MODULE – MACHINE LEARNING

PROJECT REPORT ON

**FOOD DELIVERY TIME PREDICTION**

**A MACHINE LEARNING APPROACH**

By:

Pranav Kishor Karodpati Batch – ML\_1224B1222

Under the guidance of Pavan Khandare Sir Director

3RI Technologies Pvt. Ltd., Pune



**TABLE OF CONTENTS**

[INTRODUCTION 1](#_bookmark0)

[Background 1](#_bookmark1)

[Problem Statement: 1](#_bookmark2)

[DATASET OVERVIEW 2](#_bookmark3)

[Acknowledgement: 2](#_bookmark4)

[Table – Detailing variables in the dataset 3](#_bookmark5)

[CODING IN JUPYTER NOTEBOOK 4](#_bookmark6)

[IMPORTING REQUIRED LIBRARIES 4](#_bookmark7)

[DATA EXTRACTION: 4](#_bookmark8)

[DISPLAY DATASET INFORMATION: 5](#_bookmark9)

DATA CLEANING…………………………………………………………………..5

[DATA PREPROCESSING…………………………………………………………](#_bookmark10)..15

[ML MODELS EVALUATION 17](#_bookmark11)

[LINEAR REGRESSION: 18](#_bookmark12)

[RANDOM FOREST: 19](#_bookmark13)

[ADABOOST 20](#_bookmark14)

[STACKING: 21](#_bookmark15)

[DECISION TREE REGRESOOR: 22](#_bookmark16)

[K-NEAREST NEIGHBORS (KNN): 23](#_bookmark17)

[SUPPORT VECTOR MACHINE (SVM): 24](#_bookmark18)

[CONCLUSION 26](#_bookmark21)

[LIMITATION AND FUTURE SCOPE 28](#_bookmark22)

**ACKNOWLEDGEMENT:**

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this project titled *“***Food Delivery Time Prediction using Machine Learning.***”*

First and foremost, I am deeply thankful to **Pavan Khandare Sir**, my project guide and Senior Data Science Trainer at **3RI Technologies Pvt, Ltd** for his continuous guidance, valuable insights, and encouragement throughout the course of this project.

I would also like to acknowledge **Kaggle** as the source of the dataset used in this study. The dataset, licensed under the **Apache 2.0 License**, provided a comprehensive foundation for exploring various machine learning techniques in the context of real-world food delivery challenges.

# INTRODUCTION

#### BACKGROUND:

In recent years, the food delivery industry has experienced rapid growth due to increasing consumer demand for convenience and fast service. With this surge, the need for accurate prediction of delivery times has become crucial—not only for enhancing customer satisfaction but also for optimizing logistics and resource allocation for delivery platforms. Various unpredictable factors such as traffic congestion, weather conditions, restaurant preparation time, and order complexity can significantly affect delivery time. This project leverages machine learning techniques to predict food delivery time by analyzing historical delivery data and identifying the key features influencing delays. By building predictive models, we aim to provide data-driven solutions to streamline food delivery operations and set realistic customer expectations.

This study aims to develop a **predictive model for food delivery time prediction** using machine learning techniques. The research evaluates multiple ML regressors, including **Support Vector Machines (SVM), Random Forest (RF), Linear Regression (LR), Gradient Boosting (GB)**, **KNN**, **DECISION TREE(DT)** to determine the most effective model.

#### PROBLEM STATEMENT:

Inaccurate estimation of food delivery times leads to poor customer experiences, inefficiencies in delivery management, and increased operational costs for food delivery services. The primary problem addressed in this project is the development of a machine learning-based system that can reliably predict the estimated time of delivery for food orders. This involves analyzing a dataset that includes various attributes such as restaurant location, distance, weather, traffic, and order-specific details. The goal is to build, compare, and optimize multiple machine learning models to accurately forecast delivery time and identify the most influential features affecting it.

# DATASET OVERVIEW

**Dataset Name:** Food Delivery Time Prediction

**Source:** Kaggle.Com

**Key Features**:

**TABLE – DETAILING VARIABLES IN THE DATASET**

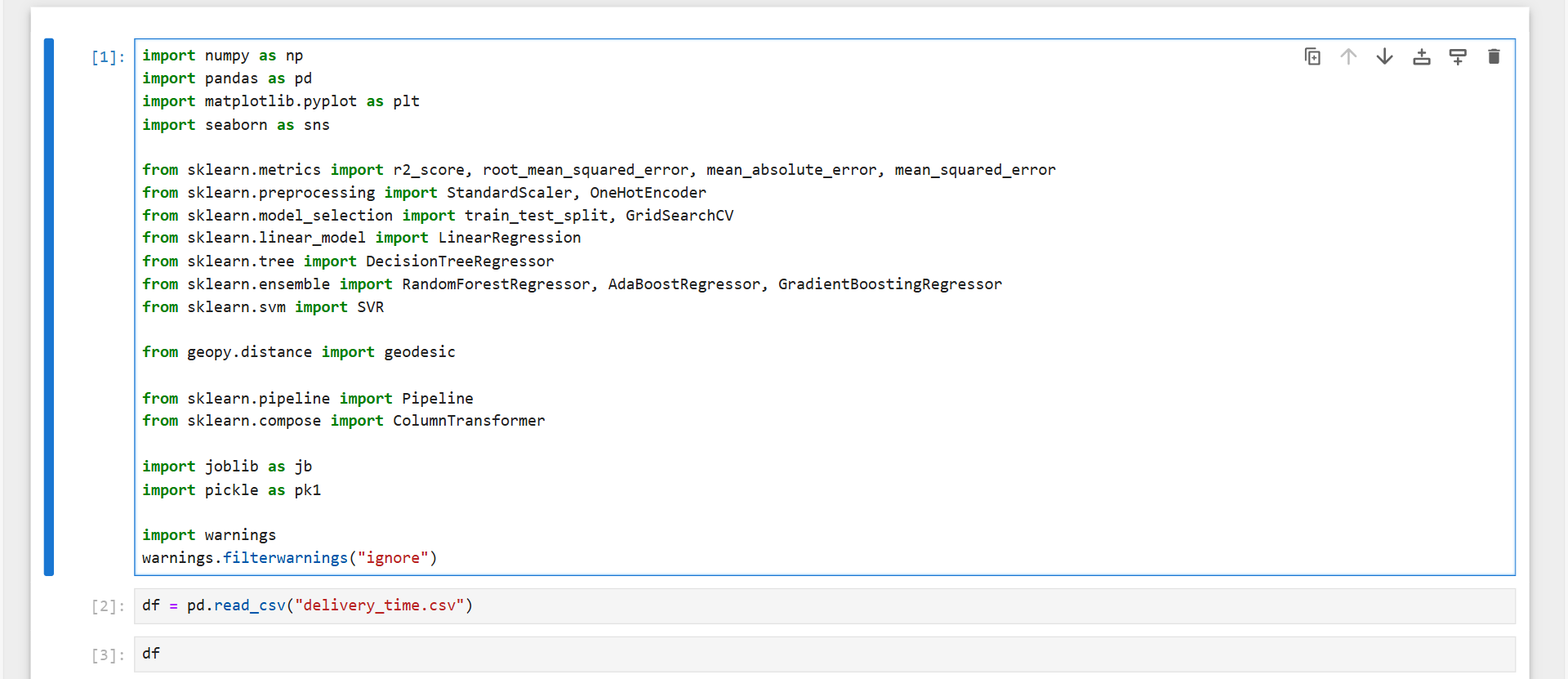
|  |  |
| --- | --- |
| **Feature Name** | **Description** |
| **ID** | A unique identifier for each food delivery transaction. |
| **Delivery\_person\_ID** | A unique identifier assigned to each delivery person. |
| **Delivery\_person\_Age** | Age of the delivery person, which may influence delivery speed and experience. |
| **Delivery\_person\_Ratings** | Customer-based rating of the delivery person, reflecting service quality. |
| **Restaurant\_latitude** | Latitude coordinate of the restaurant from where the order is picked up. |
| **Restaurant\_longitude** | Longitude coordinate of the restaurant's location. |
| **Delivery\_location\_latitude** | Latitude coordinate of the customer’s delivery address. |
| **Delivery\_location\_longitude** | Longitude coordinate of the customer’s delivery address. |
| **Type\_of\_order** | Specifies the category of food ordered (e.g., main course, snacks, dessert). |
| **Type\_of\_vehicle** | Indicates the mode of transportation used by the delivery person. |
| **Order\_ID** | A distinct identifier for each individual order in the dataset. |
| **Distance\_km** | The straight-line or route-based distance between restaurant and delivery location in kilo-meters. |
| **Weather** | Describes weather conditions during the delivery (e.g., Clear, Rainy, Foggy). |
| **Traffic\_Level** | Describes traffic congestion levels during the delivery as Low, Medium, or High. |
| **Time\_of\_Day** | Time category when the delivery occurred: Morning, Afternoon, Evening, or Night. |
| **Vehicle\_Type** | Specific type of vehicle used, such as Bike, Scooter, or Car. |
| **Preparation\_Time\_min** | Time taken by the restaurant to prepare the order in minutes. |
| **Courier\_Experience\_yrs** | Number of years the courier has been delivering professionally. |
| **Delivery\_Time\_min** | The total time taken from order pickup to final delivery (target variable). |

**Target Variable:**

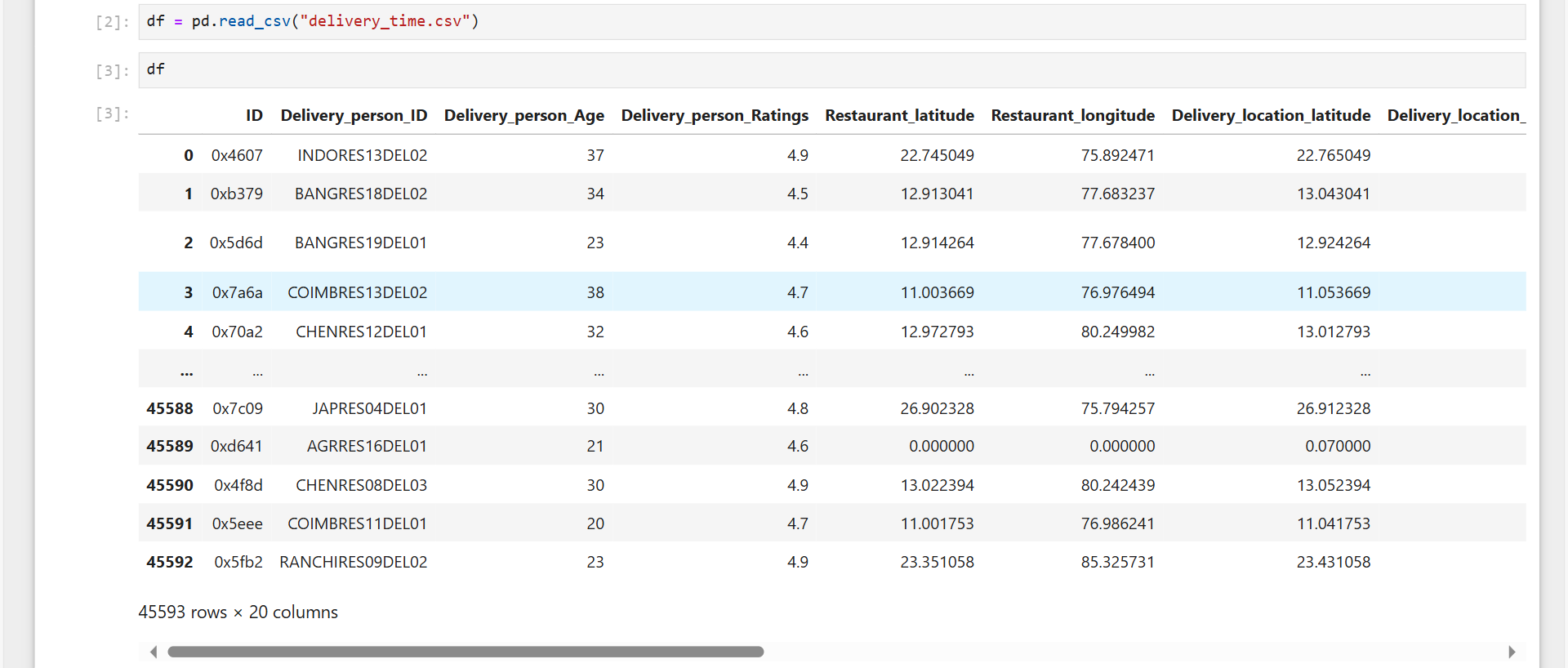
|  |  |
| --- | --- |
| **Delivery\_Time\_min** | The total time taken from order pickup to final delivery (target variable). |

# CODING IN JUPYTER NOTEBOOK

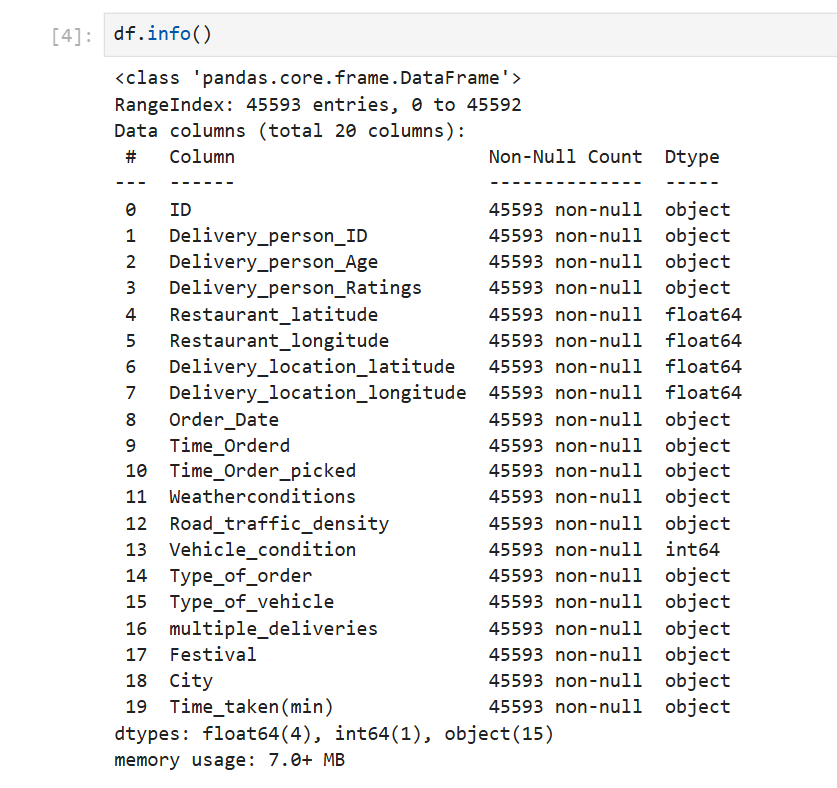
**IMPORTING REQUIRED LIBRARIES:**



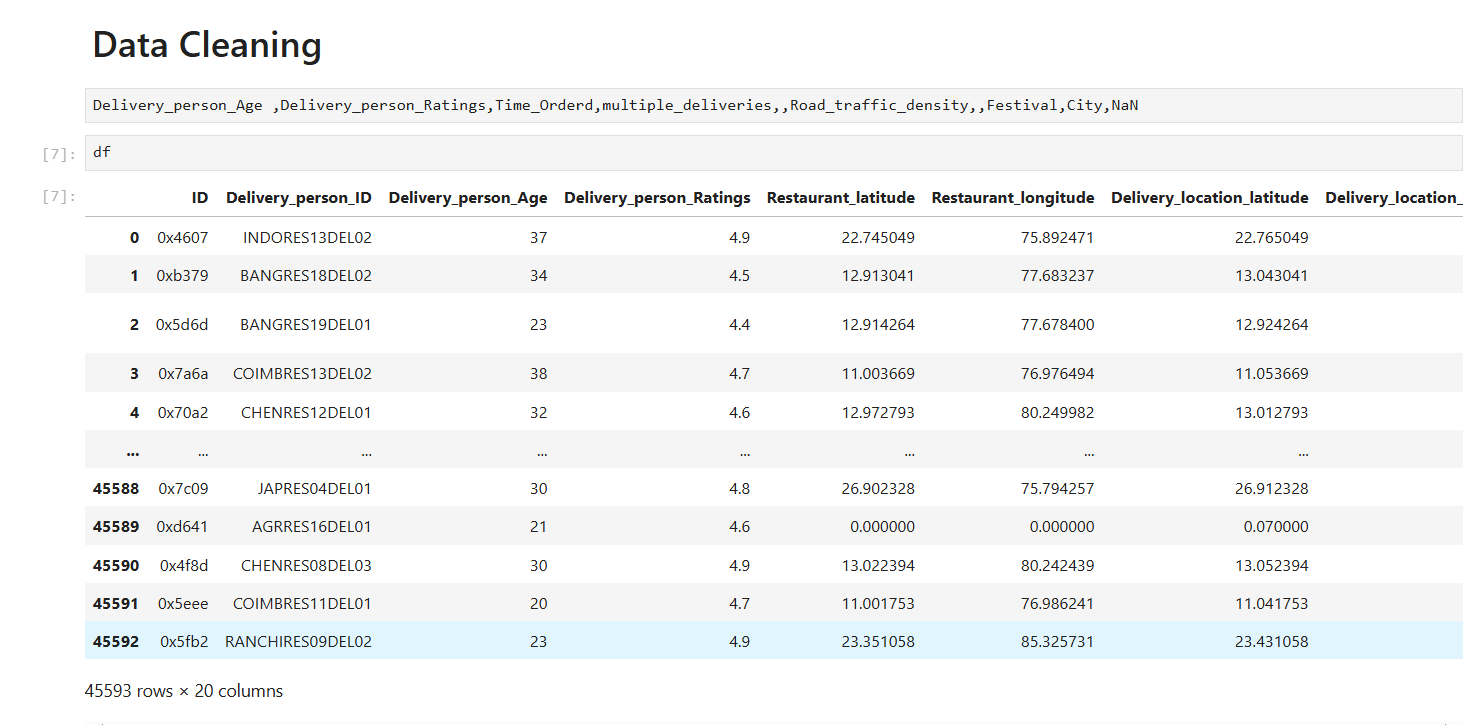
**DATA EXTRACTION:**



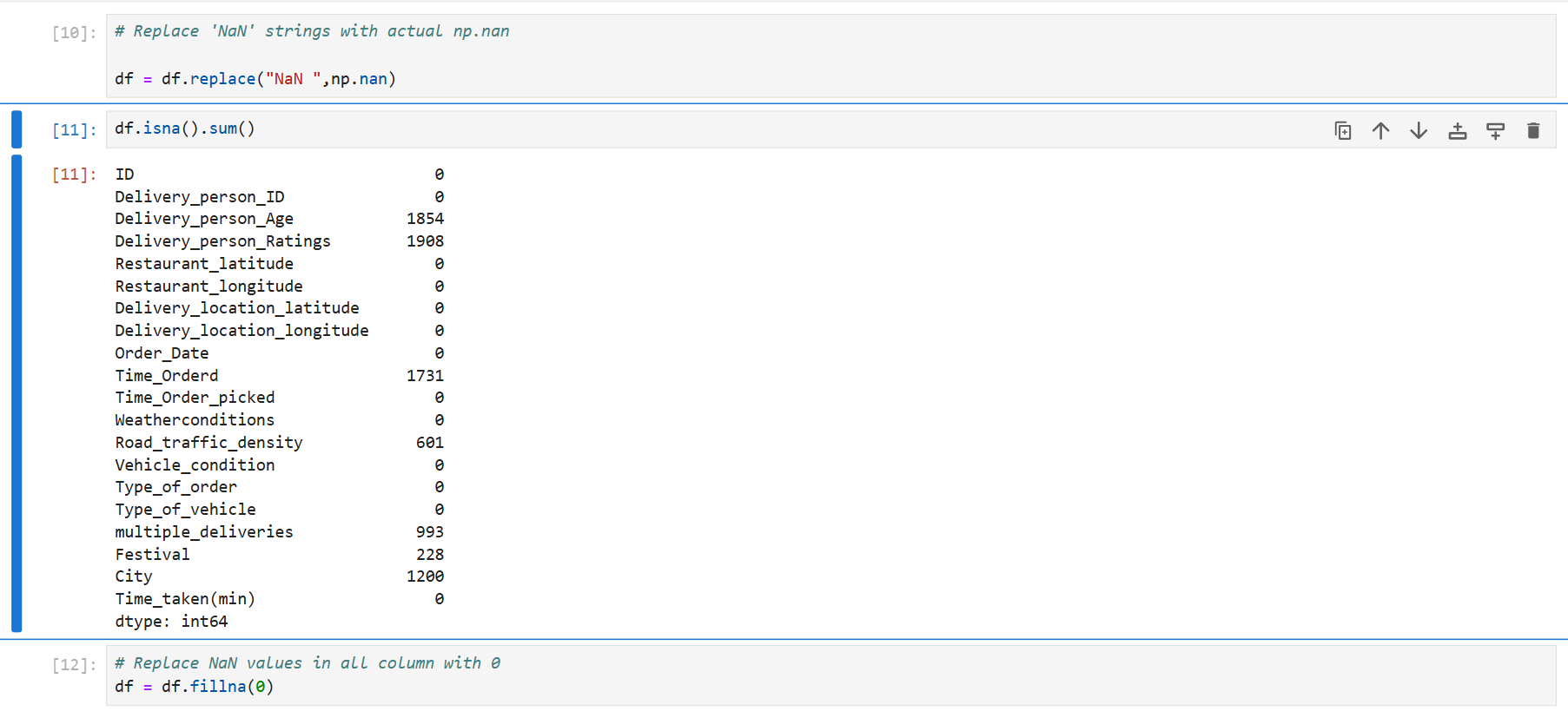
**DISPLAY DATASET INFORMATION:**



**DATA CLEANING:**

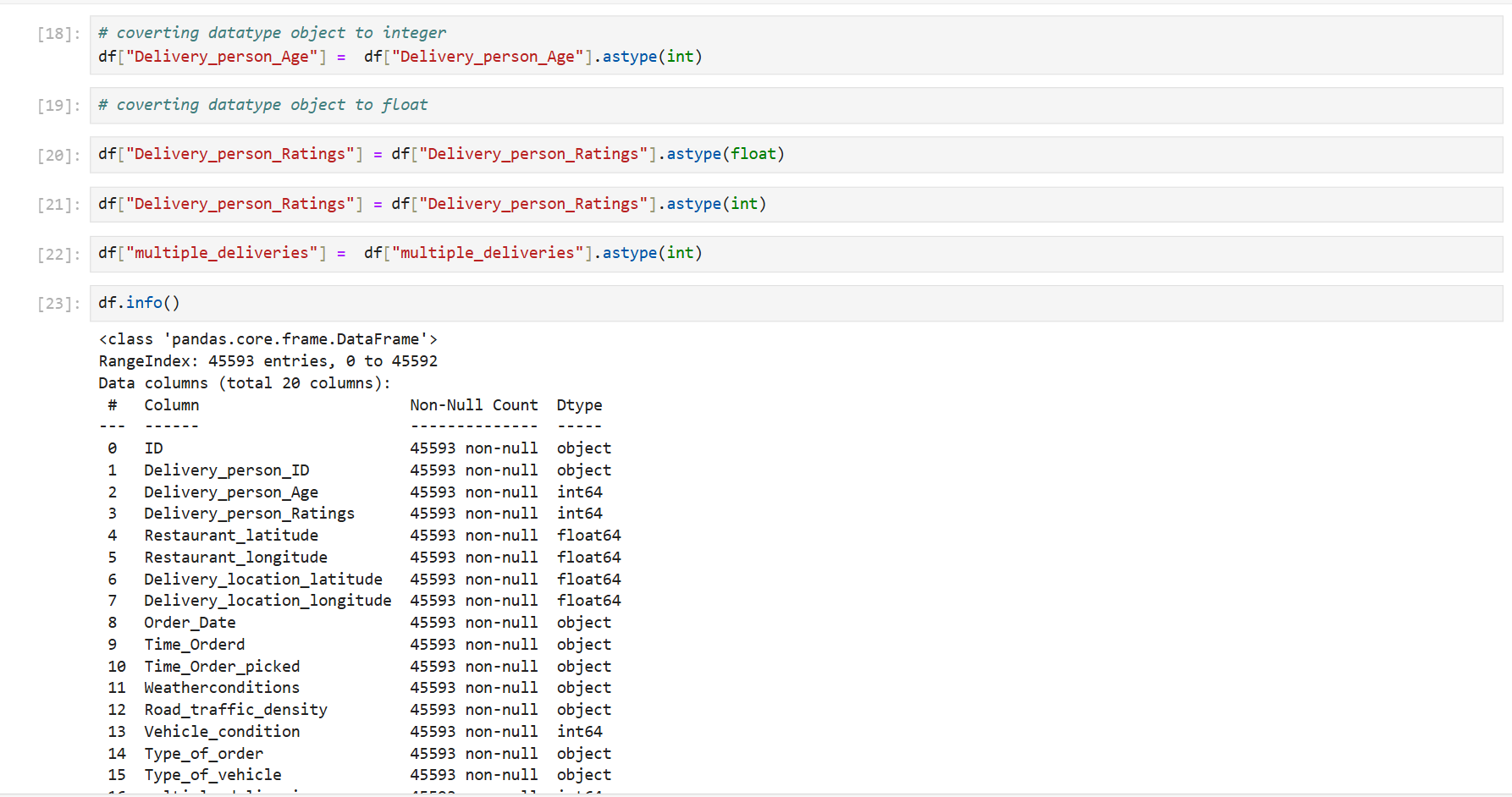


CHECKING NULL VALUES:

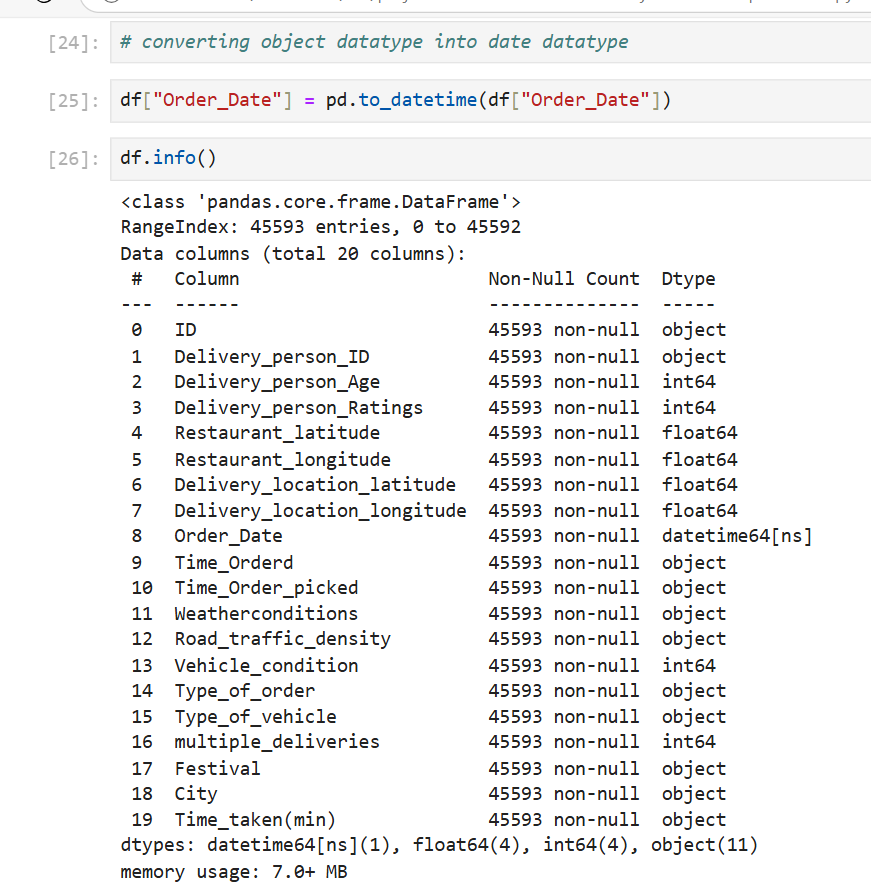


DATA TYPE CONVERSION:

OBJECT TO INTEGER:



OBJECT TO DATE:

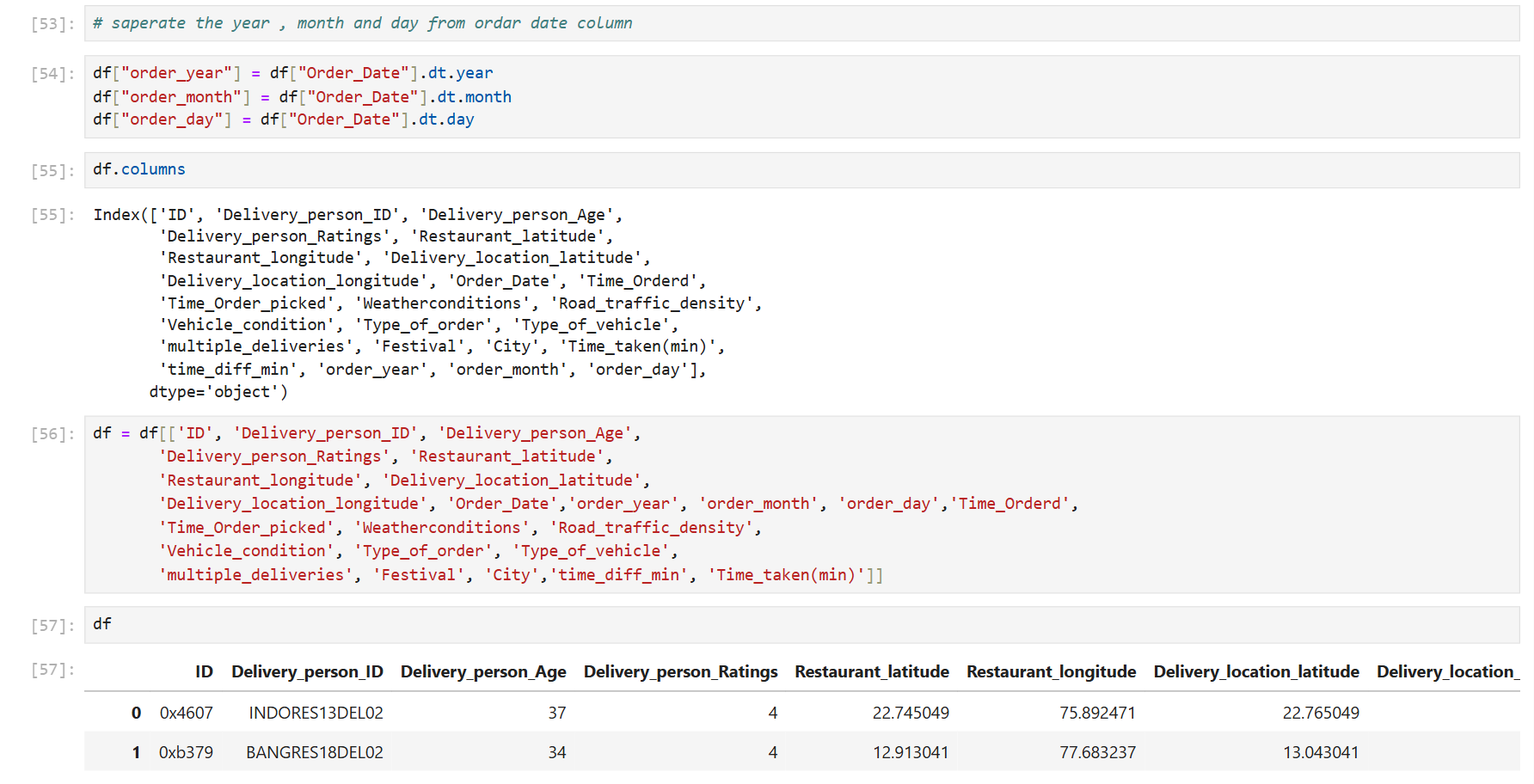




OBJECT TO INTEGER:



SAPERATING YEAR, MONTH, DAY FROM ORDER DATE COLUMN:



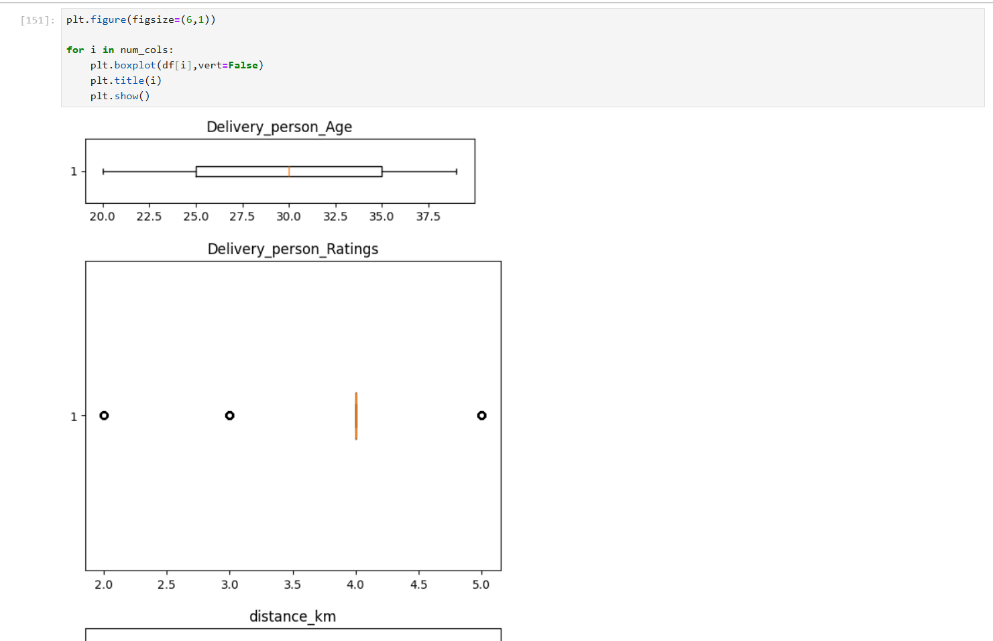
FINDING HAVERSINE DISTANCE BY USING LATITUDE’S AND LONGITUDE’S:



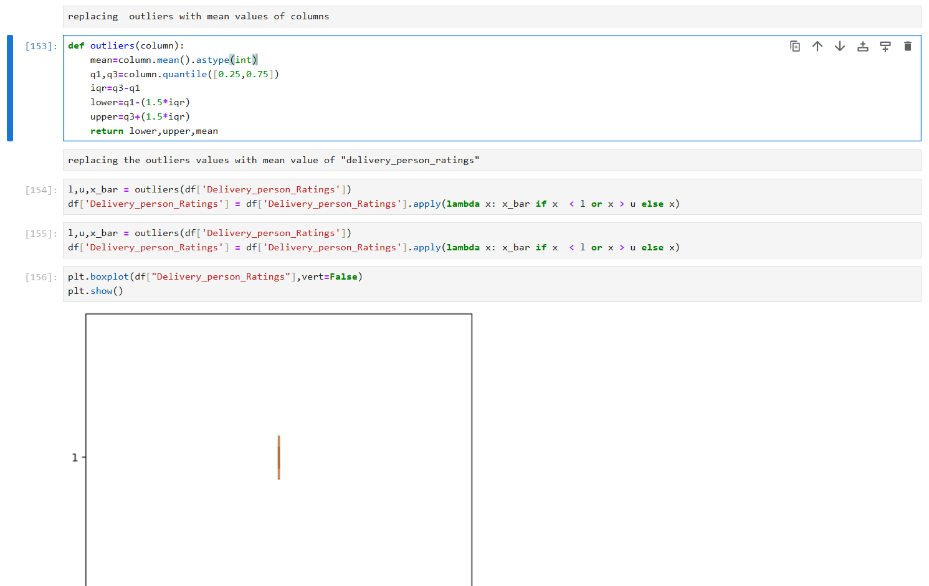
REPLACING ZERO TO MEAN VALUE:



CHECKING OUTLIERS:

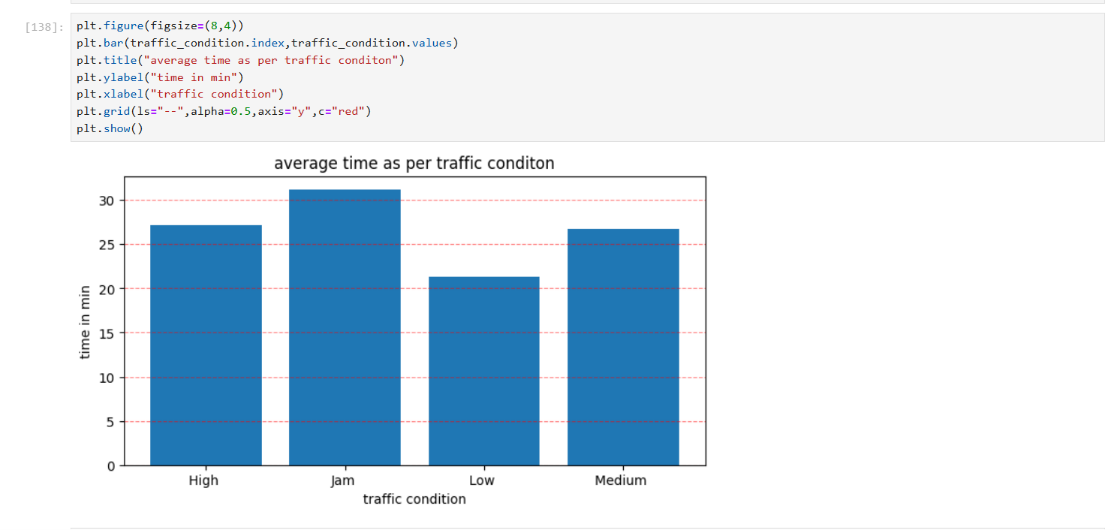


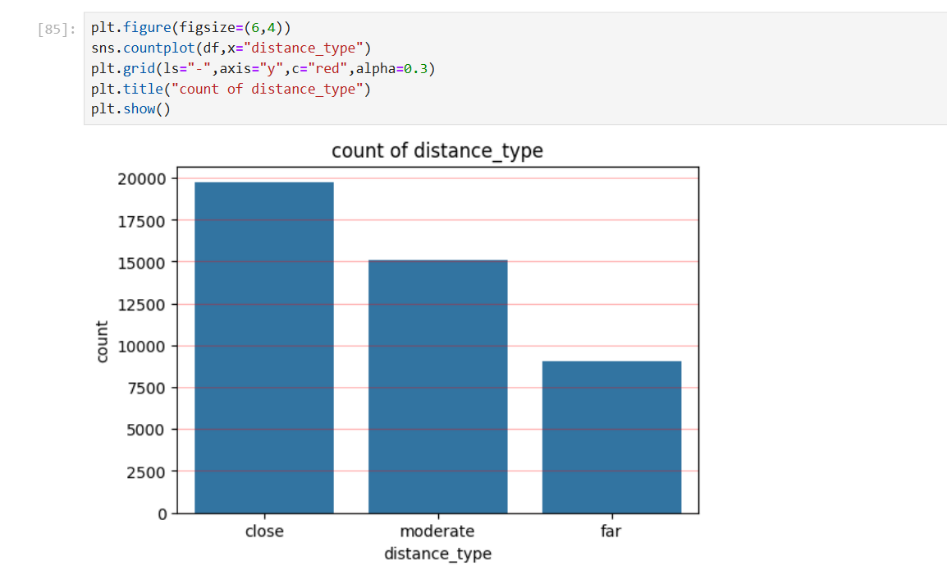
REPLACING OUTLIERS WITH MEAN VALUE:



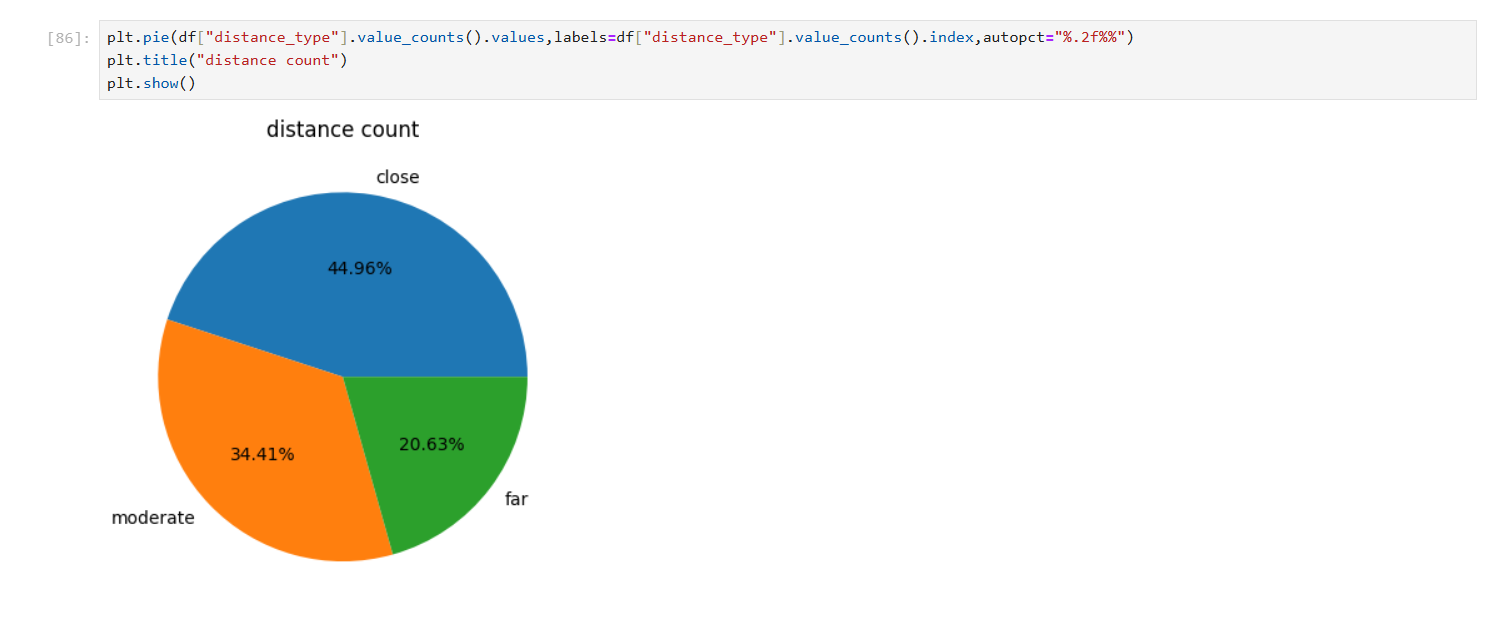
**DATA ANALYSIS:**

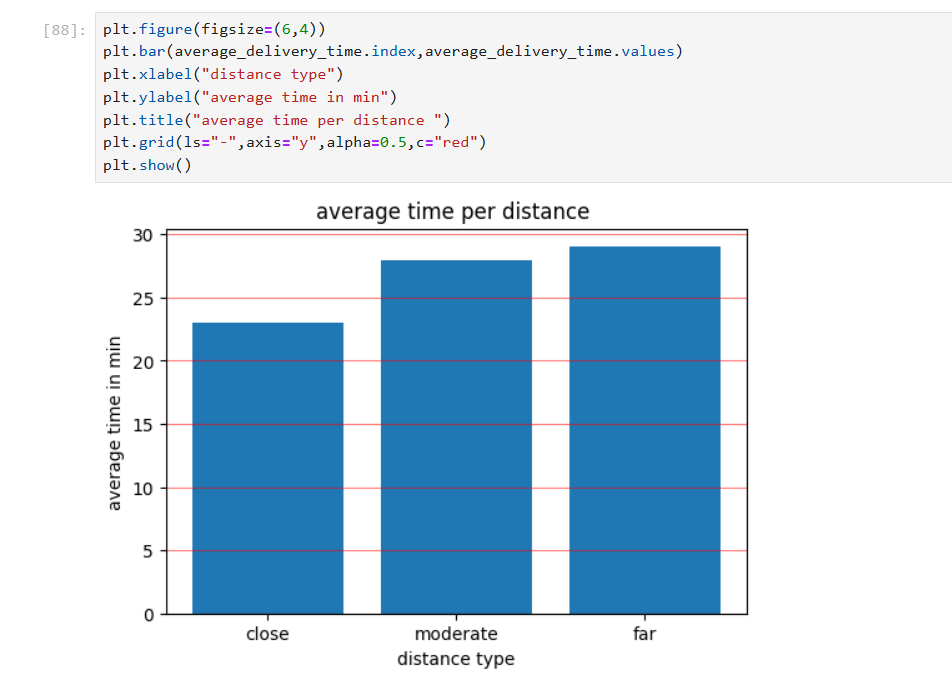


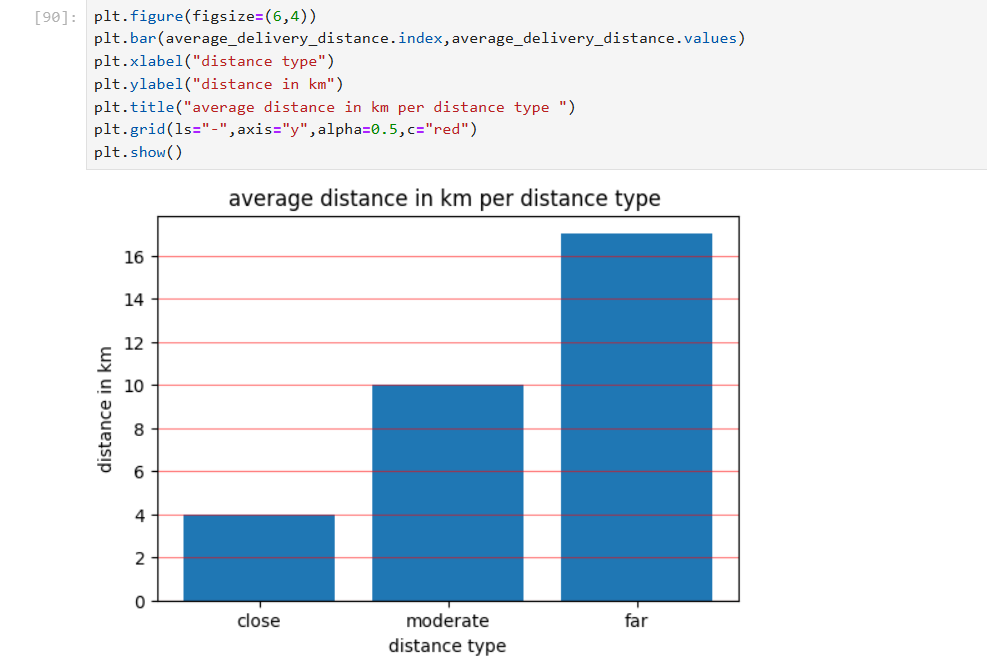


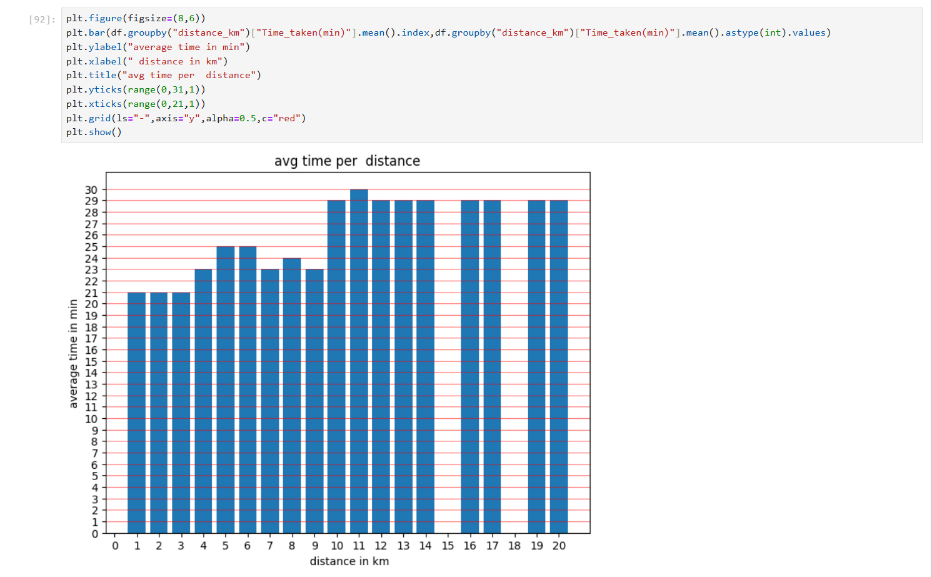


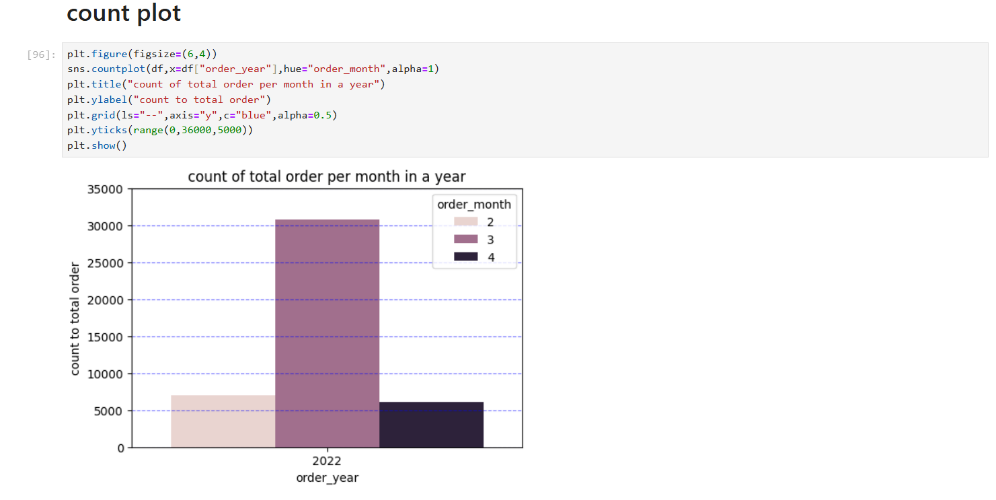


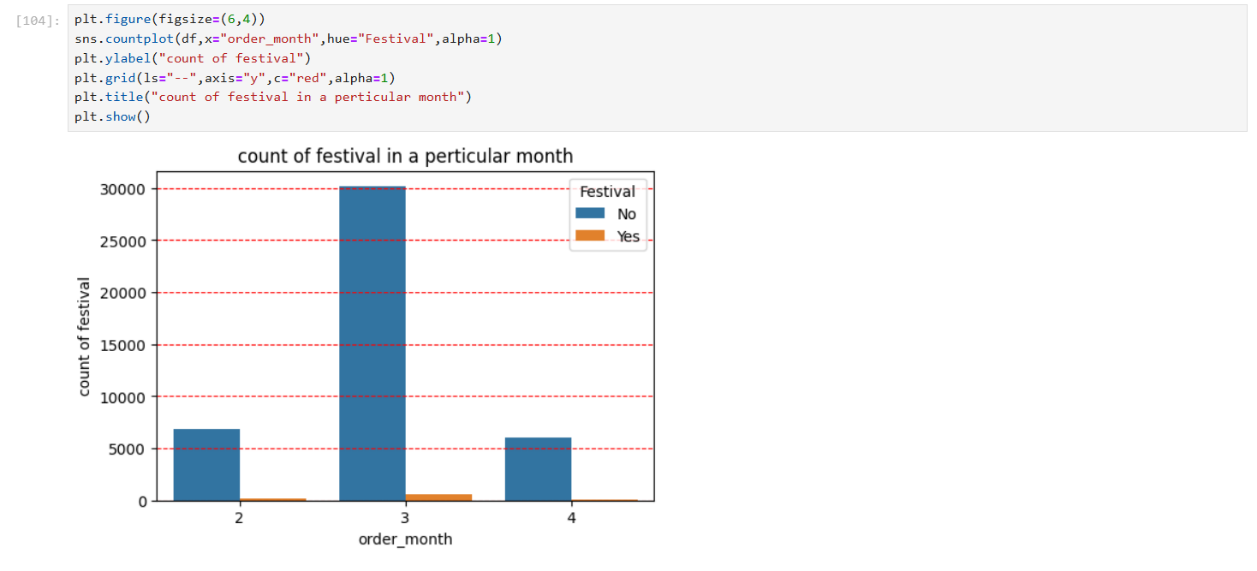


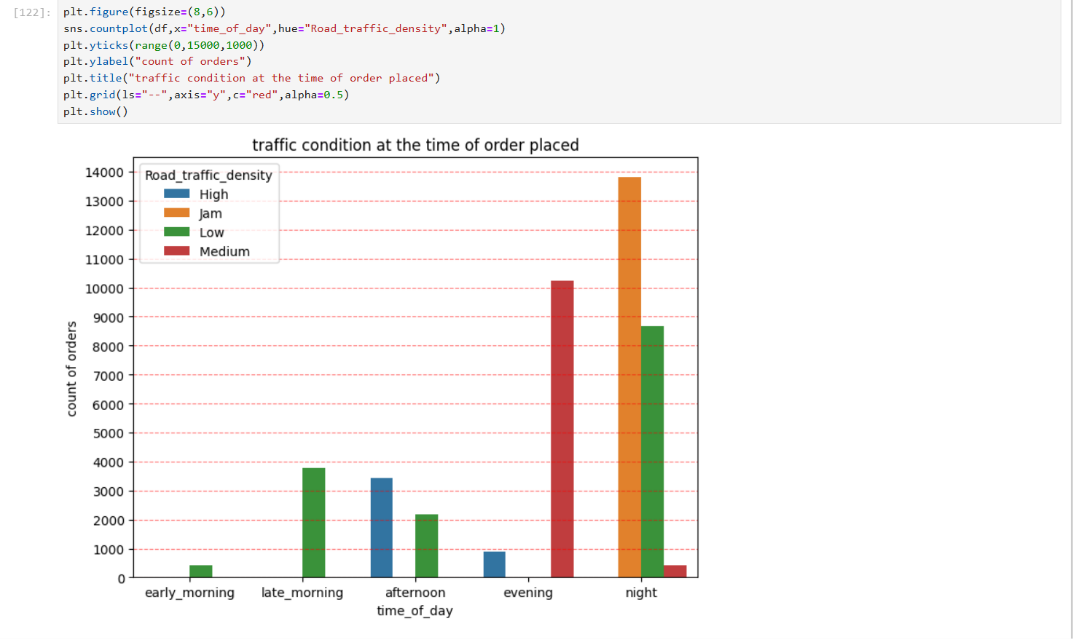
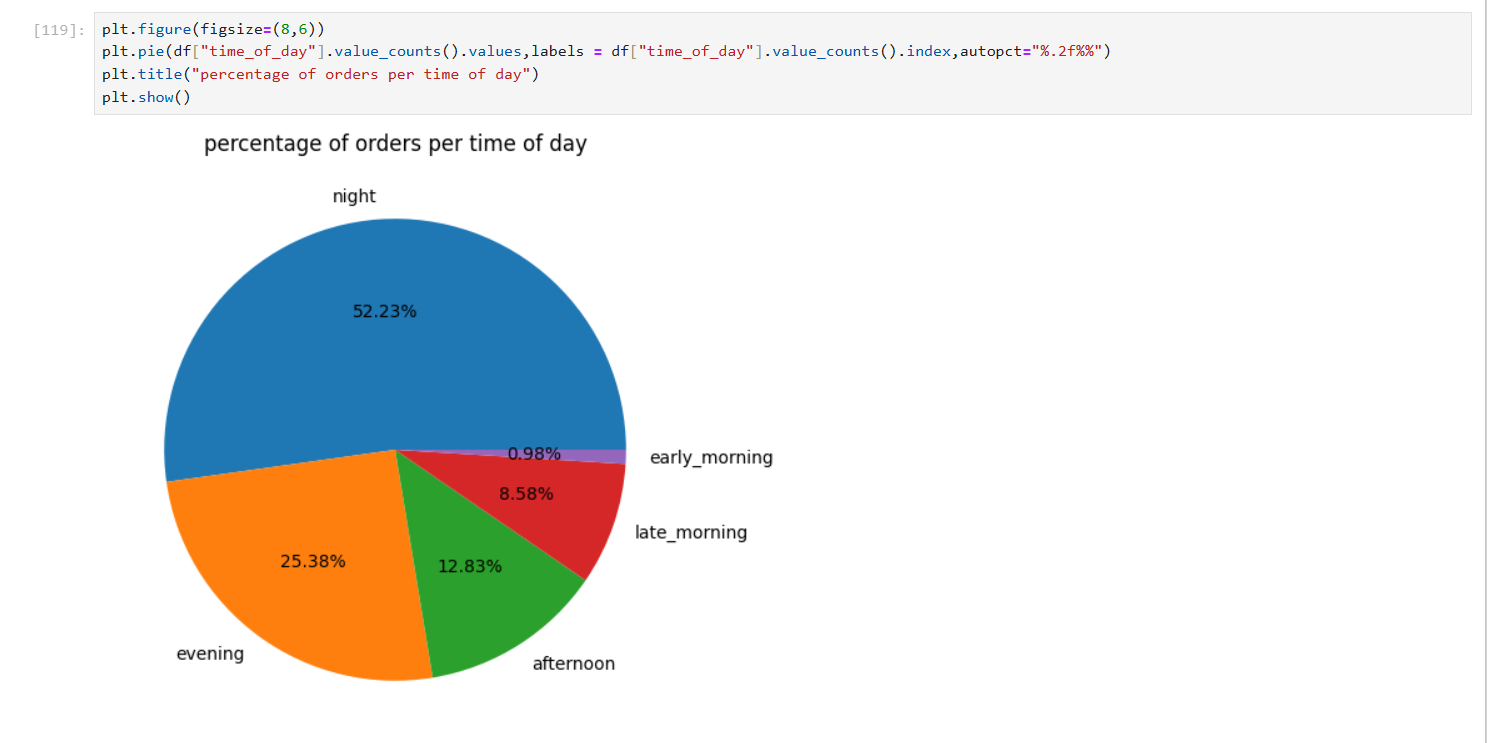












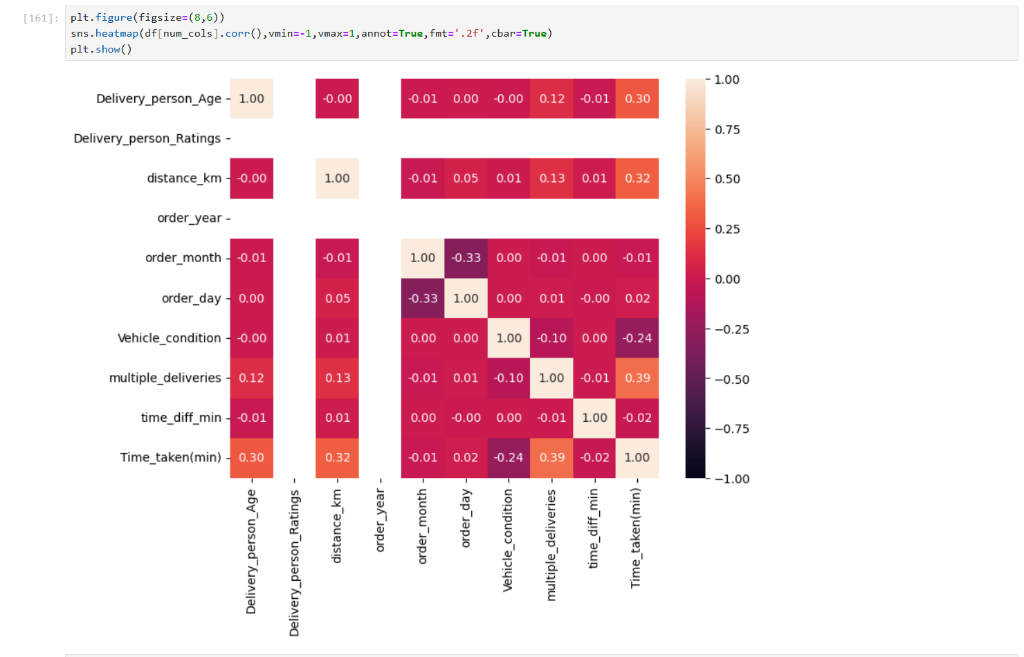


**KEY INSIGHTS FROM DATA ANALYSIS:**

1. The data analysis reveals that while most food deliveries originate from nearby areas (44.96%) with an average delivery time of 23 minutes and a distance of 4 km, delivery time and distance progressively increase for moderate (26 minutes, 10 km) and far (28 minutes, 16 km) locations, highlighting a clear correlation between distance and delivery duration.
2. Most food orders are placed at night (52.23%) and evening (25.38%), while very few come in the early morning.
3. Traffic jams are a major issue for food delivery at night compared to other times of the day
4. The majority of orders come from metropolitan and urban areas, with fewer from semi-urban and rural regions.
5. In metropolitan cities, meals are the most commonly ordered food type, more than snacks, drinks, or buffets.
6. When there is a traffic jam, it usually takes more than 30 minutes to deliver the food.

**DATA PREPROCESSING:**

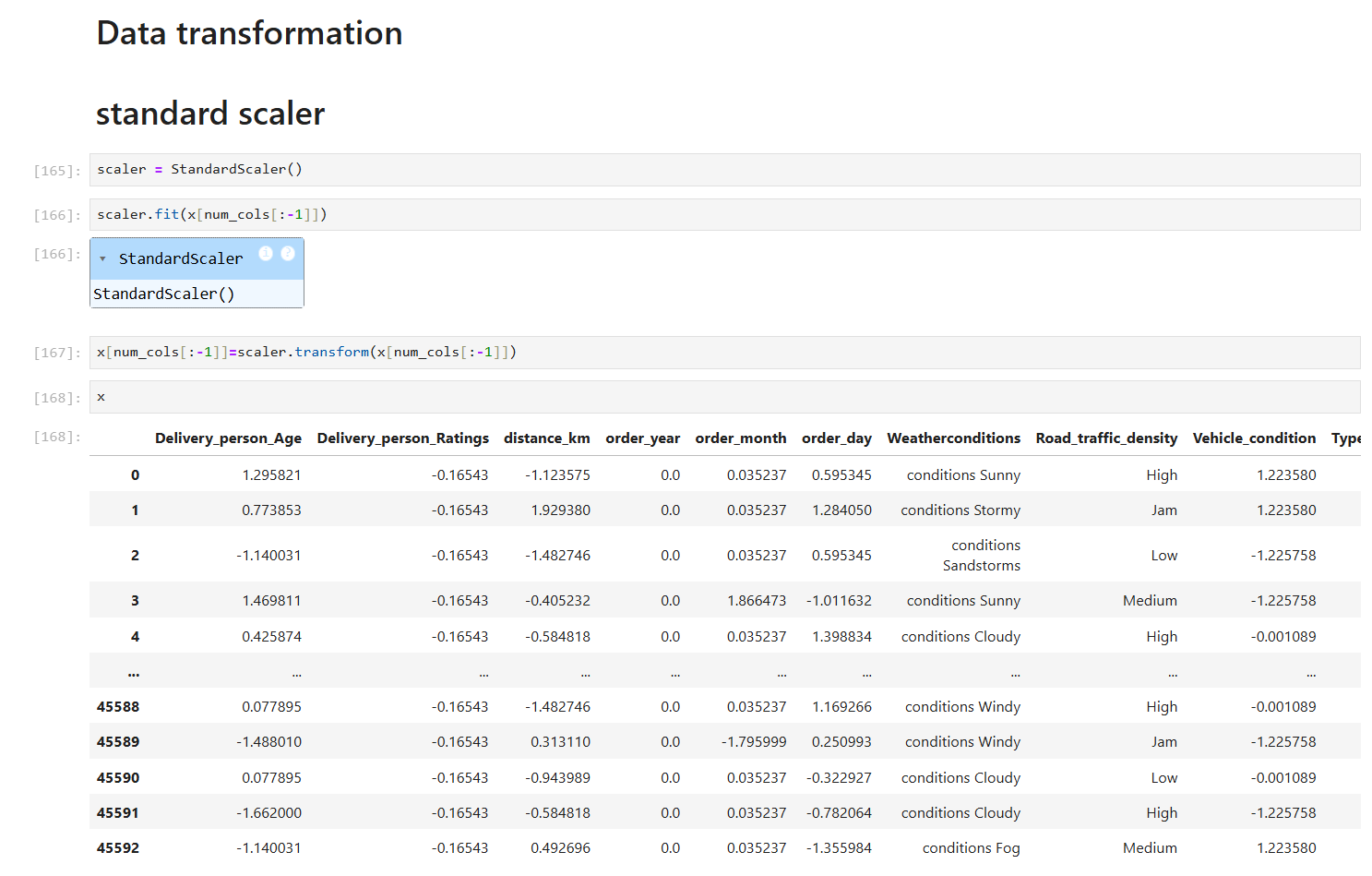
**CORRELATION METRIX:**

****

**STANDARD SCALER:**

Standardizes numerical features by calculating the mean and standard deviation of each feature, then subtracting the

mean and dividing by the standard deviation.

****

**ONE HOT ENCODER:**

PURPOSE:

Converts categorical data into a numerical format by creating a new binary column for

each unique category.

BENEFITS:

Suitable for nominal (unordered) categorical data where categories don't have a meaningful order.

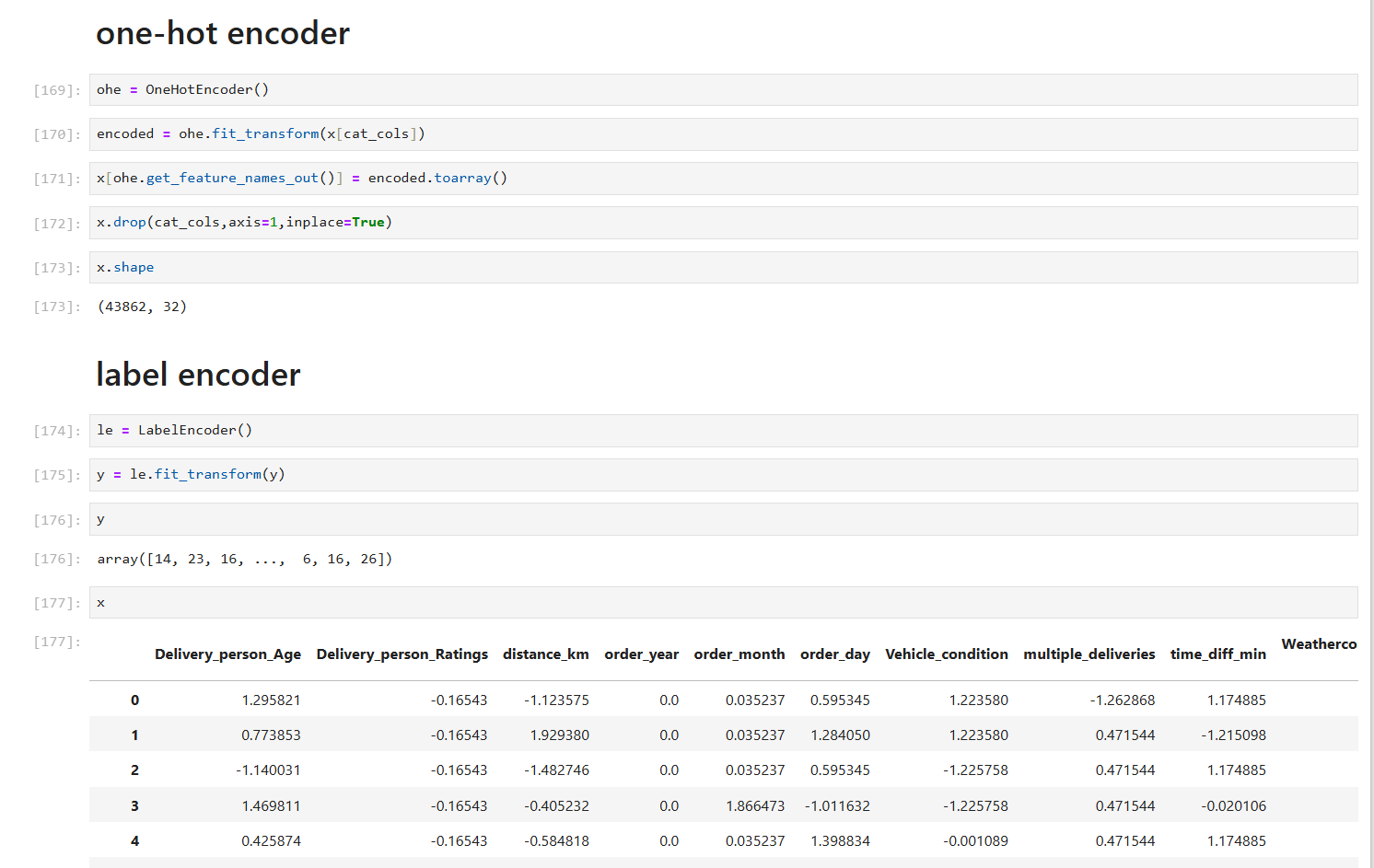
**LABEL INCODER:**

PURPOSE:

Assigns a numerical value to each unique category in a categorical column.

BENEFITS:

Simplifies categorical data for use in models that require numerical inputs.

****

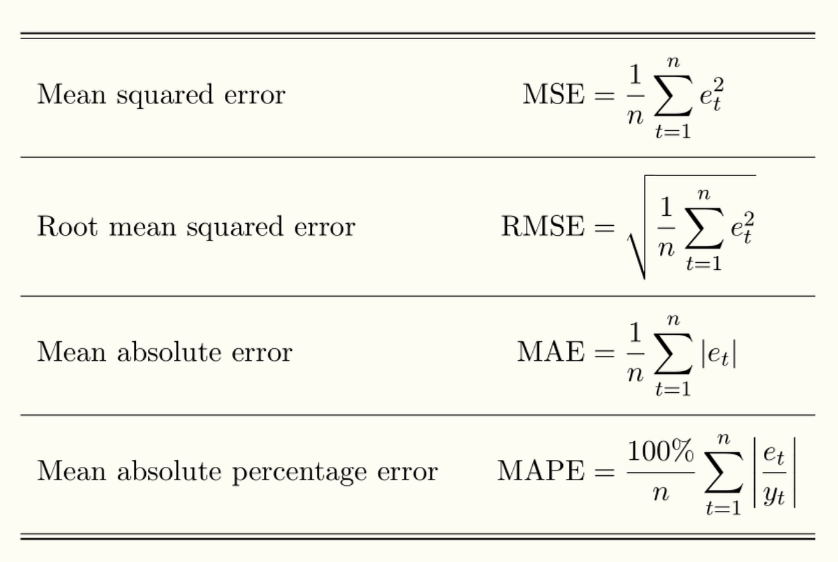
# ML MODELS EVALUATION

**THE FOLLOWING REGRESSION MODELS WERE IMPLEMENTED AND EVALUATED:**

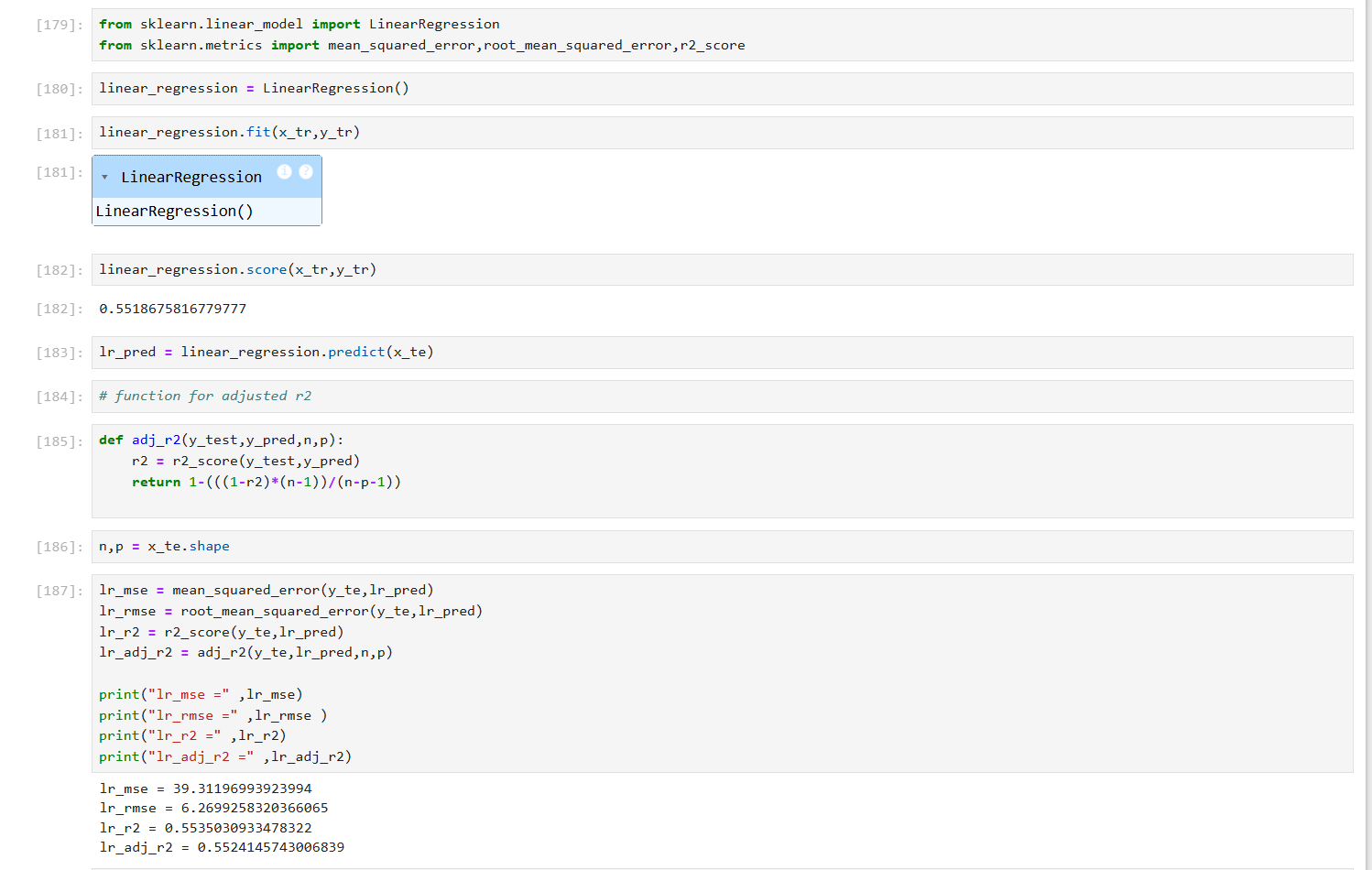
1. Linear Regression
2. Random Forest (With Hyperparameter Tuning)
3. Support Vector Machine (SVM)
4. K-Nearest Neighbors (KNN)
5. Decision Tree
6. AdaBoost
7. Stacking

**PERFORMANCE MATRIX FOR REGRESSION:**

1. **Mean Squared Error (MSE):** Squares the errors before averaging, making it more sensitive to large errors.
2. **Root Mean Squared Error (RMSE):** The square root of MSE, providing an error measure in the same units as the target variable.
3. **R-Squared (R²):** Indicates the proportion of variance in the dependent variable explained by the independent variables.
4. **Adjusted R-Squared:** A modified version of R² that accounts for the number of predictors in the model.

****

**LINEAR REGRESSION:**



##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 39.31
2. **Root Mean Squared Error (RMSE):** 6.26
3. **R-Squared (R²):** 55.35%
4. **Adjusted R-Squared:** 55.24%

**CONCLUSION:**

The linear regression model demonstrates moderate predictive power, with **R² at 55.35%,** indicating, that it explains a little over half of the variance in the dependent variable. The **Adjusted R² (55.24%)** confirms that the model remains consistent even after adjusting for the number of predictors. The  **MSE (39.31) and RMSE (6.26)** suggest that there is still some error in the predictions, indicating room for improvement.

**RANDOM FOREST REGRESSOR:**



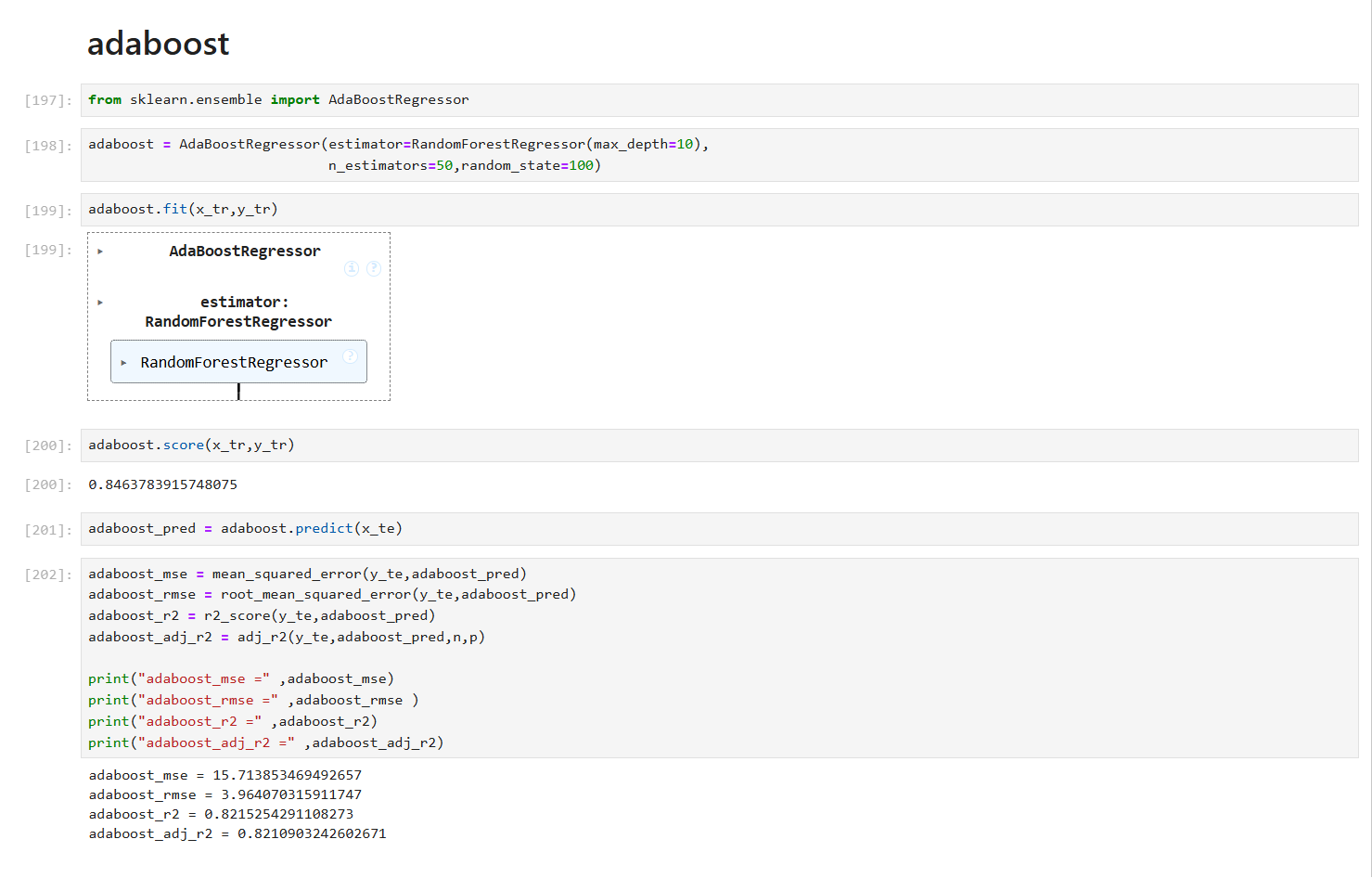
##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 16.16
2. **Root Mean Squared Error (RMSE):** 4.02
3. **R-Squared (R²):** 81.63 %
4. **Adjusted R-Squared:** 81.59 %

**CONCLUSION:**

The **random forest regression model** demonstrates **strong predictive power**, with **R² at 81.63%**, meaning it explains a significant portion of the variance in the dependent variable. The **Adjusted R² (81.59%)** confirms the model remains robust even after accounting for predictors. The **MSE (16.16)** and **RMSE (4.02)** are relatively lower than those of linear regression, indicating better accuracy and smaller errors.

**ADABOOST REGRESSOR:**



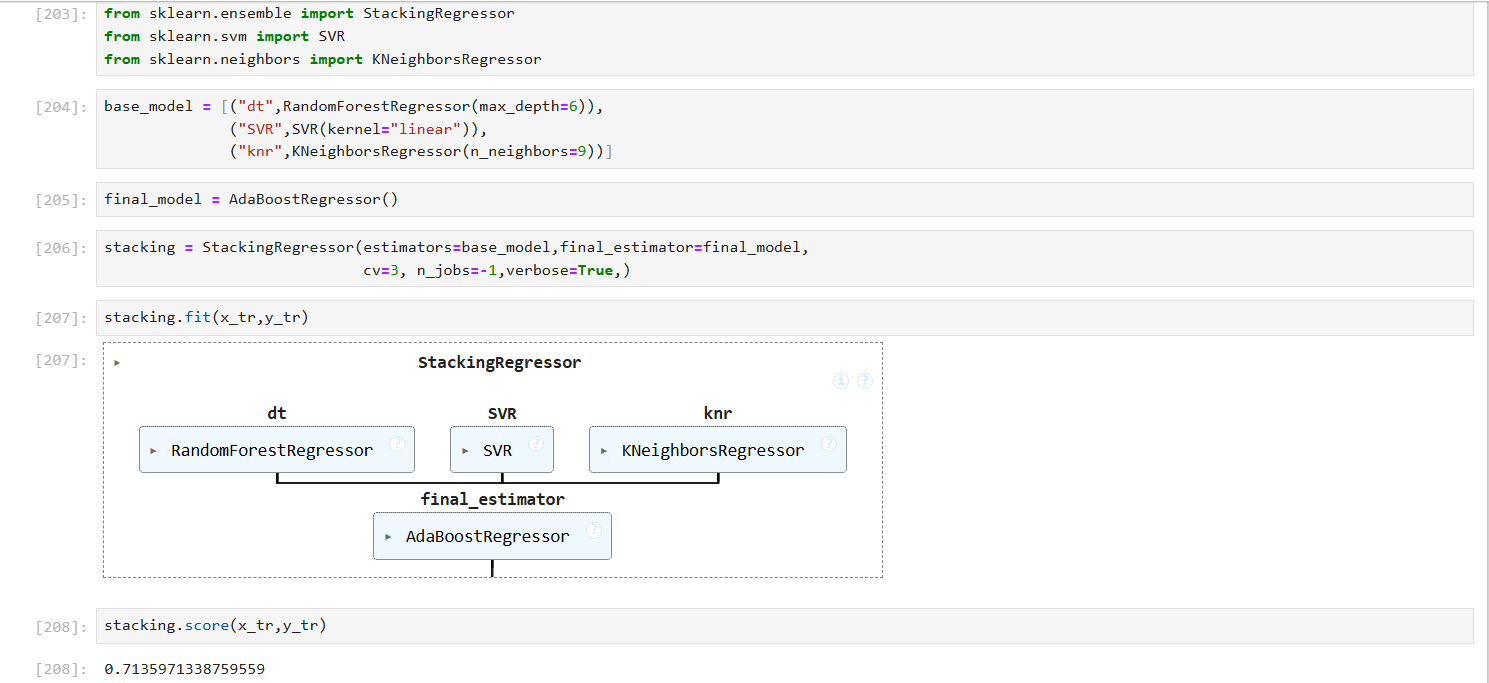
##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 15.71
2. **Root Mean Squared Error (RMSE):** 3.96
3. **R-Squared (R²):** 82.15 %
4. **Adjusted R-Squared:** 82.10 %

**CONCLUSION:**

The **AdaBoost regression model** shows **strong predictive performance**, with **R² at 82.15%**, indicating it captures a significant portion of the variance in the dependent variable. The **Adjusted R² (82.10%)** confirms the model remains robust even after accounting for predictors. The **MSE (15.71)**  and **RMSE (3.96)** are notably lower.

**STAKING REGRESSOR:**



##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 26.02
2. **Root Mean Squared Error (RMSE):** 5.10
3. **R-Squared (R²):** 70.43 %
4. **Adjusted R-Squared:** 70.36 %

**CONCLUSION:**

The **Stacking regression model** shows **moderate predictive power**, with **R² at 70.43%**, meaning it explains a fair amount of variance in the dependent variable. The **Adjusted R² (70.36%)** confirms stability even after accounting for predictors. While the **MSE (26.02)** and **RMSE (5.10)** indicate some level of error.

**DECISION TREE REGRESSOR(with hyperparameter tunning):**



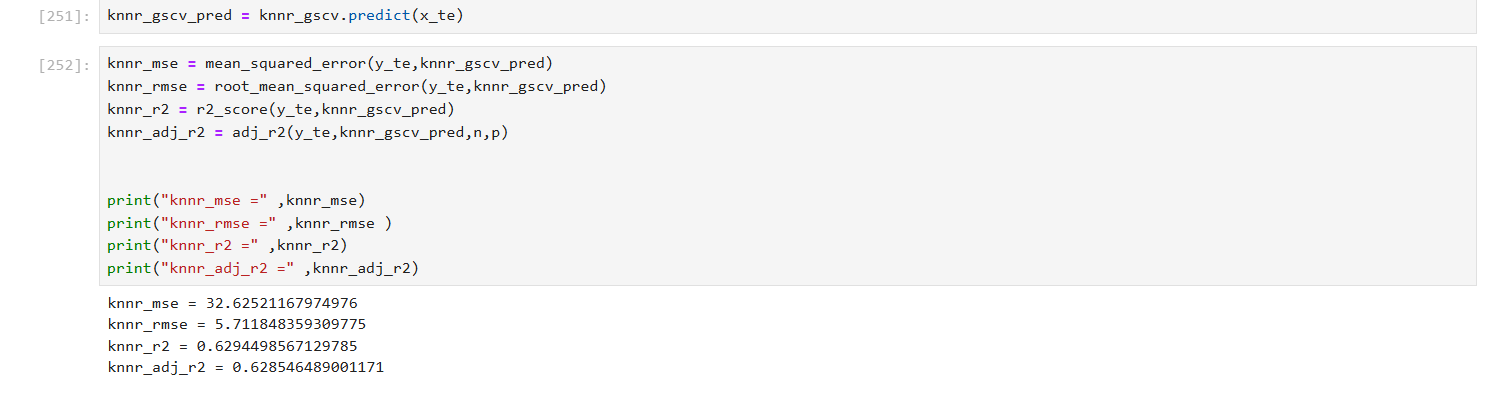
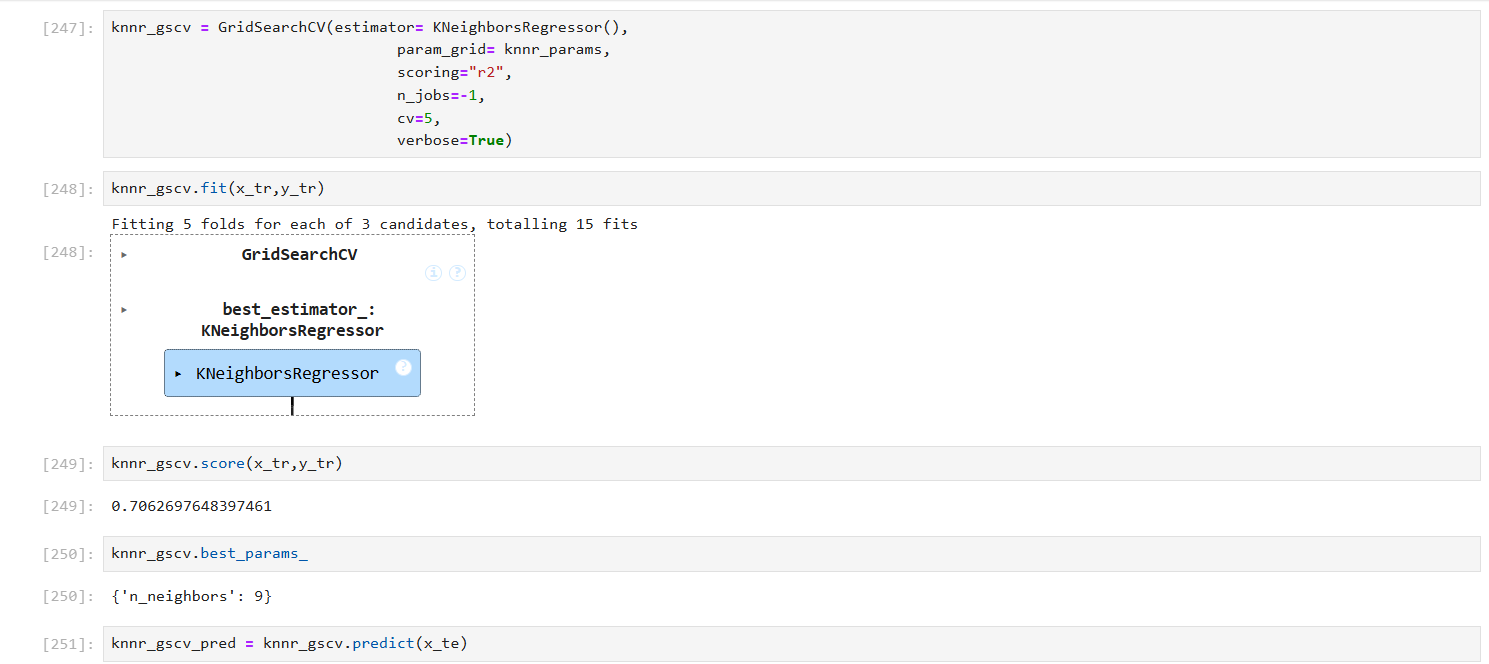
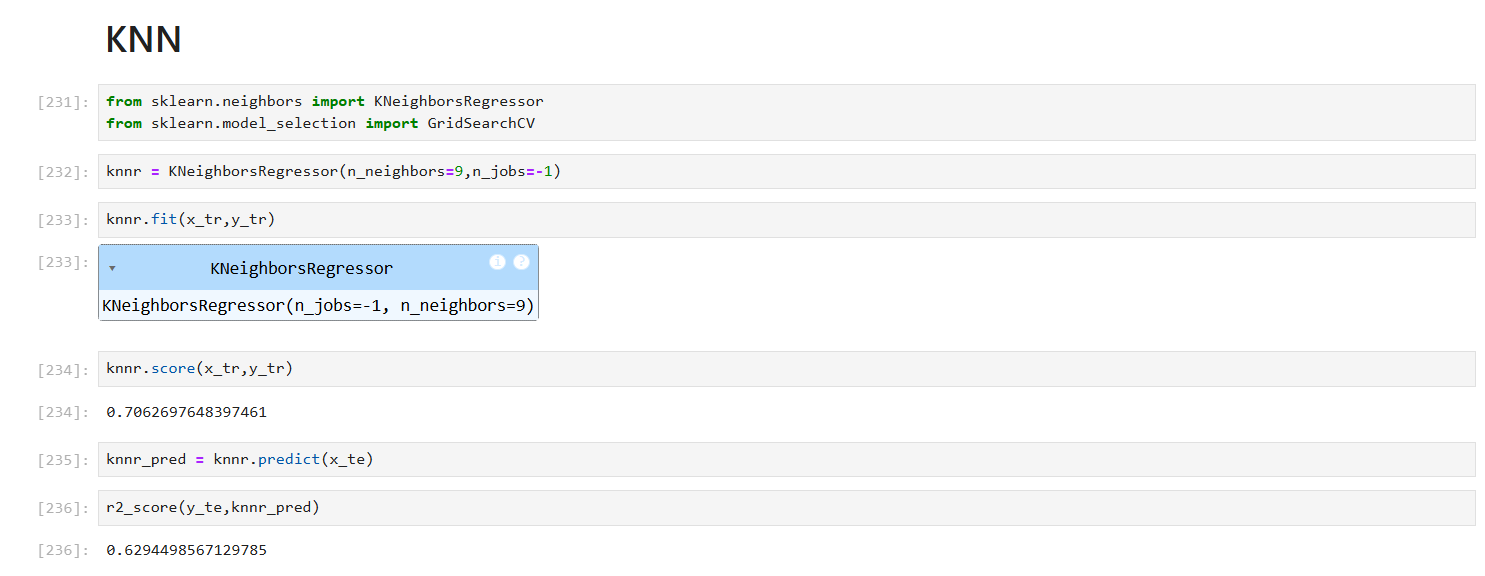
##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 17.72
2. **Root Mean Squared Error (RMSE):** 4.21
3. **R-Squared (R²):** 79.86 %
4. **Adjusted R-Squared:** 79.81 %

**CONCLUSION:**

The Decision Tree model demonstrates strong predictive performance with an **R² of 79.86%,** indicating that it explains a significant portion of the variance in the target variable. The relatively low **MSE (17.72) and RMSE (4.21)** further suggest that the model makes accurate predictions with minimal error. Overall, the model is well-suited for the given regression task.

**K NEAREST NEIGHBOURS REGRESSOR (WITH GRID SEARCH CV):**



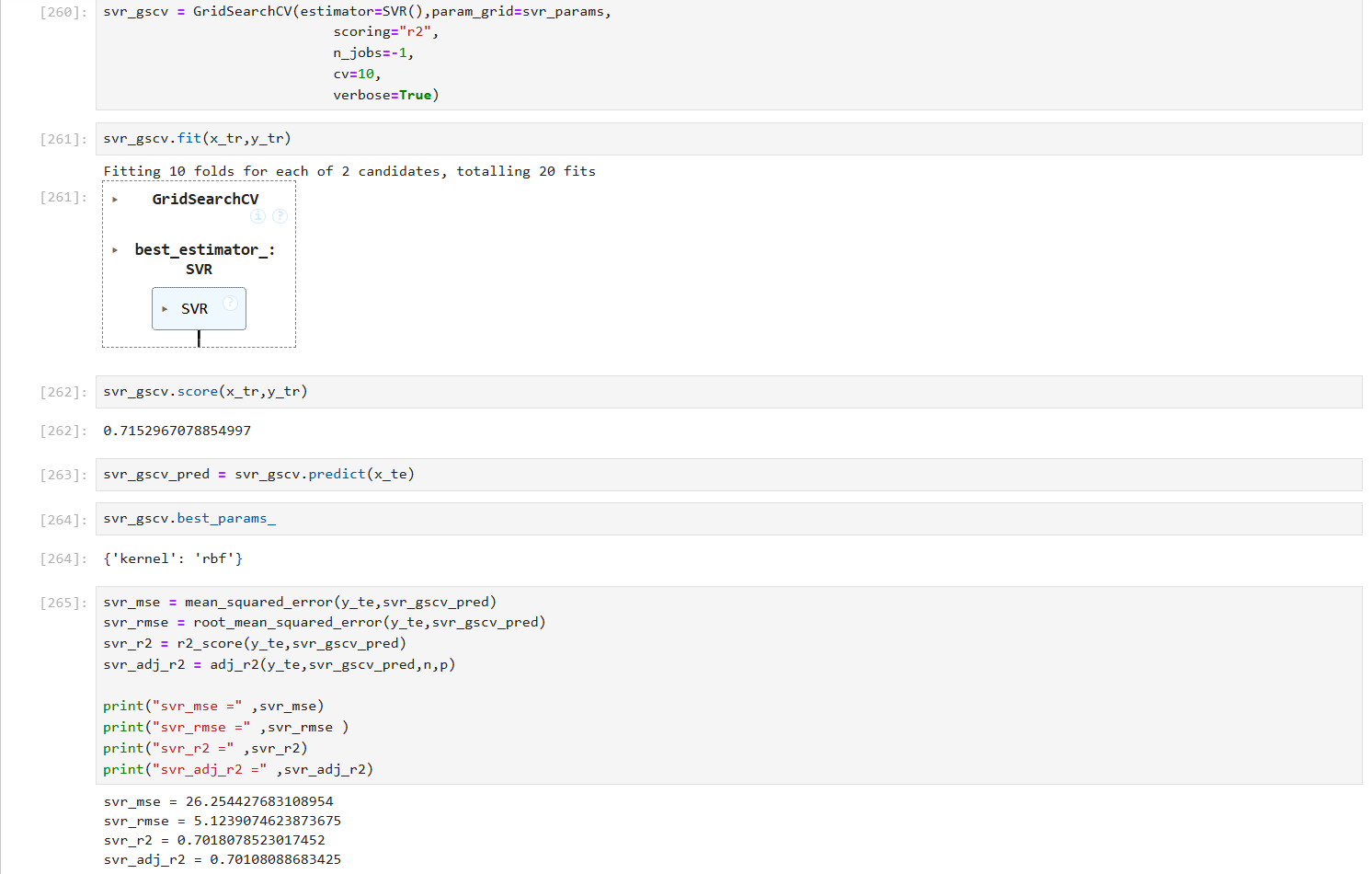
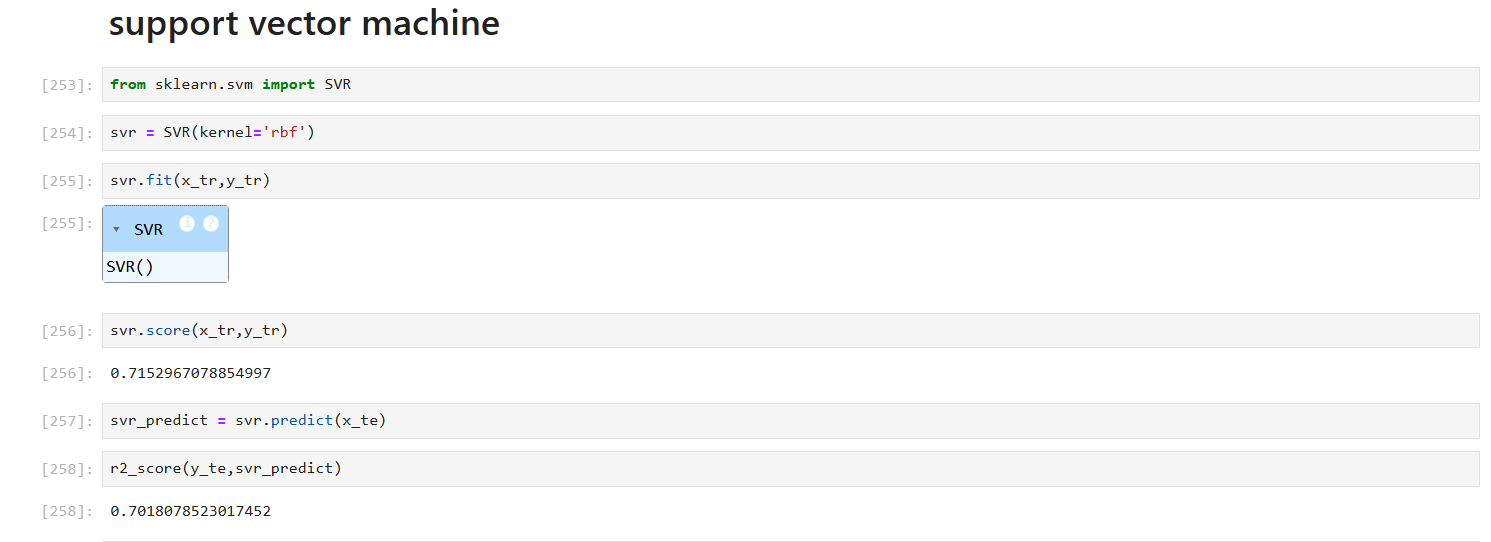
##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 32.62
2. **Root Mean Squared Error (RMSE):** 5.71
3. **R-Squared (R²):** 62.94 %
4. **Adjusted R-Squared:** 62.85 %

**CONCLUSION:**

The K-Nearest Neighbours model with GridSearchCV shows moderate predictive performance, with an **R² of 62.94%,** indicating it explains a fair amount of the variance in the target variable. However, the higher **MSE (32.62) and RMSE (5.71)** compared to the Decision Tree model suggest less accurate predictions. This model may benefit from further tuning or may not be the best choice for this regression task.

**SUPPORT VECTOR MACHIME REGRESSOR(WITH GRID SEARCH CV):**



##### REGRESSION REPORT INSIGHTS:

1. **Mean Squared Error (MSE**): 26.25
2. **Root Mean Squared Error (RMSE):** 5.12
3. **R-Squared (R²):** 70.18 %
4. **Adjusted R-Squared:** 70.10 %

**CONCLUSION:**

The **Support Vector Machine (SVM)** model demonstrates good predictive performance with an **R² of 70.18%,** indicating it captures a substantial portion of the variance in the data. The **MSE (26.25) and RMSE (5.12)** are reasonably low, reflecting better accuracy than the KNN model but slightly below the Decision Tree. Overall, SVM is a reliable model for this regression task, with room for further improvement.

# CONCLUSION

### **PERFORMANCE COMPARISON:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **MSE** | **RMSE** | **R² Score** | **Adjusted R²** |
| Linear Regression | 39.31 | 6.27 | 55.40% | 55.20% |
| Random Forest | 16.17 | 4.021 | 81.60% | 81.60% |
| K-Nearest Neighbors | 32.63 | 5.712 | 62.90% | 62.90% |
| Support Vector Machine | 26.25 | 5.124 | 70.20% | 70.10% |
| Decision Tree | 17.72 | 4.21 | 79.90% | 79.80% |
| Adaboost | 15.71 | 3.964 | 82.20% | 82.10% |
| Stacking | 26.03 | 5.102 | 70.40% | 70.40% |

* **Best Model:**
  + - AdaBoost
    - Random Forest (with Hyperparameter Tuning)
    - Decision Tree
* **Why?**

**1. AdaBoost**

* MSE: 15.714 *(lowest of all models)*
* RMSE: 3.964 *(lowest of all models)*
* R² Score: 82.2%
* Adjusted R²: 82.1%
* **Why Best:** AdaBoost achieved the highest prediction accuracy and lowest error, making it the most reliable for predicting food delivery time.

**2. Random Forest (with Hyperparameter Tuning)**

* MSE: 16.166
* RMSE: 4.021
* R² Score: 81.6%
* Adjusted R²: 81.6%
* **Why Best:** With slightly higher error than AdaBoost, Random Forest still performed very well, capturing a large portion of variance and maintaining robust generalization.

**3. Decision Tree**

* MSE: 17.724
* RMSE: 4.210
* R² Score: 79.9%
* Adjusted R²: 79.8%
* **Why Best:** Although simpler, the Decision Tree model delivered competitive accuracy, and is easy to interpret and implement, making it a practical choice.

**LIMITATIONS & FUTURE SCOPE:**

**Limitations:**

* 1. **Static Dataset:**  
     The model is trained on historical data and does not include **real-time variables** such as live traffic or current weather conditions, which can significantly affect delivery time.
  2. **Limited Feature Diversity:**  
     Although key features like distance and preparation time were considered, factors such as **driver experience, route optimization, or customer availability** were not included, possibly affecting accuracy.
  3. **Model Complexity vs Interpretability:**  
     While models like AdaBoost and Random Forest performed best, they are more **complex and less interpretable** than simpler models like Decision Tree, making them harder to explain to non-technical stakeholders.
  4. **Data Imbalance or Noise:**  
     Some features may have **inconsistent or noisy entries** (e.g., special instructions or unusual order timings), which could introduce bias or reduce generalization.

**Future Scope:**

* 1. **Integration of Real-Time Data:**  
     Incorporating **live traffic, weather updates, and GPS tracking** can make the predictions more dynamic and improve accuracy in real-world deployment.
  2. **Feature Expansion:**  
     Including additional features such as **driver metrics, delivery time windows, restaurant workload**, and **customer feedback** can enhance model depth and performance.
  3. **Model Deployment:**  
     Implementing the best-performing model (e.g., AdaBoost) into a **real-time dashboard or API** could help businesses predict delays and allocate resources proactively.
  4. **Explainable AI (XAI):**  
     Integrating tools for **model interpretability** (e.g., SHAP, LIME) will help explain predictions and build trust among end-users and business teams.
  5. **Scalability Across Regions:**  
     Testing and adapting the model for **different cities or regions** can evaluate how well it generalizes and performs under varying delivery conditions.