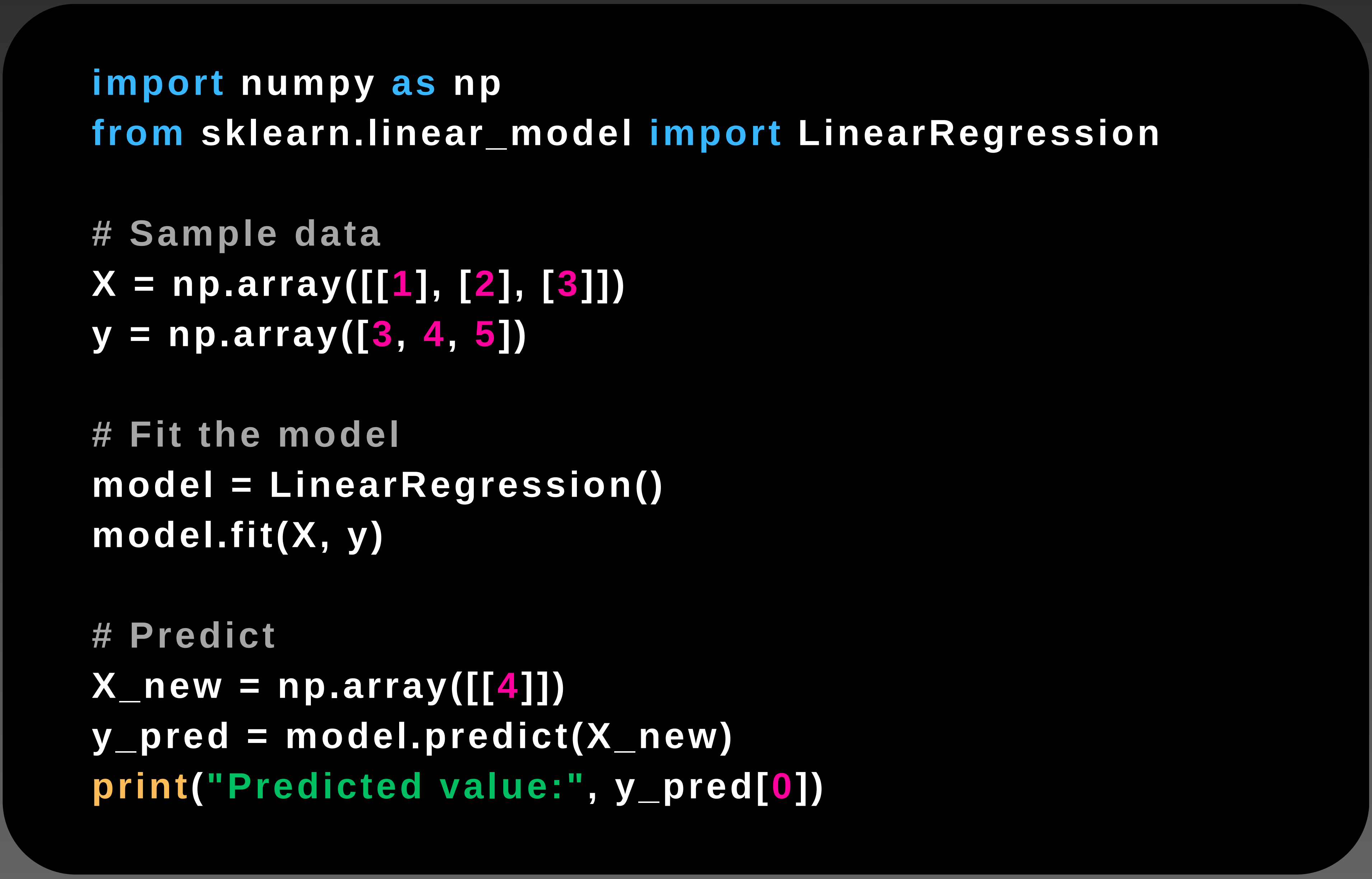
BEST MACHINE LEARNING MODELS FOR PREDICTIONS!

SHIVAM MODI
@learneverythingai

LINEAR REGRESSION

Starting with the basics! Linear Regression is perfect for predicting continuous numerical values. Its simplicity and interpretability make it a go-to model for many tasks. Here's a code snippet in Python:



```python
import numpy as np
from sklearn.linear_model import LinearRegression

# Sample data
X = np.array([[1], [2], [3]])
y = np.array([3, 4, 5])

# Fit the model
model = LinearRegression()
model.fit(X, y)

# Predict
X_new = np.array([[4]])
y_pred = model.predict(X_new)
print("Predicted value:", y_pred[0])
```

DECISION TREES

Decision Trees are great for both classification and regression tasks. They can handle complex relationships in data. Check out the Python code snippet:

```python
from sklearn.tree import DecisionTreeClassifier

# Sample data
X = [[30, 1], [25, 0], [32, 1]]
y = [0, 1, 1]

# Fit the model
model = DecisionTreeClassifier()
model.fit(X, y)

# Predict
X_new = [[28, 1]]
y_pred = model.predict(X_new)
print("Predicted class:", y_pred[0])
```

RANDOM FOREST

A powerhouse ensemble model! Random Forest combines multiple Decision Trees to improve accuracy and reduce overfitting. Here's the Python code:

```python
from sklearn.ensemble import RandomForestRegressor

# Sample data
X = [[10], [20], [30]]
y = [5, 10, 15]

# Fit the model
model = RandomForestRegressor(n_estimators=100)
model.fit(X, y)

# Predict
X_new = [[25]]
y_pred = model.predict(X_new)
print("Predicted value:", y_pred[0])
```

@LEARNEVERYTHINGAI

GRADIENT BOOSTING MACHINES

GBMs create strong predictive models by combining weak learners sequentially. It's a powerful technique widely used in competitions! Check out the Python code:

```python
from sklearn.ensemble import GradientBoostingClassifier

# Sample data
X = [[1], [2], [3]]
y = [0, 1, 0]

# Fit the model
model = GradientBoostingClassifier()
model.fit(X, y)

# Predict
X_new = [[4]]
y_pred = model.predict(X_new)
print("Predicted class:", y_pred[0])
```

SHIVAM MODI
@learneverythingai

NEURAL NETWORKS

The holy grail of ML! Neural Networks excel in handling complex tasks like image recognition, natural language processing, and more. Here's a code snippet using TensorFlow:

```python
import tensorflow as tf

# Sample data
X = tf.constant([[1.0], [2.0], [3.0]])
y = tf.constant([[2.0], [4.0], [6.0]])

# Build the model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(1)
])

# Compile and train
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X, y, epochs=100)

# Predict
X_new = tf.constant([[4.0]])
y_pred = model.predict(X_new)
print("Predicted value:", y_pred[0][0])
```

@learneverythingai

Like this Post?

- Follow Me
- Share with your friends
- Check out my previous posts

SAVE THIS

SHIVAM MODI
@learneverythingai

Follow

SHARE

www.learneverythingai.com