TP03: Full Data Analysis Project using Python & Power BI

A **Full Data Analysis Project using **Python** and **Power BI** involves multiple stages, from data collection and cleaning to analysis, visualization, and reporting. Below is a step-by-step guide to building a complete data analysis project:**

****Steps**:**

1. **Data Collection**: Load sales data from a CSV file.

2. **Data Cleaning**: Handle missing values, remove duplicates, and format dates.

3. **EDA**: Analyze sales trends, product performance, and regional sales.

4. **Advanced Analysis**: Perform customer segmentation using clustering.

5. **Power BI Dashboard**

---

1. Define the Project Scope

- **Objective**: What insights are you trying to derive? (e.g., sales trends, customer behavior, financial performance)

- **Data Sources**: Identify where the data will come from (e.g., databases, APIs, CSV files, Excel sheets).

- **Deliverables**: What will the final output look like? (e.g., a Power BI dashboard, a Python report, or both).

---

2. Data Collection

- Use Python to collect data from various sources.
- Examples:
  - CSV/Excel files
  - Web scraping (e.g., using BeautifulSoup or Scrapy)

```python
import pandas as pd

# Example: Load data from a CSV file
data = pd.read_csv('sales_data.csv')
```

3. Data Cleaning and Preprocessing

- Handle missing values, duplicates, and outliers.
- Transform data into a usable format (e.g., date formatting, categorical encoding).
- Normalize or standardize data if needed.

```python
# Drop missing values
data.dropna(inplace=True)

# Remove duplicates
data.drop_duplicates(inplace=True)

# Convert date column to datetime format
```

```
data['date'] = pd.to_datetime(data['date'])

# Handle outliers (e.g., using Z-score or IQR)
from scipy.stats import zscore
data = data[(zscore(data['sales']) < 3)]
```

---

4. Exploratory Data Analysis (EDA)**

- Perform descriptive statistics to understand the data.
- Visualize data distributions, correlations, and trends.
- Identify patterns and anomalies.

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Descriptive statistics
print(data.describe())

# Correlation matrix
corr_matrix = data.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()

# Sales trends over time
```

```
data.groupby('date')['sales'].sum().plot(kind='line')
plt.title('Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.show()
```

---

5. Advanced Analysis (Optional)**

- Perform statistical analysis or machine learning models.
- Examples:
  - Regression analysis
  - Clustering (e.g., customer segmentation)
  - Time series forecasting

```python
# Example: Linear Regression
from sklearn.linear_model import LinearRegression

X = data[['feature1', 'feature2']]
y = data['target']

model = LinearRegression()
model.fit(X, y)

# Predictions
```

predictions = model.predict(X)
```

---

**6. Data Visualization in Power BI**

- Load the cleaned and analyzed data into Power BI.
- Create interactive dashboards and reports.
- Steps:
  1. Import the dataset into Power BI (e.g., from a CSV or database).
  2. Use Power Query to perform additional transformations if needed.
  3. Create visualizations (e.g., bar charts, line charts, maps).
  4. Add filters, slicers, and interactive elements.

---

**7. Build a Power BI Dashboard**

- **Key Visualizations**:
  - Sales trends over time.
  - Top-performing products or regions.
  - Customer segmentation.
  - Key performance indicators (KPIs).
- **Interactive Features**:
  - Filters for date ranges, regions, or product categories.
  - Drill-down capabilities for detailed insights.

---
```

8. Automate the Process

- Use Python to automate data collection, cleaning, and analysis.
- Schedule regular updates using tools like **Task Scheduler** (Windows) or **Cron** (Linux/Mac).
- Use Power BI's **Automated Refresh** feature to keep dashboards up-to-date.

---

9. Share and Collaborate

- Publish the Power BI dashboard to the **Power BI Service**.
- Share the dashboard with stakeholders.
- Set up role-based access control for security.

---

10. Document the Project

- Write a report or documentation explaining:
  - The data sources and collection methods.
  - The cleaning and transformation steps.
  - The analysis performed and insights derived.
  - How to use the Power BI dashboard.

Example Project: Sales Data Analysis

Objective: Analyze sales data to identify trends, top-performing products, and regional performance.

Steps:

1. **Data Collection**: Load sales data from a CSV file.

2. **Data Cleaning**: Handle missing values, remove duplicates, and format dates.

3. **EDA**: Analyze sales trends, product performance, and regional sales.

4. **Advanced Analysis**: Perform customer segmentation using clustering.

5. **Power BI Dashboard**:

   - Visualize sales trends over time.

   - Show top 10 products by revenue.

   - Create a map showing regional sales performance.

6. **Automation**: Schedule weekly data updates and dashboard refreshes.

Tools Used:

- **Python Libraries**:

  - Pandas, NumPy (data manipulation)

  - Matplotlib, Seaborn (visualization)

  - Scikit-learn (machine learning)

- **Power BI**:

  - Data modeling and visualization

Sample Python Code for the Project

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load data
data = pd.read_csv('sales_data.csv')

# Clean data
data.dropna(inplace=True)
data['date'] = pd.to_datetime(data['date'])

# EDA
print(data.describe())
sns.pairplot(data)
plt.show()

# Sales trends
data.groupby('date')['sales'].sum().plot(kind='line')
plt.title('Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.show()

# Top products
```

```
top_products = data.groupby('product')['sales'].sum().nlargest(10)
top_products.plot(kind='bar')
plt.title('Top 10 Products by Sales')
plt.xlabel('Product')
plt.ylabel('Sales')
plt.show()
```

Power BI Dashboard Features

- **Home Page**: Overview of key metrics (total sales, average order value, etc.).
- **Sales Trends**: Line chart showing sales over time.
- **Product Performance**: Bar chart of top-selling products.
- **Regional Sales**: Map showing sales by region.
- **Filters**: Date range, product category, and region filters.

By combining **Python** for data processing and analysis with **Power BI** for visualization and reporting, you can create a powerful end-to-end data analysis project that delivers actionable insights.