**Documentation for Flask Data Analysis Script**

**Overview**

This script is a Flask-based application that performs data analysis on an Excel file containing sales data. It dynamically identifies relevant columns, cleans the data, and performs various analyses, saving the results in an Excel file with multiple sheets.

**Prerequisites**

1. Python 3.x installed.
2. Required Python libraries:
   * Flask
   * pandas
   * openpyxl
   * xlsxwriter
3. An Excel file (sales\_data.xlsx) located at C:/Users/raghu/OneDrive/Desktop/data\_analysis/.
4. data\_cleaning.py module containing the functions clean\_data and clean\_dates.

**Installation and Setup**

1. Install dependencies using pip:

pip install flask pandas openpyxl xlsxwriter

1. Ensure the data\_cleaning.py module is in the same directory or accessible in the Python path.
2. Save your sales data file in the specified directory.

**Key Functionalities**

**1. Flask Route**

* **Endpoint**: /
* **Methods**: GET, POST
* **Purpose**: Reads the input Excel file, cleans the data, performs analyses, and generates an output Excel file containing the results.

**2. Data Cleaning**

Uses the clean\_data and clean\_dates functions from data\_cleaning.py to:

* Remove duplicates.
* Normalize and standardize date columns.

**3. Dynamic Column Detection**

Automatically identifies key columns based on partial matches (e.g., city, total, invoice ID). Raises an error if any required column is missing.

**4. Analytical Functions**

* **Sales Analysis by Branch**: Groups data by branch to calculate total sales, transaction count, and average rating.
* **Sales Analysis by Product Line**: Groups data by product line to calculate total sales, transaction count, gross income, and average rating.
* **Top 3 Product Lines**: Identifies the top 3 product lines by total sales.
* **Date-Wise Sales**: Aggregates total sales and transactions for each date.
* **City-Wise Sales**: Analyzes total sales, transaction count, and average rating by city.
* **Customer Type Analysis**: Groups data by customer type to calculate total sales and average rating.
* **Top Performing Products**: Analyzes product performance by quantity sold and total sales.
* **Customer Insights**: Provides insights based on gender and customer type, including total sales and average rating.
* **Sales in Date Range**: Filters sales data within a specified date range.
* **Sales on Specific Day**: Calculates total sales for a specific date.
* **Branch Sales on Specific Day**: Analyzes branch-level sales for a given date.

**5. Output Generation**

* Results are saved in an Excel file (sales\_analysis\_output.xlsx) with multiple sheets for each analysis.
* If a file with the same name exists, a versioned file name is created (e.g., sales\_analysis\_output(1).xlsx).

**Execution**

Run the script using the following command:

python script\_name.py

* Open your browser and navigate to http://127.0.0.1:5000/.
* The results are returned as a JSON response and saved in the output Excel file.

**Error Handling**

The script gracefully handles errors and returns a JSON response with the error message if any issue occurs during execution.

**Key Functions and Their Purpose**

|  |  |
| --- | --- |
| **Function Name** | **Description** |
| analyze\_sales\_by\_branch | Analyzes sales by branch. |
| analyze\_sales\_by\_product\_line | Analyzes sales by product line. |
| get\_top\_3\_product\_lines | Identifies top 3 product lines based on sales. |
| analyze\_date\_wise\_sales | Aggregates sales data by date. |
| analyze\_city\_sales | Analyzes sales data by city. |
| analyze\_customer\_type\_sales | Analyzes sales data by customer type. |
| analyze\_top\_performing\_products | Analyzes top-performing products based on quantity and sales. |
| analyze\_customer\_insights | Provides customer insights by gender and type. |
| Filtered\_data\_by\_date\_range | Filters sales data for a specific date range. |
| specific\_day\_sale | Calculates total sales for a specific date. |
| Filter\_branch\_sales\_by\_specific\_date | Analyzes branch-level sales on a specific date. |

**Output Example**

The output file will contain the following sheets:

1. **Overall Sales by Branches**
2. **Product Line Sales**
3. **Top 3 Product Lines**
4. **Date-Wise Sales**
5. **City-Wise Sales**
6. **Customer Type**
7. **Top Selling Products**
8. **Customer Purchase Behavior**
9. **Date Range Filtered Data**
10. **Specific Day Overall Sales**
11. **Branch Sales on Specific Day**

**Contact Information**

For any queries or issues, contact the developer at raghul.officialmail@gmail.com