through simple or systematic random sampling.

Q6. What is Systematic Sampling?

Systematic sampling is a statistical technique where elements are selected from an ordered sampling frame. In systematic sampling, the list is progressed in a circular manner so once you reach the end of the list, it is progressed from the top again. The best example of systematic sampling is equal probability method.

Q7. What are Eigenvectors and Eigenvalues?

Eigenvectors are used for understanding linear transformations. In data analysis, we usually calculate the eigenvectors for a correlation or covariance matrix. Eigenvectors are the directions along which a particular linear transformation acts by flipping, compressing or stretching.

Eigenvalue can be referred to as the strength of the transformation in the direction of eigenvector or the factor by which the compression occurs.

Q8. Can you cite some examples where a false positive is important than a false negative?

Let us first understand what false positives and false negatives are

- False Positives are the cases where you wrongly classified a non-event as an event a.k.a Type I error.
- False Negatives are the cases where you wrongly classify events as non-events, a.k.a Type II error.

Example 1: In the medical field, assume you have to give chemotherapy to patients. Assume a patient comes to that hospital and he is tested positive for cancer, based on the lab prediction but he actually doesn't have cancer. This is a case of false positive. Here it is of utmost danger to start chemotherapy on this patient when he actually does not have cancer. In the absence of cancerous cell, chemotherapy will do certain damage to his normal healthy cells and might lead to severe diseases, even cancer.

Example 2: Let's say an e-commerce company decided to give \$1000 Gift voucher to the customers whom they assume to purchase at least \$10,000 worth of items. They send free voucher mail directly to 100 customers without any minimum purchase condition because they assume to make at least 20% profit on sold items above \$10,000. Now the issue is if we send the \$1000 gift vouchers to customers who have not actually purchased anything but are marked as having made \$10,000 worth of purchase.

Q9. Can you cite some examples where a false negative important than a false positive? And vice versa?

Example 1 FN: What if Jury or judge decides to make a criminal go free?

Example 2 FN: Fraud detection.

Example 3 FP: customer voucher use promo evaluation: if many used it and actually it was not true, promo sucks.

Q10. Can you cite some examples where both false positive and false negatives are equally important?

In the Banking industry giving loans is the primary source of making money but at the same time if your repayment rate is not good you will not make any profit, rather you will risk huge losses.

Banks don't want to lose good customers and at the same point in time, they don't want to acquire bad customers. In this scenario, both the false positives and false negatives become very important to measure.

Q11. Can you explain the difference between a Validation Set and a Test Set?

A Training Set:

- to fit the parameters i.e. weights

A Validation set:

- part of the training set
- for parameter selection
- to avoid overfitting

A Test set:

- for testing or evaluating the performance of a trained machine learning model, i.e. evaluating the predictive power and generalization.

Q12. Explain cross-validation.

<https://machinelearningmastery.com/k-fold-cross-validation/>

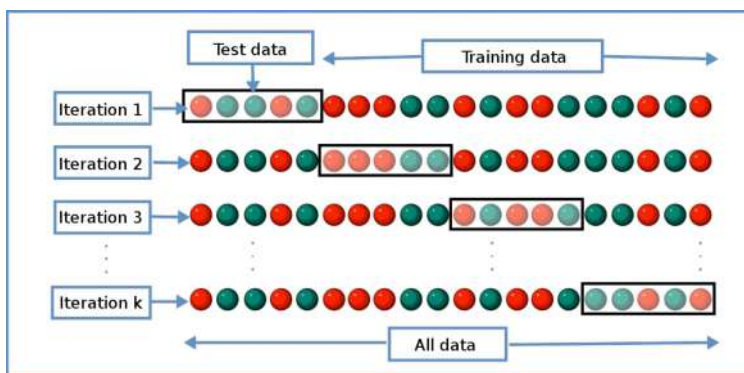
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k -fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as $k=10$ becoming 10-fold cross-validation. Mainly used in backgrounds where the objective is forecast, and one wants to estimate how accurately a model will accomplish in practice.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
 - a. Take the group as a hold out or test data set
 - b. Take the remaining groups as a training data set
 - c. Fit a model on the training set and evaluate it on the test set
 - d. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores



There is an alternative in Scikit-Learn called Stratified k fold, in which the split is shuffled to make it sure you have a representative sample of each class and a k fold in which you may not have the assurance of it (not good with a very unbalanced dataset).

Machine Learning

Q1. What is Machine Learning?

Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine Learning explores the study and construction of algorithms that can learn from and make predictions on data. You select a model to train and then manually perform feature extraction. Used to devise complex models and algorithms that lend themselves to a prediction which in commercial use is known as predictive analytics.

Q2. What is Supervised Learning?

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples.

Algorithms: Support Vector Machines, Regression, Naive Bayes, Decision Trees, K-nearest Neighbor Algorithm and Neural Networks

E.g. If you built a fruit classifier, the labels will be "this is an orange, this is an apple and this is a banana", based on showing the classifier examples of apples, oranges and bananas.

Q3. What is Unsupervised learning?

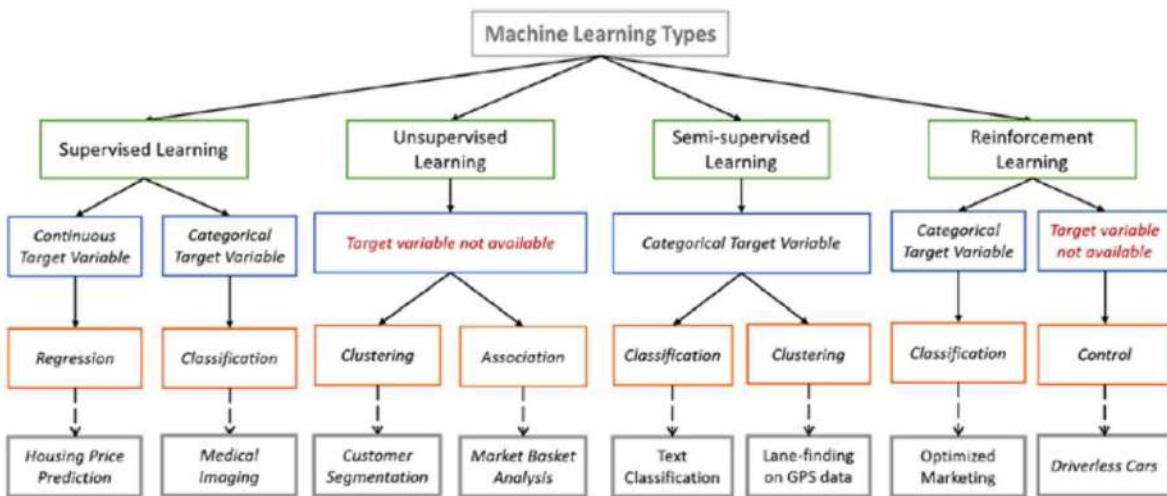
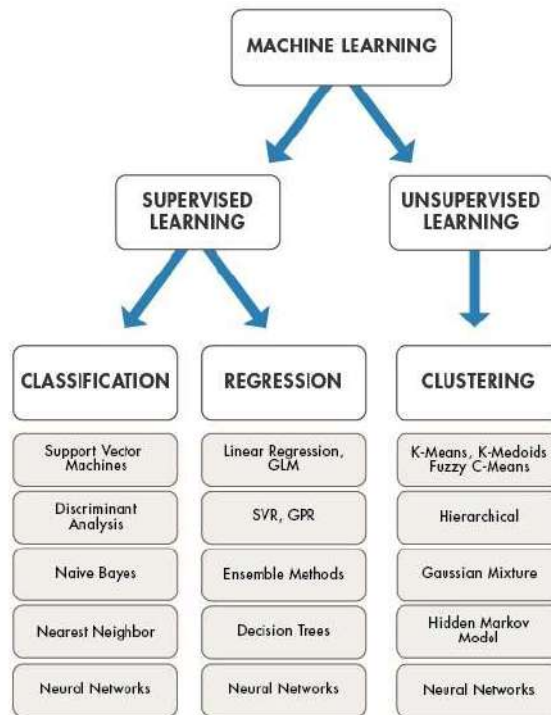
Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses.

Algorithms: Clustering, Anomaly Detection, Neural Networks and Latent Variable Models

E.g. In the same example, a fruit clustering will categorize as "fruits with soft skin and lots of dimples", "fruits with shiny hard skin" and "elongated yellow fruits".

Q4. What are the various algorithms?

There are various algorithms. Here is a list.



Q5. What is 'Naive' in a Naive Bayes?

https://en.wikipedia.org/wiki/Naive_Bayes_classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n :

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

Using the naive conditional independence assumption that each x_i is independent: for all i , this relationship is simplified to:

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)}$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(y|x_i)$; the former is then the relative frequency of class y in the training set.

$$P(y | x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i | y)$$
$$\Downarrow$$
$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y),$$

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(y|x_i)$: can be Bernoulli, Binomial, Gaussian, and so on.

Q6. What is PCA? When do you use it?

https://en.wikipedia.org/wiki/Principal_component_analysis

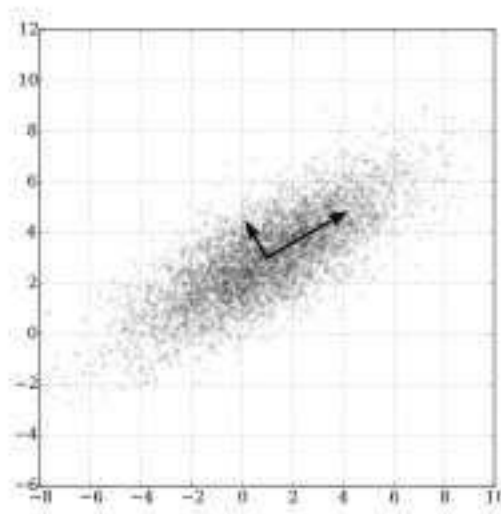
<https://blog.umetrics.com/what-is-principal-component-analysis-pca-and-how-it-is-used>

<https://blog.umetrics.com/why-preprocessing-data-creates-better-data-analytics-models>

Principal component analysis (PCA) is a statistical method used in Machine Learning. It consists in projecting data in a higher dimensional space into a lower dimensional space by maximizing the variance of each dimension.

The process works as following. We define a matrix A with n rows (the single observations of a dataset – in a tabular format, each single row) and p columns, our features. For this matrix we construct a variable space with as many dimensions as there are features. Each feature represents one coordinate axis. For

each feature, the length has been standardized according to a scaling criterion, normally by scaling to unit variance. It is determinant to scale the features to a common scale, otherwise the features with a greater magnitude will weigh more in determining the principal components. Once plotted all the observations and computed the mean of each variable, that mean will be represented by a point in the center of our plot (the center of gravity). Then, we subtract each observation with the mean, shifting the coordinate system with the center in the origin. The best fitting line resulting is the line that best accounts for the shape of the point swarm. It represents the maximum variance direction in the data. Each observation may be projected onto this line in order to get a coordinate value along the PC-line. This value is known as a score. The next best-fitting line can be similarly chosen from directions perpendicular to the first. Repeating this process yields an orthogonal basis in which different individual dimensions of the data are uncorrelated. These basis vectors are called **principal components**.



PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It is often used to visualize genetic distance and relatedness between populations.

Q7. Explain SVM algorithm in detail.

https://en.wikipedia.org/wiki/Support_vector_machine

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support-vector machines, a data point is viewed as a p -dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $(p - 1)$ -dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So, we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum-margin classifier; or equivalently, the perceptron of optimal stability. The best hyper plane that divides the data is H_3 .

We have n data $(x_1, y_1), \dots, (x_n, y_n)$ and p different features $x_i = (x_i^1, \dots, x_i^p)$ and y_i is either 1 or -1.

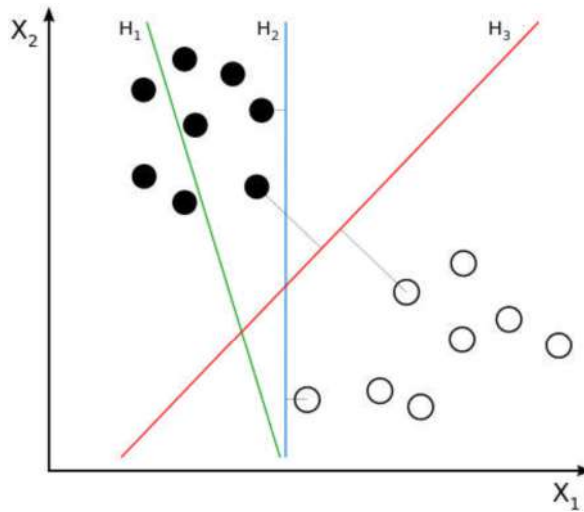
The equation of the hyperplane H_3 is as the set of points x satisfying:

$$w \cdot x - b = 0$$

where w is the (not necessarily normalized) normal vector to the hyperplane. The parameter $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin along the normal vector w .

So, for each i , either x_i is in the hyperplane of 1 or -1. Basically, x_i satisfies:

$$w \cdot x_i - b \geq 1 \quad \text{or} \quad w \cdot x_i - b \leq -1$$



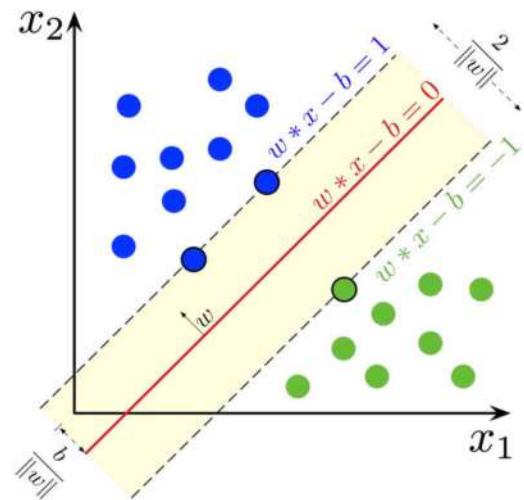
- SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings. Some methods for shallow semantic parsing are based on support vector machines.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.
- Classification of satellite data like SAR data using supervised SVM.
- Hand-written characters can be recognized using SVM.

Q8. What are the support vectors in SVM?

In the diagram, we see that the sketched lines mark the distance from the classifier (the hyper plane) to the closest data points called the support vectors (darkened data points). The distance between the two thin lines is called the margin.

To extend SVM to cases in which the data are not linearly separable, we introduce the hinge loss function,

$$\max(0, 1 - y_i(w \cdot x_i - b))$$



This function is zero if x lies on the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.

Q9. What are the different kernels in SVM?

There are four types of kernels in SVM.

1. LinearKernel
2. Polynomial kernel
3. Radial basis kernel
4. Sigmoid kernel

Q10. What are the most known ensemble algorithms?

<https://towardsdatascience.com/the-ultimate-guide-to-adaboost-random-forests-and-xgboost-7f9327061c4f>

The most popular trees are: **AdaBoost, Random Forest, and eXtreme Gradient Boosting (XGBoost).**

AdaBoost is **best used** in a dataset with low noise, when computational complexity or timeliness of results is not a main concern and when there are not enough resources for broader hyperparameter tuning due to lack of time and knowledge of the user.

Random forests should not be used when dealing with time series data or any other data where look-ahead bias should be avoided, and the order and continuity of the samples need to be ensured. This algorithm can handle noise relatively well, but more knowledge from the user is required to adequately tune the algorithm compared to AdaBoost.

The main advantages of XGBoost is its lightning speed compared to other algorithms, such as AdaBoost, and its regularization parameter that successfully reduces variance. But even aside from the regularization parameter, this algorithm leverages a learning rate (shrinkage) and subsamples from the features like random forests, which increases its ability to generalize even further. However, XGBoost is more difficult to understand, visualize and to tune compared to AdaBoost and random forests. There is a multitude of hyperparameters that can be tuned to increase performance.

Q11. Explain Decision Tree algorithm in detail.

https://en.wikipedia.org/wiki/Decision_tree_learning

<https://www.kdnuggets.com/2019/02/decision-trees-introduction.html/2>

<https://medium.com/@naeemsunesara/giniscore-entropy-and-information-gain-in-decision-trees-cbc08589852d>

A decision tree is a supervised machine learning algorithm mainly used for Regression and Classification. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision tree can handle both categorical and numerical data. The term Classification and Regression Tree (CART) analysis is an umbrella term used to refer to both of the above procedures.

Some techniques, often called *ensemble* methods, construct more than one decision tree:

- **Boosted trees** Incrementally building an ensemble by training each new instance to emphasize the training instances previously mis-modeled. A typical example is [AdaBoost](#). These can be used for regression-type and classification-type problems.
- **Bootstrap aggregated** (or bagged) decision trees, an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction.
 - A **random forest** classifier is a specific type of [bootstrap aggregating](#).
- **Rotation forest** – in which every decision tree is trained by first applying [principal component analysis](#) (PCA) on a random subset of the input features.

A special case of a decision tree is a decision list, which is a one-sided decision tree, so that every internal node has exactly 1 leaf node and exactly 1 internal node as a child (except for the bottommost node, whose only child is a single leaf node). While less expressive, decision lists are arguably easier to understand than general decision trees due to their added sparsity, permit non-greedy learning methods and monotonic constraints to be imposed.

Notable decision tree algorithms include:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)
- CART (Classification and Regression Tree)
- Chi-square automatic interaction detection (CHAID). Performs multi-level splits when computing classification trees.
- MARS: extends decision trees to handle numerical data better.
- Conditional Inference Trees. Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting. This approach results in unbiased predictor selection and does not require pruning.

Q12. What are Entropy and Information gain in Decision tree algorithm?

https://www.saedsayad.com/decision_tree.htm

<https://medium.com/@naeemsunesara/giniscor-entropy-and-information-gain-in-decision-trees-cbc08589852d>

There are a lot of algorithms which are employed to build a decision tree, ID3 (Iterative Dichotomiser 3), C4.5, C5.0, CART (Classification and Regression Trees) to name a few but at their core all of them tell us what questions to ask and when.

The below table has color and diameter of a fruit and the label tells the name of the fruit. How do we build a decision tree to classify the fruits?

Color	Diameter	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Here is how we will build the tree. We will start with a node which will ask a true or false question to split the data into two. The two resulting nodes will each ask a true or false question again to split the data further and so on.

There are 2 main things to consider with the above approach:

- Which is the best question to ask at each node
- When do we stop splitting the data further?

Let's start building the tree with the first or the topmost node. There is a list of possible questions which can be asked. The first node can ask the following questions:

- *Is the color green?*
- *Is the color yellow?*
- *Is the color red?*
- *Is the diameter ≥ 3 ?*
- *Is the diameter ≥ 1 ?*

Of these possible set of questions, which one is the best to ask so that our data is split into two sets after the first node? Remember we are trying to split or classify our data into separate classes. **Our question should be such that our data is partitioned into as unmixed or pure classes as possible. An impure set or class here refers to one which has many different types of objects** for example if we ask the question for the above data, "Is the color green?" our data will be split into two sets one of which will be pure the other will have a mixed set of labels. If we assign a label to a mixed set, we have higher chances of being incorrect. But how do we measure this impurity?

Color	Diameter	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Gini Impurity and Information Gain - CART

CART (Classification and Regression Trees) → uses *Gini Index (Classification)* as metric.

The Gini Impurity (GI) metric measures the homogeneity of a set of items. The lowest possible value of GI is 0.0. The maximum value of GI depends on the particular problem being investigated but gets close to 1.0.

Suppose for example you have 12 items — apples, grapes, lemons. If there are 0 apples, 0 grapes, 12 lemons, then you have minimal impurity (this is good for decision trees) and $GI = 0.0$. But if you have 4 apples, 4 grapes, 4 lemons, you have maximum impurity and it turns out that $GI = 0.667$.

I'll show example calculations.

Maximum GI: Apples, Grapes, Lemons

```
count = 4 4 4
p = 4/12 4/12 4/12
  = 1/3 1/3 1/3

GI = 1 - [ (1/3)^2 + (1/3)^2 + (1/3)^2 ]
  = 1 - [ 1/9 + 1/9 + 1/9 ]
  = 1 - 1/3
  = 2/3
  = 0.667
```

When the number of items is evenly distributed, as in the example above, you have maximum GI but the exact value depends on how many items there are. A bit less than maximum GI:

```
count = 3 3 6
p = 3/12 3/12 6/12
  = 1/4 1/4 1/2

GI = 1 - [ (1/4)^2 + (1/4)^2 + (1/2)^2 ]
  = 1 - [ 1/16 + 1/16 + 1/4 ]
  = 1 - 6/16
  = 10/16
  = 0.625
```

In the example above, the items are not quite evenly distributed, and the GI is slightly less (which is better when used for decision trees). Minimum GI:

```
count = 0 12 0
p = 0/12 12/12 0/12
  = 0 1 0

GI = 1 - [ 0^2 + 1^2 + 0^2 ]
  = 1 - [ 0 + 1 + 0 ]
  = 1 - 1
  = 0.00
```

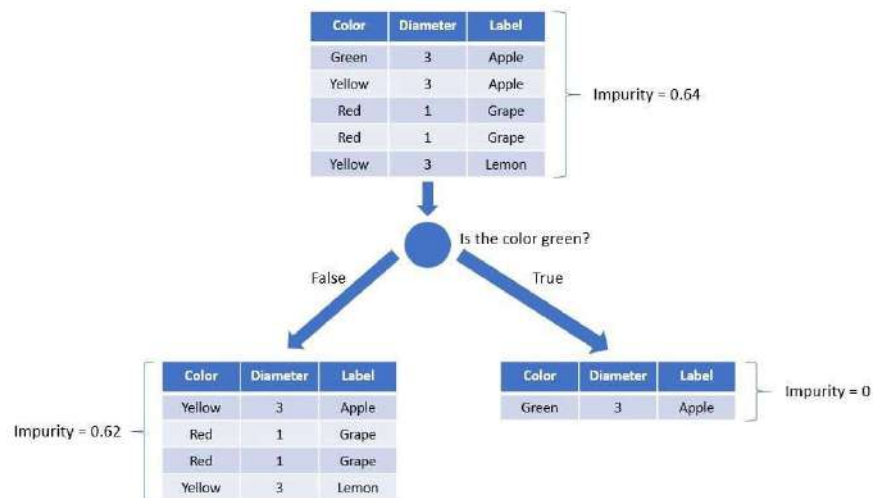
J classes $\{1, 2, \dots, J\}$, p_i fraction of elements of class i :

$$\text{Gini impurity: } I_G(p) = 1 - \sum_{i=1}^J p_i^2$$

The **Gini index** is not at all the same as a different metric called the **Gini coefficient**. The Gini impurity metric can be used when creating a decision tree but there are alternatives, including **Entropy Information gain**. The advantage of GI is its simplicity.

Information Gain

Information gain is another metric which tells us how much a question *unmixes* the labels at a node. **“Mathematically it is just a difference between impurity values before splitting the data at a node and the weighted average of the impurity after the split”**. For instance, if we go back to our data of apples, lemons and grapes and ask the question “Is the color Green?”



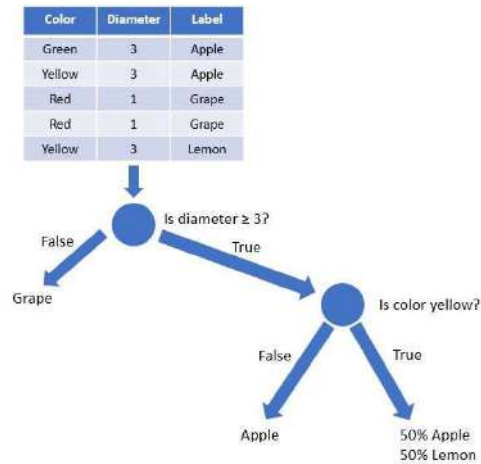
$$\text{Information Gain} = 0.64 - \left(\frac{4}{5} * 0.62 + \frac{1}{5} * 0 \right) = 0.144$$

The information gain by asking this question is 0.144. Similarly, we can ask another question from the set of possible questions split the data and compute information gain. This is also called (**Recursive Binary Splitting**).

Question	Information Gain
Is the color green?	0.14
Is diameter ≥ 3 ?	0.37
Is the color yellow?	0.17
Is the color red?	0.37
Is diameter ≥ 1 ?	0

The question where we have the highest information gain “*Is diameter ≥ 3 ?*” is the best question to ask. Note that the information gain is same for the question “*Is the color red?*” we just picked the first one at random.

Repeating the same method at the child node we can complete the tree. Note that no further questions can be asked which would increase the information gain.



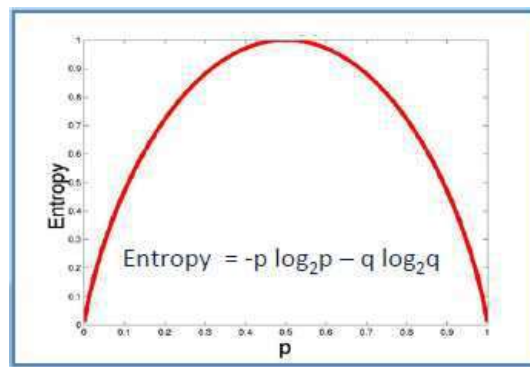
Also note that the rightmost leaf which says 50% Apple & 50% lemon means that this class cannot be divided further, and this branch can tell an apple or a lemon with 50% probability. For the grape and apple branches we stop asking further questions since the Gini Impurity is 0 for those.

Entropy and Information Gain – ID3

ID3 (Iterative Dichotomiser 3) → uses **Entropy function** and **information gain** as metrics.



If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:

a) Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

$$\begin{aligned} \text{Entropy(PlayGolf)} &= \text{Entropy}(5,9) \\ &= \text{Entropy}(0.36, 0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}$$

b) Entropy using the frequency table of two attributes:

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14



$$\begin{aligned}
 E(\text{PlayGolf, Outlook}) &= P(\text{Sunny}) * E(3,2) + P(\text{Overcast}) * E(4,0) + P(\text{Rainy}) * E(2,3) \\
 &= (5/14) * 0.971 + (4/14) * 0.0 + (5/14) * 0.971 \\
 &= 0.693
 \end{aligned}$$

Information Gain

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

Step 1: Calculate entropy of the target.

$$\begin{aligned}
 \text{Entropy}(\text{PlayGolf}) &= \text{Entropy}(5,9) \\
 &= \text{Entropy}(0.36, 0.64) \\
 &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\
 &= 0.94
 \end{aligned}$$

Step 2: The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain or decrease in entropy.

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	
	Overcast	4	0	
	Rainy	2	3	
				Gain = 0.247

		Play Golf		
		Yes	No	
Temp.	Hot	2	2	
	Mild	4	2	
	Cool	3	1	
				Gain = 0.029

		Play Golf		
		Yes	No	
Humidity	High	3	4	
	Normal	6	1	
				Gain = 0.152

		Play Golf		
		Yes	No	
Windy	False	6	2	
	True	3	3	
				Gain = 0.048

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

$$G(\text{PlayGolf, Outlook}) = E(\text{PlayGolf}) - E(\text{PlayGolf, Outlook})$$

$$= 0.940 - 0.693 = 0.247$$

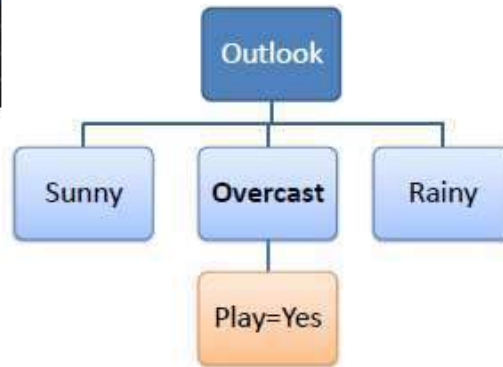
Step 3: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

Outlook	Temp	Humidity	Windy	Play Golf	
Sunny	Sunny	Mild	High	FALSE	Yes
	Sunny	Cool	Normal	FALSE	Yes
	Sunny	Cool	Normal	TRUE	No
	Sunny	Mild	Normal	FALSE	Yes
	Sunny	Mild	High	TRUE	No
Overcast	Overcast	Hot	High	FALSE	Yes
	Overcast	Cool	Normal	TRUE	Yes
	Overcast	Mild	High	TRUE	Yes
	Overcast	Hot	Normal	FALSE	Yes
Rainy	Rainy	Hot	High	FALSE	No
	Rainy	Hot	High	TRUE	No
	Rainy	Mild	High	FALSE	No
	Rainy	Cool	Normal	FALSE	Yes
	Rainy	Mild	Normal	TRUE	Yes

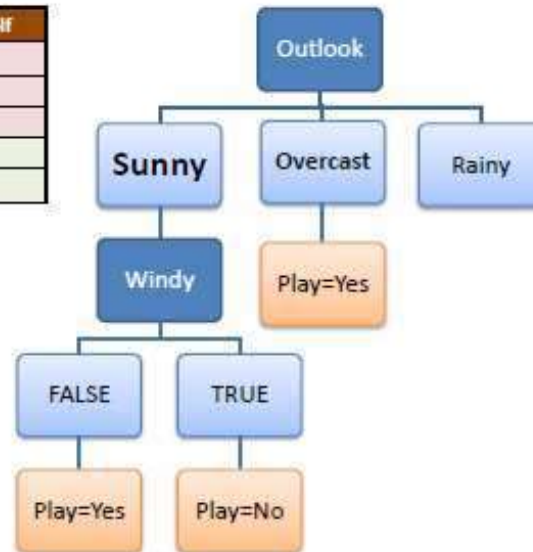
Step 4a: A branch with entropy of 0 is a leaf node.

Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Step 4b: A branch with entropy more than 0 needs further splitting.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Step 5: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Q13. What is pruning in Decision Tree?

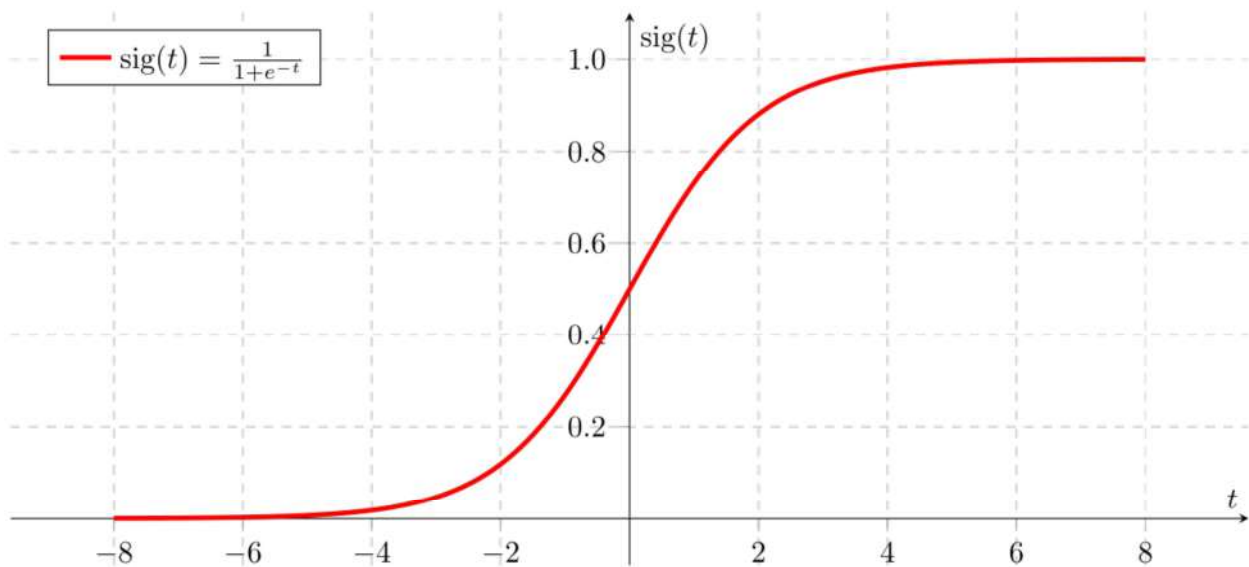
Pruning is a technique in machine learning and search algorithms that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. So, when we remove sub-nodes of a decision node, this process is called pruning or opposite process of splitting.

Q14. What is logistic regression? State an example when you have used logistic regression recently.

Logistic Regression often referred to as the logit model is a technique to predict the binary outcome from a linear combination of predictor variables. Since we are interested in a probability outcome, a line does not fit the model. Logistic Regression is a classification algorithm that works by trying to learn a function

that approximates $P(Y|X)$. It makes the central assumption that $P(X|Y)$ can be approximated as a sigmoid function applied to a linear combination of input features.

- $P(Y=1|X) = p \Rightarrow$ assume $\log \frac{p}{1-p} = b_0 + \sum_{i=1}^p b_i X_i \Leftrightarrow \frac{p}{1-p} = e^{b_0 + b^T X}$
 $\Leftrightarrow p = \frac{e^{b_0 + b^T X}}{1 + e^{b_0 + b^T X}}$
 $\Leftrightarrow p = \frac{1}{1 + e^{-\underbrace{(b_0 + b^T X)}_z}} = \text{sig}(z)$
- $P(Y=0|X) = 1 - \text{sig}(z)$

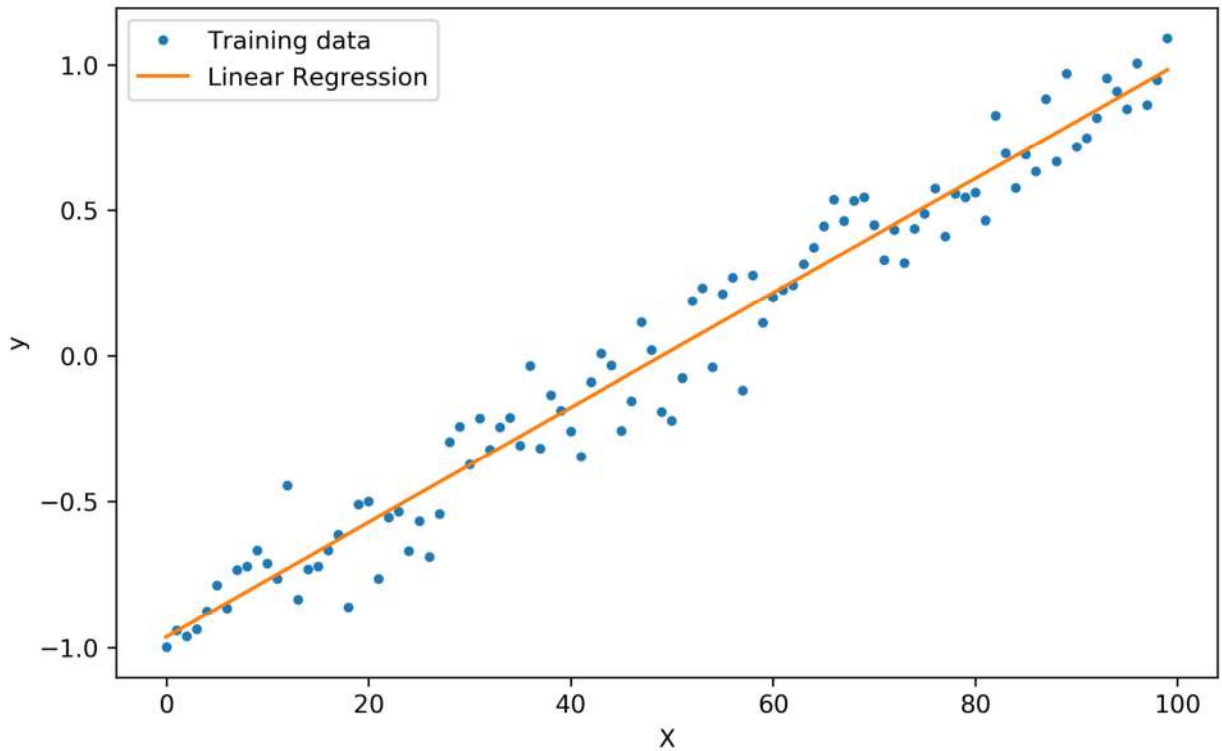


For example, if you want to predict whether a particular political leader will win the election or not. In this case, the outcome of prediction is binary i.e. 0 or 1 (Win/Lose). The predictor variables here would be the amount of money spent for election campaigning of a particular candidate, the amount of time spent in campaigning, etc.

Q15. What is Linear Regression?

Linear regression is a statistical technique where the score of a variable Y is predicted from the score of a second variable X. X is referred to as the predictor variable and Y as the criterion variable.

$$Y = b_0 + b_1 X_1 + \dots + b_p X_p$$



Q16. What Are the Drawbacks of the Linear Model?

Some drawbacks of the linear model are:

- The assumption of linearity of the model
- It can't be used for count outcomes or binary outcomes.
- There are overfitting or underfitting problems that it can't solve.

Q17. What is the difference between Regression and classification ML techniques?

Both Regression and classification machine learning techniques come under Supervised machine learning algorithms. In Supervised machine learning algorithm, we have to train the model using labelled data set, while training we have to explicitly provide the correct labels and algorithm tries to learn the pattern from input to output. If our labels are discrete values then it will be a classification problem, but if our labels are continuous values then it will be a regression problem.

Q18. What are Recommender Systems?

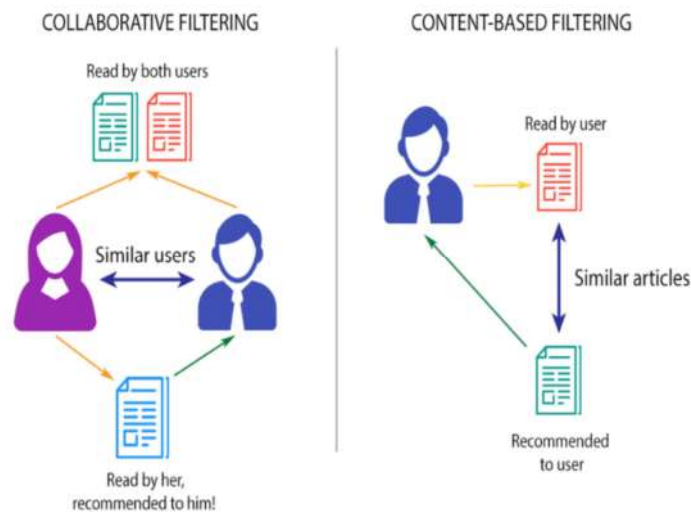
https://en.wikipedia.org/wiki/Recommender_system

Recommender Systems are a subclass of information filtering systems that are meant to predict the preferences or ratings that a user would give to a product. Recommender systems are widely used in movies, news, research articles, products, social tags, music, etc.

Examples include movie recommenders in IMDB, Netflix & BookMyShow, product recommenders in e-commerce sites like Amazon, eBay & Flipkart, YouTube video recommendations and game recommendations in Xbox.

Q19. What is Collaborative filtering? And a content based?

The process of filtering used by most of the recommender systems to find patterns or information by collaborating viewpoints, various data sources and multiple agents. Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. It looks at the items they like (usually based on rating) and combines them to create a ranked list of suggestions. Similar users are those with similar rating and on the based on that they get recommendations. In content based, we look only at the item level, recommending on similar items sold.



An example of collaborative filtering can be to predict the rating of a particular user based on his/her ratings for other movies and others' ratings for all movies. This concept is widely used in recommending movies in IMDB, Netflix & BookMyShow, product recommenders in e-commerce sites like Amazon, eBay & Flipkart, YouTube video recommendations and game recommendations in Xbox.

Q20. How can outlier values be treated?

Outlier values can be identified by using univariate or any other graphical analysis method. If the number of outlier values is few then they can be assessed individually but for a large number of outliers, the values can be substituted with either the 99th or the 1st percentile values.

All extreme values are not outlier values. The most common ways to treat outlier values:

1. Change it with a mean or median
2. Standardize the feature, changing the distribution but smoothing the outliers
3. Log transform the feature (with many outliers)
4. Drop the value

5. First/third quartile value if more than 2σ

Q21. What are the various steps involved in an analytics project?

The following are the various steps involved in an analytics project:

1. Understand the Business problem
2. Explore the data and become familiar with it
3. Prepare the data for modeling by detecting outliers, treating missing values, transforming variables, etc.
4. After data preparation, start running the model, analyze the result and tweak the approach. This is an iterative step until the best possible outcome is achieved.
5. Validate the model using a new data set.
6. Start implementing the model and track the result to analyze the performance of the model over the period of time.

Q22. During analysis, how do you treat missing values?

The extent of the missing values is identified after identifying the variables with missing values. If any patterns are identified the analyst has to concentrate on them as it could lead to interesting and meaningful business insights.

If there are no patterns identified, then the missing values can be substituted with mean or median values (imputation) or they can simply be ignored. Assigning a default value which can be mean, minimum or maximum value. Getting into the data is important.

If it is a categorical variable, the default value is assigned. The missing value is assigned a default value. If you have a distribution of data coming, for normal distribution give the mean value.

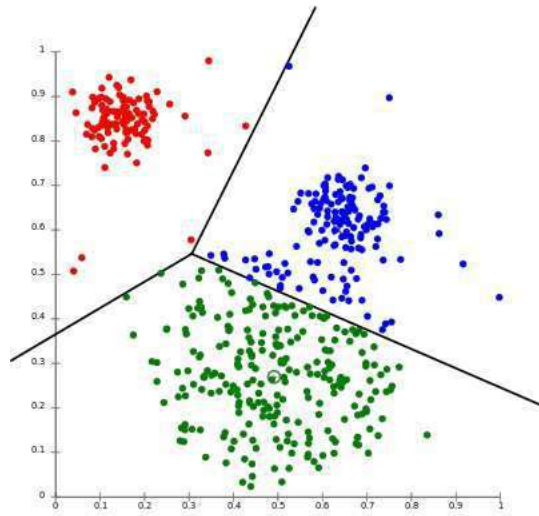
If 80% of the values for a variable are missing, then you can answer that you would be dropping the variable instead of treating the missing values.

Q23. How will you define the number of clusters in a clustering algorithm?

<https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/>

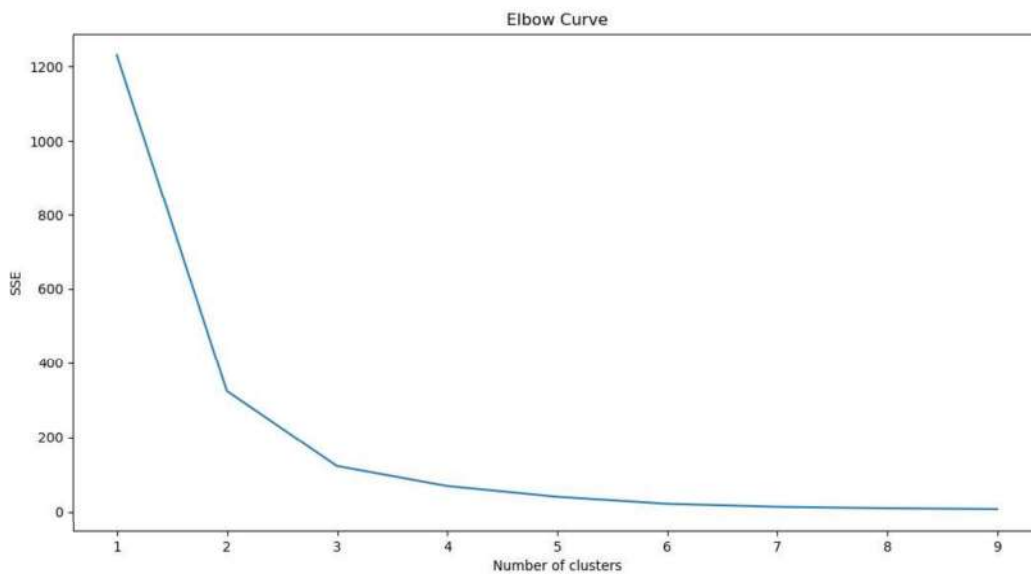
Though the Clustering Algorithm is not specified, this question is mostly in reference to K-Means clustering where "K" defines the number of clusters. The objective of clustering is to group similar entities in a way that the entities within a group are similar to each other, but the groups are different from each other.

For example, the following image shows three different groups.

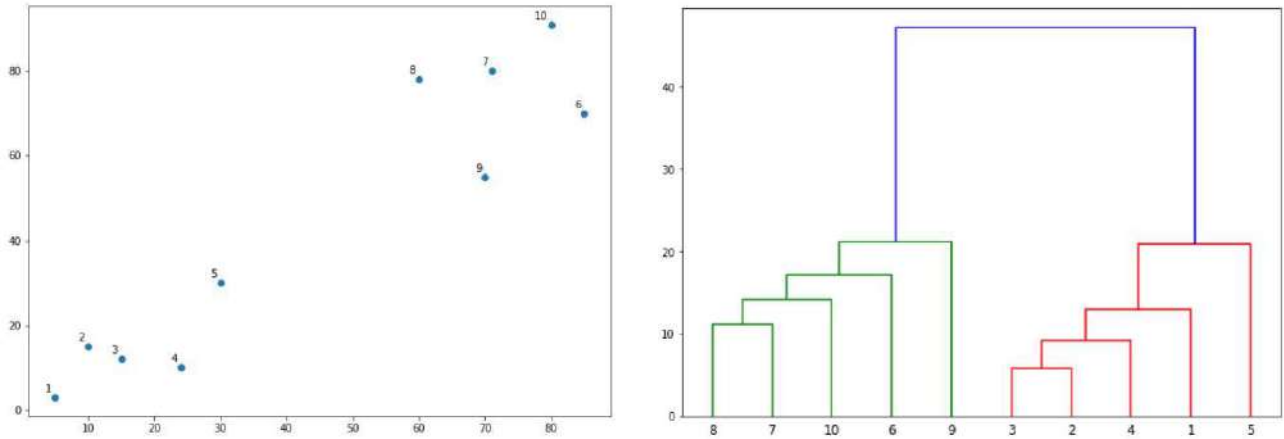


Within Sum of squares is generally used to explain the homogeneity within a cluster. If you plot WSS (as the sum of the squared distance between each member of the cluster and its centroid) for a range of number of clusters, you will get the plot shown below.

- The Graph is generally known as Elbow Curve.
- Red circled a point in above graph i.e. Number of Cluster = 3 is the point after which you don't see any decrement in WSS.
- This point is known as the bending point and taken as K in K – Means.

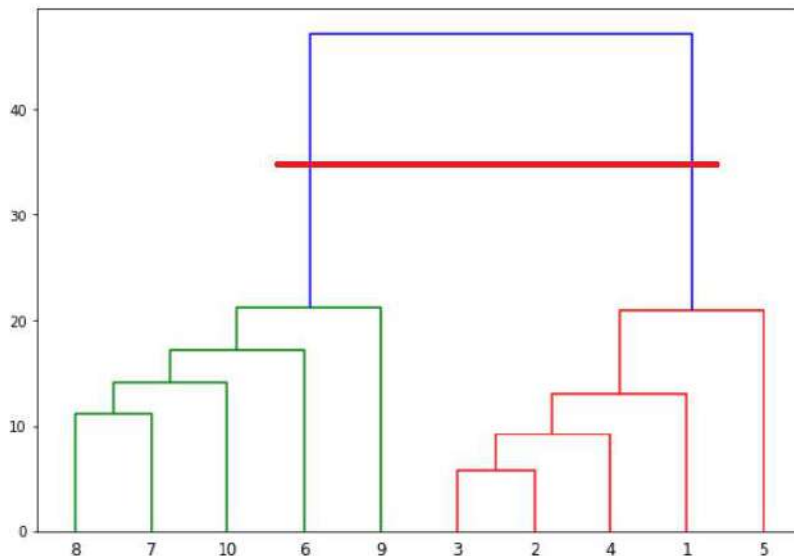


This is the widely used approach but few data scientists also use Hierarchical clustering first to create dendrograms and identify the distinct groups from there.

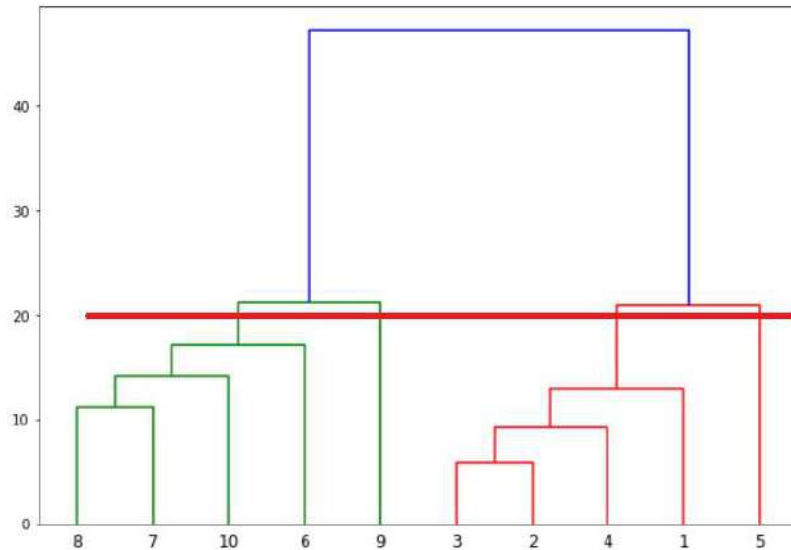


The algorithm starts by finding the two points that are closest to each other on the basis of Euclidean distance. If we look back at Graph1, we can see that points 2 and 3 are closest to each other while points 7 and 8 are closes to each other. Therefore a cluster will be formed between these two points first. In Graph2, you can see that the dendrograms have been created joining points 2 with 3, and 8 with 7. The vertical height of the dendogram shows the Euclidean distances between points. From Graph2, it can be seen that Euclidean distance between points 8 and 7 is greater than the distance between point 2 and 3. The next step is to join the cluster formed by joining two points to the next nearest cluster or point which in turn results in another cluster. If you look at Graph1, point 4 is closest to cluster of point 2 and 3, therefore in Graph2 dendrogram is generated by joining point 4 with dendrogram of point 2 and 3. This process continues until all the points are joined together to form one big cluster.

Once one big cluster is formed, the longest vertical distance without any horizontal line passing through it is selected and a horizontal line is drawn through it. The number of vertical lines this newly created horizontal line passes is equal to number of clusters. Take a look at the following plot:



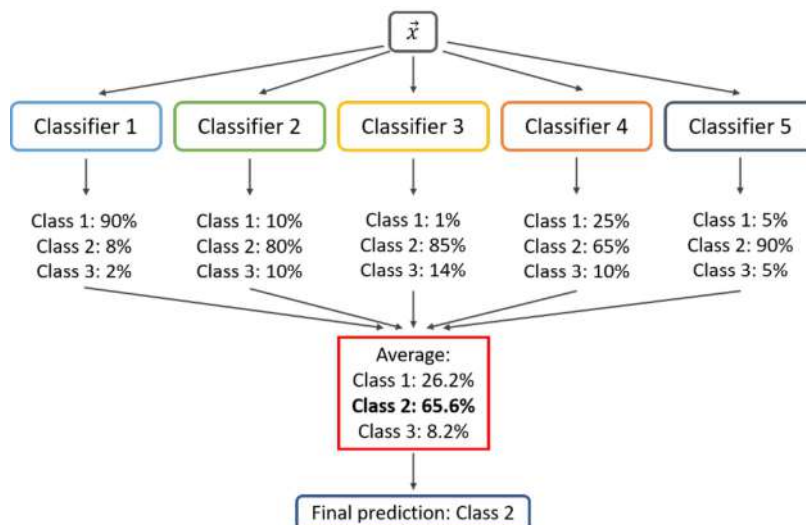
We can see that the largest vertical distance without any horizontal line passing through it is represented by blue line. So we draw a new horizontal red line that passes through the blue line. Since it crosses the blue line at two points, therefore the number of clusters will be 2. Basically the horizontal line is a threshold, which defines the minimum distance required to be a separate cluster. If we draw a line further down, the threshold required to be a new cluster will be decreased and more clusters will be formed as see in the image below:



In the above plot, the horizontal line passes through four vertical lines resulting in four clusters: cluster of points 6,7,8 and 10, cluster of points 3,2,4 and points 9 and 5 will be treated as single point clusters.

Q24. What is Ensemble Learning?

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Ensembles are a divide-and-conquer approach used to improve performance. The main principle behind ensemble methods is that a group of “weak learners” can come together to form a “strong



learner". Each classifier, individually, is a "weak learner," while all the classifiers taken together are a "strong learner".

Q25. Describe in brief any type of Ensemble Learning.

<https://medium.com/@ruhi3929/bagging-and-boosting-method-c036236376eb>

Ensemble learning has many types but two more popular ensemble learning techniques are mentioned below.

Bagging

Bagging tries to implement similar learners on small sample populations and then takes a mean of all the predictions. In generalized bagging, you can use different learners on different population. As you expect this helps us to reduce the variance error.

Pros

- Bagging method helps when we face variance or overfitting in the model. It provides an environment to deal with variance by using N learners of same size on same algorithm.
- During the sampling of train data, there are many observations which overlaps. So, the combination of these learners helps in overcoming the high variance.
- Bagging uses Bootstrap sampling method (Bootstrapping is any test or metric that uses random sampling with replacement and falls under the broader class of resampling methods.)

Cons

- Bagging is not helpful in case of bias or underfitting in the data.
- Bagging ignores the value with the highest and the lowest result which may have a wide difference and provides an average result.

Boosting

Boosting is an iterative technique which adjusts the weight of an observation based on the last classification. If an observation was classified incorrectly, it tries to increase the weight of this observation and vice versa. Boosting in general decreases the bias error and builds strong predictive models. However, they may over fit on the training data.

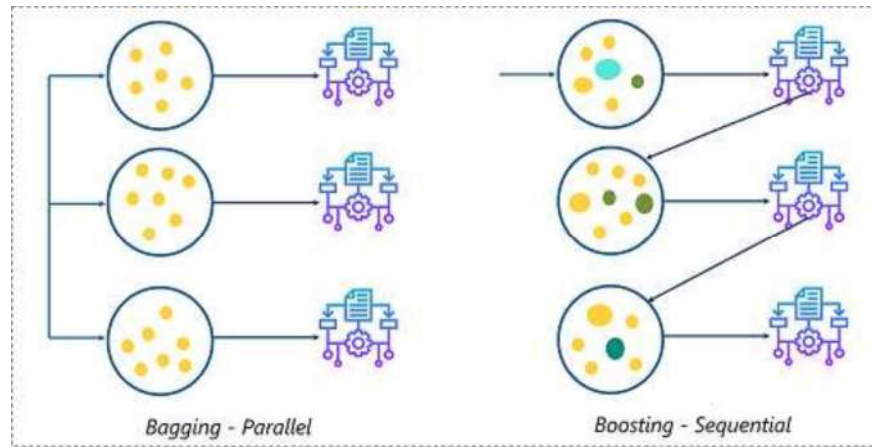
Pros

- Boosting technique takes care of the weightage of the higher accuracy sample and lower accuracy sample and then gives the combined results.
- Net error is evaluated in each learning steps. It works good with interactions.
- Boosting technique helps when we are dealing with bias or underfitting in the data set.
- Multiple boosting techniques are available. *For example: AdaBoost, LPBoost, XGBoost, GradientBoost, BrownBoost*

Cons

- Boosting technique often ignores overfitting or variance issues in the data set.

- It increases the complexity of the classification.
- Time and computation can be a bit expensive.



There are multiple areas where Bagging and Boosting technique is used to boost the accuracy.

- Banking: Loan defaulter prediction, fraud transaction
- Credit risks
- Kaggle competitions
- Fraud detection
- Recommender system for Netflix
- Malware
- Wildlife conservations and so on.

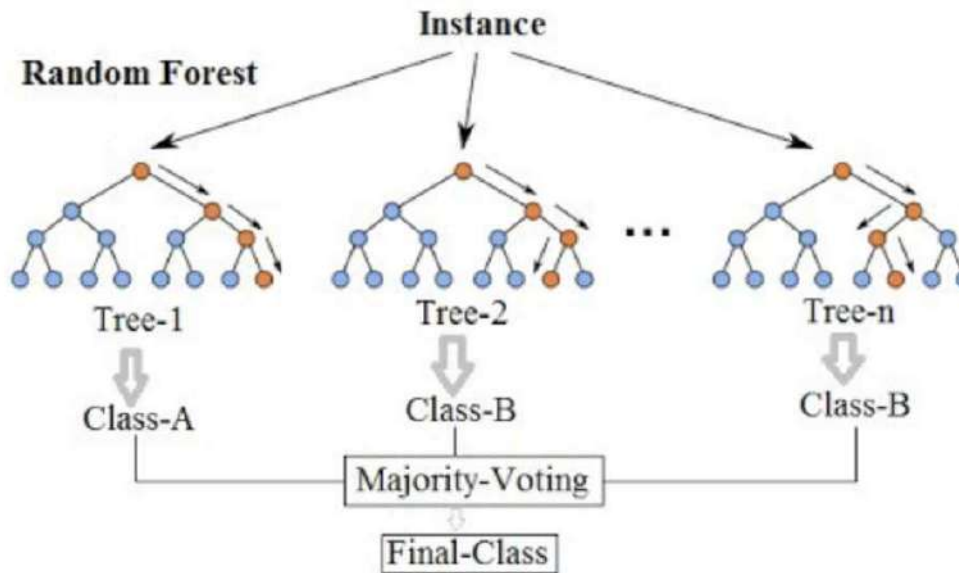
Q26. What is a Random Forest? How does it work?

Random forest is a versatile machine learning method capable of performing:

- regression
- classification
- dimensionality reduction
- treat missing values
- outlier values

It is a type of ensemble learning method, where a group of weak models combine to form a powerful model. The random forest starts with a standard machine learning technique called a “decision tree” which, in ensemble terms, corresponds to our weak learner. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets.

Random Forest Simplified



In Random Forest, we grow multiple trees as opposed to a single tree. To classify a new object based on attributes, each tree gives a classification. The forest chooses the classification having the most votes (Over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

Q27. How Do You Work Towards a Random Forest?

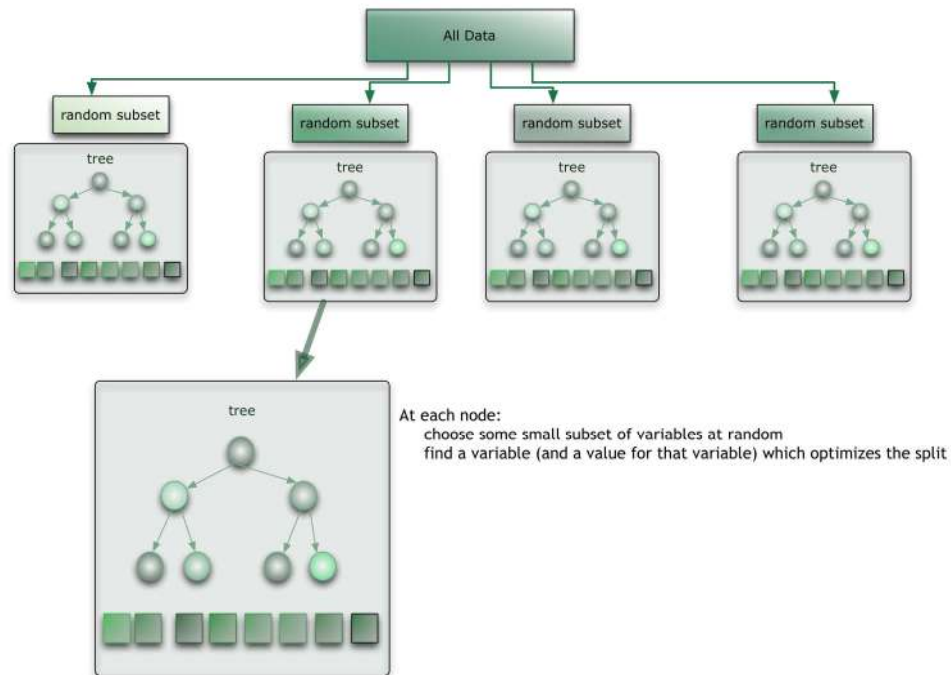
<https://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics>

The underlying principle of this technique is that several weak learners combined to provide a keen learner. Here is how such a system is trained for some number of trees T :

1. Sample N cases at random with replacement to create a subset of the data. The subset should be about 66% of the total set.
2. At each node:
 - a. For some number m (see below), m predictor variables are selected at random from all the predictor variables.
 - b. The predictor variable that provides the best split, according to some objective function, is used to do a binary split on that node.
 - c. At the next node, choose another m variables at random from all predictor variables and do the same.

Depending upon the value of m , there are three slightly different systems:

- Random splitter selection: $m = 1$
- Breiman's bagger: $m = \text{total number of predictor variables } (p)$
- Random forest: $m \ll \text{number of predictor variables}$.
 - Brieman suggests three possible values for m : $\frac{1}{2}\sqrt{p}$, \sqrt{p} , $2\sqrt{p}$



When a new input is entered into the system, it is run down all of the trees. The result may either be an average or weighted average of all of the terminal nodes that are reached, or, in the case of categorical variables, a voting majority.

Note that:

- With a large number of predictors ($p \gg 0$), the eligible predictor set (m) will be quite different from node to node.
- The greater the inter-tree correlation, the greater the random forest error rate, so one pressure on the model is to have the trees as uncorrelated as possible.
- As m goes down, both inter-tree correlation and the strength of individual trees go down. So some optimal value of m must be discovered.
- Strengths: Random forest runtimes are quite fast, and they are able to deal with unbalanced and missing data.
- Weaknesses: Random Forest used for regression cannot predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy. Of course, the best test of any algorithm is how well it works upon your own data set.

Q28. What cross-validation technique would you use on a time series data set?

Instead of using k-fold cross-validation, you should be aware of the fact that a time series is not randomly distributed data — It is inherently ordered by chronological order.

In case of time series data, you should use techniques like **forward-chaining** — Where you will be model on past data then look at forward-facing data.

fold 1: training[1], test[2]

fold 2: training[1 2], test[3]

fold 3: training[1 2 3], test[4]

fold 4: training[1 2 3 4], test[5]

Q29. What is a Box-Cox Transformation?

The dependent variable for a regression analysis might not satisfy one or more assumptions of an ordinary least squares regression. **The residuals could either curve as the prediction increases or follow the skewed distribution.** In such scenarios, it is necessary to transform the response variable so that the data meets the required assumptions. **A Box-Cox transformation is a statistical technique to transform non-normal dependent variables into a normal shape.** If the given data is not normal then most of the statistical techniques assume normality. Applying a Box-Cox transformation means that you can run a broader number of tests.

A Box-Cox transformation is a way to transform non-normal dependent variables into a normal shape. Normality is an important assumption for many statistical techniques, if your data isn't normal, applying a Box-Cox means that you are able to run a broader number of tests. The Box-Cox transformation is named after statisticians George Box and Sir David Roxbee Cox who collaborated on a 1964 paper and developed the technique.

Q30. How Regularly Must an Algorithm be Updated?

You will want to update an algorithm when:

- You want the model to evolve as data streams through infrastructure
- The underlying data source is changing
- There is a case of non-stationarity (mean, variance change over the time)
- The algorithm underperforms/results lack accuracy

Q31. If you are having 4GB RAM in your machine and you want to train your model on 10GB data set. How would you go about this problem? Have you ever faced this kind of problem in your machine learning/data science experience so far?

First of all, you have to ask which ML model you want to train.

For Neural networks: Batch size with Numpy array will work. Steps:

1. Load the whole data in the Numpy array. Numpy array has a property to create a mapping of the complete data set, it doesn't load complete data set in memory.
2. You can pass an index to Numpy array to get required data.
3. Use this data to pass to the Neural network.
4. Have a small batch size.

For SVM: Partial fit will work. Steps:

1. Divide one big data set in small size data sets.

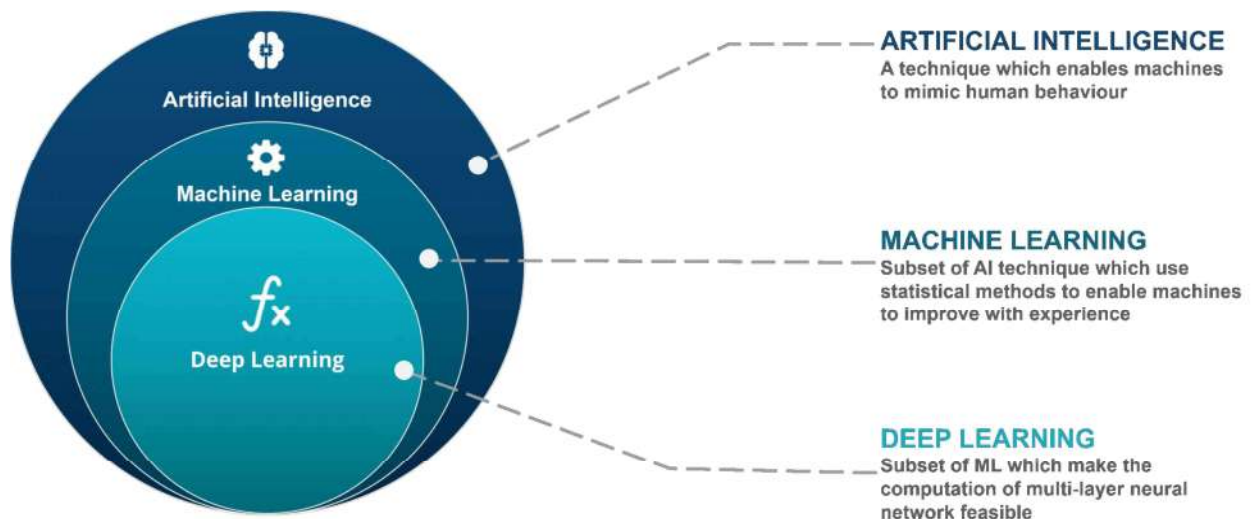
2. Use a partial fit method of SVM, it requires a subset of the complete data set.
3. Repeat step 2 for other subsets.

However, you could actually face such an issue in reality. So, you could check out the best laptop for Machine Learning to prevent that. Having said that, let's move on to some questions on deep learning.

Deep Learning

Q1. What do you mean by Deep Learning?

Deep Learning is nothing but a paradigm of machine learning which has shown incredible promise in recent years. This is because of the fact that Deep Learning shows a great analogy with the functioning of the neurons in the human brain.



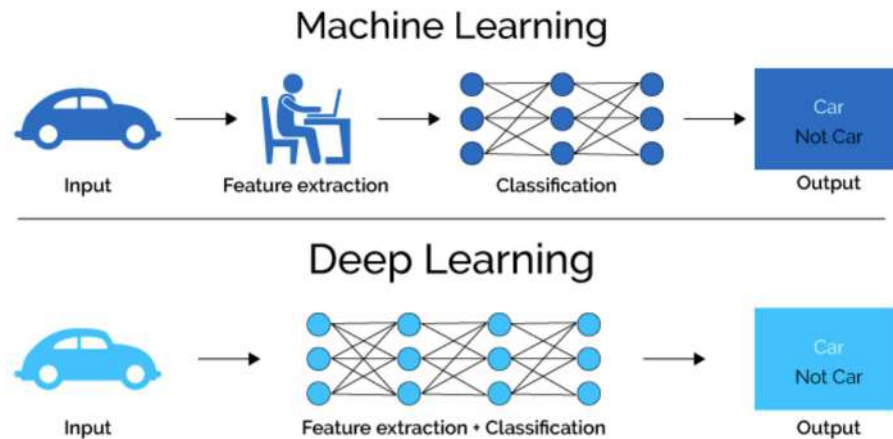
Q2. What is the difference between machine learning and deep learning?

<https://parsers.me/deep-learning-machine-learning-whats-the-difference/>

Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed. Machine learning can be categorized in the following four categories.

1. Supervised machine learning,
2. Semi-supervised machine learning,
3. Unsupervised machine learning,
4. Reinforcement learning.

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.



- The main difference between deep learning and machine learning is due to the way data is presented in the system. Machine learning algorithms almost always require structured data, while deep learning networks rely on layers of ANN (artificial neural networks).
- Machine learning algorithms are designed to “learn” to act by understanding labeled data and then use it to produce new results with more datasets. However, when the result is incorrect, there is a need to “teach them”. Because machine learning algorithms require bulleted data, they are not suitable for solving complex queries that involve a huge amount of data.
- Deep learning networks do not require human intervention, as multilevel layers in neural networks place data in a hierarchy of different concepts, which ultimately learn from their own mistakes. However, even they can be wrong if the data quality is not good enough.
- Data decides everything. It is the quality of the data that ultimately determines the quality of the result.
- Both of these subsets of AI are somehow connected to data, which makes it possible to represent a certain form of “intelligence.” However, you should be aware that deep learning requires much more data than a traditional machine learning algorithm. The reason for this is that deep learning networks can identify different elements in neural network layers only when more than a million data points interact. Machine learning algorithms, on the other hand, are capable of learning by pre-programmed criteria.

Q3. What, in your opinion, is the reason for the popularity of Deep Learning in recent times?

Now although Deep Learning has been around for many years, the major breakthroughs from these techniques came just in recent years. This is because of two main reasons:

- The increase in the amount of data generated through various sources
- The growth in hardware resources required to run these models

GPUs are multiple times faster and they help us build bigger and deeper deep learning models in comparatively less time than we required previously.

Q4. What is reinforcement learning?

Reinforcement Learning allows to take actions to max cumulative reward. It learns by trial and error through reward/penalty system. Environment rewards agent so by time agent makes better decisions. Ex: robot=agent, maze=environment. Used for complex tasks (self-driving cars, game AI).

RL is a series of time steps in a Markov Decision Process:

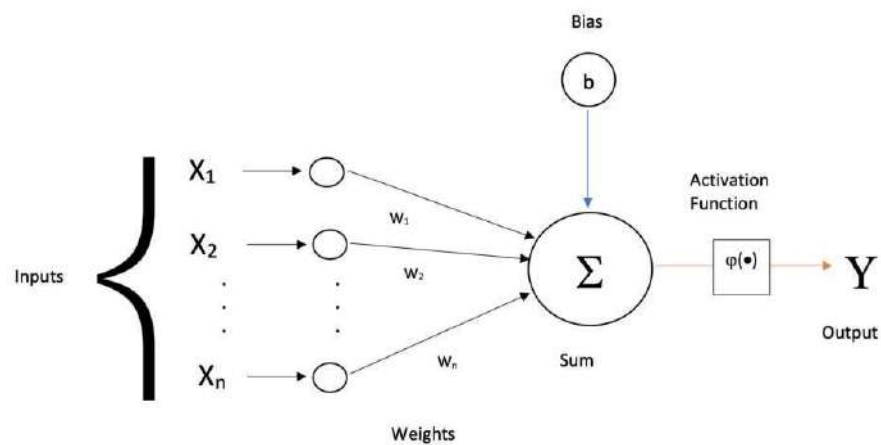
1. Environment: space in which RL operates
2. State: data related to past action RL took
3. Action: action taken
4. Reward: number taken by agent after last action
5. Observation: data related to environment: can be visible or partially shadowed

Q5. What are Artificial Neural Networks?

Artificial Neural networks are a specific set of algorithms that have revolutionized machine learning. They are inspired by biological neural networks. Neural Networks can adapt to changing the input, so the network generates the best possible result without needing to redesign the output criteria.

Q6. Describe the structure of Artificial Neural Networks?

Artificial Neural Networks works on the same principle as a biological Neural Network. It consists of inputs which get processed with weighted sums and Bias, with the help of Activation Functions.



Q7. How Are Weights Initialized in a Network?

There are two methods here: we can either initialize the weights to zero or assign them randomly.

Initializing all weights to 0: This makes your model similar to a linear model. All the neurons and every layer perform the same operation, giving the same output and making the deep net useless.

Initializing all weights randomly: Here, the weights are assigned randomly by initializing them very close to 0. It gives better accuracy to the model since every neuron performs different computations. This is the most commonly used method.

Q8. What Is the Cost Function?

Also referred to as “loss” or “error,” cost function is a measure to evaluate how good your model’s performance is. It’s used to compute the error of the output layer during backpropagation. We push that error backwards through the neural network and use that during the different training functions.

The most known one is the mean sum of squared errors.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

test set predicted value actual value

$$\hat{y}_i = \phi(\sum(w_i x_i) + b)$$

Q9. What Are Hyperparameters?

With neural networks, you’re usually working with hyperparameters once the data is formatted correctly. A hyperparameter is a parameter whose value is set before the learning process begins. It determines how a network is trained and the structure of the network (such as the number of hidden units, the learning rate, epochs, batches, etc.).

Q10. What Will Happen If the Learning Rate Is Set inaccurately (Too Low or Too High)?

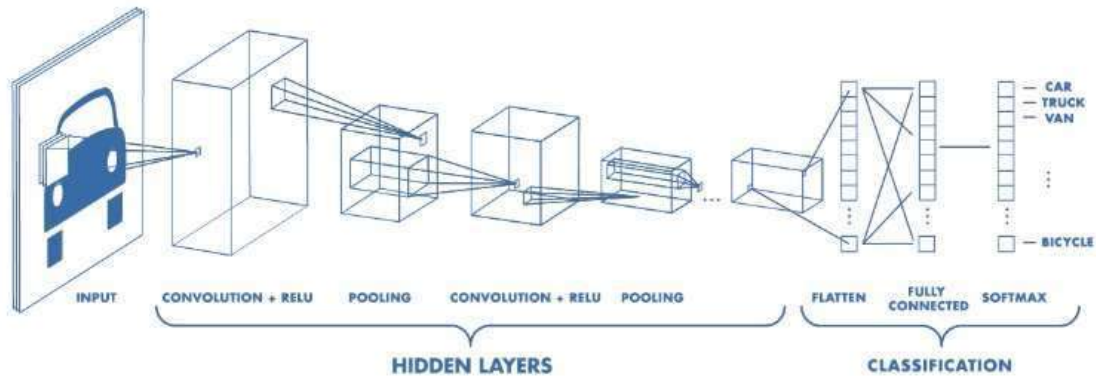
When your learning rate is too low, training of the model will progress very slowly as we are making minimal updates to the weights. It will take many updates before reaching the minimum point. If the learning rate is set too high, this causes undesirable divergent behavior to the loss function due to drastic updates in weights. It may fail to converge (model can give a good output) or even diverge (data is too chaotic for the network to train).

Q11. What Is The Difference Between Epoch, Batch, and Iteration in Deep Learning?

- Epoch – Represents one iteration over the entire dataset (everything put into the training model).
- Batch – Refers to when we cannot pass the entire dataset into the neural network at once, so we divide the dataset into several batches.
- Iteration – if we have 10,000 images as data and a batch size of 200. then an epoch should run 50 iterations (10,000 divided by 200).

Q12. What Are the Different Layers on CNN?

<https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add>



The Convolutional neural networks are regularized versions of multilayer perceptron (MLP). They were developed based on the working of the neurons of the animal visual cortex.



Let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point. RGB intensity values of the image are visualized by the computer for processing.

The objective of using the CNN:

The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (.80 for a cat, .15 for a dog, .05 for a bird, etc.). It works similar to how our brain works. When we look at a picture of a dog, we can classify it as such if the picture has identifiable features such as paws or 4 legs. In a similar way, the computer is able to perform image classification by looking for low-level features such as edges and curves and then building up to more abstract concepts through a series of convolutional layers. The computer uses low-level features obtained at the initial levels to generate high-level features such as paws or eyes to identify the object.

There are four layers in CNN:

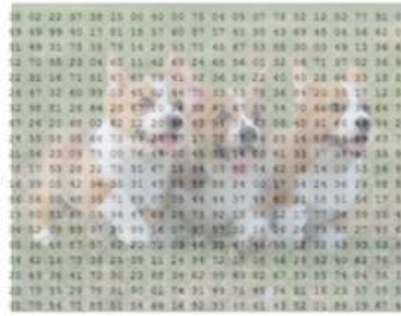
1. **Convolutional Layer** – the layer that performs a convolutional operation, creating several smaller picture windows to go over the data.
2. **Activation Layer (ReLU Layer)** – it brings non-linearity to the network and converts all the negative pixels to zero. The output is a rectified feature map. It follows each convolutional layer.
3. **Pooling Layer** – pooling is a down-sampling operation that reduces the dimensionality of the feature map. Stride = how much you slide, and you get the max of the $n \times n$ matrix
4. **Fully Connected Layer** – this layer recognizes and classifies the objects in the image.

Convolution Operation

First Layer:

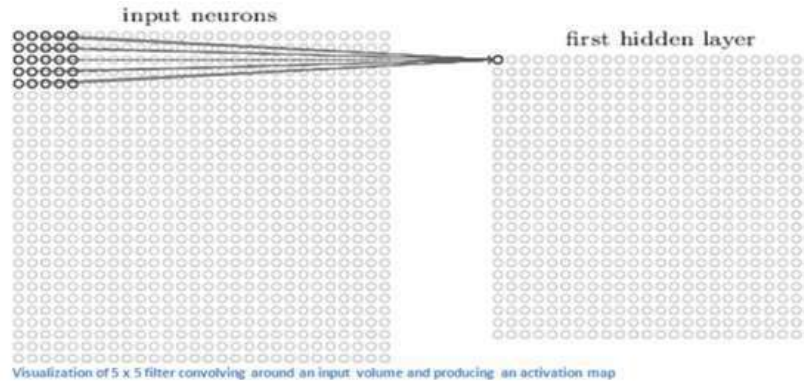
1. Input to a convolutional layer

*The image is resized to an optimal size and is fed as input to the convolutional layer.
Let us consider the input as 32x32x3 array of pixel values.*



2. There exists a filter or neuron or kernel which lays over some of the pixels of the input image depending on the dimensions of the Kernel size.

Let the dimensions of the kernel of the filter be 5x5x3.



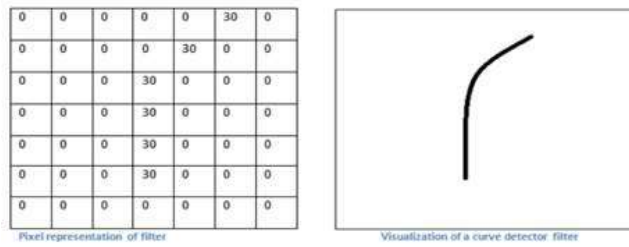
3. The Kernel actually slides over the input image; thus, it is multiplying the values in the filter with the original pixel values of the image (aka computing element-wise multiplications).

The multiplications are summed up generating a single number for that particular receptive field and hence for sliding the kernel a total of 784 numbers are mapped to 28x28 array known as the feature map.

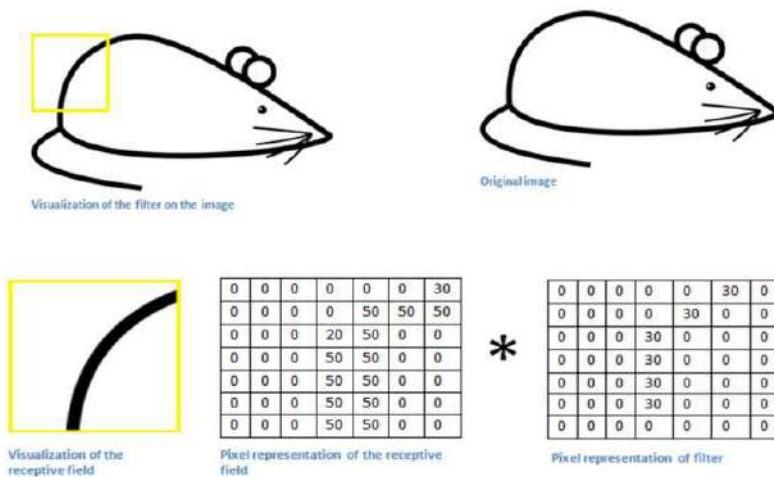
***Now if we consider two kernels of the same dimension then the obtained first layer feature map will be (28x28x2).*

High-level Perspective

- Let us take a kernel of size (7x7x3) for understanding. Each of the kernels is considered to be a feature identifier, hence say that our filter will be a curve detector.



- The original image and the visualization of the kernel on the image.



The sum of the multiplication value that is generated is = $4 * (50 * 30) + (20 * 30) = 6600$ (large number).

- Now when the kernel moves to the other part of the image.



The sum of the multiplication value that is generated is = 0 (small number).

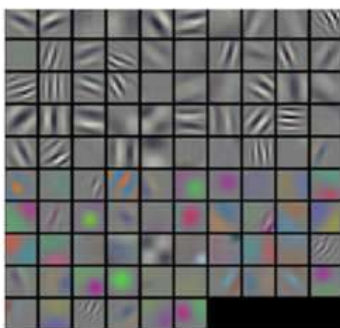
The use of the small and the large value

- The value is much lower! This is because there wasn't anything in the image section that responded to the curve detector filter. Remember, the output of this convolution layer is an activation map. So, in the simple case of a one filter convolution (and if that filter is a curve detector), the activation map will show the areas in which there at most likely to be curved in the picture.

2. In the previous example, the top-left value of our $26 \times 26 \times 1$ activation map (26 because of the 7×7 filter instead of 5×5) will be 6600. This high value means that it is likely that there is some sort of curve in the input volume that caused the filter to activate. The top right value in our activation map will be 0 because there wasn't anything in the input volume that caused the filter to activate. This is just for one filter.

3. This is just a filter that is going to detect lines that curve outward and to the right. We can have other filters for lines that curve to the left or for straight edges. The more filters, the greater the depth of the activation map, and the more information we have about the input volume.

In the picture, we can see some examples of actual visualizations of the filters of the first conv. layer of a trained network. Nonetheless, the main argument remains the same. The filters on the first layer convolve around the input image and “activate” (or compute high values) when the specific feature it is looking for is in the input volume.



Visualizations of filters

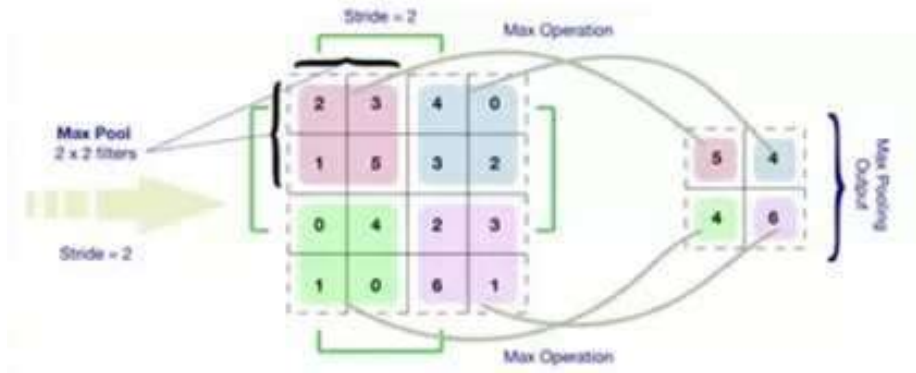
Sequential convolutional layers after the first one

1. When we go through another conv. layer, the output of the first conv. layer becomes the input of the 2nd conv. layer.
2. However, when we're talking about the 2nd conv. layer, the input is the activation map(s) that result from the first layer. So, each layer of the input is basically describing the locations in the original image for where certain low-level features appear.
3. Now when you apply a set of filters on top of that (pass it through the 2nd conv. layer), the output will be activations that represent higher-level features. Types of these features could be semicircles (a combination of a curve and straight edge) or squares (a combination of several straight edges). As you go through the network and go through more convolutional layers, you get activation maps that represent more and more complex features.
4. By the end of the network, you may have some filters that activate when there is handwriting in the image, filters that activate when they see pink objects, etc.

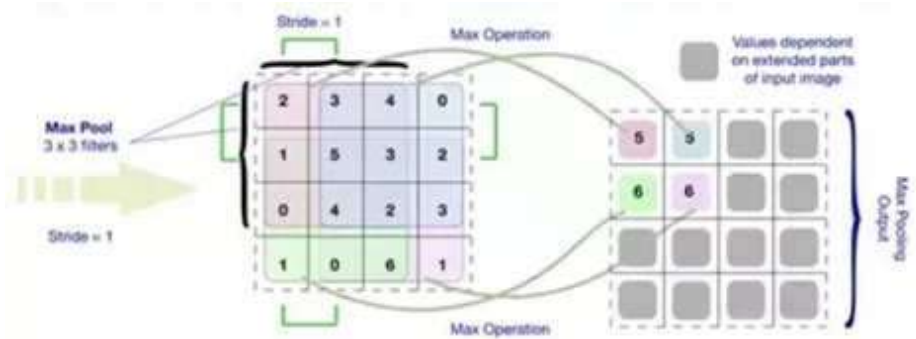
Pooling Operation

It consists in getting the largest number out of a matrix to get the most important number and reduce the dimension.

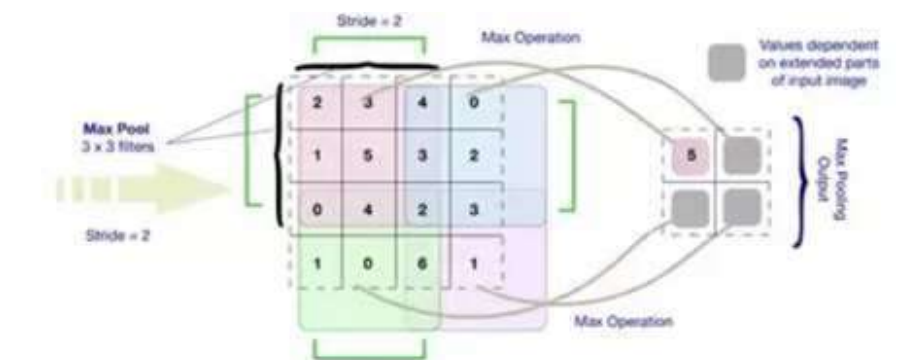
Max Pooling example



2x2 filters with stride = 2 (maximum value) is considered



3x3 filters with stride = 1 (maximum value) is considered



3x3 filters with stride = 2 (maximum value) is considered

Classification

- 1. Flatten:** The pooled matrix is converted to a vector.

2. **Fully Connected layer:** The way this fully connected layer works is that it looks at the output of the previous layer (which as we remember should represent the activation maps of high-level features) and the number of classes p (10 for digit classification). *For example, if the program is predicting that some image is a dog, it will have high values in the activation maps that represent high-level features like a paw or 4 legs, etc. Basically, an FC layer looks at what high level features most strongly correlate to a particular class and has particular weights so that when you compute the products between the weights and the previous layer, you get the correct probabilities for the different classes.*
3. **Soft-max approach:** The output of a fully connected layer is as follows [0 .1 .1 .75 0 0 0 0 .05], then this represents a 10% probability that the image is a 1, a 10% probability that the image is a 2, a 75% probability that the image is a 3, and a 5% probability that the image is a 9 (SoftMax approach) for digit classification.

Training

§We know kernels also known as feature identifiers, used for identification of specific features. But how the kernels are initialized with the specific weights or how do the filters know what values to have.

Hence comes the important step of training. The training process is also known as backpropagation, which is further separated into 4 distinct sections or processes.

- Forward Pass
- Loss Function
- Backward Pass
- Weight Update

The Forward Pass

For the first epoch or iteration of the training the initial kernels of the first convolutional layer are initialized with random values. Thus, after the first iteration output will be something like [1.1.1.1.1.1.1.1.1.1], which does not give preference to any class as the kernels don't have specific weights.

The Loss Function

The training involves images along with labels, hence the label for the digit 3 will be [0 0 0 1 0 0 0 0 0], whereas the output after a first epoch is very different, hence we will calculate loss (MSE — Mean Squared Error)

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

The objective is to minimize the loss, which is an optimization problem in calculus. It involves trying to adjust the weights to reduce the loss.

The Backward Pass

It involves determining which weights contributed most to the loss and finding ways to adjust them so that the loss decreases. It is computed using $\frac{\partial L}{\partial w}$ (or $\nabla_w L$), where L is the loss and the W is the weights of the corresponding kernel.

The weights update

This is where the weights of the kernel are updated using the following equation.

$$w = w_i - \eta \frac{dL}{dW}$$

w = Weight
w _i = Initial Weight
η = Learning Rate

Here the Learning Rate is chosen by the programmer. Larger value of the learning rate indicates much larger steps towards optimization of steps and larger time to convolve to an optimized weight.

Testing

Finally, to see whether or not our CNN works, we have a different set of images and labels (can't double dip between training and test!) and pass the images through the CNN. We compare the outputs to the ground truth and see if our network works!

Q13. What Is Pooling on CNN, and How Does It Work?

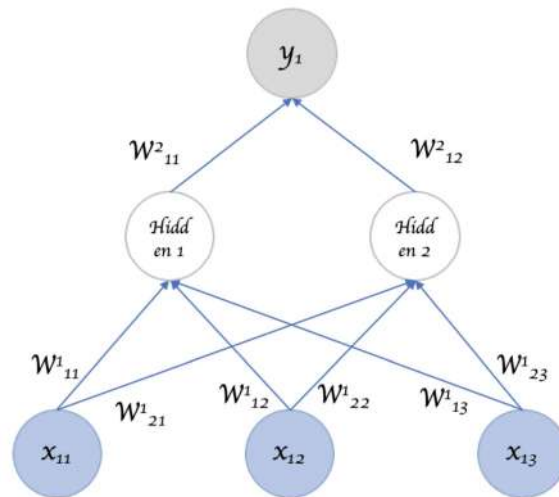
Pooling is used to reduce the spatial dimensions of a CNN. It performs down-sampling operations to reduce the dimensionality and creates a pooled feature map by sliding a filter matrix over the input matrix.

Q14. What are Recurrent Neural Networks (RNNs)?

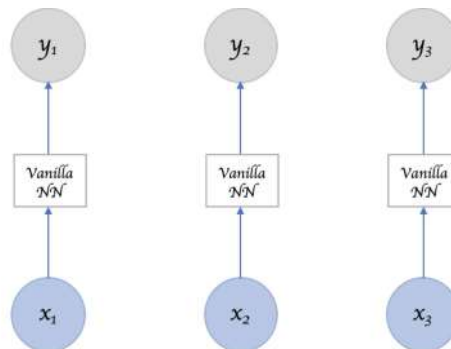
<https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce>

RNNs are a type of artificial neural networks designed to recognize the pattern from the sequence of data such as Time series, stock market and government agencies etc.

Recurrent Neural Networks (RNNs) add an interesting twist to basic neural networks. A vanilla neural network takes in a fixed size vector as input which limits its usage in situations that involve a 'series' type input with no predetermined size.



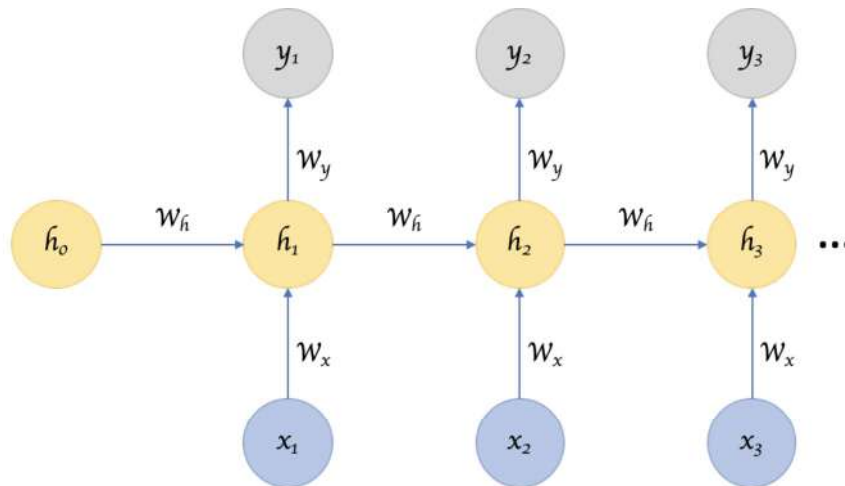
RNNs are designed to take a series of input with no predetermined limit on size. One could ask what's the big deal, I can call a regular NN repeatedly too?



Sure can, but the 'series' part of the input means something. A single input item from the series is related to others and likely has an influence on its neighbors. Otherwise it's just "many" inputs, not a "series" input (duh!).

Recurrent Neural Network remembers the past and its decisions are influenced by what it has learnt from the past. Note: Basic feed forward networks "remember" things too, but they remember things they learnt during training. For example, an image classifier learns what a "1" looks like during training and then uses that knowledge to classify things in production.

While RNNs learn similarly while training, in addition, they remember things learnt from prior input(s) while generating output(s). RNNs can take one or more input vectors and produce one or more output vectors and the output(s) are influenced not just by weights applied on inputs like a regular NN, but also by a "hidden" state vector representing the context based on prior input(s)/output(s). So, the same input could produce a different output depending on previous inputs in the series.



In summary, in a vanilla neural network, a fixed size input vector is transformed into a fixed size output vector. Such a network becomes “recurrent” when you repeatedly apply the transformations to a series of given input and produce a series of output vectors. There is no pre-set limitation to the size of the vector. And, in addition to generating the output which is a function of the input and hidden state, we update the hidden state itself based on the input and use it in processing the next input.

Parameter Sharing

You might have noticed another key difference between Figure 1 and Figure 3. In the earlier, multiple different weights are applied to the different parts of an input item generating a hidden layer neuron, which in turn is transformed using further weights to produce an output. There seems to be a lot of weights in play here. Whereas in Figure 3, we seem to be applying the same weights over and over again to different items in the input series.

I am sure you are quick to point out that we are kind of comparing apples and oranges here. The first figure deals with “a” single input whereas the second figure represents multiple inputs from a series. But nevertheless, intuitively speaking, as the number of inputs increase, shouldn’t the number of weights in play increase as well? Are we losing some versatility and depth in Figure 3?

Perhaps we are. We are sharing parameters across inputs in Figure 3. If we don’t share parameters across inputs, then it becomes like a vanilla neural network where each input node requires weights of their own. This introduces the constraint that the length of the input has to be fixed and that makes it impossible to leverage a series type input where the lengths differ and is not always known.

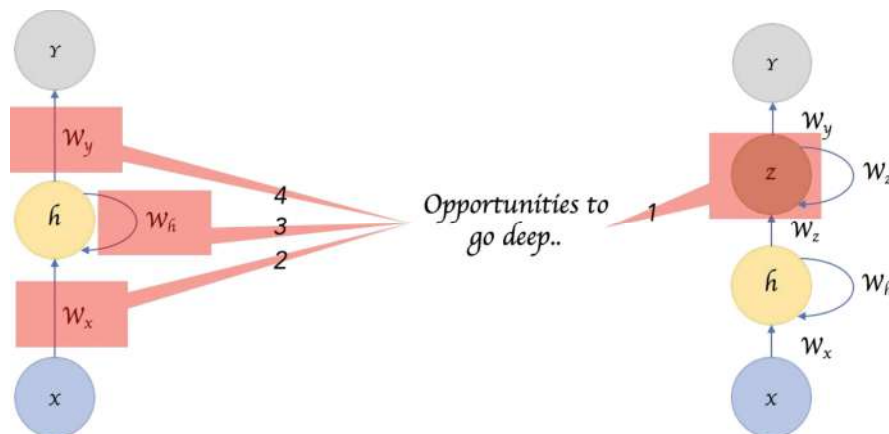
But what we seemingly lose in value here, we gain back by introducing the “hidden state” that links one input to the next. The hidden state captures the relationship that neighbors might have with each other in a serial input and it keeps changing in every step, and thus effectively every input undergoes a different transition!

Image classifying CNNs have become so successful because the 2D convolutions are an effective form of parameter sharing where each convolutional filter basically extracts the presence or absence of a feature in an image which is a function of not just one pixel but also of its surrounding neighbor pixels.

In other words, the success of CNNs and RNNs can be attributed to the concept of “parameter sharing” which is fundamentally an effective way of leveraging the relationship between one input item and its surrounding neighbors in a more intrinsic fashion compared to a vanilla neural network.

Deep RNNs

While it’s good that the introduction of hidden state enabled us to effectively identify the relationship between the inputs, is there a way we can make an RNN “deep” and gain the multi-level abstractions and representations we gain through “depth” in a typical neural network?

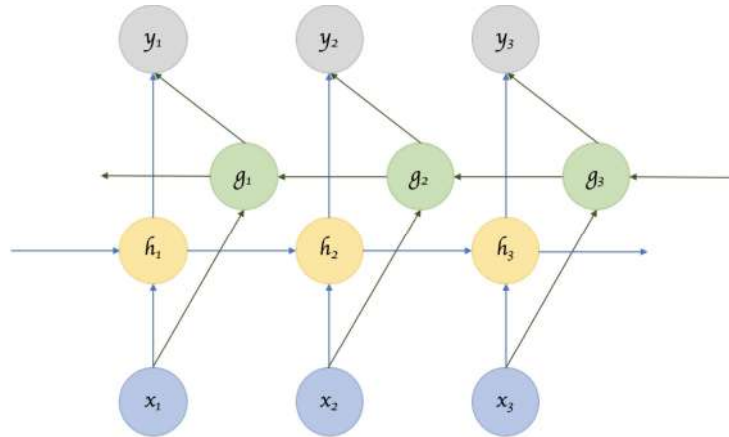


Here are four possible ways to add depth.

- 1) We can add hidden states, one on top of another, feeding the output of one to the next.
- 2) We can also add additional nonlinear hidden layers between input to hidden state.
- 3) We can increase depth in the hidden to hidden transition.
- 4) We can increase depth in the hidden to output transition.

Bidirectional RNNs

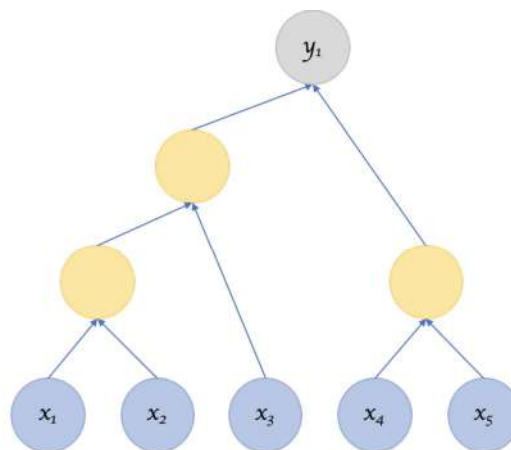
Sometimes it’s not just about learning from the past to predict the future, but we also need to look into the future to fix the past. In speech recognition and handwriting recognition tasks, where there could be considerable ambiguity given just one part of the input, we often need to know what’s coming next to better understand the context and detect the present.



This does introduce the obvious challenge of how much into the future we need to look into, because if we have to wait to see all inputs then the entire operation will become costly. And in cases like speech recognition, waiting till an entire sentence is spoken might make for a less compelling use case. Whereas for NLP tasks, where the inputs tend to be available, we can likely consider entire sentences all at once. Also, depending on the application, if the sensitivity to immediate and closer neighbors is higher than inputs that come further away, a variant that looks only into a limited future/past can be modeled.

Recursive Neural Network

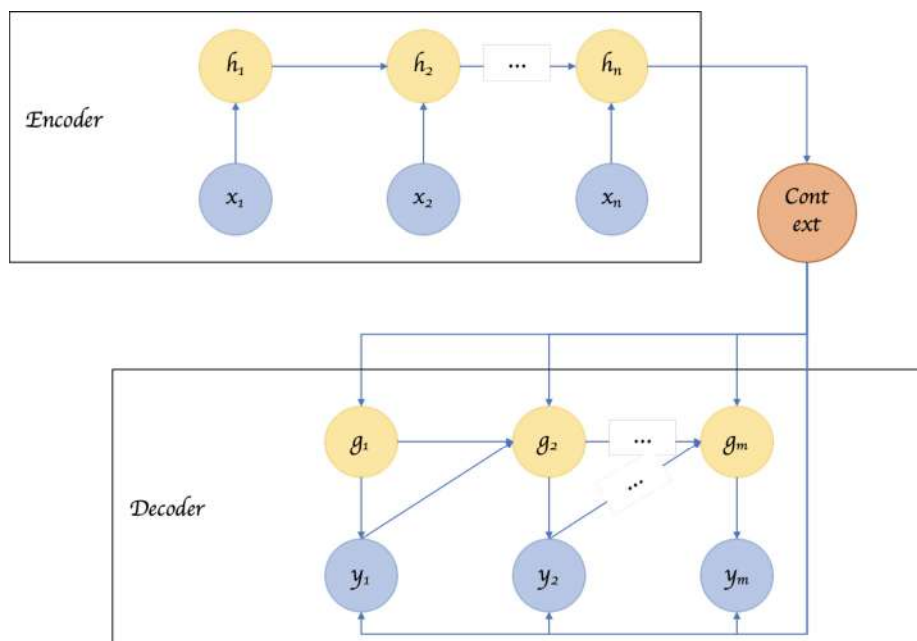
A recurrent neural network parses the inputs in a sequential fashion. A recursive neural network is similar to the extent that the transitions are repeatedly applied to inputs, but not necessarily in a sequential fashion. Recursive Neural Networks are a more general form of Recurrent Neural Networks. It can operate on any hierarchical tree structure. Parsing through input nodes, combining child nodes into parent nodes and combining them with other child/parent nodes to create a tree like structure. Recurrent Neural Networks do the same, but the structure there is strictly linear. i.e. weights are applied on the first input node, then the second, third and so on.



But this raises questions pertaining to the structure. How do we decide that? If the structure is fixed like in Recurrent Neural Networks then the process of training, backprop, makes sense in that they are similar to a regular neural network. But if the structure isn't fixed, is that learnt as well?

Encoder Decoder Sequence to Sequence RNNs

Encoder Decoder or Sequence to Sequence RNNs are used a lot in translation services. The basic idea is that there are two RNNs, one an encoder that keeps updating its hidden state and produces a final single "Context" output. This is then fed to the decoder, which translates this context to a sequence of outputs. Another key difference in this arrangement is that the length of the input sequence and the length of the output sequence need not necessarily be the same.



LSTMs

LSTM is not a different variant of RNN architecture, but rather it introduces changes to how we compute outputs and hidden state using the inputs.

In a vanilla RNN, the input and the hidden state are simply passed through a single tanh layer. LSTM (Long-Short-Term Memory) networks improve on this simple transformation and introduces additional gates and a cell state, such that it fundamentally addresses the problem of keeping or resetting context, across sentences and regardless of the distance between such context resets. There are variants of LSTMs including GRUs that utilize the gates in different manners to address the problem of long-term dependencies.

Q15. How Does an LSTM Network Work?

<http://karpathy.github.io/2015/05/21/rnn-effectiveness>
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Long-Short-Term Memory (LSTM) is a special kind of recurrent neural network capable of learning long-term dependencies, remembering information for long periods as its default behavior. There are three steps in an LSTM network:

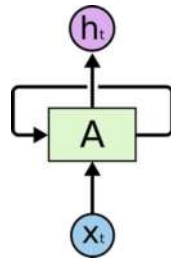
- Step 1: The network decides what to forget and what to remember.
- Step 2: It selectively updates cell state values.
- Step 3: The network decides what part of the current state makes it to the output.

Recurrent Neural Networks

Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence.

Traditional neural networks can't do this, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It's unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

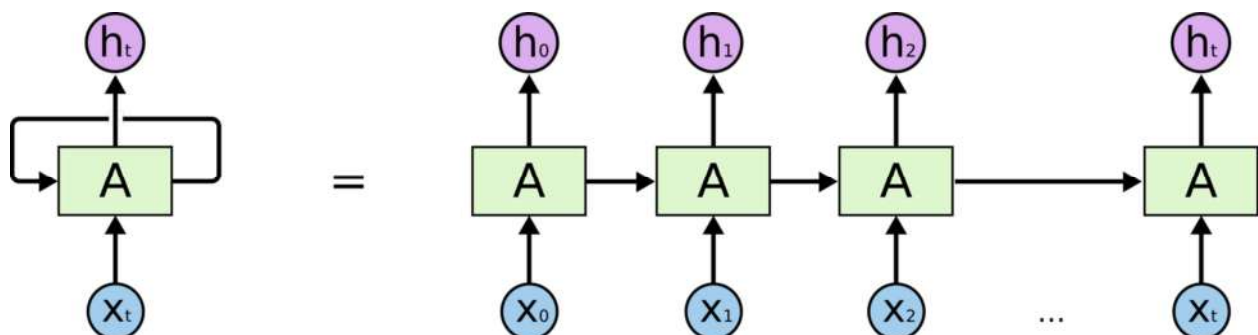
Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.



Recurrent Neural Networks have loops.

In the above diagram, a chunk of neural network, A, looks at some input x_t and outputs a value h_t . A loop allows information to be passed from one step of the network to the next.

These loops make recurrent neural networks seem kind of mysterious. However, if you think a bit more, it turns out that they aren't all that different than a normal neural network. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if we unroll the loop:



An unrolled recurrent neural network.

This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists. They're the natural architecture of neural network to use for such data.

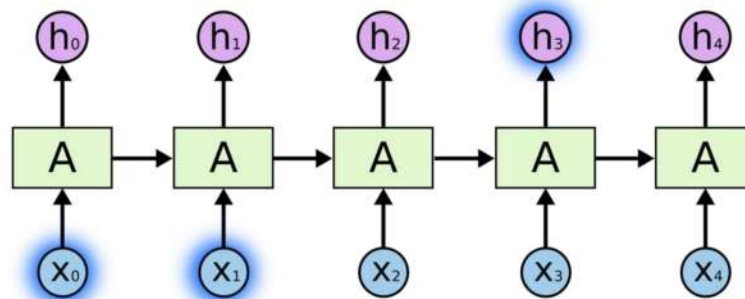
And they certainly are used! In the last few years, there have been incredible success applying RNNs to a variety of problems: speech recognition, language modeling, translation, image captioning...

Essential to these successes is the use of "LSTMs," a very special kind of recurrent neural network which works, for many tasks, much better than the standard version. Almost all exciting results based on recurrent neural networks are achieved with them.

The Problem of Long-Term Dependencies

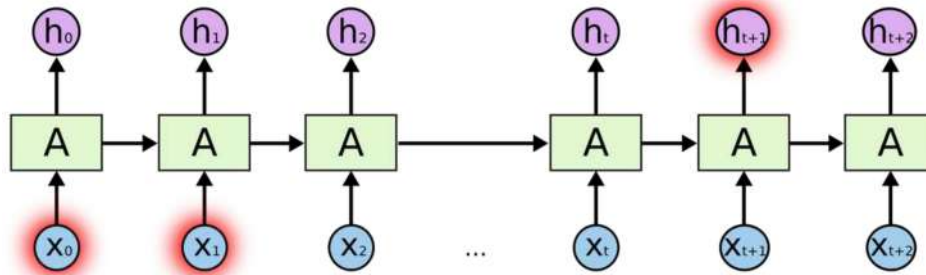
One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame. If RNNs could do this, they'd be extremely useful. But can they? It depends.

Sometimes, we only need to look at recent information to perform the present task. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in "the clouds are in the sky," we don't need any further context – it's pretty obvious the next word is going to be sky. In such cases, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information.



But there are also cases where we need more context. Consider trying to predict the last word in the text "I grew up in France... I speak fluent French." Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of France, from further back. It's entirely possible for the gap between the relevant information and the point where it is needed to become very large.

Unfortunately, as that gap grows, RNNs become unable to learn to connect the information.



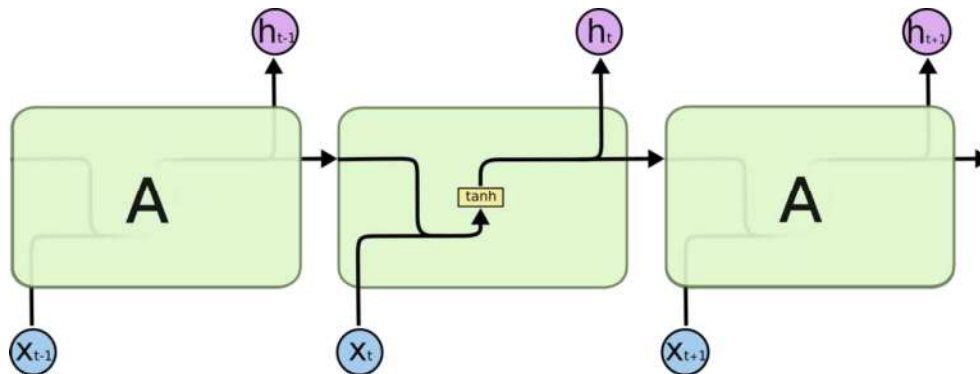
In theory, RNNs are absolutely capable of handling such “long-term dependencies.” A human could carefully pick parameters for them to solve toy problems of this form. Sadly, in practice, RNNs don’t seem to be able to learn them. Thankfully, LSTMs don’t have this problem!

LSTM Networks

Long Short-Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems and are now widely used.

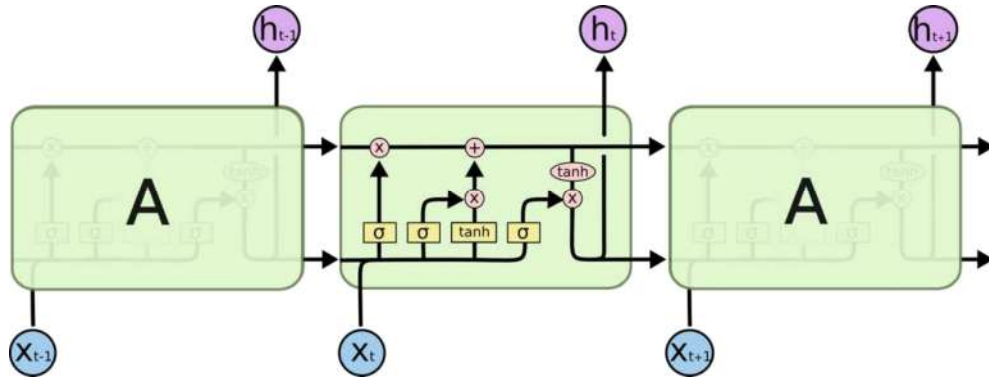
LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

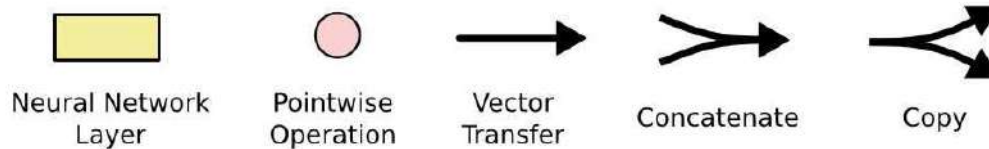


The repeating module in a standard RNN contains a single layer.

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



The repeating module in an LSTM contains four interacting layers.

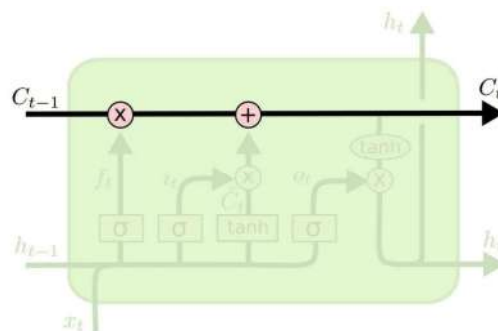


In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

The Core Idea Behind LSTMs

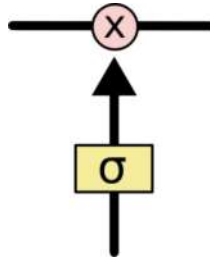
The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.

The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.



The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.



The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

An LSTM has three of these gates, to protect and control the cell state.

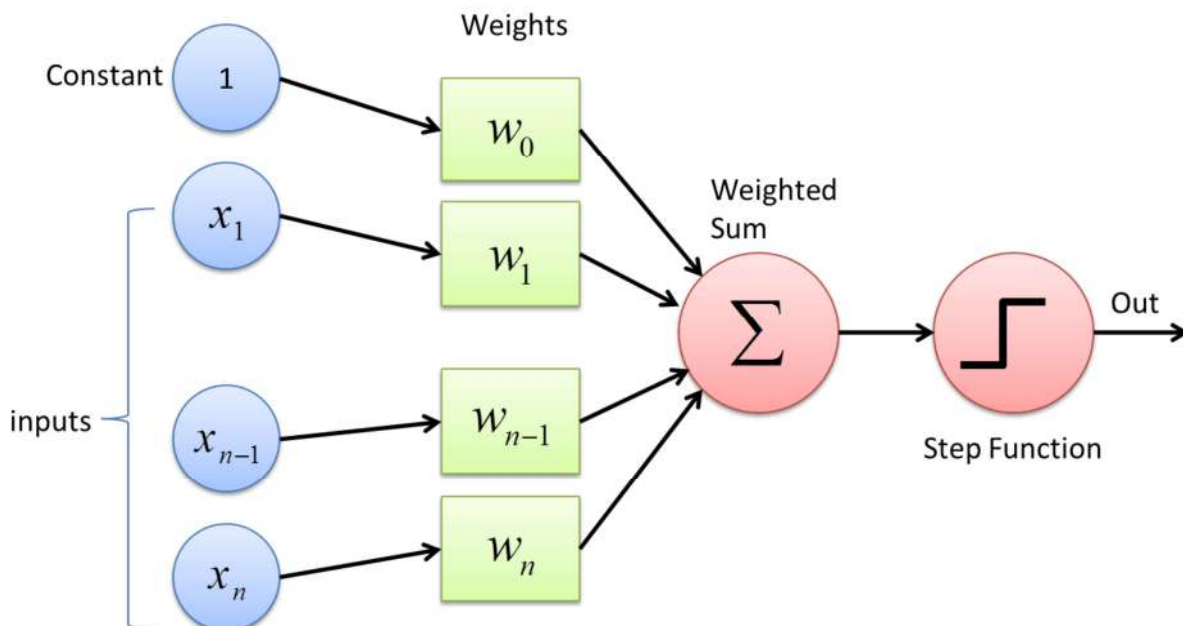
Q16. What Is a Multi-layer Perceptron (MLP)?

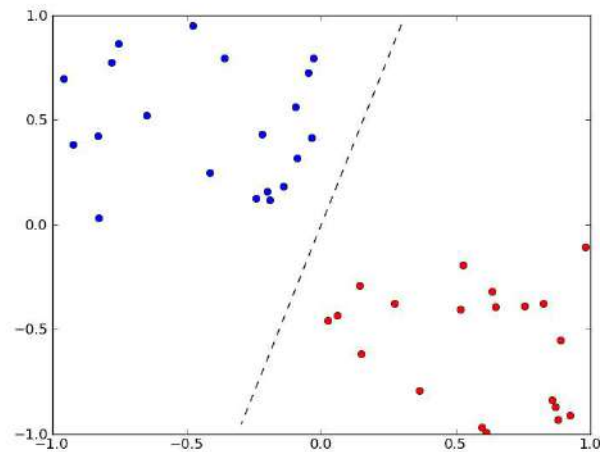
<https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>

As in Neural Networks, MLPs have an input layer, a hidden layer, and an output layer. It has the same structure as a single layer perceptron with one or more hidden layers.

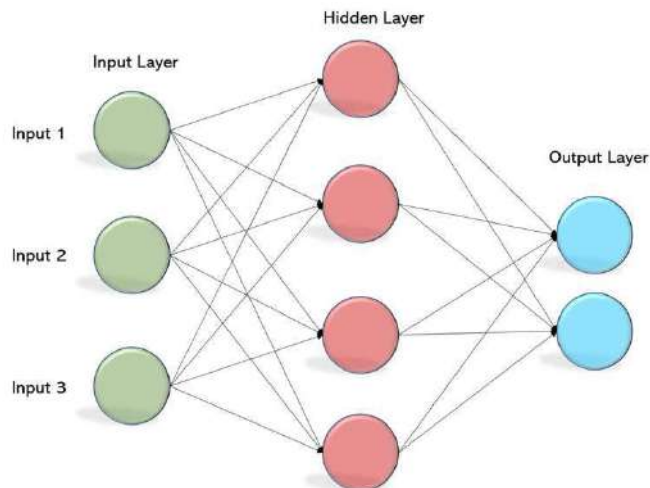
Perceptron is a single layer neural network and a multi-layer perceptron is called Neural Networks.

A (single layer) perceptron is a single layer neural network that works as a linear binary classifier. Being a single layer neural network, it can be trained without the use of more advanced algorithms like back propagation and instead can be trained by "stepping towards" your error in steps specified by a learning rate. When someone says perceptron, I usually think of the single layer version.





A single layer perceptron can classify only linear separable classes with binary output $\{0,1\}$ or $\{-1,1\}$, but MLP can classify nonlinear classes. The activation functions are used to map the input between the required values like $\{0, 1\}$ or $\{-1, 1\}$.



Except for the input layer, each node in the other layers uses a nonlinear activation function. This means the input layers, the data coming in, and the activation function is based upon all nodes and weights being added together, producing the output. MLP uses a supervised learning method called "backpropagation." In backpropagation, the neural network calculates the error with the help of cost function. It propagates this error backward from where it came (adjusts the weights to train the model more accurately).

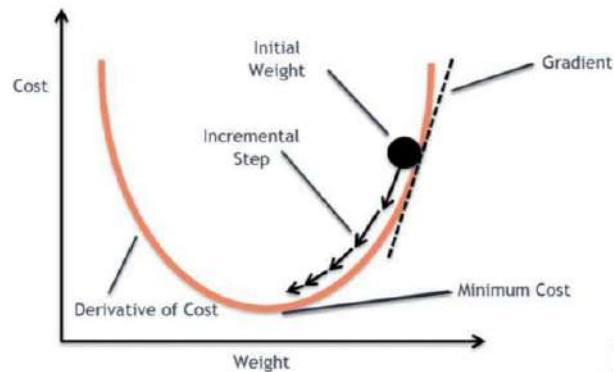
Usually, RELU is in hidden layers (it does not classify), and Soft-max or tanh is in output layers.

Q17. Explain Gradient Descent.

Let's first explain what a gradient is. A gradient is a mathematical function. When calculated on a point of a function, it gives the hyperplane (or slope) of the directions in which the function increases more. The gradient vector can be interpreted as the "direction and rate of fastest increase". If the gradient of a

function is non-zero at a point p , the direction of the gradient is the direction in which the function increases most quickly from p , and the **magnitude** of the gradient is the rate of increase in that direction. Further, the gradient is the zero vector at a point if and only if it is a **stationary point** (where the derivative vanishes).

In DS, it simply measures the change in all weights with regard to the change in error, as we are partially deriving by w the loss function.



Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function.

The goal of the gradient descent is to minimize a given function which, in our case, is the loss function of the neural network. To achieve this goal, it performs two steps iteratively.

1. Compute the slope (gradient) that is the first-order derivative of the function at the current point
2. Move-in the opposite direction of the slope increase from the current point by the computed amount

So, the idea is to pass the training set through the hidden layers of the neural network and then update the parameters of the layers by computing the gradients using the training samples from the training dataset.

Think of it like this. Suppose a man is at top of the valley and he wants to get to the bottom of the valley. So, he goes down the slope. He decides his next position based on his current position and stops when he gets to the bottom of the valley which was his goal.

Q18. What is exploding gradients?

<https://machinelearningmastery.com/exploding-gradients-in-neural-networks/>

While training an RNN, if you see exponentially growing (very large) error gradients which accumulate and result in very large updates to neural network model weights during training, they're known as exploding gradients. At an extreme, the values of weights can become so large as to overflow and result in NaN values. The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1.0.

This has the effect of your model is unstable and unable to learn from your training data.

There are some subtle signs that you may be suffering from exploding gradients during the training of your network, such as:

- The model is unable to get traction on your training data (e.g. poor loss).
- The model is unstable, resulting in large changes in loss from update to update.
- The model loss goes to NaN during training.
- The model weights quickly become very large during training.
- The error gradient values are consistently above 1.0 for each node and layer during training.

Solutions

1. Re-Design the Network Model:

- a. In deep neural networks, exploding gradients may be addressed by redesigning the network to have fewer layers. There may also be some benefit in using a [smaller batch size](#) while training the network.
- b. In RNNs, updating across fewer prior time steps during training, called [truncated Backpropagation through time](#), may reduce the exploding gradient problem.

2. Use Long Short-Term Memory Networks: In RNNs, exploding gradients can be reduced by using the [Long Short-Term Memory \(LSTM\)](#) memory units and perhaps related gated-type neuron structures. Adopting LSTM memory units is a new best practice for recurrent neural networks for sequence prediction.

3. Use Gradient Clipping: Exploding gradients can still occur in very deep Multilayer Perceptron networks with a large batch size and LSTMs with very long input sequence lengths. If exploding gradients are still occurring, you can check for and limit the size of gradients during the training of your network. This is called [gradient clipping](#). Specifically, the values of the error gradient are checked against a threshold value and clipped or set to that threshold value if the error gradient exceeds the threshold.

4. Use Weight Regularization: another approach, if exploding gradients are still occurring, is to check the size of network weights and apply a penalty to the networks [loss function](#) for large weight values. This is called weight regularization and often an L1 (absolute weights) or an L2 (squared weights) penalty can be used.

Q19. What is vanishing gradients?

While training an RNN, your slope can become either too small; this makes the training difficult. When the slope is too small, the problem is known as a Vanishing Gradient. It leads to long training times, poor performance, and low accuracy.

- Hyperbolic tangent and Sigmoid/Soft-max suffer vanishing gradient.
- RNNs suffer vanishing gradient, LSTM no (so it is perfect to predict stock prices). In fact, the propagation of error through previous layers makes the gradient get smaller so the weights are not updated.

Solutions

1. **Choose RELU**
2. **Use LSTM (for RNNs)**
3. **Use ResNet (Residual Network)** → after some layers, add x again: $F(x) \rightarrow \dots \rightarrow F(x) + x$
4. **Multi-level hierarchy:** pre-train one layer at the time through unsupervised learning, then fine-tune via backpropagation
5. **Gradient checking:** debugging strategy used to numerically track and assess gradients during training.

Q20. What is Back Propagation and Explain it Works.

Backpropagation is a training algorithm used for neural network. In this method, we update the weights of each layer from the last layer recursively, with the formula:

$$w_{previous\ layer} = w_{layer} - \eta \nabla_w L(w)$$

It has the following steps:

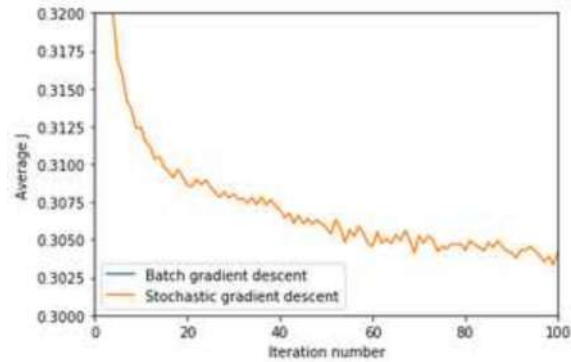
- Forward Propagation of Training Data (initializing weights with random or pre-assigned values)
- Gradients are computed using output weights and target
- Back Propagate for computing gradients of error from output activation
- Update the Weights

Q21. What are the variants of Back Propagation?

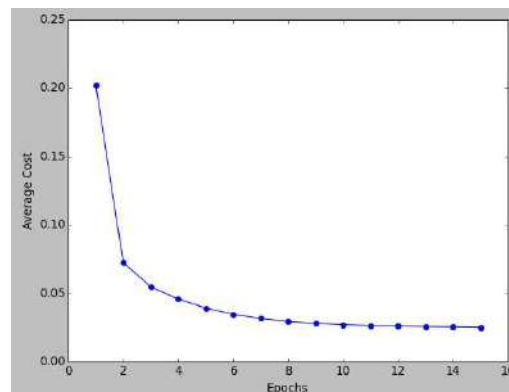
<https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a>

- **Stochastic Gradient Descent:** In Batch Gradient Descent we were considering all the examples for every step of Gradient Descent. But what if our dataset is very huge. Deep learning models crave for data. The more the data the more chances of a model to be good. *Suppose our dataset has 5 million examples, then just to take one step the model will have to calculate the gradients of all the 5 million examples. This does not seem an efficient way.* To tackle this problem, we have Stochastic Gradient Descent. **In Stochastic Gradient Descent (SGD), we consider just one example at a time to take a single step.** We do the following steps in **one epoch** for SGD:
 1. Take an example
 2. Feed it to Neural Network
 3. Calculate its gradient
 4. Use the gradient we calculated in step 3 to update the weights
 5. Repeat steps 1–4 for all the examples in training dataset

Since we are considering just one example at a time the cost will fluctuate over the training examples and it will **not** necessarily decrease. But in the long run, you will see the cost decreasing with fluctuations. Also, because the cost is so fluctuating, it will never reach the minimum, but it will keep dancing around it. SGD can be used for larger datasets. It converges faster when the dataset is large as it causes updates to the parameters more frequently.



- Batch Gradient Descent:** all the training data is taken into consideration to take a single step. We take the average of the gradients of all the training examples and then use that mean gradient to update our parameters. So that's just one step of gradient descent in one epoch. Batch Gradient Descent is great for convex or relatively smooth error manifolds. In this case, we move somewhat directly towards an optimum solution. The graph of cost vs epochs is also quite smooth because we are averaging over all the gradients of training data for a single step. The cost keeps on decreasing over the epochs.



- Mini-batch Gradient Descent:** It's one of the most popular optimization algorithms. It's a variant of Stochastic Gradient Descent and here instead of single training example, mini batch of samples is used. Batch Gradient Descent can be used for smoother curves. SGD can be used when the dataset is large. Batch Gradient Descent converges directly to minima. SGD converges faster for larger datasets. But, since in SGD we use only one example at a time, we cannot implement the vectorized implementation on it. This can slow down the computations. To tackle this problem, a mixture of Batch Gradient Descent and SGD is used. Neither we use all the dataset all at once nor we use the single example at a time. We use a batch of a fixed number of training examples which is less than the actual dataset and call it a mini-batch. Doing this helps us achieve the advantages of both the former variants we saw. So, after creating the mini-batches of fixed size, we do the following steps in **one epoch**:
 1. Pick a mini-batch
 2. Feed it to Neural Network
 3. Calculate the mean gradient of the mini-batch
 4. Use the mean gradient we calculated in step 3 to update the weights
 5. Repeat steps 1–4 for the mini-batches we created

Just like SGD, the average cost over the epochs in mini-batch gradient descent fluctuates because we are averaging a small number of examples at a time. So, when we are using the mini-batch gradient descent we are updating our parameters frequently as well as we can use vectorized implementation for faster computations.

Q22. What are the different Deep Learning Frameworks?

- **PyTorch:** PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab. It is free and open-source software released under the Modified BSD license.
- **TensorFlow:** TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. Licensed by Apache License 2.0. Developed by Google Brain Team.
- **Microsoft Cognitive Toolkit:** Microsoft Cognitive Toolkit describes neural networks as a series of computational steps via a directed graph.
- **Keras:** Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Licensed by MIT.

Q23. What is the role of the Activation Function?

The Activation function is used to introduce non-linearity into the neural network helping it to learn more complex function. Without which the neural network would be only able to learn linear function which is a linear combination of its input data. An activation function is a function in an artificial neuron that delivers an output based on inputs.

Q24. Name a few Machine Learning libraries for various purposes.

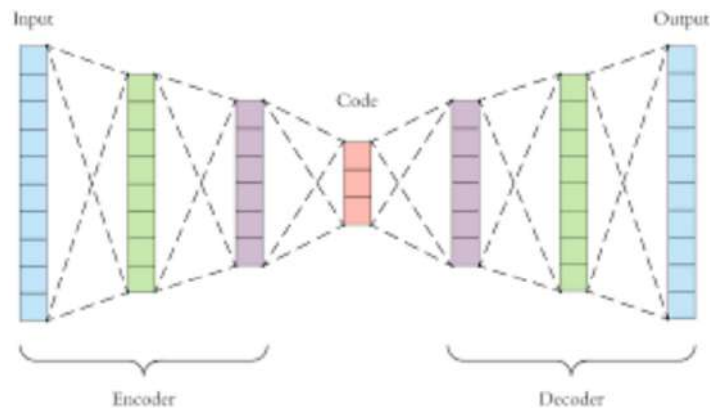
Purpose	Libraries
Scientific Computation	Numpy
Tabular Data	Pandas, GeoPandas
Data Modelling & Preprocessing	Scikit Learn
Time-Series Analysis	Statsmodels
Text processing	NLTK, Regular Expressions
Deep Learning	TensorFlow, Pytorch
Visualization	Bokeh, Seaborn
Plotting	Matplot

Q25. What is an Auto-Encoder?

<https://www.quora.com/What-is-an-autoencoder-What-are-its-applications>

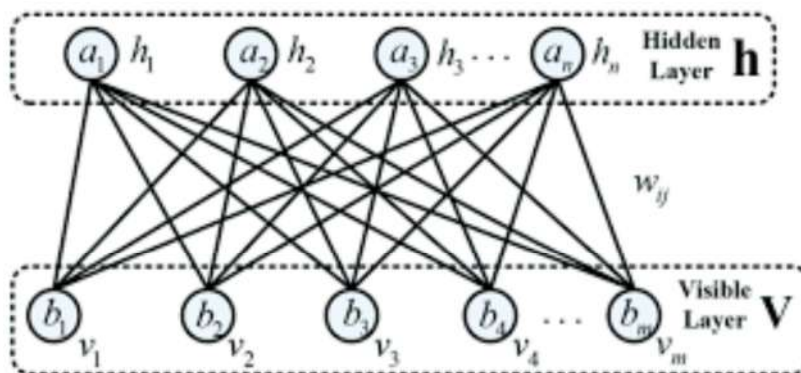
Auto-encoders are simple learning networks that aim to transform inputs into outputs with the minimum possible error. This means that we want the output to be as close to input as possible. We add a couple of layers between the input and the output, and the sizes of these layers are smaller than the input layer. The auto-encoder receives unlabeled input which is then encoded to reconstruct the input.

An **autoencoder** is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. The aim of an **autoencoder** is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise”. Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input, hence its name. Several variants exist to the basic model, with the aim of forcing the learned representations of the input to assume useful properties. Autoencoders are effectively used for solving many applied problems, from [face recognition](#) to acquiring the semantic meaning of words.



Q26. What is a Boltzmann Machine?

Boltzmann machines have a simple learning algorithm that allows them to discover interesting features that represent complex regularities in the training data. The Boltzmann machine is basically used to optimize the weights and the quantity for the given problem. The learning algorithm is very slow in networks with many layers of feature detectors. “Restricted Boltzmann Machines” algorithm has a single layer of feature detectors which makes it faster than the rest.



Q27. What Is Dropout and Batch Normalization?

Dropout is a technique of dropping out hidden and visible nodes of a network randomly to prevent overfitting of data (typically dropping 20 per cent of the nodes). It doubles the number of iterations needed to converge the network. It used to avoid overfitting, as it increases the capacity of generalization.

Batch normalization is the technique to improve the performance and stability of neural networks by normalizing the inputs in every layer so that they have mean output activation of zero and standard deviation of one.

Q28. Why Is TensorFlow the Most Preferred Library in Deep Learning?

TensorFlow provides both C++ and Python APIs, making it easier to work on and has a faster compilation time compared to other Deep Learning libraries like Keras and PyTorch. TensorFlow supports both CPU and GPU computing devices.

Q29. What Do You Mean by Tensor in TensorFlow?

A tensor is a mathematical object represented as arrays of higher dimensions. Think of a n-D matrix. These arrays of data with different dimensions and ranks fed as input to the neural network are called "Tensors."

Q30. What is the Computational Graph?

Everything in a TensorFlow is based on creating a computational graph. It has a network of nodes where each node operates. Nodes represent mathematical operations, and edges represent tensors. Since data flows in the form of a graph, it is also called a "DataFlow Graph."

Q31. How is logistic regression done?

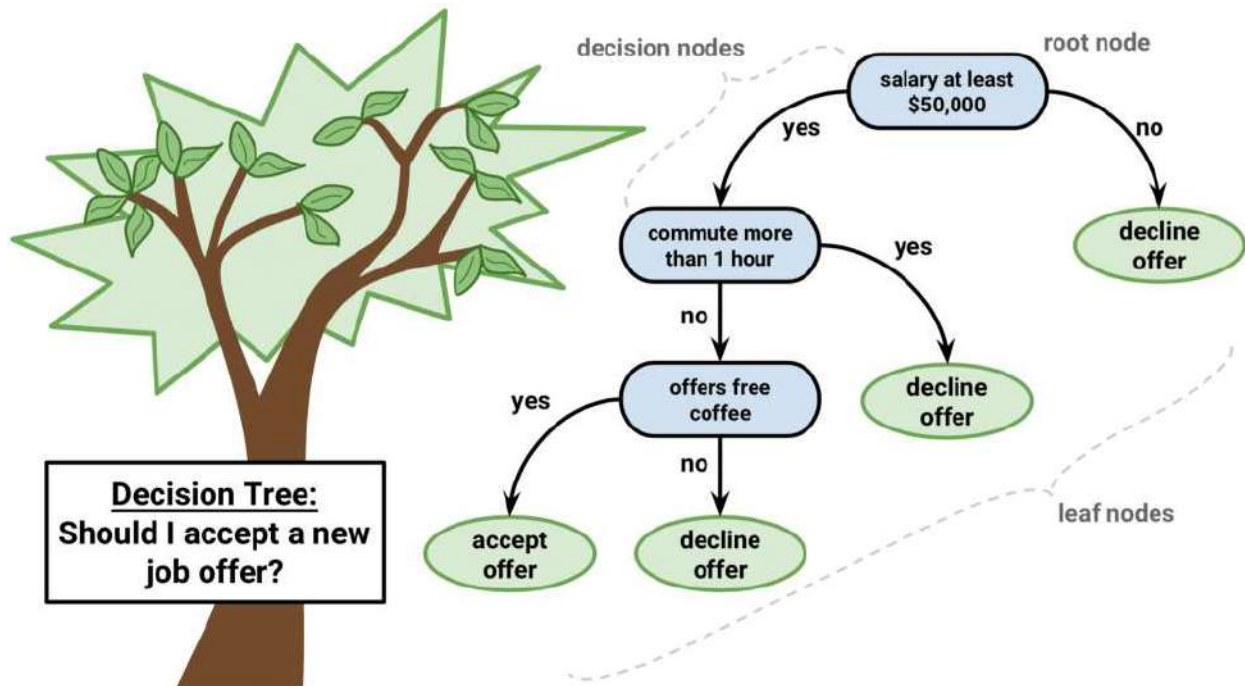
Logistic regression measures the relationship between the dependent variable (our label of what we want to predict) and one or more independent variables (our features) by estimating probability using its underlying logistic function (sigmoid).

Miscellaneous

Q1. Explain the steps in making a decision tree.

1. Take the entire data set as input
2. Calculate entropy of the target variable, as well as the predictor attributes
3. Calculate your information gain of all attributes (we gain information on sorting different objects from each other)
4. Choose the attribute with the highest information gain as the root node
5. Repeat the same procedure on every branch until the decision node of each branch is finalized

For example, let's say you want to build a decision tree to decide whether you should accept or decline a job offer. The decision tree for this case is as shown:



It is clear from the decision tree that an offer is accepted if:

- Salary is greater than \$50,000
- The commute is less than an hour
- Coffee is offered

Q2. How do you build a random forest model?

A random forest is built up of a number of decision trees. If you split the data into different packages and make a decision tree in each of the different groups of data, the random forest brings all those trees together.

Steps to build a random forest model:

1. Randomly select k features from a total of m features where $k \ll m$
2. Among the k features, calculate the node D using the best split point
3. Split the node into daughter nodes using the best split
4. Repeat steps two and three until leaf nodes are finalized
5. Build forest by repeating steps one to four for n times to create n number of trees

Q3. Differentiate between univariate, bivariate, and multivariate analysis.

Univariate

Univariate data contains only one variable. The purpose of the univariate analysis is to describe the data and find patterns that exist within it.

Example: height of students

Height (in cm)
164
167.3
170
174.2
178
180

The patterns can be studied by drawing conclusions using mean, median, mode, dispersion or range, minimum, maximum, etc.

Bivariate

Bivariate data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to determine the relationship between the two variables.

Example: temperature and ice cream sales in the summer season

Temperature (in Celsius)	Sales (in K \$)
20	2.0
25	2.1
26	2.3
28	2.7
30	3.1

Here, the relationship is visible from the table that temperature and sales are directly proportional to each other. The hotter the temperature, the better the sales.

Multivariate

Multivariate data involves three or more variables, it is categorized under multivariate. It is similar to a bivariate but contains more than one dependent variable.

Example: data for house price prediction

The patterns can be studied by drawing conclusions using mean, median, and mode, dispersion or range, minimum, maximum, etc. You can start describing the data and using it to guess what the price of the house will be.

Q4. What are the feature selection methods used to select the right variables?

There are two main methods for feature selection.

Filter Methods

This involves:

- Linear discrimination analysis
- ANOVA
- Chi-Square

The best analogy for selecting features is "bad data in, bad answer out." When we're limiting or selecting the features, it's all about cleaning up the data coming in.

Wrapper Methods

This involves:

- Forward Selection: We test one feature at a time and keep adding them until we get a good fit
- Backward Selection: We test all the features and start removing them to see what works better
- Recursive Feature Elimination: Recursively looks through all the different features and how they pair together

Wrapper methods are very labor-intensive, and high-end computers are needed if a lot of data analysis is performed with the wrapper method.

Q5. In your choice of language, write a program that prints the numbers ranging from one to 50. But for multiples of three, print "Fizz" instead of the number and for the multiples of five, print "Buzz." For numbers which are multiples of both three and five, print "FizzBuzz."

The code is shown below:

```

for x in range(51):
    if x % 3 == 0 and x % 5 == 0:
        print('fizzbuzz')

    elif x % 3 == 0:
        print('fizz')

    elif x % 5 == 0:
        print('buzz')

    else:
        print('fizzbuzz')

```

Q6. You are given a data set consisting of variables with more than 30 percent missing values. How will you deal with them?

If the data set is large, we can just simply remove the rows with missing data values. It is the quickest way; we use the rest of the data to predict the values.

For smaller data sets, we can impute missing values with the mean, median, or average of the rest of the data using pandas data frame in python. There are different ways to do so, such as:

```
df.mean(), df.fillna(mean)
```

Other option of imputation is using KNN for numeric or classification values (as KNN just uses k closest values to impute the missing value).

Q7. For the given points, how will you calculate the Euclidean distance in Python?

```
plot1 = [1,3]
plot2 = [2,5]
```

The Euclidean distance can be calculated as follows:

```
euclidean_distance = sqrt((plot1[0]-plot2[0])**2 + (plot1[1]-plot2[1])**2)
```

Q8. What are dimensionality reduction and its benefits?

Dimensionality reduction refers to the process of converting a data set with vast dimensions into data with fewer dimensions (fields) to convey similar information concisely.

This reduction helps in compressing data and reducing storage space. It also reduces computation time as fewer dimensions lead to less computing. It removes redundant features; for example, there's no point in storing a value in two different units (meters and inches).

Q9. How will you calculate eigenvalues and eigenvectors of the following 3x3 matrix?

Determinant of $A - \lambda I$ and solve to find λ .

Q10. How should you maintain a deployed model?

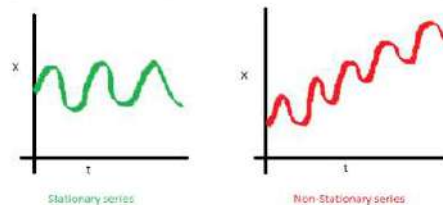
The steps to maintain a deployed model are (CREM):

- 1. Monitor:** constant monitoring of all models is needed to determine their performance accuracy. When you change something, you want to figure out how your changes are going to affect things. This needs to be monitored to ensure it's doing what it's supposed to do.
- 2. Evaluate:** evaluation metrics of the current model are calculated to determine if a new algorithm is needed.
- 3. Compare:** the new models are compared to each other to determine which model performs the best.
- 4. Rebuild:** the best performing model is re-built on the current state of data.

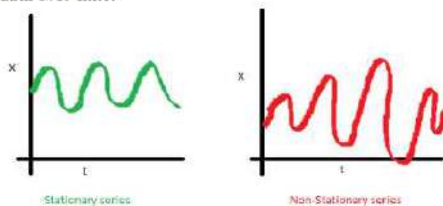
Q11. How can a time-series data be declared as stationary?

What does it mean for data to be stationary?

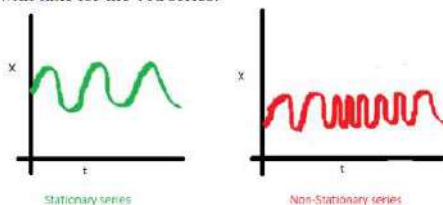
1. The mean of the series should not be a function of time. The red graph below is not stationary because the mean increases over time.



2. The variance of the series should not be a function of time. This property is known as homoscedasticity. Notice in the red graph the varying spread of data over time.



3. Finally, the covariance of the i th term and the $(i + m)$ th term should not be a function of time. In the following graph, you will notice the spread becomes closer as the time increases. Hence, the covariance is not constant with time for the 'red series'.



Q12. 'People who bought this also bought...' recommendations seen on Amazon are a result of which algorithm?

The recommendation engine is accomplished with collaborative filtering. Collaborative filtering explains the behavior of other users and their purchase history in terms of ratings, selection, etc.

The engine makes predictions on what might interest a person based on the preferences of other users.

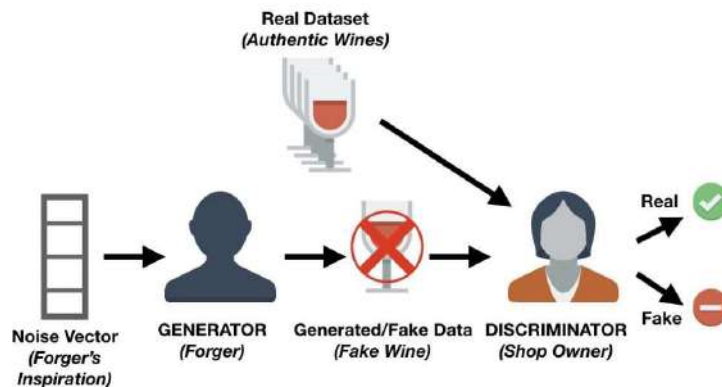
In this algorithm, item features are unknown.

For example, a sales page shows that a certain number of people buy a new phone and also buy tempered glass at the same time. Next time, when a person buys a phone, he or she may see a recommendation to buy tempered glass as well.

Q13. What is a Generative Adversarial Network?

Suppose there is a wine shop purchasing wine from dealers, which they resell later. But some dealers sell fake wine. In this case, the shop owner should be able to distinguish between fake and authentic wine. The forger will try different techniques to sell fake wine and make sure specific techniques go past the shop owner's check. The shop owner would probably get some feedback from wine experts that some of the wine is not original. The owner would have to improve how he determines whether a wine is fake or authentic.

The forger's goal is to create wines that are indistinguishable from the authentic ones while the shop owner intends to tell if the wine is real or not accurately.



- There is a noise vector coming into the forger who is generating fake wine.
- Here the forger acts as a Generator.
- The shop owner acts as a Discriminator.
- The Discriminator gets two inputs; one is the fake wine, while the other is the real authentic wine. The shop owner has to figure out whether it is real or fake.

So, there are two primary components of Generative Adversarial Network (GAN) named:

1. Generator
2. Discriminator

The generator is a CNN that keeps producing images and is closer in appearance to the real images while the discriminator tries to determine the difference between real and fake images. The ultimate aim is to make the discriminator learn to identify real and fake images.

Q14. You are given a dataset on cancer detection. You have built a classification model and achieved an accuracy of 96 percent. Why shouldn't you be happy with your model performance? What can you do about it?

Cancer detection results in imbalanced data. *In an imbalanced dataset, accuracy should not be based as a measure of performance.* It is important to focus on the remaining four percent, which represents the patients who were wrongly diagnosed. Early diagnosis is crucial when it comes to cancer detection and can greatly improve a patient's prognosis.

Hence, to evaluate model performance, we should use Sensitivity (True Positive Rate), Specificity (True Negative Rate), F measure to determine the class wise performance of the classifier.

Q15. Below are the eight actual values of the target variable in the train file. What is the entropy of the target variable? [0, 0, 0, 1, 1, 1, 1, 1]

The target variable, in this case, is 1 (the last)

The formula for calculating the entropy is, putting $p = 5$ and $n = 8$, we get:

$$Entropy = -\left(\frac{5}{8} \log\left(\frac{5}{8}\right) + \frac{3}{8} \log\left(\frac{3}{8}\right)\right)$$

Q16. We want to predict the probability of death from heart disease based on three risk factors: age, gender, and blood cholesterol level. What is the most appropriate algorithm for this case? Choose the correct option:

The most appropriate algorithm for this case is logistic regression.

Q17. After studying the behavior of a population, you have identified four specific individual types that are valuable to your study. You would like to find all users who are most similar to each individual type. Which algorithm is most appropriate for this study?

As we are looking for grouping people together specifically by four different similarities, it indicates the value of k. Therefore, K-means clustering is the most appropriate algorithm for this study.

Q18. You have run the association rules algorithm on your dataset, and the two rules {banana, apple} => {grape} and {apple, orange} => {grape} have been found to be relevant. What else must be true? Choose the right answer:

The answer is A: {grape, apple} must be a frequent itemset.

Q19. Your organization has a website where visitors randomly receive one of two coupons. It is also possible that visitors to the website will not receive a coupon. You have been asked to determine if offering a coupon to website visitors has any impact on their purchase decisions. Which analysis method should you use?

One-way ANOVA: in statistics, one-way analysis of variance is a technique that can be used to compare means of two or more samples. This technique can be used only for numerical response data, the "Y", usually one variable, and numerical or categorical input data, the "X", always one variable, hence "one-way".

The ANOVA tests the null hypothesis, which states that samples in all groups are drawn from populations with the same mean values. To do this, two estimates are made of the population variance. The ANOVA produces an F-statistic, the ratio of the variance calculated among the means to the variance within the samples. If the group means are drawn from populations with the same mean values, the variance between the group means should be lower than the variance of the samples, following the central limit theorem. A higher ratio therefore implies that the samples were drawn from populations with different mean values.

Q20. What are the feature vectors?

A feature vector is an n-dimensional vector of numerical features that represent an object. In machine learning, feature vectors are used to represent numeric or symbolic characteristics (called features) of an object in a mathematical way that's easy to analyze.

Q21. What is root cause analysis?

Root cause analysis was initially developed to analyze industrial accidents but is now widely used in other areas. **It is a problem-solving technique used for isolating the root causes of faults or problems.** A factor is called a root cause if its deduction from the problem-fault-sequence averts the final undesirable event from recurring.

Q22. Do gradient descent methods always converge to similar points?

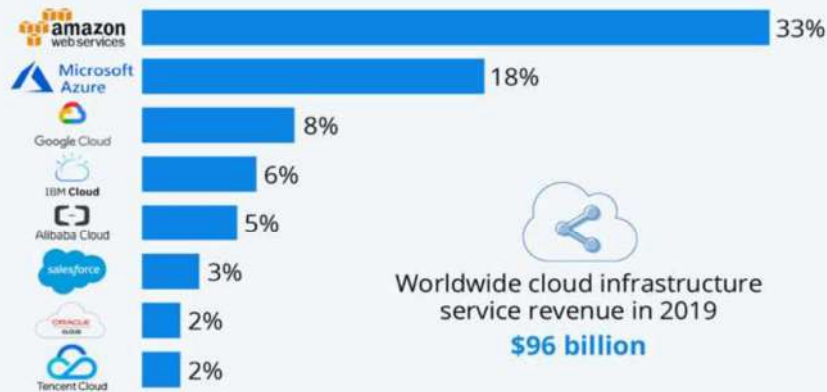
They do not, because in some cases, they reach a local minimum or a local optimum point. You would not reach the global optimum point. This is governed by the data and the starting conditions.

Q23. What are the most popular Cloud Services used in Data Science?

<https://www.zdnet.com/article/the-top-cloud-providers-of-2020-aws-microsoft-azure-google-cloud-hybrid-saas/>

Amazon Leads \$100 Billion Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q4 2019*



Worldwide cloud infrastructure service revenue in 2019
\$96 billion

* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services
Source: Synergy Research Group



statista

Q24. What is a Canary Deployment?

<https://www.split.io/glossary/canary-deployment/>

A canary deployment, or canary release, allows you to rollout your features to only a subset of users as an initial test to make sure nothing else in your system broke.

The initial steps for implementing canary deployment are:

1. create two clones of the production environment,
2. have a load balancer that initially sends all traffic to one version,
3. create new functionality in the other version.

When you deploy the new software version, you shift some percentage – say, 10% – of your user base to the new version while maintaining 90% of users on the old version. If that 10% reports no errors, you can roll it out to gradually more users, until the new version is being used by everyone. If the 10% has problems, though, you can roll it right back, and 90% of your users will have never even seen the problem.

Canary deployment benefits include zero downtime, easy rollout and quick rollback – plus the added safety from the gradual rollout process. It also has some drawbacks – the expense of maintaining multiple server instances, the difficult clone-or-don't-clone database decision.

Typically, software development teams implement blue/green deployment when they're sure the new version will work properly and want a simple, fast strategy to deploy it. Conversely, canary deployment is most useful when the development team isn't as sure about the new version and they don't mind a slower rollout if it means they'll be able to catch the bugs.

Q25. What is a Blue Green Deployment?

<https://docs.cloudfoundry.org/devguide/deploy-apps/blue-green.html>

Blue-green deployment is a technique that reduces downtime and risk by running two identical production environments called Blue and Green.

At any time, only one of the environments is live, with the live environment serving all production traffic. For this example, Blue is currently live, and Green is idle.

As you prepare a new version of your model, deployment and the final stage of testing takes place in the environment that is not live: in this example, Green. Once you have deployed and fully tested the model in Green, you switch the router, so all incoming requests now go to Green instead of Blue. Green is now live, and Blue is idle.

This technique can eliminate downtime due to app deployment and reduces risk: if something unexpected happens with your new version on Green, you can immediately roll back to the last version by switching back to Blue.

This document is based on the original document by Steve Nouri ([LinkedIn](#)).

Reviewed and corrected by Davide Callegaro ([LinkedIn](#)).

Original credits to kdnuggets, Simplilearn, Edureka, Guru99, Hackernoon,
Datacamp, Nitin Panwar, Michael Rundell.

Below some questions the reader shall view the link of the original article.

Interview Question Series #2

Python Programming

Numpy

1. Why is python numpy better than lists?

Python numpy arrays should be considered instead of a list because they are fast, consume less memory and convenient with lots of functionality.

2. Describe the map function in Python?

map function executes the function given as the first argument on all the elements of the iterable given as the second argument.

3. Generate array of '100' random numbers sampled from a standard normal distribution using Numpy

`np.random.rand(100)` will create 100 random numbers generated from standard normal distribution with mean 0 and standard deviation 1.

4. How to count the occurrence of each value in a numpy array?

Use `numpy.bincount()`

```
>>> arr = numpy.array([0, 5, 5, 0, 2, 4, 3, 0, 0, 5, 4, 1, 9, 9])
```

```
>>> numpy.bincount(arr)
```

The argument to `bincount()` must consist of booleans or positive integers. Negative integers are invalid.

5. Does Numpy Support Nan?

nan, short for "not a number", is a special floating point value defined by the IEEE-754 specification. Python numpy supports nan but the definition of nan is more system dependent and some systems don't have an all round support for it like older cray and vax computers.

6. What does `ravel()` function in numpy do?

It combines multiple numpy arrays into a single array

7. What is the meaning of `axis=0` and `axis=1`?

Axis = 0 is meant for reading rows, Axis = 1 is meant for reading columns

8. What is numpy and describe its use cases?

Numpy is a package library for Python, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions. In simple words, Numpy is an optimized version of Python lists like Financial functions, Linear Algebra, Statistics, Polynomials, Sorting and Searching etc.

9. How to remove from one array those items that exist in another?

```
>>> a = np.array([5, 4, 3, 2, 1])
>>> b = np.array([4, 8, 9, 10, 1])
# From 'a' remove all of 'b'
>>> np.setdiff1d(a,b)
# Output:
>>> array([5, 3, 2])
```

10. How to sort a numpy array by a specific column in a 2D array?

```
#Choose column 2 as an example
>>> import numpy as np
>>> arr = np.array([[1, 2, 3], [4, 5, 6], [0,0,1]])
>>> arr[arr[:,1].argsort()]
# Output
>>> array([[0, 0, 1], [1, 2, 3], [4, 5, 6]])
```

11. How to reverse a numpy array in the most efficient way?

```
>>> import numpy as np
>>> arr = np.array([9, 10, 1, 2, 0])
>>> reverse_arr = arr[::-1]
```

12. How to calculate percentiles when using numpy?

```
>>> import numpy as np
>>> arr = np.array([11, 22, 33, 44, 55, 66, 77])
>>> perc = np.percentile(arr, 40) #Returns the 40th percentile
>>> print(perc)
```

13. What Is The Difference Between Numpy And Scipy?

NumPy would contain nothing but the array data type and the most basic operations: indexing, sorting, reshaping, basic element wise functions, et cetera. All numerical code would reside in SciPy. SciPy contains more fully-featured versions of the linear algebra modules, as well as many other numerical algorithms.

14. What Is The Preferred Way To Check For An Empty (zero Element) Array?

For a numpy array, use the size attribute. The size attribute is helpful for determining the length of numpy array:

```
>>> arr = numpy.zeros((1,0))
>>> arr.size
```

15. What Is The Difference Between Matrices And Arrays?

Matrices can only be two-dimensional, whereas arrays can have any number of dimensions

16. How can you find the indices of an array where a condition is true?

Given an array a, the condition `arr > 3` returns a boolean array and since False is interpreted as 0 in Python and NumPy.

```
>>> import numpy as np
>>> arr = np.array([[9,8,7],[6,5,4],[3,2,1]])
>>> arr > 3
>>> array([[ True,  True,  True],
          [ True,  True,  True],
          [False, False, False]], dtype=bool)
```

17. How to find the maximum and minimum value of a given flattened array?

```
>>> import numpy as np
>>> a = np.arange(4).reshape((2,2))
>>> max_val = np.amax(a)
>>> min_val = np.amin(a)
```

18. Write a NumPy program to calculate the difference between the maximum and the minimum values of a given array along the second axis.

```
>>> import numpy as np
>>> arr = np.arange(16).reshape((4, 7))
>>> res = np.ptp(arr, 1)
```

19. Find median of a numpy flattened array

```
>>> import numpy as np
>>> arr = np.arange(16).reshape((4, 5))
>>> res = np.median(arr)
```

20. Write a NumPy program to compute the mean, standard deviation, and variance of a given array along the second axis

```
import numpy as np
>>> import numpy as np
>>> x = np.arange(16)
>>> mean = np.mean(x)
>>> std = np.std(x)
>>> var = np.var(x)
```

21. Calculate covariance matrix between two numpy arrays

```
>>> import numpy as np
>>> x = np.array([2, 1, 0])
>>> y = np.array([2, 3, 3])
>>> cov_arr = np.cov(x, y)
```

22. Compute Compute pearson product-moment correlation coefficients of two given numpy arrays

```
>>> import numpy as np
>>> x = np.array([0, 1, 3])
>>> y = np.array([2, 4, 5])
>>> cross_corr = np.corrcoef(x, y)
```

23. Develop a numpy program to compute the histogram of nums against the bins

```
>>> import numpy as np
>>> nums = np.array([0.5, 0.7, 1.0, 1.2, 1.3, 2.1])
>>> bins = np.array([0, 1, 2, 3])
>>> np.histogram(nums, bins)
```

24. Get the powers of an array values element-wise

```
>>> import numpy as np
>>> x = np.arange(7)
>>> np.power(x, 3)
```

25. Write a NumPy program to get true division of the element-wise array inputs

```
>>> import numpy as np
>>> x = np.arange(10)
>>> np.true_divide(x, 3)
```

Pandas

26. What is a series in pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types. The row labels of the series are called the index. By using a 'series' method, we can easily convert the list, tuple, and dictionary into series. A Series cannot contain multiple columns.

27. What features make Pandas such a reliable option to store tabular data?

Memory Efficient, Data Alignment, Reshaping, Merge and join and Time Series.

28. What is reindexing in pandas?

Reindexing is used to conform DataFrame to a new index with optional filling logic. It places NA/NaN in that location where the values are not present in the previous index. It returns a new object unless the new index is produced as equivalent to the current one, and the value of copy becomes False. It is used to change the index of the rows and columns of the DataFrame.

29. How will you create a series from dict in Pandas?

A Series is defined as a one-dimensional array that is capable of storing various data types.

```
>>> import pandas as pd
>>> info = {'x' : 0., 'y' : 1., 'z' : 2.}
>>> a = pd.Series(info)
```

30. How can we create a copy of the series in Pandas?

Use pandas.Series.copy method

```
>>> import pandas as pd
>>> pd.Series.copy(deep=True)
```

31. What is groupby in Pandas?

GroupBy is used to split the data into groups. It groups the data based on some criteria. Grouping also provides a mapping of labels to the group names. It has a lot of variations that can be defined with the parameters and makes the task of splitting the data quick and easy.

32. What is vectorization in Pandas?

Vectorization is the process of running operations on the entire array. This is done to reduce the amount of iteration performed by the functions. Pandas have a number of vectorized functions like aggregations, and string functions that are optimized to operate specifically on series and DataFrames. So it is preferred to use the vectorized pandas functions to execute the operations quickly.

33. Mention the different types of Data Structures in Pandas

Pandas provide two data structures, which are supported by the pandas library, Series, and DataFrames. Both of these data structures are built on top of the NumPy.

34. What Is Time Series In pandas

A time series is an ordered sequence of data which basically represents how some quantity changes over time. pandas contains extensive capabilities and features for working with time series data for all domains.

35. How to convert pandas dataframe to numpy array?

The function `to_numpy()` is used to convert the DataFrame to a NumPy array.

```
DataFrame.to_numpy(self, dtype=None, copy=False)
```

The `dtype` parameter defines the data type to pass to the array and the `copy` ensures the returned value is not a view on another array.

36. Write a Pandas program to get the first 5 rows of a given DataFrame

```
>>> import pandas as pd
>>> exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas'],}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
>>> df = pd.DataFrame(exam_data , index=labels)
>>> df.iloc[:5]
```

37. Develop a Pandas program to create and display a one-dimensional array-like object containing an array of data.

```
>>> import pandas as pd
>>> pd.Series([2, 4, 6, 8, 10])
```

38. Write a Python program to convert a Panda module Series to Python list and it's type.

```
>>> import pandas as pd
>>> ds = pd.Series([2, 4, 6, 8, 10])
```

```
>>> type(ds)
>>> ds.tolist()
>>> type(ds.tolist())
```

39. Develop a Pandas program to add, subtract, multiple and divide two Pandas Series.

```
>>> import pandas as pd
>>> ds1 = pd.Series([2, 4, 6, 8, 10])
>>> ds2 = pd.Series([1, 3, 5, 7, 9])
>>> sum = ds1 + ds2
>>> sub = ds1 - ds2
>>> mul = ds1 * ds2
>>> div = ds1 / ds2
```

40. Develop a Pandas program to compare the elements of the two Pandas Series.

```
>>> import pandas as pd
>>> ds1 = pd.Series([2, 4, 6, 8, 10])
>>> ds2 = pd.Series([1, 3, 5, 7, 10])
>>> ds1 == ds2
>>> ds1 > ds2
>>> ds1 < ds2
```

41. Develop a Pandas program to change the data type of given a column or a Series.

```
>>> import pandas as pd
>>> s1 = pd.Series(['100', '200', 'python', '300.12', '400'])
>>> s2 = pd.to_numeric(s1, errors='coerce')
>>> s2
```

42. Write a Pandas program to convert Series of lists to one Series

```
>>> import pandas as pd
>>> s = pd.Series(['Red', 'Black'], ['Red', 'Green', 'White'], ['Yellow'])
>>> s = s.apply(pd.Series).stack().reset_index(drop=True)
```

43. Write a Pandas program to create a subset of a given series based on value and condition

```
>>> import pandas as pd
>>> s = pd.Series([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
>>> n = 6
```



```
>>> new_s = s[s < n]
>>> new_s
```

44. Develop a Pandas code to alter the order of index in a given series

```
>>> import pandas as pd
>>> s = pd.Series(data = [1,2,3,4,5], index = ['A', 'B', 'C','D','E'])
>>> s.reindex(index = ['B','A','C','D','E'])
```

45. Write a Pandas code to get the items of a given series not present in another given series.

```
>>> import pandas as pd
>>> sr1 = pd.Series([1, 2, 3, 4, 5])
>>> sr2 = pd.Series([2, 4, 6, 8, 10])
>>> result = sr1[~sr1.isin(sr2)]
>>> result
```

46. What is the difference between the two data series df['Name'] and df.loc[:, 'Name']?

```
>>> First one is a view of the original dataframe and second one is a copy of the original dataframe.
```

47. Write a Pandas program to display the most frequent value in a given series and replace everything else as “replaced” in the series.

```
>>> import pandas as pd
>>> import numpy as np
>>> np.random.RandomState(100)
>>> num_series = pd.Series(np.random.randint(1, 5, [15]))
>>> result = num_series[~num_series.isin(num_series.value_counts().index[:1])] = 'replaced'
```

48. Write a Pandas program to find the positions of numbers that are multiples of 5 of a given series.

```
>>> import pandas as pd
>>> import numpy as np
>>> num_series = pd.Series(np.random.randint(1, 10, 9))
>>> result = np.argwhere(num_series % 5==0)
```

49. How will you add a column to a pandas DataFrame?

```
# importing the pandas library
>>> import pandas as pd
>>> info = {'one' : pd.Series([1, 2, 3, 4, 5], index=['a', 'b', 'c', 'd', 'e']),
           'two' : pd.Series([1, 2, 3, 4, 5, 6], index=['a', 'b', 'c', 'd', 'e', 'f'])}
>>> info = pd.DataFrame(info)
# Add a new column to an existing DataFrame object
>>> info['three']=pd.Series([20,40,60],index=['a','b','c'])
```

50. How to iterate over a Pandas DataFrame?

You can iterate over the rows of the DataFrame by using for loop in combination with an `iterrows()` call on the DataFrame.

Python Language

51. What type of language is python? Programming or scripting?

Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language.

52. Is python case sensitive?

Yes, python is a case sensitive language.

53. What is a lambda function in python?

An anonymous function is known as a lambda function. This function can have any number of parameters but can have just one statement.

54. What is the difference between xrange and range in python?

`xrange` and `range` are the exact same in terms of functionality. The only difference is that `range` returns a Python list object and `xrange` returns an `xrange` object.

55. What are docstrings in python?

Docstrings are not actually comments, but they are *documentation strings*. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well.

56. Whenever Python exits, why isn't all the memory deallocated?

Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces

are not always de-allocated or freed. It is impossible to de-allocate those portions of memory that are reserved by the C library. On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object.

57. What does this mean: *args, **kwargs? And why would we use it?

We use *args when we aren't sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. **kwargs is used when we don't know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments.

58. What is the difference between deep and shallow copy?

Shallow copy is used when a new instance type gets created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just like it copies the values.

Deep copy is used to store the values that are already copied. Deep copy doesn't copy the reference pointers to the objects. It makes the reference to an object and the new object that is pointed by some other object gets stored.

59. Define encapsulation in Python?

Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

60. Does python make use of access specifiers?

Python does not deprive access to an instance variable or function. Python lays down the concept of prefixing the name of the variable, function or method with a single or double underscore to imitate the behavior of protected and private access specifiers.

61. What are the generators in Python?

Generators are a way of implementing iterators. A generator function is a normal function except that it contains yield expression in the function definition making it a generator function.

62. How will you remove the duplicate elements from the given list?

The set is another type available in Python. It doesn't allow copies and provides some good functions to perform set operations like union, difference etc.

```
>>> list(set(a))
```

63. Does Python allow arguments Pass by Value or Pass by Reference?

Neither the arguments are Pass by Value nor does Python supports Pass by reference. Instead, they are Pass by assignment. The parameter which you pass is originally a reference to the object not the reference to a fixed memory location. But the reference is passed by value. Additionally, some data types like strings and tuples are immutable whereas others are mutable.

64. What is slicing in Python?

Slicing in Python is a mechanism to select a range of items from Sequence types like strings, list, tuple, etc.

65. Why is the “pass” keyword used in Python?

The “pass” keyword is a no-operation statement in Python. It signals that no action is required. It works as a placeholder in compound statements which are intentionally left blank.

66. What is PEP8 and why is it important?

PEP stands for Python Enhancement Proposal. A PEP is an official design document providing information to the Python Community, or describing a new feature for Python or its processes. PEP 8 is especially important since it documents the style guidelines for Python Code. Apparently contributing in the Python open-source community requires you to follow these style guidelines sincerely and strictly.

67. What are decorators in Python?

Decorators in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented by the `@decorator_name` in Python and are called in bottom-up fashion

68. What is the key difference between lists and tuples in python?

The key difference between the two is that while lists are mutable, tuples on the other hand are immutable objects.

69. What is self in Python?

Self is a keyword in Python used to define an instance or an object of a class. In Python, it is explicitly used as the first parameter, unlike in Java where it is optional. It helps in distinguishing between the methods and attributes of a class from its local variables.

70. What is PYTHONPATH in Python?

PYTHONPATH is an environment variable which you can set to add additional directories where Python will look for modules and packages. This is especially useful in maintaining Python libraries that you do not wish to install in the global default location.

71. What is the difference between .py and .pyc files?

.py files contain the source code of a program. Whereas, .pyc file contains the bytecode of your program. We get bytecode after compilation of .py file (source code). .pyc files are not created for all the files that you run. It is only created for the files that you import.

72. Explain how you can access a module written in Python from C?

You can access a module written in Python from C by following method,
Module = PyImport_ImportModule("<modulename>");

73. What is namespace in Python?

In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get the corresponding object.

74. What is pickling and unpickling?

Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using the dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

75. How is Python interpreted?

Python language is an interpreted language. The Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

Jupyter Notebook

76. What is the main use of a Jupyter notebook?

Jupyter Notebook is an open-source web application that allows us to create and share codes and documents. It provides an environment, where you can document your code, run it, look at the outcome, visualize data and see the results without leaving the environment.

77. How do I increase the cell width of the Jupyter/ipython notebook in my browser?

```
>>> from IPython.core.display import display, HTML
>>> display(HTML("<style>.container { width:100% !important; }</style>"))
```

78. How do I convert an IPython Notebook into a Python file via command line?

```
>>> jupyter nbconvert --to script [YOUR_NOTEBOOK].ipynb
```

79. How to measure execution time in a jupyter notebook?

```
>>> %%time is inbuilt magic command
```

80. How to run a jupyter notebook from the command line?

```
>>> jupyter nbconvert --to python nb.ipynb
```

81. How to make inline plots larger in jupyter notebooks?

Use figure size.

```
>>> fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
```

82. How to display multiple images in a jupyter notebook?

```
>>>for ima in images:
```

```
>>>plt.figure()
```

```
>>>plt.imshow(ima)
```

83. Why is the Jupyter notebook interactive code and data exploration friendly?

The ipywidgets package provides many common user interface controls for exploring code and data interactively.

84. What is the default formatting option in jupyter notebook?

Default formatting option is markdown

85. What are kernel wrappers in jupyter?

Jupyter brings a lightweight interface for kernel languages that can be wrapped in Python. Wrapper kernels can implement optional methods, notably for code completion and code inspection.

86. What are the advantages of custom magic commands?

Create IPython extensions with custom magic commands to make interactive computing even easier. Many third-party extensions and magic commands exist, for example, the %%cython magic that allows one to write Cython code directly in a notebook.

87. Is the jupyter architecture language dependent?

No. It is language independent.

88. Which tools allow jupyter notebooks to easily convert to pdf and html?

Nbconvert converts it to pdf and html while Nbviewer renders the notebooks on the web platforms.

89. What is a major disadvantage of a Jupyter notebook?

It is very hard to run long asynchronous tasks. Less Secure.

90. In which domain is the jupyter notebook widely used?

It is mainly used for data analysis and machine learning related tasks.

91. What are alternatives to jupyter notebook?

PyCharm interact, VS Code Python Interactive etc.

92. Where can you make configuration changes to the jupyter notebook?

In the config file located at ~/.ipython/profile_default/ipython_config.py

93. Which magic command is used to run python code from jupyter notebook?

%run can execute python code from .py files

94. How to pass variables across the notebooks?

The %store command lets you pass variables between two different notebooks.

```
>>> data = 'this is the string I want to pass to different notebook'
```

```
>>> %store data
```

```
# Stored 'data' (str)
```

```
# In new notebook
```

```
>>> %store -r data
```

```
>>> print(data)
```

95. Export the contents of a cell/Show the contents of an external script

Using the %%writefile magic saves the contents of that cell to an external file. %pycat does the opposite and shows you (in a popup) the syntax highlighted contents of an external file.

96. What inbuilt tool we use for debugging python code in a jupyter notebook?

Jupyter has its own interface for The Python Debugger (pdb). This makes it possible to go inside the function and investigate what happens there.

97. How to make high resolution plots in a jupyter notebook?

```
>>> %config InlineBackend.figure_format = 'retina'
```

98. How can one use latex in a jupyter notebook?

When you write [LaTeX](#) in a Markdown cell, it will be rendered as a formula using MathJax.

99. What is a jupyter lab?

It is a next generation user interface for conventional jupyter notebooks. Users can drag and drop cells, arrange code workspace and live previews. It's still in the early stage of development.

100. What is the biggest limitation for a Jupyter notebook?

Code versioning, management and debugging is not scalable in current jupyter notebook.

References

- [1] <https://www.edureka.co>
- [2] <https://www.kausavikash.in>
- [3] <https://www.wisdomjobs.com>
- [4] <https://blog.edugrad.com>
- [5] <https://stackoverflow.com>
- [6] <http://www.ezdev.org>
- [7] <https://www.techbeamers.com>
- [8] <https://www.w3resource.com>
- [9] <https://www.javatpoint.com>
- [10] <https://analyticsindiamag.com>
- [11] <https://www.onlineinterviewquestions.com>
- [12] <https://www.geeksforgeeks.org>
- [13] <https://www.springpeople.com>
- [14] <https://atraininghub.com>
- [15] <https://www.interviewcake.com>
- [16] <https://www.techbeamers.com>
- [17] <https://www.tutorialspoint.com>
- [18] <https://programmingwithmosh.com>
- [19] <https://www.interviewbit.com>
- [20] <https://www.guru99.com>
- [21] <https://hub.packtpub.com>
- [22] <https://analyticsindiamag.com>
- [23] <https://www.dataquest.io>
- [24] <https://www.infoworld.com>

Top 100 Machine Learning Questions & Answers

Steve Nouri

Q1 Explain the difference between supervised and unsupervised machine learning?

In supervised machine learning algorithms, we have to provide labeled data, for example, prediction of stock market prices, whereas in unsupervised we need not have labeled data, for example, classification of emails into spam and non-spam.

Q2 What are the parametric models? Give an example.

Parametric models are those with a finite number of parameters. To predict new data, you only need to know the parameters of the model. Examples include linear regression, logistic regression, and linear SVMs.

Non-parametric models are those with an unbounded number of parameters, allowing for more flexibility. To predict new data, you need to know the parameters of the model and the state of the data that has been observed. Examples include decision trees, k-nearest neighbors, and topic models using latent Dirichlet analysis.

Q3 What is the difference between classification and regression?

Classification is used to produce discrete results, classification is used to classify data into some specific categories. For example, classifying emails into spam and non-spam categories.

Whereas, We use regression analysis when we are dealing with continuous data, for example predicting stock prices at a certain point in time.

Q4 What Is Overfitting, and How Can You Avoid It?

Overfitting is a situation that occurs when a model learns the training set too well, taking up random fluctuations in the training data as concepts. These impact the model's ability to generalize and don't apply to new data.

When a model is given the training data, it shows 100 percent accuracy—technically a slight loss. But, when we use the test data, there may be an error and low efficiency. This condition is known as overfitting.

There are multiple ways of avoiding overfitting, such as:

- Regularization. It involves a cost term for the features involved with the objective function
- Making a simple model. With lesser variables and parameters, the variance can be reduced
- Cross-validation methods like k-folds can also be used
- If some model parameters are likely to cause overfitting, techniques for regularization like LASSO can be used that penalize these parameters

Q5 What is meant by 'Training set' and 'Test Set'?

We split the given data set into two different sections namely, 'Training set' and 'Test Set'.

'Training set' is the portion of the dataset used to train the model.

'Testing set' is the portion of the dataset used to test the trained model.

Q6 How Do You Handle Missing or Corrupted Data in a Dataset?

One of the easiest ways to handle missing or corrupted data is to drop those rows or columns or replace them entirely with some other value.

There are two useful methods in Pandas:

- `IsNull()` and `dropna()` will help to find the columns/rows with missing data and drop them
- `Fillna()` will replace the wrong values with a placeholder value

Q7 Explain Ensemble learning.

In ensemble learning, many base models like classifiers and regressors are generated and combined together so that they give better results. It is used when we build component classifiers that are accurate and independent. There are sequential as well as parallel ensemble methods.

Q8 Explain the Bias-Variance Tradeoff.

Predictive models have a tradeoff between bias (how well the model fits the data) and variance (how much the model changes based on changes in the inputs).

Simpler models are stable (low variance) but they don't get close to the truth (high bias).

More complex models are more prone to overfitting (high variance) but they are expressive enough to get close to the truth (low bias). The best model for a given problem usually lies somewhere in the middle.

Q9 What is the difference between stochastic gradient descent (SGD) and gradient descent (GD)?

Both algorithms are methods for finding a set of parameters that minimize a loss function by evaluating parameters against data and then making adjustments.

In standard gradient descent, you'll evaluate all training samples for each set of parameters. This is akin to taking big, slow steps toward the solution.

In stochastic gradient descent, you'll evaluate only 1 training sample for the set of parameters before updating them. This is akin to taking small, quick steps toward the solution.

Q10 How Can You Choose a Classifier Based on a Training Set Data Size?

When the training set is small, a model that has a right bias and low variance seems to work better because they are less likely to overfit.

For example, Naive Bayes works best when the training set is large. Models with low bias and high variance tend to perform better as they work fine with complex relationships.

Q11 What are 3 data preprocessing techniques to handle outliers?

1. Winsorize (cap at threshold).
2. Transform to reduce skew (using Box-Cox or similar).
3. Remove outliers if you're certain they are anomalies or measurement errors.

Q12 How much data should you allocate for your training, validation, and test sets?

You have to find a balance, and there's no right answer for every problem.

If your test set is too small, you'll have an unreliable estimation of model performance (performance statistic will have high variance). If your training set is too small, your actual model parameters will have a high variance.

A good rule of thumb is to use an 80/20 train/test split. Then, your train set can be further split into train/validation or into partitions for cross-validation.

Q13 What Is a False Positive and False Negative and How Are They Significant?

False positives are those cases which wrongly get classified as True but are False.

False negatives are those cases which wrongly get classified as False but are True.

In the term 'False Positive,' the word 'Positive' refers to the 'Yes' row of the predicted value in the confusion matrix. The complete term indicates that the system has predicted it as a positive, but the actual value is negative.

Q14 Explain the difference between L1 and L2 regularization.

L2 regularization tends to spread error among all the terms, while L1 is more binary/sparse, with many variables either being assigned a 1 or 0 in weighting. L1 corresponds to setting a Laplacean prior to the terms, while L2 corresponds to a Gaussian prior.

Q15 What's a Fourier transform?

A Fourier transform is a generic method to decompose generic functions into a superposition of symmetric functions. Or as this more intuitive tutorial puts it, given a smoothie, it's how we find the recipe. The Fourier transform finds the set of cycle speeds, amplitudes, and phases to match any time signal. A Fourier transform converts a signal from time to frequency domain — it's a very common way to extract features from audio signals or other time series such as sensor data.

Q16 What is deep learning, and how does it contrast with other machine learning algorithms?

Deep learning is a subset of machine learning that is concerned with neural networks: how to use backpropagation and certain principles from neuroscience to more accurately model large sets of unlabelled or semi-structured data. In that sense, deep learning represents an

unsupervised learning algorithm that learns representations of data through the use of neural nets.

Q17 What's the difference between a generative and discriminative model?

A generative model will learn categories of data while a discriminative model will simply learn the distinction between different categories of data. Discriminative models will generally outperform generative models on classification tasks.

Q18 What Are the Applications of Supervised Machine Learning in Modern Businesses?

Applications of supervised machine learning include:

- **Email Spam Detection**
Here we train the model using historical data that consists of emails categorized as spam or not spam. This labeled information is fed as input to the model.
- **Healthcare Diagnosis**
By providing images regarding a disease, a model can be trained to detect if a person is suffering from the disease or not.
- **Sentiment Analysis**
This refers to the process of using algorithms to mine documents and determine whether they're positive, neutral, or negative in sentiment.
- **Fraud Detection**
Training the model to identify suspicious patterns, we can detect instances of possible fraud.

Q19 What Is Semi-supervised Machine Learning?

Supervised learning uses data that is completely labeled, whereas unsupervised learning uses no training data.

In the case of semi-supervised learning, the training data contains a small amount of labeled data and a large amount of unlabeled data.

Q20. What Are Unsupervised Machine Learning Techniques?

There are two techniques used in unsupervised learning: clustering and association.

Clustering

- Clustering problems involve data to be divided into subsets. These subsets, also called clusters, contain data that are similar to each other. Different clusters reveal different details about the objects, unlike classification or regression.

Association

- In an association problem, we identify patterns of associations between different variables or items.
- For example, an eCommerce website can suggest other items for you to buy, based on the prior purchases that you have made, spending habits, items in your wishlist, other customers' purchase habits, and so on.

Q21 What Is 'naive' in the Naive Bayes Classifier?

The classifier is called 'naive' because it makes assumptions that may or may not turn out to be correct.

The algorithm assumes that the presence of one feature of a class is not related to the presence of any other feature (absolute independence of features), given the class variable.

For instance, a fruit may be considered to be a cherry if it is red in color and round in shape, regardless of other features. This assumption may or may not be right (as an apple also matches the description).

Q22 Explain Latent Dirichlet Allocation (LDA).

Latent Dirichlet Allocation (LDA) is a common method of topic modeling, or classifying documents by subject matter.

LDA is a generative model that represents documents as a mixture of topics that each have their own probability distribution of possible words.

The "Dirichlet" distribution is simply a distribution of distributions. In LDA, documents are distributions of topics that are distributions of words.

Q23 Explain Principle Component Analysis (PCA).

PCA is a method for transforming features in a dataset by combining them into uncorrelated linear combinations.

These new features, or principal components, sequentially maximize the variance represented (i.e. the first principal component has the most variance, the second principal component has the second most, and so on).

As a result, PCA is useful for dimensionality reduction because you can set an arbitrary variance cutoff.

Q24 What's the F1 score? How would you use it?

The F1 score is a measure of a model's performance. It is a weighted average of the precision and recall of a model, with results tending to 1 being the best, and those tending to 0 being the worst. You would use it in classification tests where true negatives don't matter much.

Q25 When should you use classification over regression?

Classification produces discrete values and dataset to strict categories, while regression gives you continuous results that allow you to better distinguish differences between individual points. You would use classification over regression if you wanted your results to reflect the belongingness of data points in your dataset to certain explicit categories (ex: If you wanted to know whether a name was male or female rather than just how correlated they were with male and female names.)

Q26 How do you ensure you're not overfitting with a model?

This is a simple restatement of a fundamental problem in machine learning: the possibility of overfitting training data and carrying the noise of that data through to the test set, thereby providing inaccurate generalizations.

There are three main methods to avoid overfitting:

- 1- Keep the model simpler: reduce variance by taking into account fewer variables and parameters, thereby removing some of the noise in the training data.
- 2- Use cross-validation techniques such as k-folds cross-validation.
- 3- Use regularization techniques such as LASSO that penalize certain model parameters if they're likely to cause overfitting.

Q27 How Will You Know Which Machine Learning Algorithm to Choose for Your Classification Problem?

While there is no fixed rule to choose an algorithm for a classification problem, you can follow these guidelines:

- If accuracy is a concern, test different algorithms and cross-validate them
- If the training dataset is small, use models that have low variance and high bias
- If the training dataset is large, use models that have high variance and little bias

Q28 How Do You Design an Email Spam Filter?

Building a spam filter involves the following process:

- The email spam filter will be fed with thousands of emails
- Each of these emails already has a label: 'spam' or 'not spam.'
- The supervised machine learning algorithm will then determine which type of emails are being marked as spam based on spam words like the lottery, free offer, no money, full refund, etc.
- The next time an email is about to hit your inbox, the spam filter will use statistical analysis and algorithms like Decision Trees and SVM to determine how likely the email is spam
- If the likelihood is high, it will label it as spam, and the email won't hit your inbox

- Based on the accuracy of each model, we will use the algorithm with the highest accuracy after testing all the models

Q29 What evaluation approaches would you work to gauge the effectiveness of a machine learning model?

You would first split the dataset into training and test sets, or perhaps use cross-validation techniques to further segment the dataset into composite sets of training and test sets within the data. You should then implement a choice selection of performance metrics: here is a fairly comprehensive list. You could use measures such as the F1 score, the accuracy, and the confusion matrix. What's important here is to demonstrate that you understand the nuances of how a model is measured and how to choose the right performance measures for the right situations.

Q30 How would you implement a recommendation system for our company's users?

A lot of machine learning interview questions of this type will involve the implementation of machine learning models to a company's problems. You'll have to research the company and its industry in-depth, especially the revenue drivers the company has, and the types of users the company takes on in the context of the industry it's in.

Q31 Explain bagging.

Bagging, or Bootstrap Aggregating, is an ensemble method in which the dataset is first divided into multiple subsets through resampling.

Then, each subset is used to train a model, and the final predictions are made through voting or averaging the component models.

Bagging is performed in parallel.

Q32 What is the ROC Curve and what is AUC (a.k.a. AUROC)?

The ROC (receiver operating characteristic) the performance plot for binary classifiers of True Positive Rate (y-axis) vs. False Positive Rate (x-axis).

AUC is the area under the ROC curve, and it's a common performance metric for evaluating binary classification models.

It's equivalent to the expected probability that a uniformly drawn random positive is ranked before a uniformly drawn random negative.

Q33 Why is Area Under ROC Curve (AUROC) better than raw accuracy as an out-of-sample evaluation metric?

AUROC is robust to class imbalance, unlike raw accuracy.

For example, if you want to detect a type of cancer that's prevalent in only 1% of the population, you can build a model that achieves 99% accuracy by simply classifying everyone has cancer-free.

Q34 What are the advantages and disadvantages of neural networks?

Advantages: Neural networks (specifically deep NNs) have led to performance breakthroughs for unstructured datasets such as images, audio, and video. Their incredible flexibility allows them to learn patterns that no other ML algorithm can learn.

Disadvantages: However, they require a large amount of training data to converge. It's also difficult to pick the right architecture, and the internal "hidden" layers are incomprehensible.

Q35 Define Precision and Recall.

Precision

- Precision is the ratio of several events you can correctly recall to the total number of events you recall (mix of correct and wrong recalls).
- $\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$

Recall

- A recall is the ratio of a number of events you can recall the number of total events.
- $\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$

Q36 What Is Decision Tree Classification?

A decision tree builds classification (or regression) models as a tree structure, with datasets broken up into ever-smaller subsets while developing the decision tree, literally in a tree-like way with branches and nodes. Decision trees can handle both categorical and numerical data.

Q37 What Is Pruning in Decision Trees, and How Is It Done?

Pruning is a technique in machine learning that reduces the size of decision trees. It reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.

Pruning can occur in:

- Top-down fashion. It will traverse nodes and trim subtrees starting at the root
- Bottom-up fashion. It will begin at the leaf nodes

There is a popular pruning algorithm called reduced error pruning, in which:

- Starting at the leaves, each node is replaced with its most popular class
- If the prediction accuracy is not affected, the change is kept
- There is an advantage of simplicity and speed

Q38 What Is a Recommendation System?

Anyone who has used Spotify or shopped at Amazon will recognize a recommendation system: It's an information filtering system that predicts what a user might want to hear or see based on choice patterns provided by the user.

Q39 What Is Kernel SVM?

Kernel SVM is the abbreviated version of the kernel support vector machine. Kernel methods are a class of algorithms for pattern analysis, and the most common one is the kernel SVM.

Q40 What Are Some Methods of Reducing Dimensionality?

You can reduce dimensionality by combining features with feature engineering, removing collinear features, or using algorithmic dimensionality reduction.

Now that you have gone through these machine learning interview questions, you must have got an idea of your strengths and weaknesses in this domain.

Q41 What Are the Three Stages of Building a Model in Machine Learning?

The three stages of building a machine learning model are:

- **Model Building** Choose a suitable algorithm for the model and train it according to the requirement
- **Model Testing** Check the accuracy of the model through the test data
- **Applying the Model** Make the required changes after testing and use the final model for real-time projects. Here, it's important to remember that once in a while, the model needs to be checked to make sure it's working correctly. It should be modified to make sure that it is up-to-date.

Q42 How is KNN different from k-means clustering?

K-Nearest Neighbors is a supervised classification algorithm, while k-means clustering is an unsupervised clustering algorithm. While the mechanisms may seem similar at first, what this really means is that in order for K-Nearest Neighbors to work, you need labeled data you want to classify an unlabeled point into (thus the nearest neighbor part). K-means clustering requires only a set of unlabeled points and a threshold: the algorithm will take unlabeled points and gradually learn how to cluster them into groups by computing the mean of the distance between different points.

Q43 Mention the difference between Data Mining and Machine learning?

Machine learning relates to the study, design, and development of the algorithms that give computers the capability to learn without being explicitly programmed. While data mining can be defined as the process in which the unstructured data tries to extract knowledge or unknown interesting patterns. During this processing machine, learning algorithms are used.

Q44 What are the different Algorithm techniques in Machine Learning?

The different types of techniques in Machine Learning are

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning
- Transduction
- Learning to Learn

Q45 You are given a data set. The data set has missing values that spread along 1 standard deviation from the median. What percentage of data would remain unaffected? Why?

This question has enough hints for you to start thinking! Since the data is spread across the median, let's assume it's a normal distribution. We know, in a normal distribution, ~68% of the data lies in 1 standard deviation from mean (or mode, median), which leaves ~32% of the data unaffected. Therefore, ~32% of the data would remain unaffected by missing values.

Q46 What are PCA, KPCA, and ICA used for?

PCA (Principal Components Analysis), KPCA (Kernel-based Principal Component Analysis) and ICA (Independent Component Analysis) are important feature extraction techniques used for dimensionality reduction.

Q47 What are support vector machines?

Support vector machines are supervised learning algorithms used for classification and regression analysis.

Q48 What is batch statistical learning?

Statistical learning techniques allow learning a function or predictor from a set of observed data that can make predictions about unseen or future data. These techniques provide guarantees on the performance of the learned predictor on the future unseen data based on a statistical assumption on the data generating process.

Q49 What is the bias-variance decomposition of classification error in the ensemble method?

The expected error of a learning algorithm can be decomposed into bias and variance. A bias term measures how closely the average classifier produced by the learning algorithm matches the target function. The variance term measures how much the learning algorithm's prediction fluctuates for different training sets.

Q50 When is Ridge regression favorable over Lasso regression?

You can quote ISLR's authors Hastie, Tibshirani who asserted that, in the presence of few variables with medium / large sized effect, use lasso regression. In presence of many variables with small/medium-sized effects, use ridge regression.

Conceptually, we can say, lasso regression (L1) does both variable selection and parameter shrinkage, whereas Ridge regression only does parameter shrinkage and end up including all the coefficients in the model. In the presence of correlated variables, ridge regression might be the preferred choice. Also, ridge regression works best in situations where the least square estimates have higher variance. Therefore, it depends on our model objective.

Q51 You've built a random forest model with 10000 trees. You got delighted after getting training error as 0.00. But, the validation error is 34.23. What is going on? Haven't you trained your model perfectly?

The model has overfitted. Training error 0.00 means the classifier has mimicked the training data patterns to an extent, that they are not available in the unseen data. Hence, when this classifier was run on an unseen sample, it couldn't find those patterns and returned predictions with higher error. In a random forest, it happens when we use a larger number of trees than necessary. Hence, to avoid this situation, we should tune the number of trees using cross-validation.

Q50 What is a convex hull?

In the case of linearly separable data, the convex hull represents the outer boundaries of the two groups of data points. Once the convex hull is created, we get maximum margin hyperplane (MMH) as a perpendicular bisector between two convex hulls. MMH is the line which attempts to create the greatest separation between two groups.

Q51 What do you understand by Type I vs Type II error?

Type I error is committed when the null hypothesis is true and we reject it, also known as a 'False Positive'. Type II error is committed when the null hypothesis is false and we accept it, also known as 'False Negative'.

In the context of the confusion matrix, we can say Type I error occurs when we classify a value as positive (1) when it is actually negative (0). Type II error occurs when we classify a value as negative (0) when it is actually positive(1).

Q52. In k-means or kNN, we use euclidean distance to calculate the distance between nearest neighbors. Why not manhattan distance?

We don't use manhattan distance because it calculates distance horizontally or vertically only. It has dimension restrictions. On the other hand, the euclidean metric can be used in any space to calculate distance. Since the data points can be present in any dimension, euclidean distance is a more viable option.

Example: Think of a chessboard, the movement made by a bishop or a rook is calculated by manhattan distance because of their respective vertical & horizontal movements.

Q53 Do you suggest that treating a categorical variable as a continuous variable would result in a better predictive model?

For better predictions, the categorical variable can be considered as a continuous variable only when the variable is ordinal in nature.

Q54 OLS is to linear regression. The maximum likelihood is logistic regression. Explain the statement.

OLS and Maximum likelihood are the methods used by the respective regression methods to approximate the unknown parameter (coefficient) value. In simple words, Ordinary least square(OLS) is a method used in linear regression which approximates the parameters resulting in minimum distance between actual and predicted values. Maximum Likelihood helps in choosing the values of parameters which maximizes the likelihood that the parameters are most likely to produce observed data.

Q55 When does regularization becomes necessary in Machine Learning?

Regularization becomes necessary when the model begins to overfit/underfit. This technique introduces a cost term for bringing in more features with the objective function. Hence, it tries to push the coefficients for many variables to zero and hence reduce the cost term. This helps to reduce model complexity so that the model can become better at predicting (generalizing).

Q56 What is Linear Regression?

Linear Regression is a supervised Machine Learning algorithm. It is used to find the linear relationship between the dependent and the independent variables for predictive analysis.

Q57 What is the Variance Inflation Factor?

Variance Inflation Factor (VIF) is the estimate of the volume of multicollinearity in a collection of many regression variables.

$VIF = \text{Variance of the model} / \text{Variance of the model with a single independent variable}$

We have to calculate this ratio for every independent variable. If VIF is high, then it shows the high collinearity of the independent variables.

Q58 We know that one hot encoding increases the dimensionality of a dataset, but label encoding doesn't. How?

When we use **one-hot encoding**, there is an increase in the dimensionality of a dataset. The reason for the increase in dimensionality is that, for every class in the categorical variables, it forms a different variable.

Q59 What is a Decision Tree?

A decision tree is used to explain the sequence of actions that must be performed to get the desired output. It is a hierarchical diagram that shows the actions.

Q60 What is the Binarizing of data? How to Binarize?

In most of the Machine Learning Interviews, apart from theoretical questions, interviewers focus on the implementation part. So, this ML Interview Questions focused on the implementation of the theoretical concepts.

Converting data into binary values on the basis of threshold values is known as the binarizing of data. The values that are less than the threshold are set to 0 and the values that are greater than the threshold are set to 1. This process is useful when we have to perform feature engineering, and we can also use it for adding unique features.

Q61 What is cross-validation?

Cross-validation is essentially a technique used to assess how well a model performs on a new independent dataset. The simplest example of cross-validation is when you split your data into two groups: training data and testing data, where you use the training data to build the model and the testing data to test the model.

Q62 When would you use random forests Vs SVM and why?

There are a couple of reasons why a random forest is a better choice of the model than a support vector machine:

- Random forests allow you to determine the feature importance. SVM's can't do this.
- Random forests are much quicker and simpler to build than an SVM.
- For multi-class classification problems, SVMs require a one-vs-rest method, which is less scalable and more memory intensive.

Q63 What are the drawbacks of a linear model?

There are a couple of drawbacks of a linear model:

- A linear model holds some strong assumptions that may not be true in the application. It assumes a linear relationship, multivariate normality, no or little multicollinearity, no auto-correlation, and homoscedasticity
- A linear model can't be used for discrete or binary outcomes.
- You can't vary the model flexibility of a linear model.

Q64 Do you think 50 small decision trees are better than a large one? Why?

Another way of asking this question is “Is a random forest a better model than a decision tree?” And the answer is yes because a random forest is an ensemble method that takes many weak decision trees to make a strong learner. Random forests are more accurate, more robust, and less prone to overfitting.

Q65 What is a kernel? Explain the kernel trick

A kernel is a way of computing the dot product of two vectors x and y in some (possibly very high dimensional) feature space, which is why kernel functions are sometimes called “generalized dot product”

The kernel trick is a method of using a linear classifier to solve a non-linear problem by transforming linearly inseparable data to linearly separable ones in a higher dimension.

Q66 State the differences between causality and correlation?

Causality applies to situations where one action, say X , causes an outcome, say Y , whereas Correlation is just relating one action (X) to another action (Y) but X does not necessarily cause Y .

Q67 What is the exploding gradient problem while using the backpropagation technique?

When large error gradients accumulate and result in large changes in the neural network weights during training, it is called the exploding gradient problem. The values of weights can become so large as to overflow and result in NaN values. This makes the model unstable and the learning of the model to stall just like the vanishing gradient problem.

Q68 What do you mean by Associative Rule Mining (ARM)?

Associative Rule Mining is one of the techniques to discover patterns in data like features (dimensions) which occur together and features (dimensions) which are correlated.

Q69 What is Marginalisation? Explain the process.

Marginalization is summing the probability of a random variable X given the joint probability distribution of X with other variables. It is an application of the law of total probability.

Q70 Why is the rotation of components so important in Principle Component Analysis(PCA)?

Rotation in PCA is very important as it maximizes the separation within the variance obtained by all the components because of which interpretation of components would become easier. If the components are not rotated, then we need extended components to describe the variance of the components.

Q71 What is the difference between regularization and normalisation?

Normalisation adjusts the data; regularisation adjusts the prediction function. If your data is on very different scales (especially low to high), you would want to normalise the data. Alter each column to have compatible basic statistics. This can be helpful to make sure there is no loss of accuracy. One of the goals of model training is to identify the signal and ignore the noise if the model is given free rein to minimize error, there is a possibility of suffering from overfitting. Regularization imposes some control on this by providing simpler fitting functions over complex ones.

Q72 When does the linear regression line stop rotating or finds an optimal spot where it is fitted on data?

A place where the highest RSquared value is found, is the place where the line comes to rest. RSquared represents the amount of variance captured by the virtual linear regression line with respect to the total variance captured by the dataset.

Q73 How does the SVM algorithm deal with self-learning?

SVM has a learning rate and expansion rate which takes care of this. The learning rate compensates or penalises the hyperplanes for making all the wrong moves and expansion rate deals with finding the maximum separation area between classes.

Q74 How do you handle outliers in the data?

Outlier is an observation in the data set that is far away from other observations in the data set. We can discover outliers using tools and functions like box plot, scatter plot, Z-Score, IQR score etc. and then handle them based on the visualization we have got. To handle outliers, we can cap at some threshold, use transformations to reduce skewness of the data and remove outliers if they are anomalies or errors.

Q75 Name and define techniques used to find similarities in the recommendation system.

Pearson correlation and Cosine correlation are techniques used to find similarities in recommendation systems.

Q76 Why would you Prune your tree?

In the context of data science or AIML, pruning refers to the process of reducing redundant branches of a decision tree. Decision Trees are prone to overfitting, pruning the tree helps to reduce the size and minimizes the chances of overfitting. Pruning involves turning branches of a decision tree into leaf nodes and removing the leaf nodes from the original branch. It serves as a tool to perform the tradeoff.

Q77 Mention some of the EDA Techniques?

Exploratory Data Analysis (EDA) helps analysts to understand the data better and forms the foundation of better models.

Visualization

- Univariate visualization
- Bivariate visualization
- Multivariate visualization

Missing Value Treatment – Replace missing values with Either Mean/Median

Outlier Detection – Use Boxplot to identify the distribution of Outliers, then Apply IQR to set the boundary for IQR

Q78 What is data augmentation? Can you give some examples?

Data augmentation is a technique for synthesizing new data by modifying existing data in such a way that the target is not changed, or it is changed in a known way.

CV is one of the fields where data augmentation is very useful. There are many modifications that we can do to images:

- Resize
- Horizontal or vertical flip
- Rotate
- Add noise
- Deform
- Modify colors

Each problem needs a customized data augmentation pipeline. For example, on OCR, doing flips will change the text and won't be beneficial; however, resizes and small rotations may help.

Q79 What is Inductive Logic Programming in Machine Learning (ILP)?

Inductive Logic Programming (ILP) is a subfield of machine learning which uses logic programming representing background knowledge and examples.

Q80 What is the difference between inductive machine learning and deductive machine learning?

The difference between inductive machine learning and deductive machine learning are as follows: machine-learning where the model learns by examples from a set of observed instances to draw a generalized conclusion whereas in deductive learning the model first draws the conclusion and then the conclusion is drawn.

Q81 Difference between machine learning and deep learning

Machine learning is a branch of computer science and a method to implement artificial intelligence. This technique provides the ability to automatically learn and improve from experiences without being explicitly programmed.

Deep learning can be said as a subset of machine learning. It is mainly based on the artificial neural network where data is taken as an input and the technique makes intuitive decisions using the artificial neural network.

Q82 What Are The Steps Involved In Machine Learning Project?

As you plan for doing a machine learning project. There are several important steps you must follow to achieve a good working model and they are data collection, data preparation, choosing a machine learning model, training the model, model evaluation, parameter tuning and lastly prediction.

Q83 Differences between Artificial Intelligence and Machine Learning?

Artificial intelligence is a broader prospect than machine learning. Artificial intelligence mimics the cognitive functions of the human brain. The purpose of AI is to carry out a task in an intelligent manner based on algorithms. On the other hand, machine learning is a subclass of artificial intelligence. To develop an autonomous machine in such a way so that it can learn without being explicitly programmed is the goal of machine learning.

Q84 Steps Needed to Choose the Appropriate Machine Learning Algorithm for your Classification problem.

Firstly, you need to have a clear picture of your data, your constraints, and your problems before heading towards different machine learning algorithms. Secondly, you have to understand which type and kind of data you have because it plays a primary role in deciding which algorithm you have to use.

Following this step is the data categorization step, which is a two-step process – categorization by input and categorization by output. The next step is to understand your constraints; that is, what is your data storage capacity? How fast the prediction has to be? etc.

Finally, find the available machine learning algorithms and implement them wisely. Along with that, also try to optimize the hyperparameters which can be done in three ways – grid search, random search, and Bayesian optimization.

Q85 Explain Backpropagation in Machine Learning.

A very important question for your machine learning interview. **Backpropagation** is the algorithm for computing artificial neural networks (ANN). It is used by the gradient descent optimization that exploits the chain rule. By calculating the gradient of the loss function, the weight of the neurons is adjusted to a certain value. To train a multi-layered neural network is the prime motivation of backpropagation so that it can learn the appropriate internal demonstrations. This will help them learn to map any input to its respective output arbitrarily.

Q86 What is the Convex Function?

This question is very often asked in machine learning interviews. A convex function is a continuous function, and the value of the midpoint at every interval in its given domain is less than the numerical mean of the values at the two ends of the interval.

Q87 What's the Relationship between True Positive Rate and Recall?

The True positive rate in machine learning is the percentage of the positives that have been properly acknowledged, and recall is just the count of the results that have been correctly identified and are relevant. Therefore, they are the same things, just having different names. It is also known as sensitivity.

Q88 List some Tools for Parallelizing Machine Learning Algorithms.

Although this question may seem very easy, make sure not to skip this one because it is also very closely related to artificial intelligence and thereby, AI interview questions. Almost all machine learning algorithms are easy to serialize. Some of the basic tools for parallelizing are Matlab, Weka, R, Octave, or the Python-based sci-kit learn.

Q89 What do you mean by Genetic Programming?

Genetic Programming (GP) is almost similar to an Evolutionary Algorithm, a subset of machine learning. Genetic programming software systems implement an algorithm that uses random mutation, a fitness function, crossover, and multiple generations of evolution to resolve a user-defined task. The genetic programming model is based on testing and choosing the best option among a set of results.

Q90 What do you know about Bayesian Networks?

Bayesian Networks also referred to as 'belief networks' or 'casual networks', are used to represent the graphical model for probability relationship among a set of variables.

For example, a Bayesian network can be used to represent the probabilistic relationships between diseases and symptoms. As per the symptoms, the network can also compute the probabilities of the presence of various diseases.

Efficient algorithms can perform inference or learning in Bayesian networks. Bayesian networks which relate the variables (e.g., speech signals or protein sequences) are called dynamic Bayesian networks.

Q91 Which are the two components of the Bayesian logic program?

A Bayesian logic program consists of two components:

- **Logical** It contains a set of Bayesian Clauses, which capture the qualitative structure of the domain.
- **Quantitative** It is used to encode quantitative information about the domain.

Q92 How is machine learning used in day-to-day life?

Most of the people are already using machine learning in their everyday life. Assume that you are engaging with the internet, you are actually expressing your preferences, likes, dislikes through your searches. All these things are picked up by cookies coming on your computer, from this, the behavior of a user is evaluated. It helps to increase the progress of a user through the internet and provide similar suggestions.

The navigation system can also be considered as one of the examples where we are using machine learning to calculate a distance between two places using optimization techniques. Surely, people are going to more engage with machine learning in the near future

Q93 Define Sampling. Why do we need it?

Answer: Sampling is a process of choosing a subset from a target population that would serve as its representative. We use the data from the sample to understand the pattern in the community as a whole. Sampling is necessary because often, we can not gather or process the complete data within a reasonable time.

Q94 What does the term decision boundary mean?

Answer: A decision boundary or a decision surface is a hypersurface which divides the underlying feature space into two subspaces, one for each class. If the decision boundary is a hyperplane, then the classes are linearly separable.

Q95 Define entropy?

Answer: Entropy is the measure of uncertainty associated with random variable Y. It is the expected number of bits required to communicate the value of the variable.

Q96 Indicate the top intents of machine learning?

Answer: The top intents of machine learning are stated below,

- The system gets information from the already established computations to give well-founded decisions and outputs.
- It locates certain patterns in the data and then makes certain predictions on it to provide answers on matters.

Q97 Highlight the differences between the Generative model and the Discriminative model?

The aim of the Generative model is to generate new samples from the same distribution and new data instances, Whereas, the Discriminative model highlights the differences between different kinds of data instances. It tries to learn directly from the data and then classifies the data.

Q98 Identify the most important aptitudes of a machine learning engineer?

Machine learning allows the computer to learn itself without being decidedly programmed. It helps the system to learn from experience and then improve from its mistakes. The intelligence system, which is based on machine learning, can learn from recorded data and past incidents. In-depth knowledge of statistics, probability, data modelling, programming language, as well as CS, Application of ML Libraries and algorithms, and software design is required to become a successful machine learning engineer.

Q99 What is feature engineering? How do you apply it in the process of modelling?

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.

Q100 How can learning curves help create a better model?

Learning curves give the indication of the presence of overfitting or underfitting.

In a learning curve, the training error and cross-validating error are plotted against the number of training data points.

References

1 springboard.com 2 [simplilearn.com](https://www.simplilearn.com) 3 geeksforgeeks.org 4 [elitedatascience.com](https://www.elitedatascience.com) 5 [analyticsvidhya.com](https://www.analyticsvidhya.com) 6 [guru99.com](https://www.guru99.com) 7 [intellipaat.com](https://www.intellipaat.com) 8 towardsdatascience.com 9 mygreatlearning.com 10 [mindmajix.com](https://www.mindmajix.com) 11 toptal.com 12 [glassdoor.co.in](https://www.glassdoor.co.in) 13 [udacity.com](https://www.udacity.com) 14 [educba.com](https://www.educba.com) 15 [analyticsindiamag.com](https://www.analyticsindiamag.com) 16 [ubuntupit.com](https://www.ubuntupit.com) 17 [javatpoint.com](https://www.javatpoint.com) 18 quora.com 19 hackr.io 20 kaggle.com

Top 100 NLP Questions

Steve Nouri

Q1. Which of the following techniques can be used for keyword normalization in NLP, the process of converting a keyword into its base form?

- a. Lemmatization
- b. Soundex
- c. Cosine Similarity
- d. N-grams

Answer : a) Lemmatization helps to get to the base form of a word, e.g. are playing -> play, eating -> eat, etc. Other options are meant for different purposes.

Q2. Which of the following techniques can be used to compute the distance between two word vectors in NLP?

- a. Lemmatization
- b. Euclidean distance
- c. Cosine Similarity
- d. N-grams

Answer : b) and c)

Distance between two word vectors can be computed using Cosine similarity and Euclidean Distance. Cosine Similarity establishes a cosine angle between the vector of two words. A cosine angle close to each other between two word vectors indicates the words are similar and vice versa.

E.g. cosine angle between two words "Football" and "Cricket" will be closer to 1 as compared to angle between the words "Football" and "New Delhi"

Q3. What are the possible features of a text corpus in NLP?

- a. Count of the word in a document
- b. Vector notation of the word
- c. Part of Speech Tag
- d. Basic Dependency Grammar
- e. All of the above

Answer : e) All of the above can be used as features of the text corpus.

Q4. You created a document term matrix on the input data of 20K documents for a Machine learning model. Which of the following can be used to reduce the dimensions of data?

1. Keyword Normalization
2. Latent Semantic Indexing
3. Latent Dirichlet Allocation

- a. only 1
- b. 2, 3
- c. 1, 3
- d. 1, 2, 3

Answer : d)

Q5. Which of the text parsing techniques can be used for noun phrase detection, verb phrase detection, subject detection, and object detection in NLP.

- a. Part of speech tagging
- b. Skip Gram and N-Gram extraction
- c. Continuous Bag of Words
- d. Dependency Parsing and Constituency Parsing

Answer : d)

Q6. Dissimilarity between words expressed using cosine similarity will have values significantly higher than 0.5

- a. True
- b. False

Answer : a)

Q7. Which one of the following are keyword Normalization techniques in NLP

- a. Stemming
- b. Part of Speech
- c. Named entity recognition
- d. Lemmatization

Answer : a) and d)

Part of Speech (POS) and Named Entity Recognition(NER) are not keyword Normalization techniques. Named Entity help you extract Organization, Time, Date, City, etc..type of entities from the given sentence, whereas Part of Speech helps you extract Noun, Verb, Pronoun, adjective, etc..from the given sentence tokens.

Q8. Which of the below are NLP use cases?

- a. Detecting objects from an image
- b. Facial Recognition
- c. Speech Biometric
- d. Text Summarization

Answer : (d)

a) And b) are Computer Vision use cases, and c) is Speech use case.
Only d) Text Summarization is an NLP use case.

Q9. In a corpus of N documents, one randomly chosen document contains a total of T terms and the term “hello” appears K times.

What is the correct value for the product of TF (term frequency) and IDF (inverse-document-frequency), if the term “hello” appears in approximately one-third of the total documents?

- a. $KT * \text{Log}(3)$
- b. $T * \text{Log}(3) / K$
- c. $K * \text{Log}(3) / T$
- d. $\text{Log}(3) / KT$

Answer : (c)

formula for TF is K/T

formula for IDF is $\log(\text{total docs} / \text{no of docs containing "data"})$

$= \log(1 / (1/3))$

$= \log(3)$

Hence correct choice is $K\log(3)/T$

Q10. In NLP, The algorithm decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents

- a. Term Frequency (TF)
- b. Inverse Document Frequency (IDF)
- c. Word2Vec
- d. Latent Dirichlet Allocation (LDA)

Answer : b)

Q11. In NLP, The process of removing words like “and”, “is”, “a”, “an”, “the” from a sentence is called as

- a. Stemming
- b. Lemmatization
- c. Stop word
- d. All of the above

Answer : c) In Lemmatization, all the stop words such as a, an, the, etc.. are removed. One can also define custom stop words for removal.

Q12. In NLP, The process of converting a sentence or paragraph into tokens is referred to as Stemming

- a. True
- b. False

Answer : b) The statement describes the process of tokenization and not stemming, hence it is False.

Q13. In NLP, Tokens are converted into numbers before giving to any Neural Network

- a. True
- b. False

Answer : a) In NLP, all words are converted into a number before feeding to a Neural Network.

Q14 Identify the odd one out

- a. nltk
- b. scikit learn
- c. SpaCy
- d. BERT

Answer : d) All the ones mentioned are NLP libraries except BERT, which is a word embedding

Q15 TF-IDF helps you to establish?

- a. most frequently occurring word in the document
- b. most important word in the document

Answer : b) TF-IDF helps to establish how important a particular word is in the context of the document corpus. TF-IDF takes into account the number of times the word appears in the document and offset by the number of documents that appear in the corpus.

- TF is the frequency of term divided by a total number of terms in the document.
- IDF is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient.

- Tf.idf is then the multiplication of two values TF and IDF.

Q16 In NLP, The process of identifying people, an organization from a given sentence, paragraph is called

- a. Stemming
- b. Lemmatization
- c. Stop word removal
- d. Named entity recognition

Answer : d)

Q17 Which one of the following is not a pre-processing technique in NLP

- a. Stemming and Lemmatization
- b. converting to lowercase
- c. removing punctuations
- d. removal of stop words
- e. Sentiment analysis

Answer : e) Sentiment Analysis is not a pre-processing technique. It is done after pre-processing and is an NLP use case. All other listed ones are used as part of statement pre-processing.

Q18 In text mining, converting text into tokens and then converting them into an integer or floating-point vectors can be done using

- a. CountVectorizer
- b. TF-IDF
- c. Bag of Words
- d. NERs

Answer : a) CountVectorizer helps do the above, while others are not applicable.

text =["Rahul is an avid writer, he enjoys studying understanding and presenting. He loves to play"]

```
vectorizer = CountVectorizer()
```

```
vectorizer.fit(text)
```

```
vector = vectorizer.transform(text)
```

```
print(vector.toarray())
```

output

```
[[1 1 1 1 2 1 1 1 1 1 1 1 1]]
```

The second section of the interview questions covers advanced NLP techniques such as Word2Vec, GloVe word embeddings, and advanced models such as GPT, ELMO, BERT, XLNET based questions, and explanations.

Q19. In NLP, Words represented as vectors are called as Neural Word Embeddings

- a. True
- b. False

Answer : a) Word2Vec, GloVe based models build word embedding vectors that are multidimensional.

Q20. In NLP, Context modeling is supported with which one of the following word embeddings

- 1. a. Word2Vec
- 2. b) GloVe
- 3. c) BERT
- 4. d) All of the above

Answer : c) Only BERT (Bidirectional Encoder Representations from Transformer) supports context modelling where the previous and next sentence context is taken into consideration. In Word2Vec, GloVe only word embeddings are considered and previous and next sentence context is not considered.

Q21. In NLP, Bidirectional context is supported by which of the following embedding

- a. Word2Vec
- b. BERT
- c. GloVe
- d. All the above

Answer : b) Only BERT provides a bidirectional context. The BERT model uses the previous and the next sentence to arrive at the context. Word2Vec and GloVe are word embeddings, they do not provide any context.

Q22. Which one of the following Word embeddings can be custom trained for a specific subject in NLP

- a. Word2Vec
- b. BERT

- c. GloVe
- d. All the above

Answer : b) BERT allows Transform Learning on the existing pre-trained models and hence can be custom trained for the given specific subject, unlike Word2Vec and GloVe where existing word embeddings can be used, no transfer learning on text is possible.

Q23. Word embeddings capture multiple dimensions of data and are represented as vectors

- a. True
- b. False

Answer : a)

Q24. In NLP, Word embedding vectors help establish distance between two tokens

- a. True
- b. False

Answer : a) One can use Cosine similarity to establish distance between two vectors represented through Word Embeddings

Q25. Language Biases are introduced due to historical data used during training of word embeddings, which one amongst the below is not an example of bias

- a. New Delhi is to India, Beijing is to China
- b. Man is to Computer, Woman is to Homemaker

Answer : a)

Statement b) is a bias as it buckets Woman into Homemaker, whereas statement a) is not a biased statement.

Q26. Which of the following will be a better choice to address NLP use cases such as semantic similarity, reading comprehension, and common sense reasoning

- a. ELMo
- b. Open AI's GPT
- c. ULMFit

Answer : b) Open AI's GPT is able to learn complex pattern in data by using the Transformer models Attention mechanism and hence is more suited for complex use cases such as semantic similarity, reading comprehensions, and common sense reasoning.

Q27. Transformer architecture was first introduced with?

- a. GloVe
- b. BERT

- c. Open AI's GPT
- d. ULMFit

Answer : c) ULMFit has an LSTM based Language modeling architecture. This got replaced into Transformer architecture with Open AI's GPT

Q28. Which of the following architecture can be trained faster and needs less amount of training data

- a. LSTM based Language Modelling
- b. Transformer architecture

Answer : b) Transformer architectures were supported from GPT onwards and were faster to train and needed less amount of data for training too.

Q29. Same word can have multiple word embeddings possible with _____?

- a. GloVe
- b. Word2Vec
- c. ELMo
- d. nltk

Answer : c) ELMo word embeddings supports same word with multiple embeddings, this helps in using the same word in a different context and thus captures the context than just meaning of the word unlike in GloVe and Word2Vec. Nltk is not a word embedding.

Q30 For a given token, its input representation is the sum of embedding from the token, segment and position embedding

- a. ELMo
- b. GPT
- c. BERT
- d. ULMFit

Answer : c) BERT uses token, segment and position embedding.

Q31. Trains two independent LSTM language model left to right and right to left and shallowly concatenates them

- a. GPT
- b. BERT
- c. ULMFit
- d. ELMo

Answer : d) ELMo tries to train two independent LSTM language models (left to right and right to left) and concatenates the results to produce word embedding.

Q32. Uses unidirectional language model for producing word embedding

- a. BERT
- b. GPT
- c. ELMo
- d. Word2Vec

Answer : b) GPT is a unidirectional model and word embedding are produced by training on information flow from left to right. ELMo is bidirectional but shallow. Word2Vec provides simple word embedding.

Q33. In this architecture, the relationship between all words in a sentence is modelled irrespective of their position. Which architecture is this?

- a. OpenAI GPT
- b. ELMo
- c. BERT
- d. ULMFit

Answer : c)BERT Transformer architecture models the relationship between each word and all other words in the sentence to generate attention scores. These attention scores are later used as weights for a weighted average of all words' representations which is fed into a fully-connected network to generate a new representation.

Q34. List 10 use cases to be solved using NLP techniques?

- Sentiment Analysis
- Language Translation (English to German, Chinese to English, etc..)
- Document Summarization
- Question Answering
- Sentence Completion
- Attribute extraction (Key information extraction from the documents)
- Chatbot interactions
- Topic classification
- Intent extraction
- Grammar or Sentence correction
- Image captioning
- Document Ranking
- Natural Language inference

Q35. Transformer model pays attention to the most important word in Sentence

- a. True
- b. False

Answer : a) Attention mechanisms in the Transformer model are used to model the relationship between all words and also provide weights to the most important word.

Q36. Which NLP model gives the best accuracy amongst the following?

- a. BERT
- b. XLNET
- c. GPT-2
- d. ELMo

Answer : b) XLNET has given best accuracy amongst all the models. It has outperformed BERT on 20 tasks and achieves state of art results on 18 tasks including sentiment analysis, question answering, natural language inference, etc.

Q37. Permutation Language models is a feature of

- a. BERT
- b. EMMo
- c. GPT
- d. XLNET

Answer : d) XLNET provides permutation-based language modelling and is a key difference from BERT. In permutation language modeling, tokens are predicted in a random manner and not sequential. The order of prediction is not necessarily left to right and can be right to left. The original order of words is not changed but a prediction can be random.

The conceptual difference between BERT and XLNET can be seen from the following diagram.

Q38. Transformer XL uses relative positional embedding

- a. True
- b. False

a) Instead of embedding having to represent the absolute position of a word, Transformer XL uses an embedding to encode the relative distance between the words. This embedding is used to compute the attention score between any 2 words that could be separated by n words before or after.

Q39. What is Naive Bayes algorithm, When we can use this algorithm in NLP?

Naive Bayes algorithm is a collection of classifiers which works on the principles of the Bayes' theorem. This series of NLP model forms a family of algorithms that can be used for a wide range of classification tasks including sentiment prediction, filtering of spam, classifying documents and more.

Naive Bayes algorithm converges faster and requires less training data. Compared to other discriminative models like logistic regression, Naive Bayes model it takes lesser time to train. This algorithm is perfect for use while working with multiple classes and text classification where the data is dynamic and changes frequently.

Q40. Explain Dependency Parsing in NLP?

Dependency Parsing, also known as Syntactic parsing in NLP is a process of assigning syntactic structure to a sentence and identifying its dependency parses. This process is crucial to understand the correlations between the “head” words in the syntactic structure.

The process of dependency parsing can be a little complex considering how any sentence can have more than one dependency parses. Multiple parse trees are known as ambiguities. Dependency parsing needs to resolve these ambiguities in order to effectively assign a syntactic structure to a sentence.

Dependency parsing can be used in the semantic analysis of a sentence apart from the syntactic structuring.

Q41. What is text Summarization?

Text summarization is the process of shortening a long piece of text with its meaning and effect intact. Text summarization intends to create a summary of any given piece of text and outlines the main points of the document. This technique has improved in recent times and is capable of summarizing volumes of text successfully.

Text summarization has proved to a blessing since machines can summarise large volumes of text in no time which would otherwise be really time-consuming. There are two types of text summarization:

- Extraction-based summarization
- Abstraction-based summarization

Q42. What is NLTK? How is it different from Spacy?

NLTK or Natural Language Toolkit is a series of libraries and programs that are used for symbolic and statistical natural language processing. This toolkit contains some of the most powerful libraries that can work on different ML techniques to break down and understand human language. NLTK is used for Lemmatization, Punctuation, Character count, Tokenization, and Stemming. The difference between NLTK and Spacy are as follows:

- While NLTK has a collection of programs to choose from, Spacy contains only the best-suited algorithm for a problem in its toolkit
- NLTK supports a wider range of languages compared to Spacy (Spacy supports only 7 languages)
- While Spacy has an object-oriented library, NLTK has a string processing library
- Spacy can support word vectors while NLTK cannot

Q43. What is information extraction?

Information extraction in the context of Natural Language Processing refers to the technique of extracting structured information automatically from unstructured sources to ascribe meaning to it. This can include extracting information regarding attributes of entities, relationship between different entities and more. The various models of information extraction includes:

- Tagger Module
- Relation Extraction Module
- Fact Extraction Module
- Entity Extraction Module

- Sentiment Analysis Module
- Network Graph Module
- Document Classification & Language Modeling Module

Q44. What is Bag of Words?

Bag of Words is a commonly used model that depends on word frequencies or occurrences to train a classifier. This model creates an occurrence matrix for documents or sentences irrespective of its grammatical structure or word order.

Q45. What is Pragmatic Ambiguity in NLP?

Pragmatic ambiguity refers to those words which have more than one meaning and their use in any sentence can depend entirely on the context. Pragmatic ambiguity can result in multiple interpretations of the same sentence. More often than not, we come across sentences which have words with multiple meanings, making the sentence open to interpretation. This multiple interpretation causes ambiguity and is known as Pragmatic ambiguity in NLP.

Q46. What is a Masked Language Model?

Masked language models help learners to understand deep representations in downstream tasks by taking an output from the corrupt input. This model is often used to predict the words to be used in a sentence.

Q48. What are the best NLP Tools?

Some of the best NLP tools from open sources are:

- SpaCy
- TextBlob
- Textacy
- Natural language Toolkit
- Retext
- NLP.js
- Stanford NLP
- CogcompNLP

Q49. What is POS tagging?

Parts of speech tagging better known as POS tagging refers to the process of identifying specific words in a document and group them as part of speech, based on its context. POS tagging is also known as grammatical tagging since it involves understanding grammatical structures and identifying the respective component.

POS tagging is a complicated process since the same word can be different parts of speech depending on the context. The same generic process used for word mapping is quite ineffective for POS tagging because of the same reason.

Q50. What is NES?

Name entity recognition is more commonly known as NER is the process of identifying specific entities in a text document which are more informative and have a unique context. These often denote places, people, organisations, and more. Even though it seems like these entities are

proper nouns, the NER process is far from identifying just the nouns. In fact, NER involves entity chunking or extraction wherein entities are segmented to categorise them under different predefined classes. This step further helps in extracting information.

Q51 Explain the Masked Language Model?

Masked language modelling is the process in which the output is taken from the corrupted input. This model helps the learners to master the deep representations in downstream tasks. You can predict a word from the other words of the sentence using this model.

Q52 What is pragmatic analysis in NLP?

Pragmatic Analysis: It deals with outside word knowledge, which means knowledge that is external to the documents and/or queries. Pragmatics analysis that focuses on what was described is reinterpreted by what it actually meant, deriving the various aspects of language that require real-world knowledge.

Q53 What is perplexity in NLP?

The word "perplexed" means "puzzled" or "confused", thus Perplexity in general means the inability to tackle something complicated and a problem that is not specified. Therefore, Perplexity in NLP is a way to determine the extent of uncertainty in predicting some text.

In NLP, perplexity is a way of evaluating language models. Perplexity can be high and low; Low perplexity is ethical because the inability to deal with any complicated problem is less while high perplexity is terrible because the failure to deal with a complicated is high.

Q54 What is ngram in NLP?

N-gram in NLP is simply a sequence of n words, and we also conclude the sentences which appeared more frequently, for example, let us consider the progression of these three words:

- New York (2 gram)
- The Golden Compass (3 gram)
- She was there in the hotel (4 gram)

Now from the above sequence, we can easily conclude that sentence (a) appeared more frequently than the other two sentences, and the last sentence(c) is not seen that often. Now if we assign probability in the occurrence of an n-gram, then it will be advantageous. It would help in making next-word predictions and in spelling error corrections.

Q55 Explain differences between AI, Machine Learning and NLP

Artificial Intelligence	Machine Learning	Natural Language Processing
It is the technique to create smarter machines	Machine Learning is the term used for systems that learn from experience.	This is the set of system that has the ability to understand the language
AI includes human intervention	Machine Learning purely involves the working of computers and no human intervention.	NLP links both computer and human languages.
Artificial intelligence is a broader concept than Machine Learning	ML is a narrow concept and is a subset of AI.	

Q56 Why self-attention is awesome?

“In terms of computational complexity, self-attention layers are faster than recurrent layers when the sequence length n is smaller than the representation dimensionality d , which is most often the case with sentence representations used by state-of-the-art models in machine translations, such as word-piece and byte-pair representations.” — from Attention is all you need

Q57 What are stop words?

Stop words are said to be useless data for a search engine. Words such as articles, prepositions, etc. are considered as stop words. There are stop words such as was, were, is, am, the, a, an, how, why, and many more. In Natural Language Processing, we eliminate the stop words to understand and analyze the meaning of a sentence. The removal of stop words is one of the most important tasks for search engines. Engineers design the algorithms of search engines in such a way that they ignore the use of stop words. This helps show the relevant search result for a query.

Q58 What is Latent Semantic Indexing (LSI)?

Latent semantic indexing is a mathematical technique used to improve the accuracy of the information retrieval process. The design of LSI algorithms allows machines to detect the hidden (latent) correlation between semantics (words). To enhance information understanding, machines generate various concepts that associate with the words of a sentence.

The technique used for information understanding is called singular value decomposition. It is generally used to handle static and unstructured data. The matrix obtained for singular value decomposition contains rows for words and columns for documents. This method best suits to identify components and group them according to their types.

The main principle behind LSI is that words carry a similar meaning when used in a similar context. Computational LSI models are slow in comparison to other models. However, they are good at contextual awareness that helps improve the analysis and understanding of a text or a document.

Q60 What are Regular Expressions?

A regular expression is used to match and tag words. It consists of a series of characters for matching strings.

Suppose, if A and B are regular expressions, then the following are true for them:

- If $\{\epsilon\}$ is a regular language, then ϵ is a regular expression for it.
- If A and B are regular expressions, then $A + B$ is also a regular expression within the language $\{A, B\}$.
- If A and B are regular expressions, then the concatenation of A and B ($A.B$) is a regular expression.
- If A is a regular expression, then A^* (A occurring multiple times) is also a regular expression.

Q61 What are unigrams, bigrams, trigrams, and n-grams in NLP?

When we parse a sentence one word at a time, then it is called a unigram. The sentence parsed two words at a time is a bigram.

When the sentence is parsed three words at a time, then it is a trigram. Similarly, n-gram refers to the parsing of n words at a time.

Example: To understand unigrams, bigrams, and trigrams, you can refer to the below diagram:

Q62 What are the steps involved in solving an NLP problem?

Below are the steps involved in solving an NLP problem:

1. Gather the text from the available dataset or by web scraping

2. Apply stemming and lemmatization for text cleaning
3. Apply feature engineering techniques
4. Embed using word2vec
5. Train the built model using neural networks or other Machine Learning techniques
6. Evaluate the model's performance
7. Make appropriate changes in the model
8. Deploy the model

Q63. There have some various common elements of natural language processing. Those elements are very important for understanding NLP properly, can you please explain the same in details with an example?

Answer:

There have a lot of components normally using by natural language processing (NLP). Some of the major components are explained below:

- Extraction of Entity: It actually identifying and extracting some critical data from the available information which help to segmentation of provided sentence on identifying each entity. It can help in identifying one human that it's fictional or real, same kind of reality identification for any organization, events or any geographic location etc.
- The analysis in a syntactic way: it mainly helps for maintaining ordering properly of the available words.

Q64 In the case of processing natural language, we normally mentioned one common terminology NLP and binding every language with the same terminology properly. Please explain in details about this NLP terminology with an example?

Answer:

This is the basic NLP Interview Questions asked in an interview. There have some several factors available in case of explaining natural language processing. Some of the key factors are given below:

- Vectors and Weights: Google Word vectors, length of TF-IDF, varieties documents, word vectors, TF-IDF.
- Structure of Text: Named Entities, tagging of part of speech, identifying the head of the sentence.
- Analysis of sentiment: Know about the features of sentiment, entities available for the sentiment, sentiment common dictionary.
- Classification of Text: Learning supervising, set off a train, set of validation in Dev, Set of define test, a feature of the individual text, LDA.
- Reading of Machine Language: Extraction of the possible entity, linking with an individual entity, DBpedia, some libraries like Pikes or FRED.

Q65 Explain briefly about word2vec

Word2Vec embeds words in a lower-dimensional vector space using a shallow neural network. The result is a set of word-vectors where vectors close together in vector space have similar meanings based on context, and word-vectors distant to each other have differing meanings. For example, apple and orange would be close together and apple and gravity would be relatively far.

There are two versions of this model based on skip-grams (SG) and continuous-bag-of-words (CBOW).

Q66 What are the metrics used to test an NLP model?

Accuracy, Precision, Recall and F1. Accuracy is the usual ratio of the prediction to the desired output. But going just by accuracy is naive considering the complexities involved.

Q67 What are some ways we can preprocess text input?

Here are several preprocessing steps that are commonly used for NLP tasks:

- case normalization: we can convert all input to the same case (lowercase or uppercase) as a way of reducing our text to a more canonical form
- punctuation/stop word/white space/special characters removal: if we don't think these words or characters are relevant, we can remove them to reduce the feature space
- lemmatizing/stemming: we can also reduce words to their inflectional forms (i.e. walks → walk) to further trim our vocabulary
- generalizing irrelevant information: we can replace all numbers with a <NUMBER> token or all names with a <NAME> token

Q68 How does the encoder-decoder structure work for language modelling?

The encoder-decoder structure is a deep learning model architecture responsible for several state of the art solutions, including Machine Translation.

The input sequence is passed to the encoder where it is transformed to a fixed-dimensional vector representation using a neural network. The transformed input is then decoded using another neural network. Then, these outputs undergo another transformation and a softmax layer. The final output is a vector of probabilities over the vocabularies. Meaningful information is extracted based on these probabilities.

Q69 What are attention mechanisms and why do we use them?

This was a followup to the encoder-decoder question. Only the output from the last time step is passed to the decoder, resulting in a loss of information learned at previous time steps. This information loss is compounded for longer text sequences with more time steps.

Attention mechanisms are a function of the hidden weights at each time step. When we use attention in encoder-decoder networks, the fixed-dimensional vector passed to the decoder becomes a function of all vectors outputted in the intermediary steps.

Two commonly used attention mechanisms are additive attention and multiplicative attention. As the names suggest, additive attention is a weighted sum while multiplicative attention is a weighted multiplier of the hidden weights. During the training process, the model also learns weights for the attention mechanisms to recognize the relative importance of each time step.

Q70 How would you implement an NLP system as a service, and what are some pitfalls you might face in production?

This is less of a NLP question than a question for productionizing machine learning models. There are however certain intricacies to NLP models.

Without diving too much into the productionization aspect, an ideal Machine Learning service will have:

- endpoint(s) that other business systems can use to make inference
- a feedback mechanism for validating model predictions
- a database to store predictions and ground truths from the feedback
- a workflow orchestrator which will (upon some signal) re-train and load the new model for serving based on the records from the database + any prior training data
- some form of model version control to facilitate rollbacks in case of bad deployments
- post-production accuracy and error monitoring

Q71 How can we handle misspellings for text input?

By using word embeddings trained over a large corpus (for instance, an extensive web scrape of billions of words), the model vocabulary would include common misspellings by design. The model can then learn the relationship between misspelled and correctly spelled words to recognize their semantic similarity.

We can also preprocess the input to prevent misspellings. Terms not found in the model vocabulary can be mapped to the “closest” vocabulary term using:

- edit distance between strings
- phonetic distance between word pronunciations
- keyword distance to catch common typos

Q72 Which of the following models can perform tweet classification with regards to context mentioned above?

- A) Naive Bayes
- B) SVM
- C) None of the above

Solution: (C)

Since, you are given only the data of tweets and no other information, which means there is no target variable present. One cannot train a supervised learning model, both svm and naive bayes are supervised learning techniques.

Q73 You have created a document term matrix of the data, treating every tweet as one document. Which of the following is correct, in regards to document term matrix?

1. Removal of stopwords from the data will affect the dimensionality of data
2. Normalization of words in the data will reduce the dimensionality of data

3. Converting all the words in lowercase will not affect the dimensionality of the data

- A) Only 1
- B) Only 2
- C) Only 3
- D) 1 and 2
- E) 2 and 3
- F) 1, 2 and 3

Solution: (D)

Choices A and B are correct because stopword removal will decrease the number of features in the matrix, normalization of words will also reduce redundant features, and, converting all words to lowercase will also decrease the dimensionality.

Q74 Which of the following features can be used for accuracy improvement of a classification model?

- A) Frequency count of terms
- B) Vector Notation of sentence
- C) Part of Speech Tag
- D) Dependency Grammar
- E) All of these

Solution: (E)

All of the techniques can be used for the purpose of engineering features in a model.

Q75 What percentage of the total statements are correct with regards to Topic Modeling?

1. It is a supervised learning technique
2. LDA (Linear Discriminant Analysis) can be used to perform topic modeling
3. Selection of number of topics in a model does not depend on the size of data
4. Number of topic terms are directly proportional to size of the data

- A) 0
- B) 25
- C) 50
- D) 75
- E) 100

Solution: (A)

LDA is unsupervised learning model, LDA is latent Dirichlet allocation, not Linear discriminant analysis. Selection of the number of topics is directly proportional to the size of the data, while number of topic terms is not directly proportional to the size of the data. Hence none of the statements are correct.

Q76 In Latent Dirichlet Allocation model for text classification purposes, what does alpha and beta hyperparameter represent-

- A) Alpha: number of topics within documents, beta: number of terms within topics False
- B) Alpha: density of terms generated within topics, beta: density of topics generated within terms False
- C) Alpha: number of topics within documents, beta: number of terms within topics False
- D) Alpha: density of topics generated within documents, beta: density of terms generated within topics True

Solution: (D)

Option D is correct

Q77 What is the problem with ReLu?

- Exploding gradient(Solved by gradient clipping)
- Dying ReLu — No learning if the activation is 0 (Solved by parametric relu)
- Mean and variance of activations is not 0 and 1.(Partially solved by subtracting around 0.5 from activation. Better explained in fastai videos)

Q78 What is the difference between learning latent features using SVD and getting embedding vectors using deep network?

SVD uses linear combination of inputs while a neural network uses nonlinear combination.

Q79 What is the information in the hidden and cell state of LSTM?

Hidden stores all the information till that time step and cell state stores particular information that might be needed in the future time step.

Number of parameters in an LSTM model with bias

$4(mh+h^2+h)$ where m is input vectors size and h is output vectors size a.k.a. hidden

The point to see here is that mh dictates the model size as $m \gg h$. Hence it's important to have a small vocab.

Time complexity of LSTM

$seq_length * hidden^2$

Time complexity of transformer

$seq_length^2 * hidden$

When hidden size is more than the seq_length (which is normally the case), transformer is faster than LSTM.

Q80 When is self-attention not faster than recurrent layers?

When the sequence length is greater than the representation dimensions. This is rare.

Q81 What is the benefit of learning rate warm-up?

Learning rate warm-up is a learning rate schedule where you have low (or lower) learning rate at the beginning of training to avoid divergence due to unreliable gradients at the beginning. As the model becomes more stable, the learning rate would increase to speed up convergence.

Q82 What's the difference between hard and soft parameter sharing in multi-task learning?

Hard sharing is where we train for all the task at the same time and update our weights using all the losses whereas soft sharing is where we train for one task at a time.

Q83 What's the difference between BatchNorm and LayerNorm?

BatchNorm computes the mean and variance at each layer for every minibatch whereas LayerNorm computes the mean and variance for every sample for each layer independently. Batch normalisation allows you to set higher learning rates, increasing speed of training as it reduces the instability of initial starting weights.

Q84 Difference between BatchNorm and LayerNorm?

BatchNorm — Compute the mean and var at each layer for every minibatch

LayerNorm — Compute the mean and var for every single sample for each layer independently

Q85 Why does the transformer block have LayerNorm instead of BatchNorm?

Looking at the advantages of LayerNorm, it is robust to batch size and works better as it works at the sample level and not batch level.

Q86 What changes would you make to your deep learning code if you knew there are errors in your training data?

We can do label smoothing where the smoothing value is based on % error. If any particular class has known error, we can also use class weights to modify the loss.

Q87 What are the tricks used in ULMFiT? (Not a great questions but checks the awareness)

- LM tuning with task text
- Weight dropout
- Discriminative learning rates for layers
- Gradual unfreezing of layers
- Slanted triangular learning rate schedule

This can be followed up with a question on explaining how they help.

Q88 Tell me a language model which doesn't use dropout

ALBERT v2 — This throws a light on the fact that a lot of assumptions we take for granted are not necessarily true. The regularisation effect of parameter sharing in ALBERT is so strong that dropouts are not needed. (ALBERT v1 had dropouts.)

Q89 What are the differences between GPT and GPT-2? (From Lilian Weng)

- [Layer normalization](#) was moved to the input of each sub-block, similar to a residual unit of type [“building block”](#) (differently from the original type [“bottleneck”](#), it has batch normalization applied before weight layers).
- An additional layer normalization was added after the final self-attention block.
- A modified initialization was constructed as a function of the model depth.
- The weights of residual layers were initially scaled by a factor of $1/\sqrt{n}$ where n is the number of residual layers.
- Use larger vocabulary size and context size.

Q90 What are the differences between GPT and BERT?

- GPT is not bidirectional and has no concept of masking
- BERT adds next sentence prediction task in training and so it also has a segment embedding

Q91 What are the differences between BERT and ALBERT v2?

- Embedding matrix factorisation(helps in reducing no. of parameters)
- No dropout
- Parameter sharing(helps in reducing no. of parameters and regularisation)

Q92 How does parameter sharing in ALBERT affect the training and inference time?

No effect. Parameter sharing just decreases the number of parameters.

Q93 How would you reduce the inference time of a trained NN model?

- Serve on GPU/TPU/FPGA
- 16 bit quantisation and served on GPU with fp16 support
- Pruning to reduce parameters
- Knowledge distillation (To a smaller transformer model or simple neural network)
- Hierarchical softmax/Adaptive softmax
- You can also cache results as explained here.

Q94 Would you use BPE with classical models?

Of course! BPE is a smart tokeniser and it can help us get a smaller vocabulary which can help us find a model with less parameters.

Q95 How would you make an arxiv papers search engine? (I was asked — How would you make a plagiarism detector?)

Get top k results with TF-IDF similarity and then rank results with

- semantic encoding + cosine similarity
- a model trained for ranking

Q96 Get top k results with TF-IDF similarity and then rank results with

- semantic encoding + cosine similarity
- a model trained for ranking

Q97 How would you make a sentiment classifier?

This is a trick question. The interviewee can say all things such as using transfer learning and latest models but they need to talk about having a neutral class too otherwise you can have really good accuracy/f1 and still, the model will classify everything into positive or negative.

The truth is that a lot of news is neutral and so the training needs to have this class. The interviewee should also talk about how he will create a dataset and his training strategies like the selection of language model, language model fine-tuning and using various datasets for multi-task learning.

Q98 What is the difference between regular expression and regular grammar?

A regular expression is the representation of natural language in the form of mathematical expressions containing a character sequence. On the other hand, regular grammar is the generator of natural language, defining a set of defined rules and syntax which the strings in the natural language must follow.

Q99 Why should we use Batch Normalization?

Once the interviewer has asked you about the fundamentals of deep learning architectures, they would move on to the key topic of improving your deep learning model's performance.

Batch Normalization is one of the techniques used for reducing the training time of our deep learning algorithm. Just like normalizing our input helps improve our logistic regression model, we can normalize the activations of the hidden layers in our deep learning model as well:

Q100 How is backpropagation different in RNN compared to ANN?

In Recurrent Neural Networks, we have an additional loop at each node:

This loop essentially includes a time component into the network as well. This helps in capturing sequential information from the data, which could not be possible in a generic artificial neural network.

This is why the backpropagation in RNN is called Backpropagation through Time, as in backpropagation at each time step.

Top 100 Questions on Computer Vision

By Steve Nouri

Q1 Which of the following is a challenge when dealing with computer vision problems?

Variations due to geometric changes (like pose, scale, etc), Variations due to photometric factors (like illumination, appearance, etc) and Image occlusion. All the above-mentioned options are challenges in computer vision

Q2 Consider an image with width and height as 100×100. Each pixel in the image can have a color from Grayscale, i.e. values. How much space would this image require for storing?

The answer will be 8x100x100 because 8 bits will be required to represent a number from 0-256

Q3 Why do we use convolutions for images rather than just FC layers?

Firstly, convolutions preserve, encode, and actually use the spatial information from the image. If we used only FC layers we would have no relative spatial information. Secondly, Convolutional Neural Networks (CNNs) have a partially built-in translation in-variance, since each convolution kernel acts as it's own filter/feature detector.

Q4 What makes CNN's translation-invariant?

As explained above, each convolution kernel acts as it's own filter/feature detector. So let's say you're doing object detection, it doesn't matter where in the image the object is since we're going to apply the convolution in a sliding window fashion across the entire image anyways.

Q5 Why do we have max-pooling in classification CNNs?

for a role in Computer Vision. Max-pooling in a CNN allows you to reduce computation since your feature maps are smaller after the pooling. You don't lose too much semantic information since you're taking the maximum activation. There's also a theory that max-pooling contributes a bit to giving CNN's more translation in-variance. Check out this great video from Andrew Ng on the [benefits of max-pooling](#).

Q6 Why do segmentation CNN's typically have an encoder-decoder style/structure?

The encoder CNN can basically be thought of as a feature extraction network, while the decoder uses that information to predict the image segments by "decoding" the features and upscaling to the original image size.

Q7 What is the significance of Residual Networks?

The main thing that residual connections did was allow for direct feature access from previous layers. This makes information propagation throughout the network much easier. One very interesting paper about this shows how using local skip connections gives the network a type of ensemble multi-path structure, giving features multiple paths to propagate throughout the network.

Q8 What is batch normalization and why does it work?

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. The idea is then to normalize the inputs of each layer in such a way that they have a mean output activation of zero and a standard deviation of one. This is done for each individual mini-batch at each layer i.e compute the mean and variance of that mini-batch alone, then normalize. This is analogous to how the inputs to networks are standardized. How does this help? We know that normalizing the inputs to a network helps it learn. But a network is just a series of layers, where the output of one layer becomes the input to the next. That means we can think of any layer in a neural network as the first layer of a smaller subsequent network. Thought of as a series of neural networks feeding into each other, we normalize the output of one layer before applying the activation function and then feed it into the following layer (sub-network).

Q9 Why would you use many small convolutional kernels such as 3x3 rather than a few large ones?

This is very well explained in the [VGGNet paper](#). There are 2 reasons: First, you can use several smaller kernels rather than few large ones to get the same receptive field and capture more spatial context, but with the smaller kernels you are using less parameters and computations. Secondly, because with smaller kernels you will be using more filters, you'll be able to use more activation functions and thus have a more discriminative mapping function being learned by your CNN.

Q10 What is Precision?

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances

Precision = true positive / (true positive + false positive)

Q11 What is Recall?

Recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Recall = true positive / (true positive + false negative)

Q12 Define F1-score.

It is the weighted average of precision and recall. It considers both false positive and false negatives into account. It is used to measure the model's performance.

$F1\text{-Score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$

Q13 What is cost function?

The cost function is a scalar function that Quantifies the error factor of the Neural Network. Lower the cost function better than the Neural network. Eg: MNIST Data set to classify the image, the input image is digit 2 and the Neural network wrongly predicts it to be 3

Q14 List different activation neurons or functions

- Linear Neuron
- Binary Threshold Neuron
- Stochastic Binary Neuron
- Sigmoid Neuron
- Tanh function
- Rectified Linear Unit (ReLU)

Q15 Define Learning rate.

The learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect to the loss gradient.

Q16 What is Momentum (w.r.t NN optimization)?

Momentum lets the optimization algorithm remembers its last step, and adds some proportion of it to the current step. This way, even if the algorithm is stuck in a flat region, or a small local minimum, it can get out and continue towards the true minimum.

Q17 What is the difference between Batch Gradient Descent and Stochastic Gradient Descent?

Batch gradient descent computes the gradient using the whole dataset. This is great for convex or relatively smooth error manifolds. In this case, we move somewhat directly towards an optimum solution, either local or global. Additionally, batch gradient descent, given an annealed learning rate, will eventually find the minimum located in its basin of attraction.

Stochastic gradient descent (SGD) computes the gradient using a single sample. SGD works well (Not well, I suppose, but better than batch gradient descent) for error manifolds that have lots of local maxima/minima. In this case, the somewhat noisier gradient calculated using the reduced number of samples tends to jerk the model out of local minima into a region that hopefully is more optimal.

Q18 Epoch vs Batch vs Iteration.

Epoch: one forward pass and one backward pass of all the training examples

Batch: examples processed together in one pass (forward and backward)

Iteration: number of training examples / Batch size

Q19 What is the vanishing gradient?

As we add more and more hidden layers, backpropagation becomes less and less useful in passing information to the lower layers. In effect, as information is passed back, the gradients begin to vanish and become small relative to the weights of the networks.

Q20 What are dropouts?

Dropout is a simple way to prevent a neural network from overfitting. It is the dropping out of some of the units in a neural network. It is similar to the natural reproduction process, where nature produces offsprings by combining distinct genes (dropping out others) rather than strengthening the co-adapting of them.

Q21 Can you explain the differences between supervised, unsupervised, and reinforcement learning?

In supervised learning, we train a model to learn the relationship between input data and output data. We need to have labeled data to be able to do supervised learning.

With unsupervised learning, we only have unlabeled data. The model learns a representation of the data. Unsupervised learning is frequently used to initialize the parameters of the model when we have a lot of unlabeled data and a small fraction of labeled data. We first train an unsupervised model and, after that, we use the weights of the model to train a supervised model. In reinforcement learning, the model has some input data and a reward depending on the output of the model. The model learns a policy that maximizes the reward. Reinforcement learning has been applied successfully to strategic games such as Go and even classic Atari video games.

Q22 What is data augmentation? Can you give some examples?

Data augmentation is a technique for synthesizing new data by modifying existing data in such a way that the target is not changed, or it is changed in a known way. Computer vision is one of the fields where data augmentation is very useful. There are many modifications that we can do to images:

- Resize
- Horizontal or vertical flip
- Rotate, Add noise, Deform
- Modify colors Each problem needs a customized data augmentation pipeline. For example, on OCR, doing flips will change the text and won't be beneficial; however, resizes and small rotations may help.

Q23 What are the components of GAN?

- Generator
- Discriminator

Q24 What's the difference between a generative and discriminative model?

A generative model will learn categories of data while a discriminative model will simply learn the distinction between different categories of data. Discriminative models will generally outperform generative models on classification tasks.

Q25 What is Linear Filtering?

[Linear filtering](#) is a neighborhood operation, which means that the output of a pixel's value is decided by the weighted sum of the values of the input pixels.

Q26 How can you achieve Blurring through Gaussian Filter?

This is the most common technique for blurring or smoothing an image. This filter [improves the resulting pixel](#) found at the center and slowly minimizes the effects as pixels move away from the center. This filter can also help in removing noise in an image

Q27 What is Non-Linear Filtering? How it is used?

Linear filtering is easy to use and implement. In some cases, this method is enough to get the necessary output. However, an increase in performance can be obtained through non-linear filtering. Through non-linear filtering, we can have more control and achieve better results when we encounter a more complex computer vision task.

Q28 Explain Median Filtering.

The median filter is an example of a non-linear filtering technique. This technique is commonly used for minimizing the noise in an image. It operates by inspecting the image pixel by pixel and taking the place of each pixel's value with the value of the neighboring pixel median.

Some techniques in [detecting and matching](#) features are:

- Lucas-Kanade
- Harris
- Shi-Tomasi
- SUSAN (smallest uni value segment assimilating nucleus)
- MSER (maximally stable extremal regions)
- SIFT (scale-invariant feature transform)
- HOG (histogram of oriented gradients)
- FAST (features from accelerated segment test)
- SURF (speeded-up robust features)

Q29 Describe the Scale Invariant Feature Transform (SIFT) algorithm

SIFT solves the problem of detecting the corners of an object even if it is scaled. Steps to implement this algorithm:

- Scale-space extrema detection – This step will identify the locations and scales that can still be recognized from different angles or views of the same object in an image.
- Keypoint localization – When possible key points are located, they would be refined to get accurate results. This would result in the elimination of points that are low in contrast or points that have edges that are deficiently localized.
- Orientation assignment – In this step, a consistent orientation is assigned to each key point to attain invariance when the image is being rotated.
- Keypoint matching – In this step, the key points between images are now linked to recognizing their nearest neighbors.

Q30 Why Speeded-Up Robust Features (SURF) came into existence?

SURF was introduced to as a speed-up version of SIFT. Though SIFT can detect and describe key points of an object in an image, still this algorithm is slow.

Q31 What is Oriented FAST and rotated BRIEF (ORB)?

This algorithm is a great possible substitute for SIFT and SURF, mainly because it performs better in computation and matching. It is a combination of fast keypoint detector and brief descriptor, which contains a lot of alterations to improve performance. It is also a great alternative in terms of cost because the SIFT and SURF algorithms are patented, which means that you need to buy them for their utilization.

Q32 What is image segmentation?

In computer vision, [segmentation](#) is the process of extracting pixels in an image that is related. Segmentation algorithms usually take an image and produce a group of contours (the boundary of an object that has well-defined edges in an image) or a mask where a set of related pixels are assigned to a unique color value to identify it.

Popular image segmentation techniques:

- Active contours
- Level sets
- Graph-based merging
- Mean Shift
- Texture and intervening contour-based normalized cuts

Q33 What is the purpose of semantic segmentation?

The [purpose of semantic segmentation](#) is to categorize every pixel of an image to a certain class or label. In semantic segmentation, we can see what is the class of a pixel by simply looking directly at the color, but one downside of this is that we cannot identify if two colored masks belong to a certain object.

Q34 Explain instance segmentation.

In semantic segmentation, the only thing that matters to us is the class of each pixel. This would somehow lead to a problem that we cannot identify if that class belongs to the same object or not. Semantic segmentation cannot identify if two objects in an image are separate entities. So to solve this problem, instance segmentation was created. This segmentation can identify two different objects of the same class. For example, if an image has two sheep in it, the sheep will be detected and masked with different colors to differentiate what instance of a class they belong to.

Q35 How is panoptic segmentation different from semantic/instance segmentation?

Panoptic segmentation is basically a union of semantic and instance segmentation. In [panoptic segmentation](#), every pixel is classified by a certain class and those pixels that have several instances of a class are also determined. For example, if an image has two cars, these cars will

be masked with different colors. These colors represent the same class — car — but point to different instances of a certain class.

Q36 Explain the problem of recognition in computer vision.

Recognition is one of the toughest challenges in the concepts in computer vision. Why is recognition hard? For the human eyes, recognizing an object's features or attributes would be very easy. Humans can recognize multiple objects with very small effort. However, this does not apply to a machine. It would be very hard for a machine to recognize or detect an object because these objects vary. They vary in terms of viewpoints, sizes, or scales. Though these things are still challenges faced by most computer vision systems, they are still making advancements or approaches for solving these daunting tasks.

Q37 What is Object Recognition?

Object recognition is used for indicating an object in an image or video. This is a product of machine learning and deep learning algorithms. Object recognition tries to acquire this innate human ability, which is to understand certain features or visual detail of an image.

Q38 What is Object Detection and it's real-life use cases?

[Object detection](#) in computer vision refers to the ability of machines to pinpoint the location of an object in an image or video. A lot of companies have been using object detection techniques in their system. They use it for face detection, web images, and security purposes.

Q39 Describe Optical Flow, its uses, and assumptions.

Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is a 2D vector field where each vector is a displacement vector showing the movement of points from the first frame to the second

Optical flow has many applications in areas like :

- Structure from Motion
- Video Compression
- Video Stabilization ...

Optical flow works on several assumptions:

1. The pixel intensities of an object do not change between consecutive frames.
2. Neighboring pixels have similar motion.

Q40 What is Histogram of Oriented Gradients (HOG)?

HOG stands for Histograms of Oriented Gradients. HOG is a type of “feature descriptor”. The intent of a feature descriptor is to generalize the object in such a way that the same object (in this case a person) produces as close as possible to the same feature descriptor when viewed under different conditions. This makes the classification task easier.

Q41 What is BOV: Bag-of-visual-words (BOV)?

BOV also called the bag of keypoints, is based on vector quantization. Similar to HOG features, BOV features are histograms that count the number of occurrences of certain patterns within a patch of the image.

Q42 What is Poselets? Where are poselets used?

Poselets rely on manually added extra keypoints such as “right shoulder”, “left shoulder”, “right knee” and “left knee”. They were originally used for human pose estimation

Q43 Explain Textons in context of CNNs

A texton is the minimal building block of vision. The computer vision literature does not give a strict definition for textons, but edge detectors could be one example. One might argue that deep learning techniques with Convolution Neuronal Networks (CNNs) learn textons in the first filters.

Q44 What is Markov Random Fields (MRFs)?

MRFs are undirected probabilistic graphical models which are a wide-spread model in computer vision. The overall idea of MRFs is to assign a random variable for each feature and a random variable for each pixel

Q45 Explain the concept of superpixel?

A superpixel is an image patch that is better aligned with intensity edges than a rectangular patch. Superpixels can be extracted with any segmentation algorithm, however, most of them produce highly irregular superpixels, with widely varying sizes and shapes. A more regular space tessellation may be desired.

Q46 What is Non-maximum suppression(NMS) and where is it used?

NMS is often used along with edge detection algorithms. The image is scanned along the image gradient direction, and if pixels are not part of the local maxima they are set to zero. It is widely used in object detection algorithms.

Q47 Describe the use of Computer Vision in Healthcare.

Computer vision has also been an important part of advances in health-tech. Computer vision algorithms can help automate tasks such as detecting cancerous moles in skin images or finding symptoms in x-ray and MRI scans.

Q48 Describe the use of Computer Vision in Augmented Reality & Mixed Reality

Computer vision also plays an important role in augmented and mixed reality, the technology that enables computing devices such as smartphones, tablets, and smart glasses to overlay and embed virtual objects on real-world imagery. Using computer vision, AR gear detects objects in the real world in order to determine the locations on a device's display to place a virtual object. For instance, computer vision algorithms can help AR applications detect planes such as

tabletops, walls, and floors, a very important part of establishing depth and dimensions and placing virtual objects in the physical world.

Q49 Describe the use of Computer Vision in Facial Recognition

Computer vision also plays an important role in facial recognition applications, the technology that enables computers to match images of people's faces to their identities. Computer vision algorithms detect facial features in images and compare them with databases of face profiles. Consumer devices use facial recognition to authenticate the identities of their owners. Social media apps use facial recognition to detect and tag users. Law enforcement agencies also rely on facial recognition technology to identify criminals in video feeds.

Q50 Describe the use of Computer Vision in Self-Driving Cars

Computer vision enables self-driving cars to make sense of their surroundings. Cameras capture video from different angles around the car and feed it to computer vision software, which then processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects, and pedestrians. The self-driving car can then steer its way on streets and highways, avoid hitting obstacles, and (hopefully) safely drive its passengers to their destination.

Q51 Explain famous Computer Vision tasks using a single image example.

Many popular computer vision applications involve trying to recognize things in photographs; for example:

Object Classification: What broad category of object is in this photograph?

Object Identification: Which type of a given object is in this photograph?

Object Verification: Is the object in the photograph?

Object Detection: Where are the objects in the photograph?

Object Landmark Detection: What are the key points for the object in the photograph?

Object Segmentation: What pixels belong to the object in the image?

Object Recognition: What objects are in this photograph and where are they?

Q52 Explain the distinction between Computer Vision and Image Processing.

Computer vision is distinct from image processing.

[Image processing](#) is the process of creating a new image from an existing image, typically simplifying or enhancing the content in some way. It is a type of digital signal processing and is not concerned with understanding the content of an image.

A given computer vision system may require image processing to be applied to raw input, e.g. pre-processing images.

Examples of image processing include:

- Normalizing photometric properties of the image, such as brightness or color.
- Cropping the bounds of the image, such as centering an object in a photograph.
- Removing digital noise from an image, such as digital artifacts from low light levels.

Q53 Explain business use cases in computer vision.

- Optical character recognition (OCR)
- Machine inspection
- Retail (e.g. automated checkouts)
- 3D model building (photogrammetry)
- Medical imaging
- Automotive safety
- Match move (e.g. merging CGI with live actors in movies)
- Motion capture (mocap)
- Surveillance
- Fingerprint recognition and biometrics

Q54 What is the Boltzmann Machine?

One of the most basic Deep Learning models is a Boltzmann Machine, resembling a simplified version of the Multi-Layer Perceptron. This model features a visible input layer and a hidden layer -- just a two-layer neural net that makes stochastic decisions as to whether a neuron should be on or off. Nodes are connected across layers, but no two nodes of the same layer are connected.

Q56 What Is the Role of Activation Functions in a Neural Network?

At the most basic level, an activation function decides whether a neuron should be fired or not. It accepts the weighted sum of the inputs and bias as input to any activation function. Step function, Sigmoid, ReLU, Tanh, and Softmax are examples of activation functions.

Q57 What Is the Difference Between a Feedforward Neural Network and Recurrent Neural Network?

A Feedforward Neural Network signals travel in one direction from input to output. There are no feedback loops; the network considers only the current input. It cannot memorize previous inputs (e.g., CNN).

Q58 What Are the Applications of a Recurrent Neural Network (RNN)?

The [RNN](#) can be used for sentiment analysis, text mining, and image captioning. Recurrent Neural Networks can also address time series problems such as predicting the prices of stocks in a month or quarter.

Q59 What Are the Softmax and ReLU Functions?

Softmax is an activation function that generates the output between zero and one. It divides each output, such that the total sum of the outputs is equal to one. Softmax is often used for output layers.

Q60 What Are Hyperparameters?

With neural networks, you're usually working with hyperparameters once the data is formatted correctly. A hyperparameter is a parameter whose value is set before the learning process begins. It determines how a network is trained and the structure of the network (such as the number of hidden units, the learning rate, epochs, etc.).

Q61 What Will Happen If the Learning Rate Is Set Too Low or Too High?

When your learning rate is too low, training of the model will progress very slowly as we are making minimal updates to the weights. It will take many updates before reaching the minimum point. If the learning rate is set too high, this causes undesirable divergent behavior to the loss function due to drastic updates in weights. It may fail to converge (model can give a good output) or even diverge (data is too chaotic for the network to train).

Q62 How Are Weights Initialized in a Network?

There are two methods here: we can either initialize the weights to zero or assign them randomly. Initializing all weights to 0: This makes your model similar to a linear model. All the neurons and every layer perform the same operation, giving the same output and making the deep net useless. Initializing all weights randomly: Here, the weights are assigned randomly by initializing them very close to 0. It gives better accuracy to the model since every neuron performs different computations. This is the most commonly used method.

Q63 What Are the Different Layers on CNN?

There are four layers in CNN:

1. Convolutional Layer - the layer that performs a convolutional operation, creating several smaller picture windows to go over the data.
2. ReLU Layer - it brings non-linearity to the network and converts all the negative pixels to zero. The output is a rectified feature map.
3. Pooling Layer - pooling is a down-sampling operation that reduces the dimensionality of the feature map.
4. Fully Connected Layer - this layer recognizes and classifies the objects in the image.

Q64 What is Pooling on CNN, and How Does It Work?

Pooling is used to reduce the spatial dimensions of a CNN. It performs down-sampling operations to reduce the dimensionality and creates a pooled feature map by sliding a filter matrix over the input matrix.

Q65 How Does an LSTM Network Work?

Long-Short-Term Memory (LSTM) is a special kind of recurrent neural network capable of learning long-term dependencies, remembering information for long periods as its default behavior. There are three steps in an LSTM network:

- Step 1: The network decides what to forget and what to remember.
- Step 2: It selectively updates cell state values.
- Step 3: The network decides what part of the current state makes it to the output.

Q66 What Is the Difference Between Epoch, Batch, and Iteration in Deep Learning?

- Epoch - Represents one iteration over the entire dataset (everything put into the training model).
- Batch - Refers to when we cannot pass the entire dataset into the neural network at once, so we divide the dataset into several batches.
- Iteration - if we have 10,000 images as data and a batch size of 200. then an epoch should run 50 iterations (10,000 divided by 50).

Q67 Why Is Tensorflow the Most Preferred Library in Deep Learning?

[Tensorflow](#) provides both C++ and Python APIs, making it easier to work on and has a faster compilation time compared to other Deep Learning libraries like Keras and Torch. Tensorflow supports both CPU and GPU computing devices.

Q68 What Do You Mean by Tensor in Tensorflow?

A tensor is a mathematical object represented as arrays of higher dimensions. These arrays of data with different dimensions and ranks fed as input to the neural network are called "Tensors."

Q69 Explain a Computational Graph.

Everything in TensorFlow is based on creating a computational graph. It has a network of nodes where each node operates, Nodes represent mathematical operations, and edges represent tensors. Since data flows in the form of a graph, it is also called a "DataFlow Graph."

Q70 What Is an Auto-encoder?

This Neural Network has three layers in which the input neurons are equal to the output neurons. The network's target outside is the same as the input. It uses dimensionality reduction to restructure the input. It works by compressing the image input to a latent space representation then reconstructing the output from this representation.

Q71 Can we have the same bias for all neurons of a hidden layer?

Essentially, you can have a different bias value at each layer or at each neuron as well. However, it is best if we have a bias matrix for all the neurons in the hidden layers as well. A point to note is that both these strategies would give you very different results.

Q72 In a neural network, what if all the weights are initialized with the same value?

In simplest terms, if all the neurons have the same value of weights, each hidden unit will get exactly the same signal. While this might work during forward propagation, the derivative of the cost function during backward propagation would be the same every time.

In short, there is no learning happening by the network! What do you call the phenomenon of the model being unable to learn any patterns from the data? Yes, [underfitting](#). Therefore, if all weights have the same initial value, this would lead to underfitting.

Q73 What is the role of weights and bias in a neural network?

This is a question best explained with a real-life example. Consider that you want to go out today to play a cricket match with your friends. Now, a number of factors can affect your decision-making, like:

- How many of your friends can make it to the game?
- How much equipment can all of you bring?
- What is the temperature outside?

And so on. These factors can change your decision greatly or not too much. For example, if it is raining outside, then you cannot go out to play at all. Or if you have only one bat, you can share it while playing as well. The magnitude by which these factors can affect the game is called the weight of that factor.

Factors like the weather or temperature might have a higher weight, and other factors like equipment would have a lower weight.

Q74 Why does a Convolutional Neural Network (CNN) work better with image data?

The key to this question lies in the Convolution operation. Unlike humans, the machine sees the image as a matrix of pixel values. Instead of interpreting a shape like a petal or an ear, it just identifies curves and edges.

Thus, instead of looking at the entire image, it helps to just read the image in parts. Doing this for a 300 x 300-pixel image would mean dividing the matrix into smaller 3 x 3 matrices and dealing with them one by one. This is convolution.

Q75 Why do RNNs work better with text data?

The main component that differentiates Recurrent Neural Networks (RNN) from the other models is the addition of a loop at each node. This loop brings the recurrence mechanism in RNNs. In a basic Artificial Neural Network (ANN), each input is given the same weight and fed to the network at the same time. So, for a sentence like “I saw the movie and hated it”, it would be difficult to capture the information which associates “it” with the “movie”.

Q76 In a CNN, if the input size 5 X 5 and the filter size is 7 X 7, then what would be the size of the output?

This is a pretty intuitive answer. As we saw above, we perform the convolution on ‘x’ one step at a time, to the right, and in the end, we got Z with dimensions 2 X 2, for X with dimensions 3 X 3. Thus, to make the input size similar to the filter size, we make use of padding – adding 0s to the input matrix such that its new size becomes at least 7 X 7. Thus, the output size would be using the formula:

Dimension of image = (n, n) = 5 X 5

Dimension of filter = (f,f) = 7 X 7

Padding = 1 (adding 1 pixel with value 0 all around the edges)

Dimension of output will be (n+2p-f+1) X (n+2p-f+1) = 1 X 1

Q77 What's the difference between valid and same padding in a CNN?

This question has more chances of being a follow-up question to the previous one. Or if you have explained how you used CNNs in a computer vision task, the interviewer might ask this question along with the details of the padding parameters.

- Valid Padding: When we do not use any padding. The resultant matrix after convolution will have dimensions $(n - f + 1) \times (n - f + 1)$
- Same padding: Adding padded elements all around the edges such that the output matrix will have the same dimensions as that of the input matrix

Q78 What are the applications of transfer learning in Deep Learning?

I am sure you would have a doubt as to why a relatively simple question was included in the Intermediate Level. The reason is the sheer volume of subsequent questions it can generate! The use of [transfer learning](#) has been one of the key milestones in deep learning. Training a large model on a huge dataset, and then using the final parameters on smaller simpler datasets has led to defining breakthroughs in the form of Pretrained Models. Be it Computer Vision or NLP, pretrained models have become the norm in research and in the industry. Some popular examples include BERT, ResNet, GPT-2, VGG-16, etc, and many more.

Q79 Why is GRU faster as compared to LSTM?

As you can see, the LSTM model can become quite complex. In order to still retain the functionality of retaining information across time and yet not make a too complex model, we need GRUs. Basically, in GRUs, instead of having an additional Forget gate, we combine the input and Forget gates into a single Update Gate:

Q80 How is the transformer architecture better than RNN?

Advancements in deep learning have made it possible to solve many tasks in Natural Language Processing. Networks/Sequence models like RNNs, LSTMs, etc. are specifically used for this purpose – so as to capture all possible information from a given sentence, or a paragraph. However, sequential processing comes with its caveats:

- It requires high processing power
- It is difficult to execute in parallel because of its sequential nature

Q81 How Can We Scale GANs Beyond Image Synthesis?

Aside from applications like image-to-image translation and domain-adaptation most GAN successes have been in image synthesis. Attempts to use GANs beyond images have focused on three domains: Text, Structured Data and Audio

Q82 How Should we Evaluate GANs and When Should We Use Them?

When it comes to evaluating GANs, there are many proposals but little consensus. Suggestions include:

- Inception Score and FID - Both these scores use a pre-trained image classifier and both have known issues. A common criticism is that these scores measure 'sample quality' and don't really capture 'sample diversity'.
- MS-SSIM - propose using MS-SSIM to separately evaluate diversity, but this technique has some issues and hasn't really caught on.
- AIS - propose putting a Gaussian observation model on the outputs of a GAN and using annealed importance sampling to estimate the log-likelihood under this model, but show that estimates computed this way are inaccurate in the case where the GAN generator is also a flow model. The generator being a flow model allows for the computation of exact log-likelihoods in this case.
- Geometry Score - suggest computing geometric properties of the generated data manifold and comparing those properties to the real data.
- Precision and Recall - attempt to measure both the 'precision' and 'recall' of GANs.
- Skill Rating - have shown that trained GAN discriminators can contain useful information with which evaluation can be performed.

Q83 What should we use GANs for?

If you want an actual density model, GANs probably isn't the best choice. There is now good experimental evidence that GANs learn a 'low support' representation of the target dataset, which means there may be substantial parts of the test set to which a GAN (implicitly) assigns zero likelihood.

Q84 How should we evaluate GANs on these perceptual tasks?

Ideally, we would just use a human judge, but this is expensive. A cheap proxy is to see if a classifier can distinguish between real and fake examples. This is called a classifier two-sample test (C2STs). The main issue with C2STs is that if the Generator has even a minor defect that's systematic across samples (e.g.,) this will dominate the evaluation.

Q85 Explain the problem of Vanishing Gradients in GANs

[Research](#) has suggested that if your discriminator is too good, then generator training can fail due to [vanishing gradients](#). In effect, an optimal discriminator doesn't provide enough information for the generator to make progress.

Attempts to Remedy

- Wasserstein loss: The [Wasserstein loss](#) is designed to prevent vanishing gradients even when you train the discriminator to optimality.
- Modified minimax loss: The [original GAN paper](#) proposed a [modification to minimax loss](#) to deal with vanishing gradients.

Q86 What is Mode Collapse and why it is a big issue?

Usually, you want your GAN to produce a wide variety of outputs. You want, for example, a different face for every random input to your face generator.

However, if a generator produces an especially plausible output, the generator may learn to produce only that output. In fact, the generator is always trying to find the one output that seems most plausible to the discriminator.

If the generator starts producing the same output (or a small set of outputs) over and over again, the discriminator's best strategy is to learn to always reject that output. But if the next generation of discriminator gets stuck in a local minimum and doesn't find the best strategy, then it's too easy for the next generator iteration to find the most plausible output for the current discriminator. Each iteration of generator over-optimizes for a particular discriminator and the discriminator never manages to learn its way out of the trap. As a result, the generators rotate through a small set of output types. This form of GAN failure is called mode collapse.

Q87 Explain Progressive GANs

In a progressive GAN, the generator's first layers produce very low resolution images, and subsequent layers add details. This technique allows the GAN to train more quickly than comparable non-progressive GANs, and produces higher resolution images.

Q88 Explain Conditional GANs

Conditional GANs train on a labeled data set and let you specify the label for each generated instance. For example, an unconditional MNIST GAN would produce random digits, while a conditional MNIST GAN would let you specify which digit the GAN should generate.

Instead of modeling the joint probability $P(X, Y)$, conditional GANs model the conditional probability $P(X | Y)$.

For more information about conditional GANs, see [Mirza et al, 2014](#).

Q89 Explain Image-to-Image Translation

Image-to-Image translation GANs take an image as input and map it to a generated output image with different properties. For example, we can take a mask image with blob of color in the shape of a car, and the GAN can fill in the shape with photorealistic car details.

Q90 Explain CycleGAN

CycleGANs learn to transform images from one set into images that could plausibly belong to another set. For example, a CycleGAN produced the righthand image below when given the lefthand image as input. It took an image of a horse and turned it into an image of a zebra.

Q91 What is Super-resolution?

Super-resolution GANs increase the resolution of images, adding detail where necessary to fill in blurry areas. For example, the blurry middle image below is a downsampled version of the original image on the left. Given the blurry image, a GAN produced the sharper image on the right:

Q92 Explain different problems in GANs

Many GAN models suffer the following major problems:

- Non-convergence: the model parameters oscillate, destabilize and never converge,
- Mode collapse: the generator collapses which produces limited varieties of samples,

- Diminished gradient: the discriminator gets too successful that the generator gradient vanishes and learns nothing,
- Unbalance between the generator and discriminator causing overfitting, &
- Highly sensitive to the hyperparameter selections.

Q93 Describe Cost v.s. image quality in GANS?

In a discriminative model, the loss measures the accuracy of the prediction and we use it to monitor the progress of the training. However, the loss in GAN measures how well we are doing compared with our opponent. Often, the generator cost increases but the image quality is actually improving. We fall back to examine the generated images manually to verify the progress. This makes model comparison harder which leads to difficulties in picking the best model in a single run. It also complicates the tuning process.

Q94 Why Singular Value Decomposition (SVD) is used in Computer Vision?

The singular value decomposition is the most common and useful decomposition in computer vision. The goal of computer vision is to explain the three-dimensional world through two-dimensional pictures.

Q95 What Is Image Transform?

An image can be expanded in terms of a discrete set of basis arrays called basis images. Hence, these basis images can be generated by unitary matrices. An $N \times N$ image can be viewed as an $N^2 \times 1$ vector. It provides a set of coordinates or basis vectors for vector space.

Q96 List The Hardware Oriented Color Models?

They are as follows.

- RGB model
- CMY model
- YIQ model
- HSI model

Q96 What Is The Need For Transform?

Answer: The need for transform is most of the signals or images are time-domain signal (ie) signals can be measured with a function of time. This representation is not always best. Any person of the mathematical transformations is applied to the signal or images to obtain further information from that signal. Particularly, for image processing.

Q97 What is FPN?

Feature Pyramid Network (FPN) is a feature extractor designed with a feature pyramid concept to improve accuracy and speed. Images are first to pass through the CNN pathway, yielding semantically rich final layers. Then to regain better resolution, it creates a top-down pathway by upsampling this feature map.