@learneverythingai

APPLICATION OF PYTHON IN DATA SCIENCE

SHIVAM MODI
@learneverythingai

DATA MANIPULATION

Python's powerful libraries like Pandas allow data scientists to efficiently clean, reshape, and explore data, making it a cornerstone for any data-driven project.

```python
import pandas as pd

# Load data from a CSV file
data = pd.read_csv('data.csv')

# Perform data cleaning and filtering
cleaned_data = data.dropna()

# Explore data using basic statistics
mean_age = cleaned_data['Age'].mean()
print(f"Mean Age: {mean_age}")
```

SHIVAM MODI
@learneverythingai

MACHINE LEARNING

Python's extensive libraries like Scikit-learn, TensorFlow, and Keras empower data scientists to build and deploy sophisticated machine learning models for tasks like classification, regression, and clustering.

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Prepare data for training and testing
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2)

# Initialize the logistic regression model
model = LogisticRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on test data
predictions = model.predict(X_test)
```

SHIVAM MODI
@learneverythingai

DATA VISUALIZATION

With libraries like Matplotlib and Seaborn, Python enables data scientists to create stunning visualizations, making complex data easily understandable and impactful.

```python
import matplotlib.pyplot as plt

# Create a bar chart
plt.bar(categories, counts)
plt.xlabel('Categories')
plt.ylabel('Counts')
plt.title('Data Distribution')
plt.show()
```

SHIVAM MODI
@learneverythingai

NATURAL LANGUAGE PROCESSING (NLP)

Python offers libraries like NLTK and spaCy, allowing data scientists to process and analyze human language data, powering applications like sentiment analysis and language translation.

```
import nltk

# Tokenize the text
tokens = nltk.word_tokenize(text)

# Perform part-of-speech tagging
pos_tags = nltk.pos_tag(tokens)

# Extract named entities
named_entities = nltk.chunk.ne_chunk(pos_tags)
```

SHIVAM MODI
@learneverythingai

BIG DATA ANALYSIS

Python can work seamlessly with big data frameworks like Apache Spark, making it an ideal language for processing and analyzing massive datasets efficiently.

```python
from pyspark import SparkContext
from pyspark.sql import SparkSession

# Create a Spark context
sc = SparkContext('local', 'BigDataAnalysis')
spark = SparkSession(sc)

# Load data from a large dataset
data = spark.read.csv('big_data.csv')

# Perform data transformations and analysis
result = data.groupBy('category').agg({'sales': 'sum'})
```

SHIVAM MODI
@learneverythingai

TIME SERIES ANALYSIS

Python libraries like Statsmodels and Prophet enable data scientists to perform time series forecasting, critical for financial and demand forecasting.

```
import statsmodels.api as sm

# Prepare data for time series analysis
time_series = pd.Series(data, index=dates)

# Perform time series forecasting
model = sm.tsa.ARIMA(time_series, order=(1, 1, 1))
results = model.fit()
forecast = results.forecast(steps=10)
```

SHIVAM MODI
@learneverythingai

WEB SCRAPING

Python's libraries such as Beautiful Soup and Scrapy facilitate data extraction from websites, an essential skill for gathering information from the web.

```python
import requests
from bs4 import BeautifulSoup

# Send a request to the website
url = 'https://example.com'
response = requests.get(url)

# Parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# Extract data from the website
data = soup.find('div', class_='content').text
```

SHIVAM MODI
@learneverythingai

DEEP LEARNING

Python's frameworks like PyTorch and TensorFlow provide a robust foundation for building and training deep neural networks, revolutionizing image recognition, language processing, and more.

```python
import torch
import torch.nn as nn

# Create a simple neural network
class SimpleNN(nn.Module):
    def __init__(self):
        super(SimpleNN, self).__init__()
        self.fc1 = nn.Linear(784, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

SHIVAM MODI
@learneverythingai

@learneverythingai

SHIVAM MODI
@learneverythingai

LIKE THIS POST?

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

www.learneverythingai.com

Follow

SHARE

SHIVAM MODI
@learneverythingai