**Introduction to Python for Data Analytics**

Python is a general-purpose, high-level programming language that is widely used for data analytics due to its simplicity, readability, and vast ecosystem of libraries.

**Key Features of Python:**

* **General-Purpose Language:** Can be used across various domains.
* **High-Level Language:** Uses human-readable syntax.
* **Interpreted Language:** Executes code directly without compilation.
* **Dynamically Typed:** No need to declare data types explicitly.

**Example:**

python

CopyEdit

x = 10 # No need to specify int type

print("Hello, Python!")

**2. Python Libraries for Data Analytics**

Several libraries make Python a preferred choice for data analytics by offering pre-built functions and structures to handle data effectively.

**a) NumPy (Numerical Python)**

Used for numerical computations and handling large datasets efficiently.

* **Key Features:**
  + Supports multi-dimensional arrays.
  + Offers mathematical operations (addition, multiplication, statistical functions).
  + Faster processing compared to Python lists.

**Example:**

python

CopyEdit

import numpy as np

arr = np.array([1, 2, 3, 4])

print(arr.mean()) # Output: 2.5

**b) Pandas (Data Manipulation)**

Pandas provides powerful data structures like DataFrames and Series for data manipulation and analysis.

* **Key Features:**
  + Handles missing data efficiently.
  + Provides operations for merging, grouping, and filtering data.
  + Works well with different data sources like CSV, Excel, SQL.

**Example:**

python

CopyEdit

import pandas as pd

data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}

df = pd.DataFrame(data)

print(df.head()) # View top rows

**c) Matplotlib (Data Visualization)**

Used to create static, animated, and interactive visualizations in Python.

* **Key Features:**
  + Generates charts like line, bar, scatter, and histograms.
  + Customizable with labels, colors, and styles.

**Example:**

python

CopyEdit

import matplotlib.pyplot as plt

plt.plot([1, 2, 3], [4, 5, 6])

plt.title("Sample Plot")

plt.show()

**d) Seaborn (Statistical Data Visualization)**

Built on top of Matplotlib, Seaborn provides beautiful statistical visualizations.

* **Key Features:**
  + Works well with Pandas DataFrames.
  + Provides advanced visualizations like heatmaps, pair plots.

**Example:**

python

CopyEdit

import seaborn as sns

tips = sns.load\_dataset('tips')

sns.boxplot(x='day', y='total\_bill', data=tips)

plt.show()

**3. Data Handling and Preprocessing**

Data preprocessing is a critical step in analytics to clean and prepare data for analysis.

**Key Steps in Preprocessing:**

1. **Handling Missing Values:**

python

CopyEdit

df.fillna(df.mean(), inplace=True) # Replace missing values with column mean

1. **Filtering Data:**

python

CopyEdit

filtered\_df = df[df['Age'] > 25]

1. **Converting Data Types:**

python

CopyEdit

df['Age'] = df['Age'].astype(int)

**4. Exploratory Data Analysis (EDA)**

EDA is used to understand the underlying patterns and distributions in data.

**Common EDA Tasks:**

* **Summarizing Data:**

python

CopyEdit

print(df.describe()) # Statistical summary of numerical columns

* **Checking Correlations:**

python

CopyEdit

print(df.corr()) # Identify relationships between numerical columns

* **Visualizing Distributions:**

python

CopyEdit

sns.histplot(df['Age'], bins=10)

plt.show()

**5. Working with Large Datasets**

Handling large datasets efficiently is crucial in data analytics.

**Tips:**

* Use **chunksize** to load data in parts.
* Use **vectorized operations** in Pandas instead of loops.
* Convert data to efficient formats like Parquet for faster processing.

**Example:**

python

CopyEdit

df\_chunk = pd.read\_csv('large\_file.csv', chunksize=1000)

for chunk in df\_chunk:

print(chunk.head())

**6. Grouping and Aggregation**

Grouping data helps in summarizing large datasets based on specific criteria.

**Example:**

python

grouped\_df = df.groupby('Category')['Sales'].sum()

print(grouped\_df)

**7. Merging and Joining Data**

Combining data from different sources or tables.

**Example:**

python

CopyEdit

df1 = pd.DataFrame({'ID': [1, 2], 'Name': ['Alice', 'Bob']})

df2 = pd.DataFrame({'ID': [1, 2], 'Salary': [5000, 6000]})

merged\_df = pd.merge(df1, df2, on='ID')

print(merged\_df)

**8. Basic SQL Operations in Pandas**

Pandas can mimic SQL-like operations such as selection, filtering, and grouping.

**Example:**

python

CopyEdit

df[df['Age'] > 30] # SQL equivalent: SELECT \* FROM df WHERE Age > 30

**9. Pivot Tables in Pandas**

Pivot tables summarize data similar to Excel.

**Example:**

python

CopyEdit

pivot\_df = df.pivot\_table(index='Category', values='Sales', aggfunc='sum')

print(pivot\_df)

**10. Automation with Python**

Python can automate repetitive data analysis tasks such as data extraction, transformation, and loading (ETL).

**Example:**

python

CopyEdit

def automate\_task():

df = pd.read\_csv('data.csv')

df\_cleaned = df.dropna()

df\_cleaned.to\_csv('cleaned\_data.csv', index=False)

automate\_task()