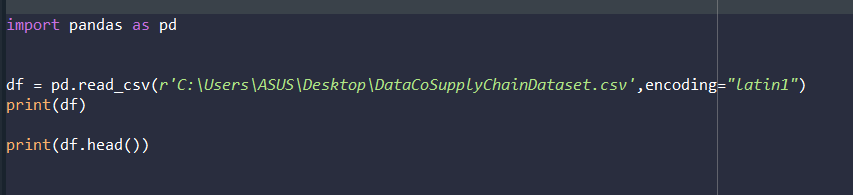
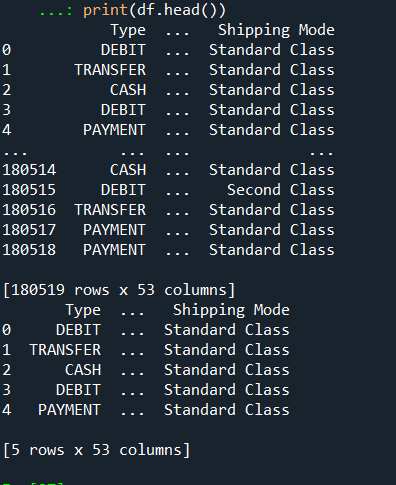
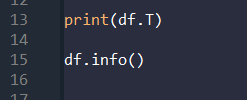
**Supply chain analysis project report**  
  
Intoduction :   
Project Description : Our supply chain data analysis project aims to analyze supply chain data to uncover actionable insights that can enhance efficiency, reduce shipment time, improve performance across the world, and maximize profit and reduce costs, understand customers’ behavior and follow their visits patterns in online store by which we can reach the categories customers like most, and also products, and how we can improve based on data-driven insights.  
  
Project objectives :  
1. Shipment Duration Insights:  
 By investigating average shipping times and discrepancies between scheduled and actual shipments, we aim to identify bottlenecks and streamline logistics processes. Understanding variations by shipping mode, vendor, and region allows us to propose tailored improvements.  
 2. Schedule Compliance Rate:  
 We explore how closely real shipping days match scheduled ones, identifying departments and vendors that consistently underperform. This analysis helps in formulating strategies to improve adherence to shipping schedules and reduce late delivery risks.   
3. Customer Profitability Assessment:   
Through evaluating the Customer Profitability Index, we aim to pinpoint the most profitable customer segments. By understanding distribution patterns, we seek to enhance strategies that increase profit while identifying high-risk or low-profit segments.  
 4. Department Performance Analysis:   
By examining departmental profit contributions and delivery accuracy, we will identify which departments align best with on-time delivery metrics. This insight guides resource allocation to optimize departmental output.  
 5. Market and Revenue Mapping:   
We analyze market sales comparisons to understand which regions generate the most revenue and identify market-level demand patterns. This analysis helps in regional strategy planning and leveraging geographic sales concentrations.  
6. Order and Payment Trends: order behaviors and payment methods, we seek to uncover correlations between payment types and delivery reliability, and assess how discount strategies impact order volumes.   
7. Online Shop logs tracking : tracking the time , date , and selected products to see in the online shop. So we can understand the customer interests and behavior.

**Data Overview** :  
-- Our Main Data is Collected from Kaggle from this link : [DATA](https://www.kaggle.com/datasets/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis/data)  
Our data consists of 2 Main data sets and 2 helping data sets that we got :  
1. Main data which has orders data with its details about 180,600 row   
2. Tokenized access logs on the online store about 470,000 row   
3. Ips geo info. (we collected it from web to help us in leverage benefits of logs data)   
4. ISO 2 digit Country code data set (we collect this from web this in cleaning and optimization phases )   
  
**-- Our Analysis Process’ Flow :**  
1) Data Overviewing ( using Python )  
2) Data Preprocessing, Removing Errors, and Cleaning ( using Python, Excel )  
3) Data Normalization ( using Power Query Editor )  
4) Data Cleaning, Transformation and Optimization ( using SQL )  
5) Developing our Database Schema ( using SQL )  
6) EDA, Data Visualization and Questions Answering Phase ( using SQL, Python )  
7) Machine Learning Preprocessing, Models Implementation, and Forecasting Questions Phase (using Python )  
9) Develop Interactive Dashboard that contains Main Insights ( using Power BI )  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1) Data Overviewing ( using Python )  
\*\* import pandas and overviewing data**

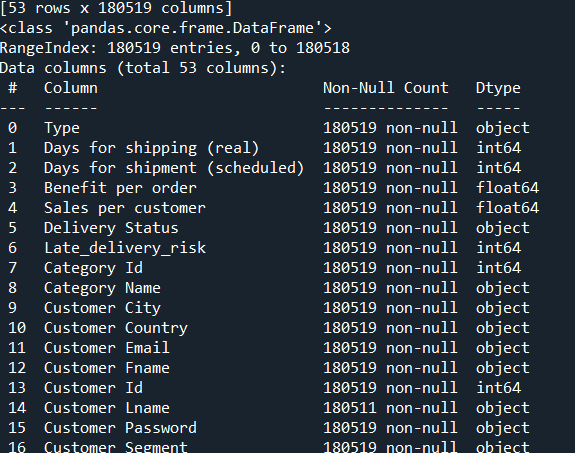
****

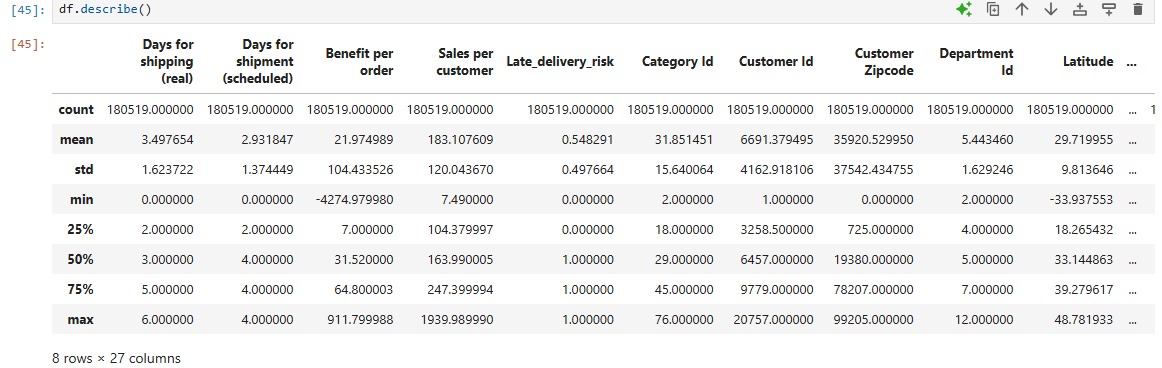
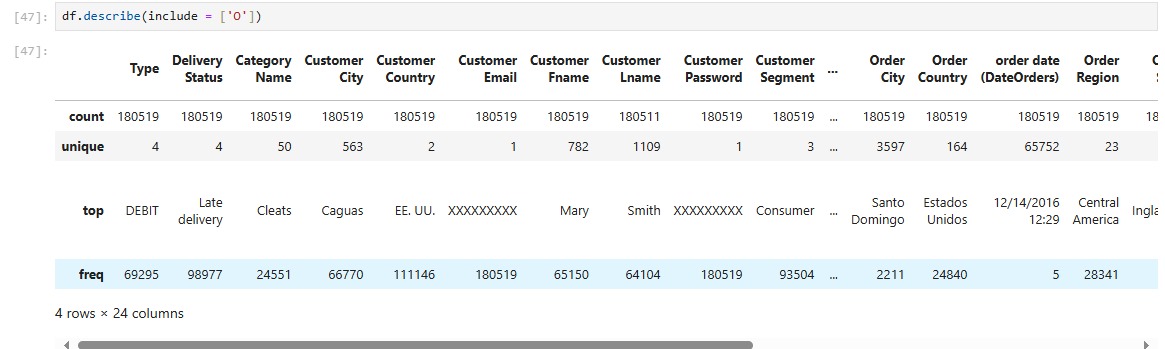
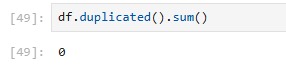
**\*\* Output :**

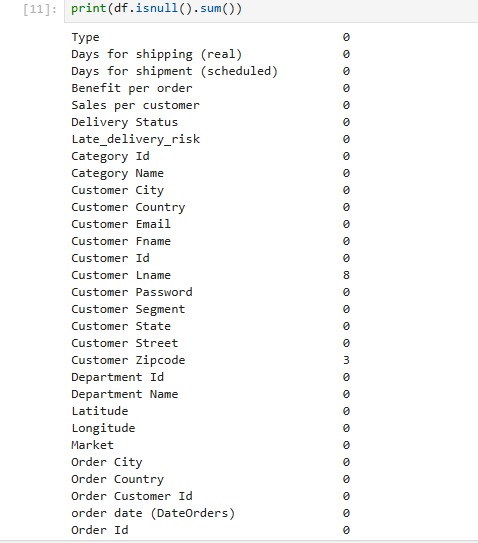
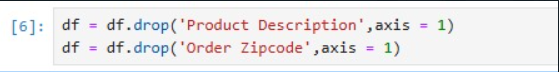
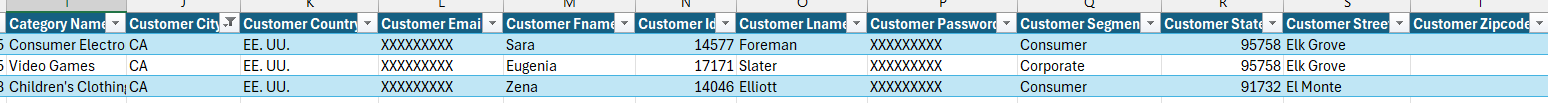
* **Show all the columns with its data type:**

**## Comment :**

* **we have 53 columns \* 180518 rows   
     
   \*\* with data types :  
   - object ( 24 column)  
   - int (14 column)  
   - float (13 column)**

**\*\* output :**

**  
-- Showing Numeric and Object Data Types:   
>>> There are data errors in the Customer Email & Password Column  
-- Checking duplicates and null values :  
## Duplicates Check >>>> There is no exact match duplicates in the whole data**

**  
## Nulls Check >>>> we have two columns have almost of their values are nulls & Customer Zip code with 3 null values.  
  
  
 >>>>>>>>>>>>>> Let’s go to the next Phase  
  
2) Data Preprocessing, Removing Errors, and Cleaning ( using Python, SQL, Excel ):  
-- Drop Null columns :  
  
-- Finding and Replacing Errors in Customer Zipcode >>  
>>>>>> It’s done manually because it's much faster than coding.  
**

**A blue and white box with white text

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A close-up of a table

AI-generated content may be incorrect.A blue box with white text

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.Removing Columns with Errors ( Customer Email & Password ):  
  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
-- Transform and replace Spanish Text to English to avoid any errors in plotting and visualization - Data Audit and Validation Process : ( Using Excel Formula )  
  
3) Data Normalization ( using Power Query Editor ) :  
We conduct Data Normalization according to data Normalization standards and techniques in order to maintain consistency and make our Analysis process faster :  
by separating our row data after cleaning to related columns as follows by Duplicate Queries :  
- Full Data ( To help in machine learning )  
- Orders  
- Order Items   
- Customers  
- Categories  
- Departments  
- logs ( Visits on the Online store )  
A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.  
-- After that, we conduct more Cleaning, Removing Errors, Transformation and Optimization using SQL: Create table OrderItems With Composite Primary key**

**A white background with black text

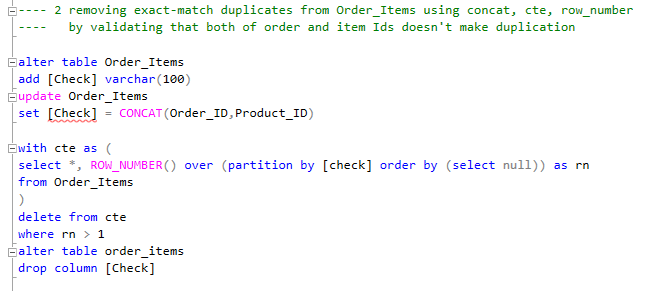
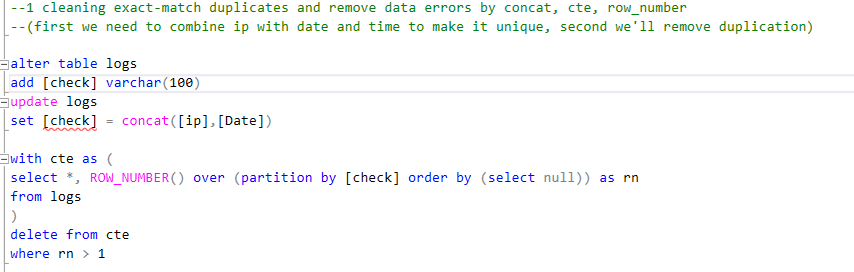
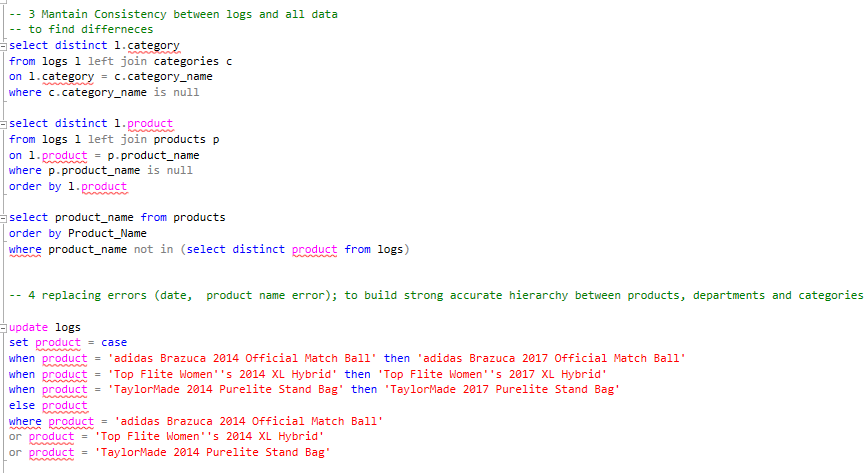
AI-generated content may be incorrect.A screenshot of a computer

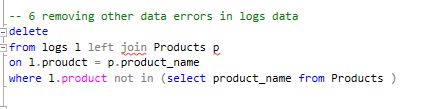
AI-generated content may be incorrect.4) Data Cleaning, Transformation and Optimization ( using SQL ): 1 ) Main Data  
## Because our Categories and Departments data are conflicting as shown   
We corrected it using a double check to give accurate insights and avoid data misleading info.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

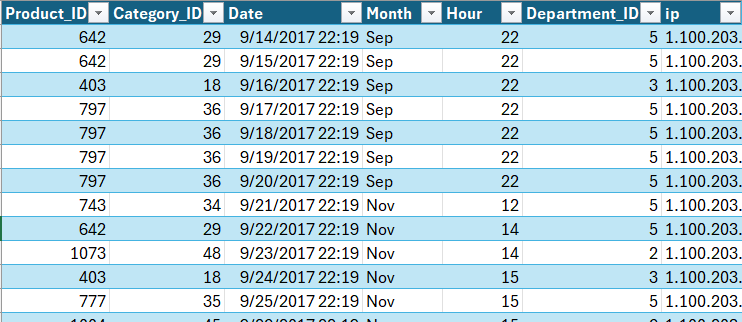
* **A screenshot of a computer code

  AI-generated content may be incorrect.Remove Duplicates Errors in Order Items ( Logically, no order id with order Item id should make duplication ) : using SQL ( CTE, RowNumber )  
    
  -- Adding Important for analysis columns & Adding Aggregated columns to Orders based on calculations on order\_items table ( using SQL subqueries )  
    
  2) Logs Data Cleaining, Transformation, and Database Optimization:  
  -- Find Differences and Replacing Errors ( Using SQL Case When )**

****

**A screenshot of a computer code

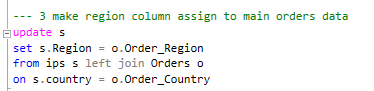
AI-generated content may be incorrect.## Powerful Database Optimization :  
replacing Text data to Integer with IDs to enhance query Performance, Optimize database and storage, and make the retrieval process more rapid  
BY : SQL Update, Alter columns data Types, Stored Procedures:**

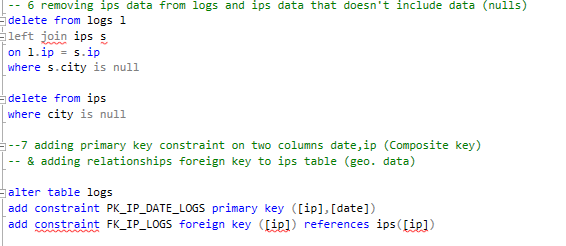
**## The Result:  
  
  
  
  
  
  
3) Create new data set with Ips geo info. And make relationship between logs data :  
A table of names and numbers

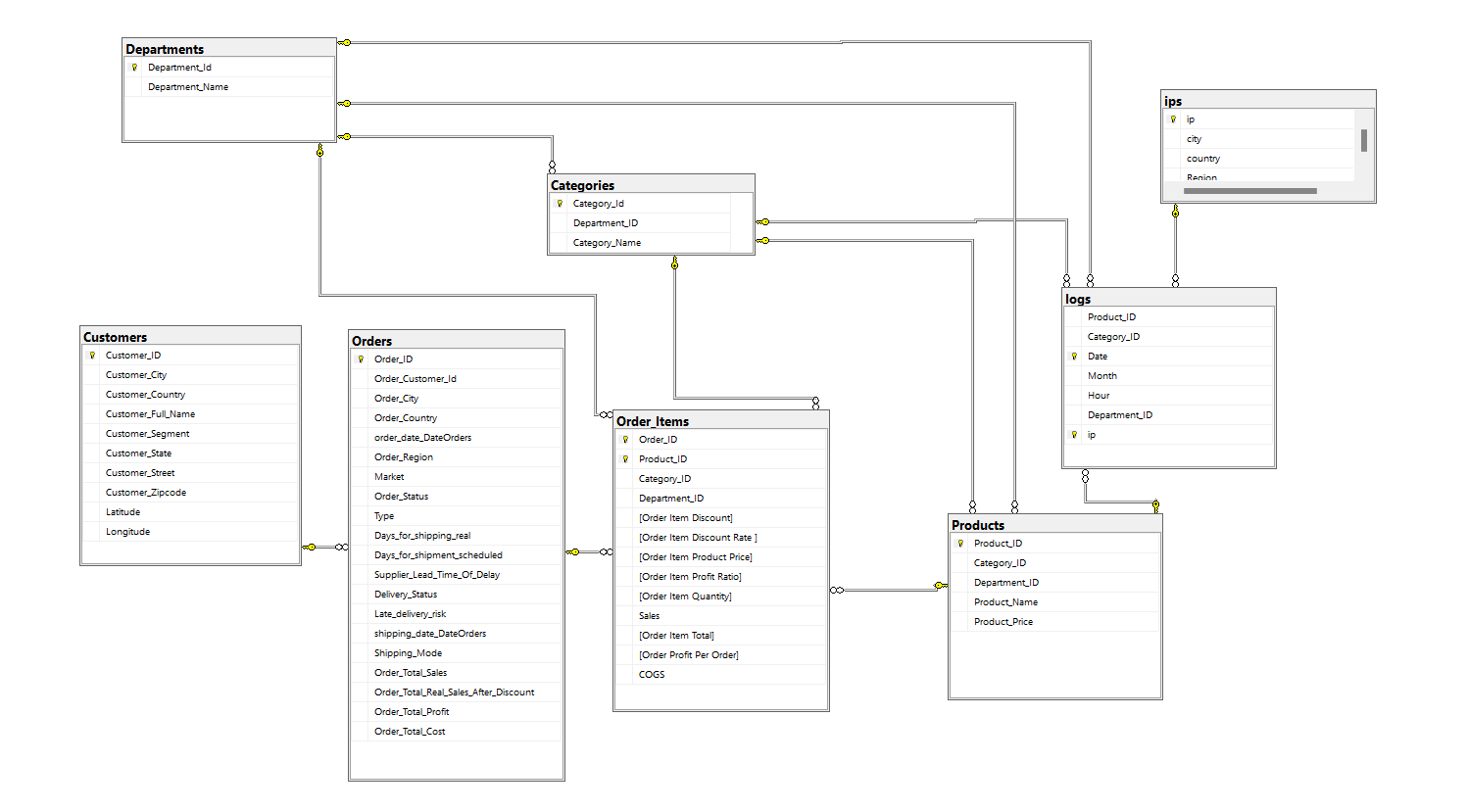
AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.( Using Web Smart Data Collecting ):**

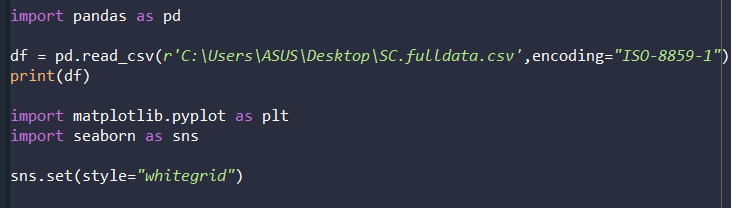
* **A screenshot of a computer code

  AI-generated content may be incorrect.To maintain Consistency between Ips and all of data :  
  We had to convert Country ISO code to Country Name, So we got the ISO country dataset and started our work as follows: ( Using SQL )  
  - Make Region column assign with our main data to maintain consistency : ( Using SQL )  
  **
* **Removing Errors and adding Composite Key to IPs data ( End Of Cleaning, Transformation, Optimization Process )**

****

**5) Developing our Database Schema ( using SQL ):  
## After All the Cleaning, transformation and optimization we did,  
we got optimal Database Diagram as follows:**

**- With Tables and Relationships:  
# Orders >>>** (Primary Key : Order ID, Foreign key : Order\_Customer\_ID from Customers) **#Customers >>>** (Primary Key : Customer\_ID)  
**# Order\_Items >>>** (Primary Key : Order\_ID, Product\_ID, Foreign keys : Order\_ID from Orders, Product\_ID from Products, Category\_ID from Categories, Department\_ID from Deprtments)  
**# Products >>>** (Primary Key : Product\_ID, Foreign keys : Category\_ID from Categories, Department\_ID from Departments)  
**# Categories >** (Primary Key : Category\_ID, Foreign key : Department\_ID from Departments)  
**# Departments >>>** (Primary Key : Department\_ID)  
**# Logs >>>** (Primary Key : ID, Foreign keys : Product\_ID from Products, Department\_ID from Departments)  
**# Ips >>>** (Primary Key : Ip)

**6) EDA, Data Visualization and Questions Answering Phase ( using Python ) :  
  
We Conduct Exploratry Data Analysis by creating visuals in Python using   
( pandas, Matplotlib, seaborn ), and we answering questions using SQL.  
  
-- Here is EDA Data Visulization using Python: \*\* Preparing Data To visualization  
**

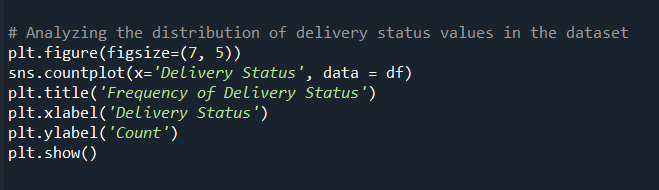
**A screen shot of a computer code

AI-generated content may be incorrect.**

**-- 1:**

A graph of a bar chart

AI-generated content may be incorrect.**Output :**

**--2 : **

**Output :**A bar graph with blue rectangles

AI-generated content may be incorrect. **--3 :** **A computer screen with text

AI-generated content may be incorrect.**

A graph of a distribution of days for shipping

AI-generated content may be incorrect.**Output :**

A screenshot of a computer program

AI-generated content may be incorrect.**--4 :**

A graph of sales per customer

AI-generated content may be incorrect.

**-5 : Heat Map, Correlation Matrix :  
A screen shot of a computer program

AI-generated content may be incorrect.**

A screenshot of a data analysis

AI-generated content may be incorrect.

**A computer screen shot of a program

AI-generated content may be incorrect.-6 :**

A graph of a shipping distribution

AI-generated content may be incorrect.

A chart with different colored rectangles

AI-generated content may be incorrect.**--7 A screenshot of a computer program

AI-generated content may be incorrect.**

**-8 : A screen shot of a computer code

AI-generated content may be incorrect.**

**Output :**

A bar chart with different colored squares

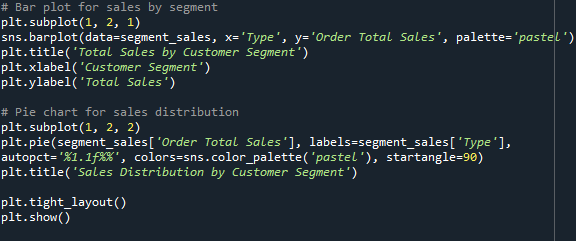
AI-generated content may be incorrect.

**-9 :**

**Output :**

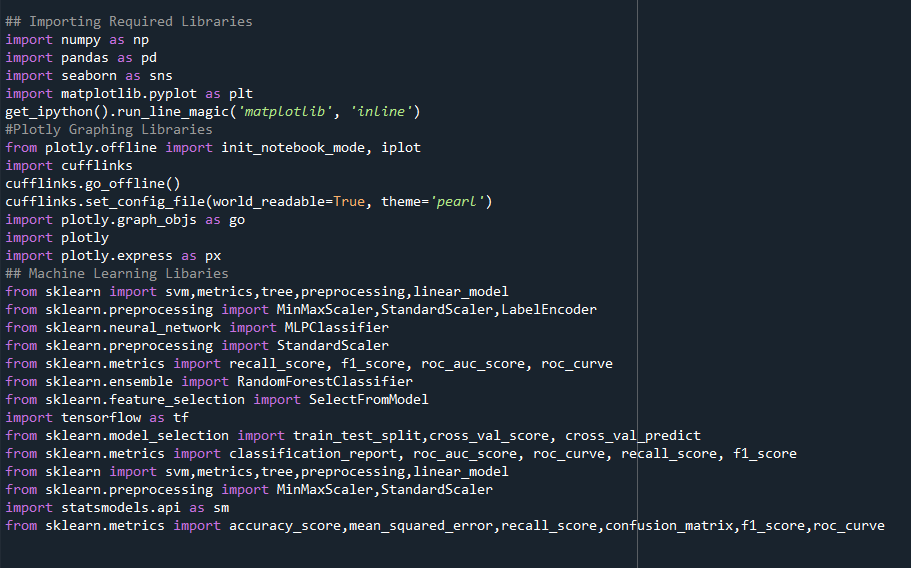
A graph of different colored bars

AI-generated content may be incorrect.

**-10 :**

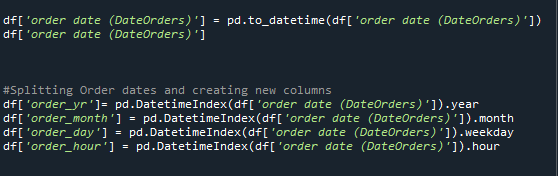
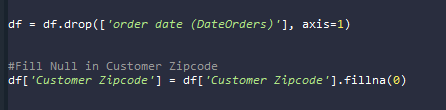
A close-up of a graph

AI-generated content may be incorrect. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7) Machine Learning Preprocessing, Models Implementation, and Forecasting Questions Phase (using Python )**  
**Machine Learning Models Report: ( Codes >> Results and Comments >> Description )  
 We conduct two phases machine learning prediction model as demonstrated bellow:  
-- First Phase :  
KNN and Random Forest Classification Models on Late Delivery Risk : (Codes and Steps )  
\*\* Importing required libraries to use in the models :  
  
**

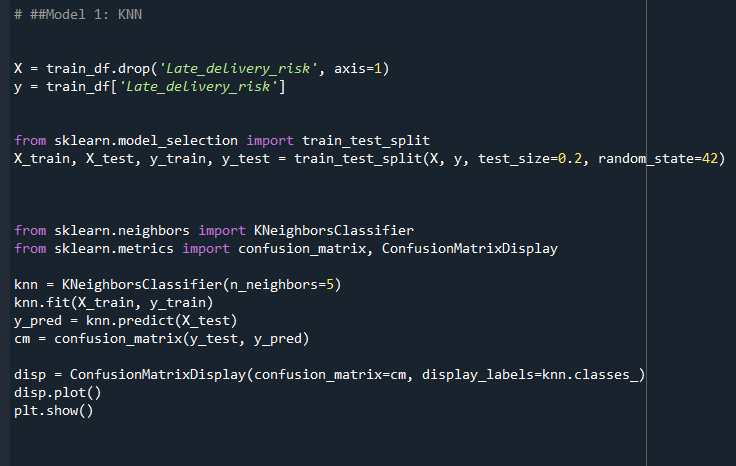
**A screen shot of a computer program

AI-generated content may be incorrect.  
\*\* Conduct required data preprocessing in order to train and run the ML models:  
A screenshot of a computer program

AI-generated content may be incorrect.  
A computer screen shot of a program code

AI-generated content may be incorrect.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
  
-- Data Distribution after Preprocessing:**A pie chart with numbers and a number

AI-generated content may be incorrect. **This pie chart shows the distribution of the target variable after preprocessing, indicating a roughly balanced number of late delivery risk (1) and non-risk (0) cases, which supports the training process of the models.**

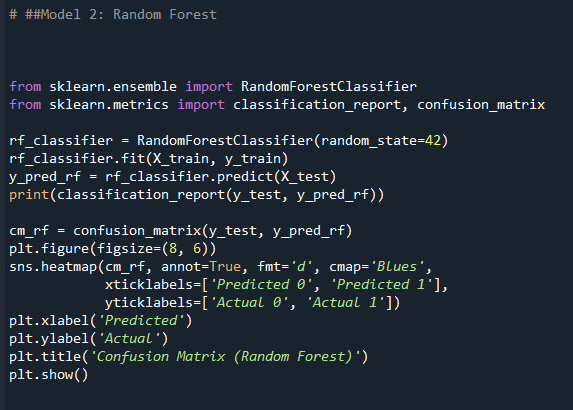
**\*\* Training and Running the first ML model ( KNN ) :**

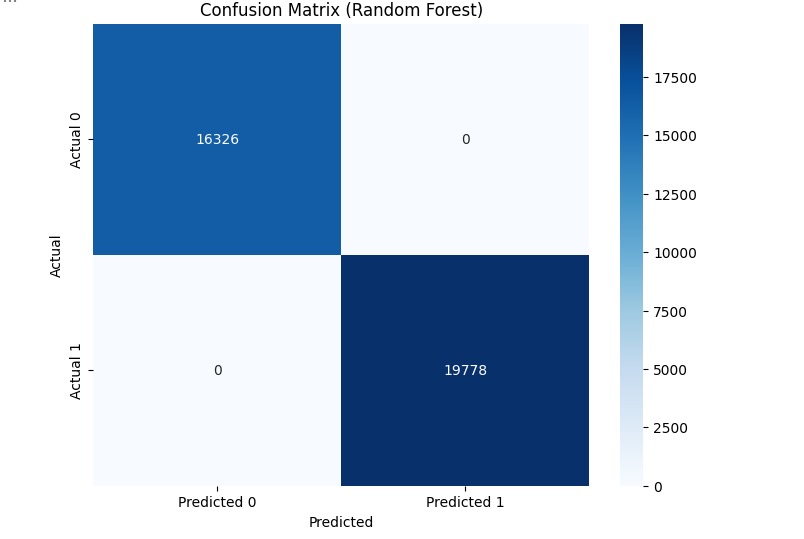
**\*\* Results of KNN Model :A chart of a number of labels

AI-generated content may be incorrect. (The comment in the next page)**

**## Comment on the KNN Model’s Results: The confusion matrix  
  
 The KNN model correctly classified many samples, with 11,667 true negatives and 15,364 true positives. However, it made notable errors, with 4,659 false positives and 4,414 false negatives. This indicates the model struggles somewhat to clearly distinguish between the two classes, leading to misclassifications, especially in borderline cases. Overall, the model shows acceptable performance but would benefit from further tuning or advanced techniques to reduce classification errors.  
  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\*\* Training and Running The Second ML Model ( Random Forest ):**

****

**  
\*\* Results of Random Forest Model : ( Confusion Matrix )**

**## Comment on The Random Forest Model’s Results:**

**The Random Forest model achieved perfect classification on the test set, with zero errors in predicting both classes. It correctly identified all 16,326 negative cases and all 19,778 positive cases. This flawless performance demonstrates its strong ability to capture complex data patterns and accurately differentiate between the classes. This makes Random Forest the preferred deployment model, with the caveat that data leakage or overfitting should be thoroughly checked to validate these perfect results.**

* **Data ML Description & Comparison between KNN and Random Forest Models:  
    
  1 ) Data ML Distribution and description**
* **The target variable, Late\_delivery\_risk, is almost evenly distributed between the classes:**
  + **Class 0 (No late delivery risk)**
  + **Class 1 (Late delivery risk)**
* **This balance is confirmed by the pie chart visualization, showing a nearly 50-50 split, which is ideal for model training and evaluation without strong class bias.**

A screenshot of a computer screen

AI-generated content may be incorrect.**Model Evaluation Metrics:**

* **The Random Forest classifier achieved perfect performance on the test set, as summarized below:**

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| **0** | **1.00** | **1.00** | **1.00** | **16,326** |
| **1** | **1.00** | **1.00** | **1.00** | **19,778** |

* **Accuracy: 1.00 (100%)**
* **Macro Average: Precision = 1.00, Recall = 1.00, F1-Score = 1.00**
* **Weighted Average: Precision = 1.00, Recall = 1.00, F1-Score = 1.00**

**Confusion Matrix Interpretation:**

* **The confusion matrix confirms there were no misclassifications:**
  + **All actual Class 0 instances were correctly predicted.**
  + **All actual Class 1 instances were correctly predicted.**
* **This flawless classification implies the model predictions matched perfectly with the true labels on the test set.**

**Analysis and Implications:**

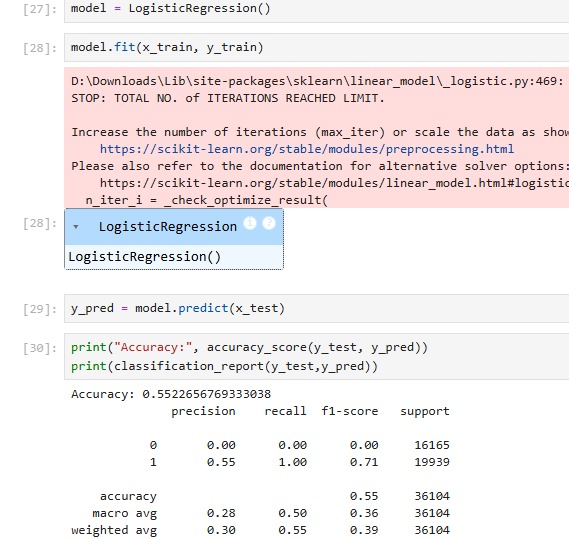
* **The outstanding results highlight that the Random Forest model can accurately distinguish late delivery risk in this dataset.**
* **A perfect score with no errors suggests:**
  + **The data quality and feature encoding were handled well.**
  + **The feature set is highly predictive of the target variable.**
* **However, such perfect results may sometimes indicate potential overfitting or leakage between training and test data, so it is essential to:**
  + **Validate using cross-validation techniques.**
  + **Check for data leakage or duplicates between training and test sets.**
  + **Review feature engineering steps to ensure no label information was inadvertently included.**
* **If validated, this Random Forest model serves as a robust predictive tool for anticipating delivery delays, enabling proactive logistical decisions.**

**Recommendations:**

* **Conduct cross-validation to confirm the model's generalizability.**
* **Perform feature importance analysis to understand key drivers of late delivery risk.**
* **Explore model deployment for real-time late delivery risk assessment.**
* **Consider comparing with other models and ensemble techniques for robustness.**

**2) ML Two Models’ comparison :**  
**The first model used for predicting "Late Delivery Risk" is the K-Nearest Neighbors (KNN) algorithm. KNN is a simple yet powerful machine learning model that classifies data points based on their proximity to other points in the feature space. In this setup, the algorithm uses 5 neighbors (n\_neighbors=5) to determine the class of each test sample. The data is split into training and testing sets with an 80-20 split (test\_size=0.2), and the random\_state=42 ensures the split remains consistent across runs. From the confusion matrix, we observe that the model correctly identifies 11,667 instances of "no risk" (true negatives) and 15,364 instances of "late delivery risk" (true positives). However, the model also misclassifies 4,659 samples as false positives and 4,414 as false negatives. These misclassifications highlight limitations in KNN's ability to distinguish between the two classes in this dataset effectively.**

* **The second model employed is the Random Forest Classifier, an ensemble method that combines multiple decision trees to deliver high predictive accuracy while minimizing overfitting. The model was trained with the same data split and random\_state=42 for reproducibility. The results for this model are exemplary, as shown by the confusion matrix and classification report. The model achieves perfect predictions, with 16,326 instances of "no risk" (true negatives) and 19,778 instances of "late delivery risk" (true positives). There are no false positives or false negatives, and the precision, recall, and F1-scores for both classes are all 1.00. This indicates that the Random Forest model performs flawlessly on this dataset, capturing all patterns and distinctions between the classes.**
* **Comparing the two models, it is evident that the Random Forest classifier outperforms the KNN model by a significant margin. While KNN is simpler to implement and understand, its performance is hindered by the presence of misclassifications, which could result from challenges like high-dimensional feature spaces or unbalanced data distributions. On the other hand, the Random Forest model handles these challenges with ease, leveraging its ensemble nature to effectively account for feature interactions and variability. While KNN might still be a useful baseline, Random Forest demonstrates superior robustness and accuracy, making it the more suitable choice for predicting late delivery risks in this supply chain dataset.**

**-- Second ML Phase:  
  
- Logistic Regression Model for Predicting Late Delivery Risk in Orders : (Codes and Steps)  
  
  
**

* **Model Description  
  Logistic Regression classification is a model to predict the risk of late delivery in a large-scale sales dataset. The dataset includes over 180,000 records and 48 features, covering various aspects of the order lifecycle such as product details, customer demographics, geographic information, shipping modes, and order timelines.**

**The binary target variable, Late\_delivery\_risk, indicates whether a given order is at risk of being delivered late:**

* **1 represents high risk of late delivery.**
* **0 represents low or no risk of late delivery.**

**This model aims to support business operations by enabling early identification of potentially delayed orders, allowing for timely interventions in the supply chain process.**

**Data Preprocessing**

**Before training the model, several preprocessing steps were performed:**

* **All categorical features (e.g., Delivery Status, Customer City, Shipping Mode) were label-encoded using LabelEncoder.**
* **Date columns were also label-encoded as the model used a basic logistic regression which requires numerical inputs.**
* **The dataset was split into features (X) and target (y), and further split into training and testing sets using an 80/20 ratio.**
* **Model Training and Evaluation**

**A Logistic Regression model was trained using the default solver and parameters. Upon evaluation on the test set, the following results were observed:**

* **Accuracy: 55.1%**
* **Precision (for class 1): 55%**
* **Recall (for class 1): 100%**
* **F1-Score (for class 1): 71%**
* **Precision, Recall, F1-Score (for class 0): 0%**

**The classification report highlights a major imbalance in prediction, where the model predicts nearly all instances as 1 (late delivery risk). This leads to:**

* **Very high recall for the positive class (1) but**
* **No predictive power for the negative class (0), causing poor precision and F1-score for that class.**

**Additionally, a convergence warning was raised during training, indicating that the default number of iterations was insufficient for the model to fully converge.**

**Insights and Recommendations**

* **The model exhibits strong sensitivity (recall) to late delivery cases, meaning it successfully flags almost all high-risk orders.**
* **However, it lacks specificity, failing to correctly identify on-time deliveries, which limits its usability for balanced decision-making.**
* **This may be due to class imbalance in the dataset or inadequate feature scaling.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
  
 ## Comparison of Three Models for Predicting Late Delivery Risk :**

**1. K-Nearest Neighbors (KNN)**

**Model Overview:  
KNN assigns a label based on the majority class among the 5 nearest neighbors.**

**Performance Highlights:**

**Correctly classifies many samples from both classes.**

**Shows a notable number of false positives and false negatives.**

**Struggles to distinguish between classes in complex data.**

**Strengths:**

**Simple and easy to implement.**

**Serves as a useful baseline model.**

**Limitations:**

**Sensitive to data distribution and overlapping classes.**

**Limited capacity for capturing complex patterns.**

**Computationally inefficient with large or high-dimensional datasets.**

**Summary:  
Effective as a baseline model, but lacks robustness for critical use cases.**

**2. Random Forest Classifier**

**Model Overview:  
An ensemble method combining multiple decision trees to enhance predictive accuracy and robustness.**

**Performance Highlights:**

**Achieved perfect classification on the test data (no errors).**

**Effectively captures complex relationships and feature interactions.**

**Strengths:**

**Robust against noise and overfitting.**

**Handles large, mixed-type datasets efficiently.**

**Offers interpretable feature importance insights.**

**Limitations:**

**Perfect accuracy may indicate overfitting or potential data leakage.**

**More complex and computationally demanding than simpler models.**

**Summary:  
Delivers outstanding and balanced performance; highly suitable for deployment after proper validation.**

**3. Logistic Regression**

**Model Overview:  
A linear model used to estimate the probability of a binary outcome.**

**Performance Highlights:**

**Achieves high recall for the late delivery risk class.**

**Suffers from poor precision and F1-score for the no-risk class.**

**Overall accuracy is approximately 55%.**

**Strengths:**

**Fast, simple, and easy to interpret.**

**Effective at identifying high-risk deliveries.**

**Limitations:**

**High false positive rate.**

**Low precision and specificity.**

**Requires scaling, parameter tuning, and potentially addressing class imbalance.**

**Summary:  
Good sensitivity to risky deliveries, but low reliability due to frequent false alarms and limited accuracy.**

* **Overall Comparison and Recommendation**

**Best Model: Random Forest**

**Provides the highest accuracy and best balance between precision and recall.**

**No misclassifications observed, making it the most reliable and deployment-ready model.**

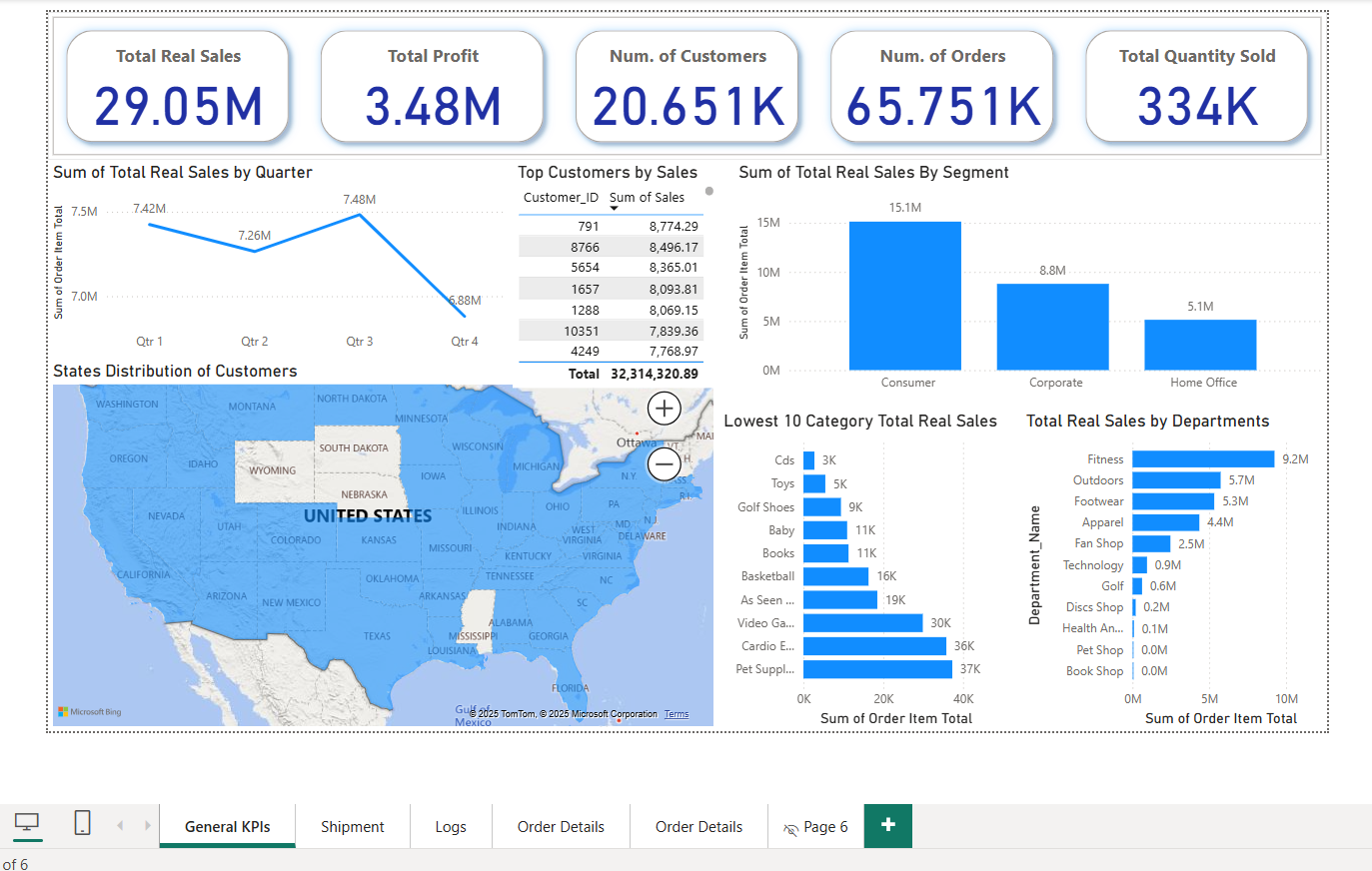
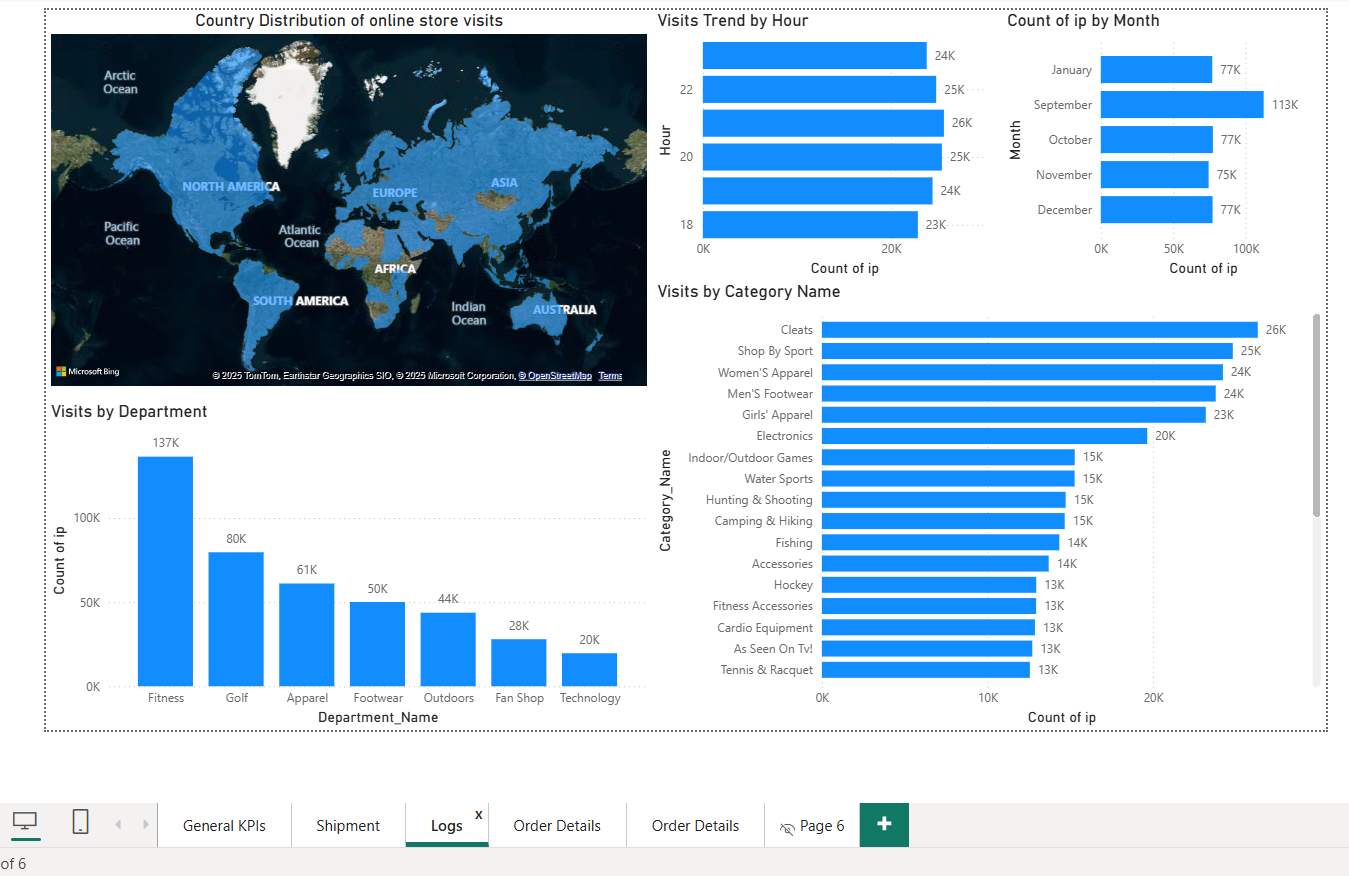
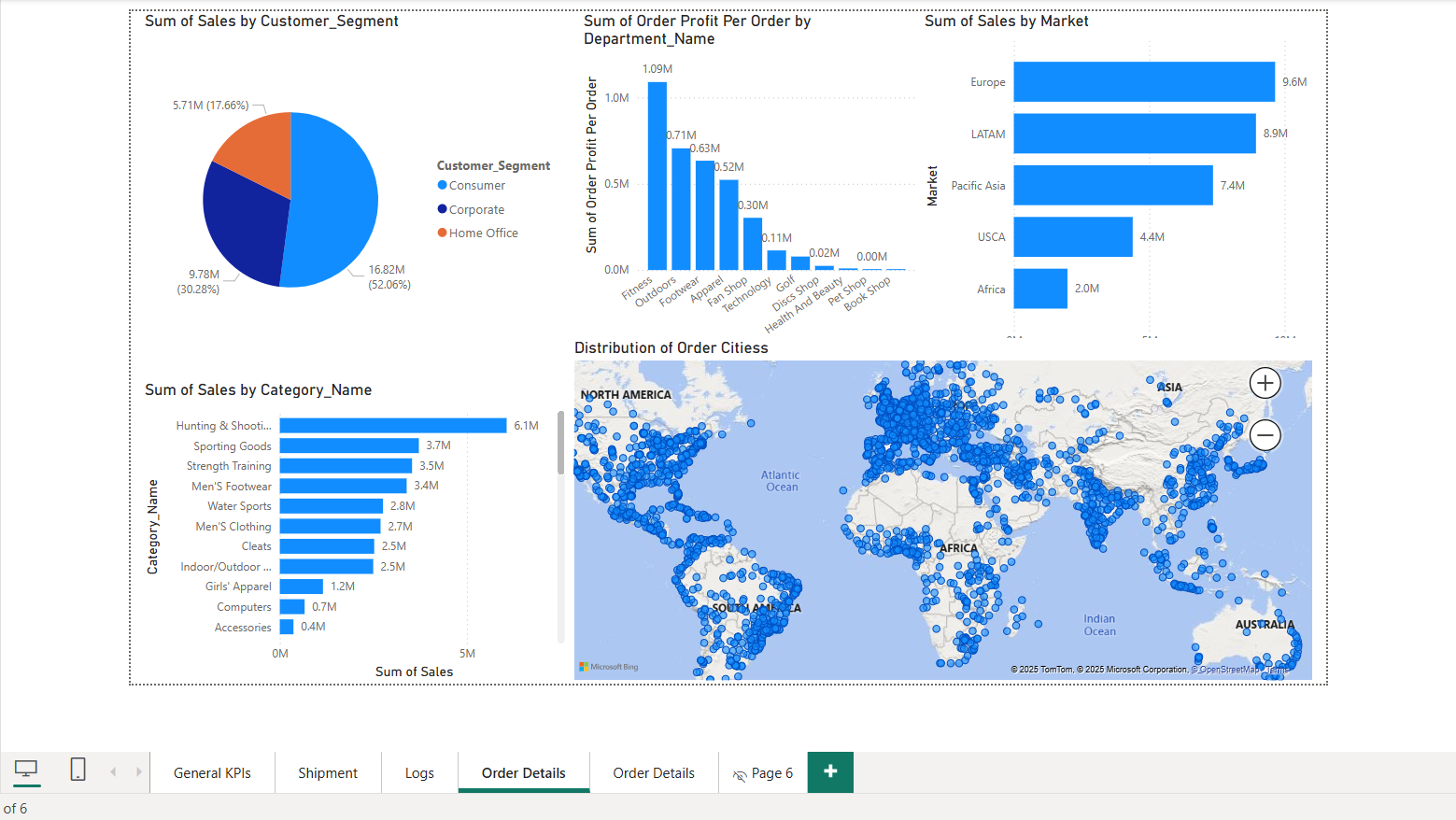
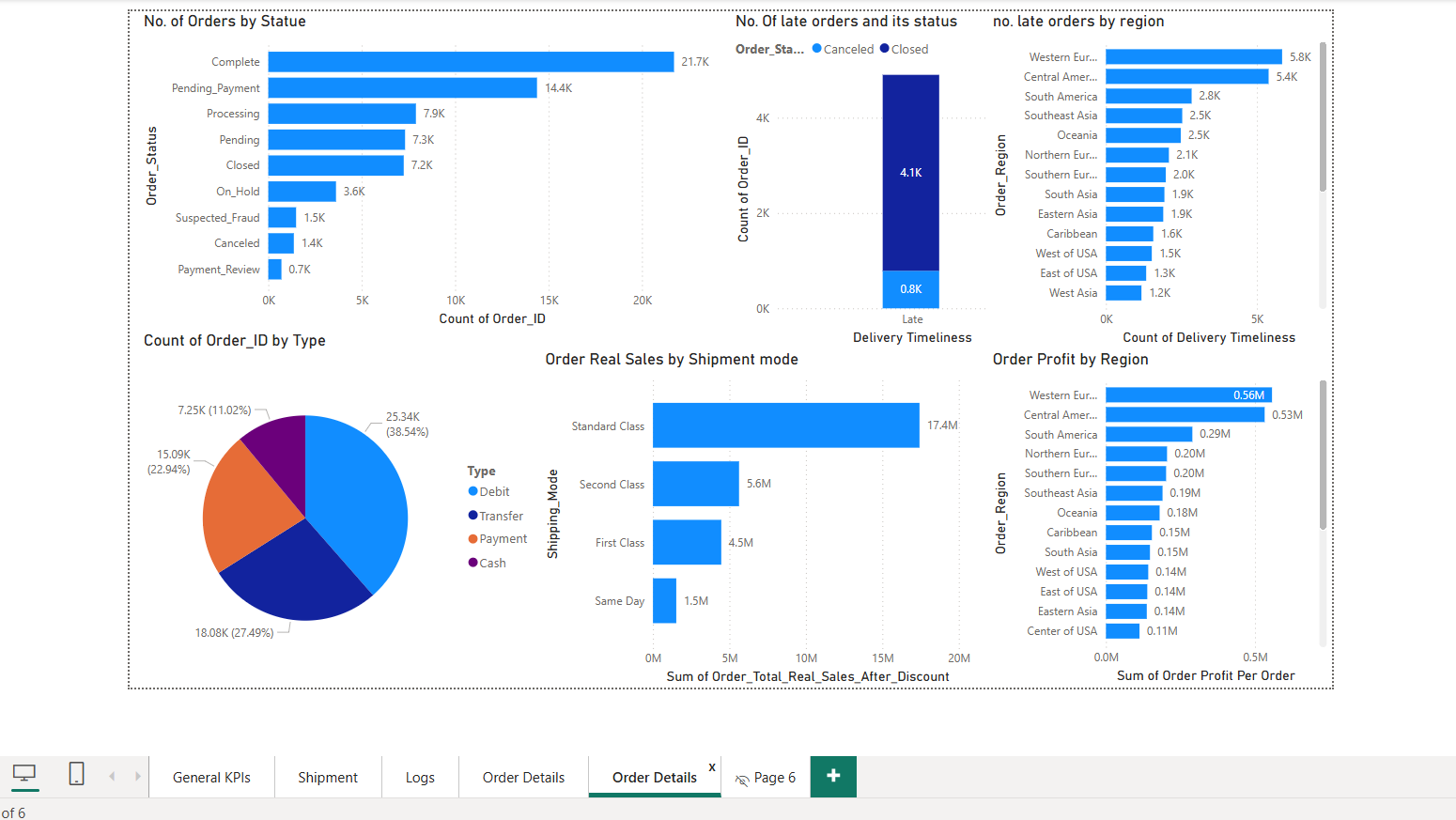
**Second Best: K-Nearest Neighbors (KNN)**

**Performs better than Logistic Regression in managing false positives and negatives.**

**Less robust than Random Forest, but a reasonable alternative baseline.**

**Least Suitable: Logistic Regression**

**While it detects risky cases well, it generates too many false positives.**

**Shows the lowest overall accuracy and requires significant enhancements before use.  
  
  
  
  
  
  
9) Develop Interactive Dashboard that contains Main Insights ( using Power BI** ): **  
**