Python Project- Sales Outlook

The "Sales Outlook Project" involves analyzing sales data to derive insights and make informed business decisions. It includes tasks such as data cleaning, exploratory data analysis, aggregation, visualization, and database management. Students will use Python libraries like Pandas, NumPy, and Matplotlib to manipulate and analyze the data, perform SQL queries for data retrieval, and create database tables to store the results.

Data Tables:

1. **Sales Table**
    - Columns:
        - sales_id (Primary Key)
        - product_id (Foreign Key)
        - customer_id (Foreign Key)
        - sales_amount
        - date
        - region_id (Foreign Key)
2. **Products Table**
    - Columns:
        - product_id (Primary Key)
        - product_name
        - category
        - price
3. **Customers Table**
    - Columns:
        - customer_id (Primary Key)
        - customer_name
        - email
        - address
4. **Regions Table**
    - Columns:
        - region_id (Primary Key)
        - region_name

Here are sample CSV files for the "Sales Outlook Project":

sales.csv:

sales_id,product_id,customer_id,sales_amount,date,region_id

```
1,101,201,100,2022-01-01,1
2,102,202,150,2022-01-02,2
3,103,203,200,2022-01-03,1
4,104,204,120,2022-01-04,2
5,105,205,180,2022-01-05,1
```

products.csv:

```
product_id,product_name,category,price
101,Product A,Electronics,500
102,Product B,Home Appliances,200
103,Product C,Apparel,50
104,Product D,Electronics,800
105,Product E,Home Decor,300
```

customers.csv:

```
customer_id,customer_name,email,address
201,John Doe,john@example.com,123 Main St, City
202,Jane Smith,jane@example.com,456 Elm St, Town
203,Michael Brown,michael@example.com,789 Oak St, Village
204,Emily Johnson,emily@example.com,101 Pine St, County
205,David Wilson,david@example.com,111 Maple St, Borough
```

regions.csv:

```
region_id,region_name
1,North
2,South
3,East
4,West
```

Level 1: Data Loading and Basic Analysis

1. Load the sales dataset (e.g., 'sales.csv') into a Pandas DataFrame.
2. Display the first 5 rows of the dataset.
3. Check the shape (number of rows and columns) of the dataset.
4. Display basic statistics (mean, median, min, max) for the 'Sales' column.
5. Determine the number of unique products sold.

Level 2: Data Cleaning and Preprocessing

1. Handle missing values by filling them with appropriate values (e.g., mean or median).
2. Convert categorical variables to numerical representation using one-hot encoding or label encoding.
3. Check for and remove any duplicate rows in the dataset.
4. Normalize the 'Sales' column to a scale between 0 and 1.
5. Identify and remove outliers from the dataset using appropriate techniques.

Level 3: Exploratory Data Analysis (EDA)

1. Visualize the distribution of sales amounts using a histogram.
2. Create a boxplot to identify any outliers in the sales data.
3. Explore the relationship between sales and other variables using scatter plots.
4. Calculate and visualize the correlation matrix between numerical variables.
5. Analyze the sales trend over time using line plots or time series plots.

Level 4: Data Aggregation and Grouping

1. Group the sales data by product category and calculate the total sales amount for each category.
2. Group the sales data by month and year and calculate the average sales amount for each month.
3. Aggregate the sales data by region and calculate the total sales amount for each region.
4. Group the sales data by customer segment and calculate the average sales amount for each segment.
5. Aggregate the sales data by sales representative and calculate the total sales amount for each representative.

Level 5: Data Visualization with Matplotlib

1. Create a bar chart to visualize the total sales amount by product category.
2. Generate a line plot to show the sales trend over time.
3. Create a pie chart to represent the distribution of sales across different regions.

4. Generate a boxplot to compare the sales performance of different customer segments.
5. Visualize the relationship between sales and advertising expenditure using a scatter plot.

Level 6: Advanced Data Manipulation with Numpy

1. Use Numpy to calculate the mean, median, and standard deviation of the sales data.
2. Perform element-wise arithmetic operations on the sales data (e.g., addition, subtraction, multiplication).
3. Use Numpy to reshape the sales data into a different dimension.
4. Apply broadcasting to perform operations on arrays with different shapes.
5. Use Numpy to perform matrix multiplication on sales data arrays.

Level 7: Advanced Pandas Queries

1. Use Pandas to filter the sales data for a specific time period (e.g., quarter or year).
2. Apply boolean indexing to select rows based on multiple conditions.
3. Use Pandas groupby and aggregate functions to calculate custom metrics.
4. Combine multiple DataFrames using merge or join operations.

Level 8: Creating Database Tables

1. Design and create a database table to store sales data, including columns for product ID, sales amount, date, and customer ID.
2. Create a separate table to store product information, including product ID, name, category, and price.
3. Design a table to store customer information, including customer ID, name, email, and address.
4. Define relationships between the tables using foreign key constraints.
5. Populate the database tables with sample data from the sales dataset.

Level 9: SQL Queries for Data Retrieval

1. Write an SQL query to retrieve total sales amount for each product category.
2. Perform a join operation to retrieve sales data along with product information.
3. Write a subquery to find customers who made the highest purchase.
4. Use the GROUP BY clause to calculate the average sales amount by month and year.
5. Write a complex SQL query to find the top-selling products by region.

Level 10: Advanced Data Analysis Techniques with NumPy and Matplotlib

1. Use historical sales data to create a time series array.
2. Visualize the original and smoothed time series using Matplotlib line plots.

3. Create word frequency arrays to identify common themes in customer reviews.
4. Visualize the word frequency distribution using Matplotlib histograms or word clouds.
5. Visualize statistical results using Matplotlib box plots, violin plots, or histograms.

Level 11: Building Interactive Data Dashboard with NumPy and Matplotlib – Use (Tkinter /Python Script / JavaScript)

1. Create interactive line plots with NumPy and Matplotlib to visualize sales trends over time.
   - Use sliders or dropdown menus to allow users to select different time periods or product categories.
   - Update the line plot dynamically based on user input.
2. Design interactive scatter plots with NumPy and Matplotlib to explore relationships between sales variables.
   - Implement hover functionality to display detailed information when users hover over data points.
   - Add interactive legends to allow users to toggle the visibility of different data series.
3. Develop interactive bar charts with NumPy and Matplotlib to compare sales performance across different regions or product categories.
   - Incorporate tooltips to provide additional information when users hover over bar segments.
   - Include buttons or checkboxes for users to filter data based on specific criteria.

Level 12: Building Interactive Data Charts with NumPy and Matplotlib – Use (Tkinter /Python Script / JavaScript)

1. Build interactive pie charts with NumPy and Matplotlib to visualize the distribution of sales across different product categories or customer segments.
   - Enable users to click on pie chart segments to drill down into detailed information.
   - Implement animation effects to enhance user engagement and visualization aesthetics.
2. Create interactive heatmaps with NumPy and Matplotlib to analyze sales patterns based on geographical regions or time periods.
   - Allow users to adjust color maps and heatmap resolution for better visualization.
   - Include zoom and pan functionality to explore detailed heatmap data.