**A**

**Course End Project Report on**

**DELIVER ON TIME**

**Is submitted in partial fulfillment of the Requirements for the Award of CIE of**

**DATA ANALYSIS AND VISUALIZATION-22ADE01**

**in**

**B.E, IV-SEM, INFORMATION TECHNOLOGY**

**Submitted by**

**KOLLI HARSHITHA, 160122737010**

**MARRI SHIVANI, 160122737012**

**COURSE TAUGHT BY:**

**Dr Ramakrishna Kolikipogu Professor, Dept of IT.**



**DEPARTMENT OF INFORMATION TECHNOLOGY CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)**

**(Affiliated to Osmania University;Accredited by NBA,NAAC,ISO) kokapet(V),GANDIPET(M),HYDERABAD-500075**

**W**[**ebsite:www.cbit.ac.in**](http://www.cbit.ac.in/)

**2023-2024**



### CERTIFICATE

This is to certify that the course end project work entitled **”DELIVER ON TIME”** is submitted by **KOLLI HARSHITHA(160122737010), MARRI SHIVANI(160122737012)** in partial fulfillment of the requirements for the award of CIE Marks of **DATA ANALYSIS AND VISUALIZATION (22ADE01)** of **B.E, IV-SEM, INFORMATION TECHNOLOGY** to CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A) affiliated to

OSMANIA UNIVERSITY, Hyderabad is a record of bonofide work carried out by them under my supervision and guidance. The results embodied in this report have not been submitted to any other University or Institute for the award of any other Degree or Diploma.

**Signature of Course Faculty Dr Ramakrishna Kolikipogu Professor of IT**

Kokapet(V),Gandipet(M),Ranga Reddy (Dist.)–500075, Hyderabad, T.S.

## Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr Ramakrishna Kolikipogu, Professor of IT** for his able guidance and useful suggestions, which helped us in completing the Course End Project in time.

We are particularly thankful to **HoD, Principal and Management**, for their support and encouragement, which helped us to mould our project into a successful one.

We also thank all the staff members of IT Department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

**KOLLI HARSHITHA, 160122737010**

**MARRI SHIVANI, 160122737012**

## Abstract

## This project aims to analyse the time dynamics of online delivery services through comprehensive data analysis and exploration. Leveraging a dataset containing various attributes related to delivery operations, including delivery personnel information, weather conditions, and order details, the project unfolds in several stages. First, a thorough understanding of the dataset is established, elucidating the meaning and significance of each attribute. Subsequently, data cleaning procedures are employed to address missing values and ensure data integrity. Following this, exploratory data analysis (EDA) techniques are applied to delve into the distribution of the target variable, Time taken, and to uncover insights into the relationships between numerical and categorical attributes and delivery time. Moreover, feature engineering is conducted to create additional features that may impact delivery time, such as time of day and distance between the restaurant and delivery location. Graphical analysis is then employed to assess the relationships between different attributes and delivery time. Through these systematic steps, this project aims to provide valuable insights into the factors influencing delivery time in online delivery services.

**Keywords**: Online delivery services, Time dynamics, Data analysis ,Delivery operations, Numerical attributes, Categorical attributes , Feature engineering, Graphical analysis.

.

### Table of Contents

**Title Page No.**

[Acknowledgement](#_bookmark0)………………………………………………………………………………………… i

[Abstract](#_bookmark1)…………………………………………………………………………………………………………… ii

[List of Tables](#_bookmark2)………………………………………………………………………………………………… iii

[List of Figures](#_bookmark3)………………………………………………………………………………………………. vi

[Abbreviations](#_bookmark4)…………………………………………………………………………………………………. v

CHAPTER 1 [Introduction](#_bookmark5)……………………………………………………………………… 1

* 1. [Definition of Problem](#_bookmark8)…………………………………………………………………………. 1
  2. [Objectives](#_bookmark9)…………………………………………………………………………………………….. 2

**CHAPTER 2** [**Methodology**](#_bookmark20)…………………………………………………………………… 3

* 1. [Data collection and Dataset description](#_bookmark23)………………………………………….. 3

2.2 [Data cleaning and preprocessing](#_bookmark25)………………………………………………………. 5

**CHAPTER 3** [**System Architecture and Implementation**](#_bookmark30)…………….. 6

3.1 [Google Colab](#_bookmark31)………………………………………………………………………………………. 6

* + 1. [What is Google Colab?](#_bookmark33)............................................................ 7

3.2 [Benefits of Google Colab](#_bookmark34)…………………………………………………………………… 7

3.2.1 [Why Choose Google Colab?](#_bookmark35).………………………………………………… 7

3.2.2 [Notebook in Google Colab](#_bookmark36)……………………………………………………. 7

3.2.3 [Google Colab Features](#_bookmark37)………………………………………………………….. 8

|  |  |  |
| --- | --- | --- |
| [3.3 Code](#_bookmark43) | [Snippets](#_bookmark43)…………………………………………………………………………………. | 9 |
| [3.3.1](#_bookmark44) | [Importing libraries and Data loading](#_bookmark44)………………………………………… | 10 |
| [3.3.2](#_bookmark47) | [Data cleaning and preprocessing](#_bookmark47)….………………………………………….. | 14 |
| [3.3.3](#_bookmark49) | Feature Engineering…….…………………………………………………………………. | 15 |
| [3.3.4](#_bookmark49) | | [Exploratory Data Analysis](#_bookmark49)..………………….…………………………………… | 21 |
| 3.3.5 | | Conclusion………………………....…………..…….…………………………………… | 23 |
|  |  |  |
| [**References**](#_bookmark59) | …………………………………………………………………………………………………… | 24 |

­­­

### List of Figures

* 1. Dataset 3
  2. Google Colab 6
  3. [importing libraries and Dataset loading](#_bookmark45) 9
  4. [Staring cells of our Data set](#_bookmark46) 9
  5. Identifying number of rows and columns 10
  6. Statistics for Numerical features 10
  7. Removing Unneccessary columns 11
  8. Addressing Missing values 12
  9. [Handling](#_bookmark54) Duplicate Values 12
  10. Addressing Datatypes 13
  11. Handling Outliners 13
  12. Handling Zero values 14
  13. Feature Engineering 15
  14. Average Time Taken vs. Weather Conditions 16
  15. Average Time Taken vs. Road Traffic Density 17
  16. Average Time Taken vs. Vechicle Condition 18
  17. Average Time Taken vs. Type of Vechicle 18
  18. Average Time Taken vs. Festival 19
  19. Average Time Taken vs. City 20

3.19 Correlation Matrix 21

#### Abbreviations

**Abbreviation Description**

DAV Data Analysis and Visualization

EDA Exploratory Data Analysis

# CHAPTER 1

## Introduction

### Definition of Problem

The problem addressed in this project revolves around the effectiveness of online delivery services by understanding and analyzing the time dynamics involved in the delivery process. It involves leveraging a comprehensive dataset encompassing diverse attributes pertinent to delivery operations, including but not limited to, delivery personnel information, prevailing weather conditions, and other order details.

It involves dissecting the dataset, thereby establishing a profound understanding of each attribute's essence and relevance within the context of delivery services. This foundational comprehension lays the groundwork for subsequent data cleaning endeavors, where various procedures are implemented to rectify missing values and uphold the sanctity of data integrity. Next, the project moves into exploratory data analysis (EDA), using various methods to examine the distribution of the "Time taken" variable and to discover important insights about how both numerical and categorical features relate to delivery time.

Moreover, the project delves into the realm of feature engineering, a stage where innovative features are crafted to encapsulate latent variables that potentially exert substantial influence on delivery time. Notable examples include temporal factors such as the geographical distance between the origin (restaurant) and destination (delivery location).

Finally, the project uses graphical analysis to explore the relationships between different factors and their impact on delivery time. By creating detailed visual aids, the project aims to uncover important insights about what influences delivery time in online delivery services, helping to make better decisions and improve operations in the industry.

### Objectives and Outcomes

* + - Objective 1 (Bloom's Taxonomy Level 3 - Application): Apply data analysis techniques to identify and address data quality issues in the online food delivery dataset.

Outcome 1: Demonstrate proficiency in using Pandas to load the dataset, identify missing data, outliers, zero values, and duplicates, and apply appropriate methods to handle these issues effectively.

* + - Objective 2 (Bloom's Taxonomy Level 4 - Analysis): Analyse temporal patterns and trends in delivery times across different variables to gain insights into factors influencing service efficiency and customer satisfaction.

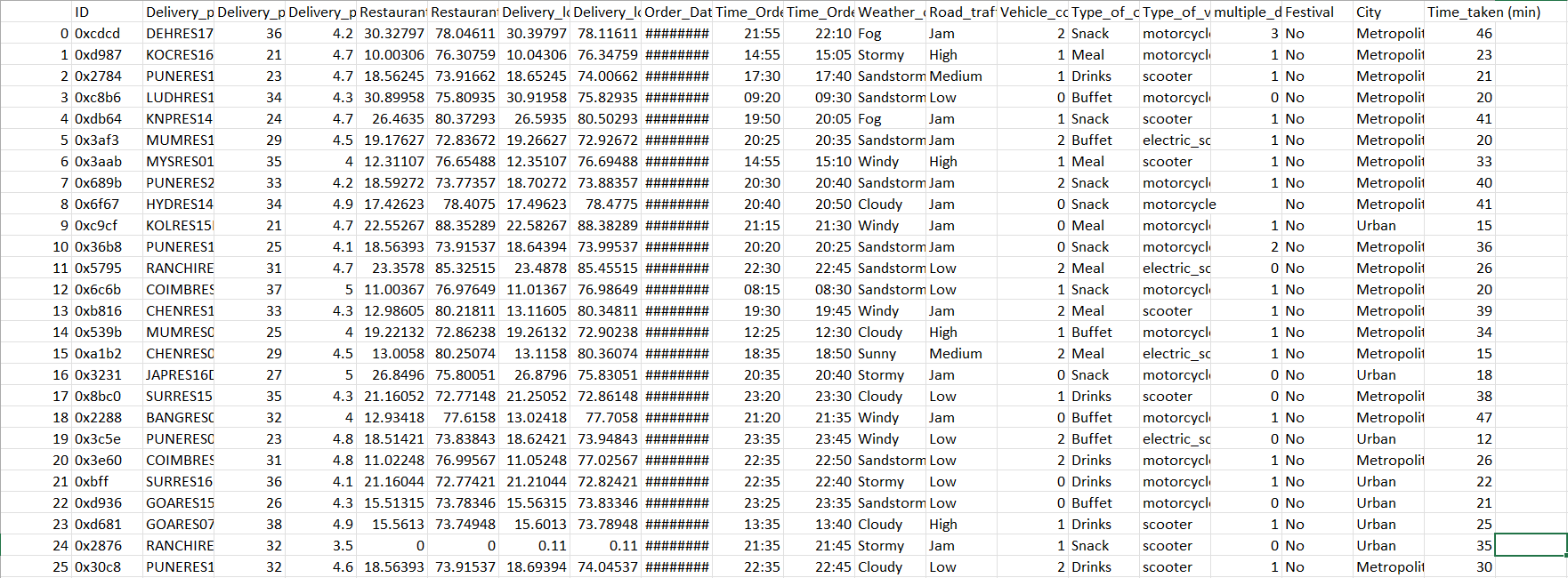
Outcome 2: Utilize statistical methods and visualization techniques (e.g., scatter plots, line plots) to explore relationships between delivery times and variables such as day of the week, time of day, geographical location, and food type, thereby identifying key factors impacting delivery performance.

# CHAPTER 2

## Methodology

### Data collection and Dataset description

The dataset we're using, sourced from GitHub, contains valuable information about online food delivery. It covers details such as delivery personnel, weather, and order specifics, forming the basis of our analysis on delivery times. Each delivery record is unique and provides insights into different parts of the delivery process. This includes details about delivery staff, restaurant and delivery locations, order times, and weather conditions. Additionally, it includes information about delivery vehicles, order types, transportation methods, multiple deliveries, and festivals impacting deliveries. It also specifies the city for each delivery. Importantly, it logs delivery times, reflecting how well the service is working and how satisfied customers are. With this dataset, we aim to thoroughly study how these factors interact, revealing what affects delivery times and service quality in online food delivery.



**Figure 2.1:** Dataset

### Data cleaning and preprocessing

We delve into the critical aspects of data cleaning and preprocessing, laying the groundwork for the subsequent analysis. Data cleaning is a crucial step in the data analysis process, ensuring that our dataset is accurate, reliable, and free from errors. In this stage, we meticulously sift through the data to identify and rectify any inconsistencies, inaccuracies, or missing values that could potentially skew our analysis.

The data cleaning process for our dataset was carried out in a structured manner to ensure the data's quality and reliability for subsequent analysis. The first step involved removing unnecessary columns. This began with an initial review of the dataset to understand the significance of each column. Any columns deemed irrelevant to the analysis or redundant, that did not affect delivery time, were identified and removed using the Pandas drop function. This streamlined the dataset, making it more focused and manageable.

Next, we addressed missing values, which is crucial to prevent biases or inaccuracies in our analysis. Using Pandas’ isnull functions, we identified columns with missing data. Depending on the extent and nature of these missing values, we employed different strategies. For instance, if a column had a few missing values, we filled them with the mean and mode of that column using Pandas’ fillna function. However, if a column had a significant portion of missing values and was not critical to the analysis, we considered dropping it entirely using the dropna method. This ensured that our dataset maintained its integrity and represented a complete set of information.

Handling duplicate values was the next step, aimed at ensuring that each record in the dataset was unique. We used Pandas’ duplicated function to identify any duplicate rows. Once identified, these duplicates were removed with the drop\_duplicates method, ensuring that each entry in the dataset was distinct and avoiding any skewed analysis results from repetitive data.

Subsequently, we addressed data types to ensure each column was in the appropriate format for accurate analysis. We used Pandas’ dtypes attribute to review the data types and converted columns as needed with the astype method. For example, columns containing delivery times stored as strings were converted to numerical types or datetime objects to facilitate proper time series analysis. This step was crucial for enabling accurate computations and comparisons within the dataset.

Outliers, which could distort the analysis, were then handled. We detected outliers using various statistical methods . Values falling outside the range ,depending on the context, we either removed these outliers or capped them to a specified limit using Pandas and NumPy functions, ensuring they did not unduly influence the analysis.

Finally, we addressed zero values in columns where zeros were not logically valid, such as the distance between the restaurant and delivery location. Using Pandas, we identified zero values and replaced them with a more appropriate measure, like the mean of the column, through the replace method. This step was vital to maintain the data's logical consistency and validity.

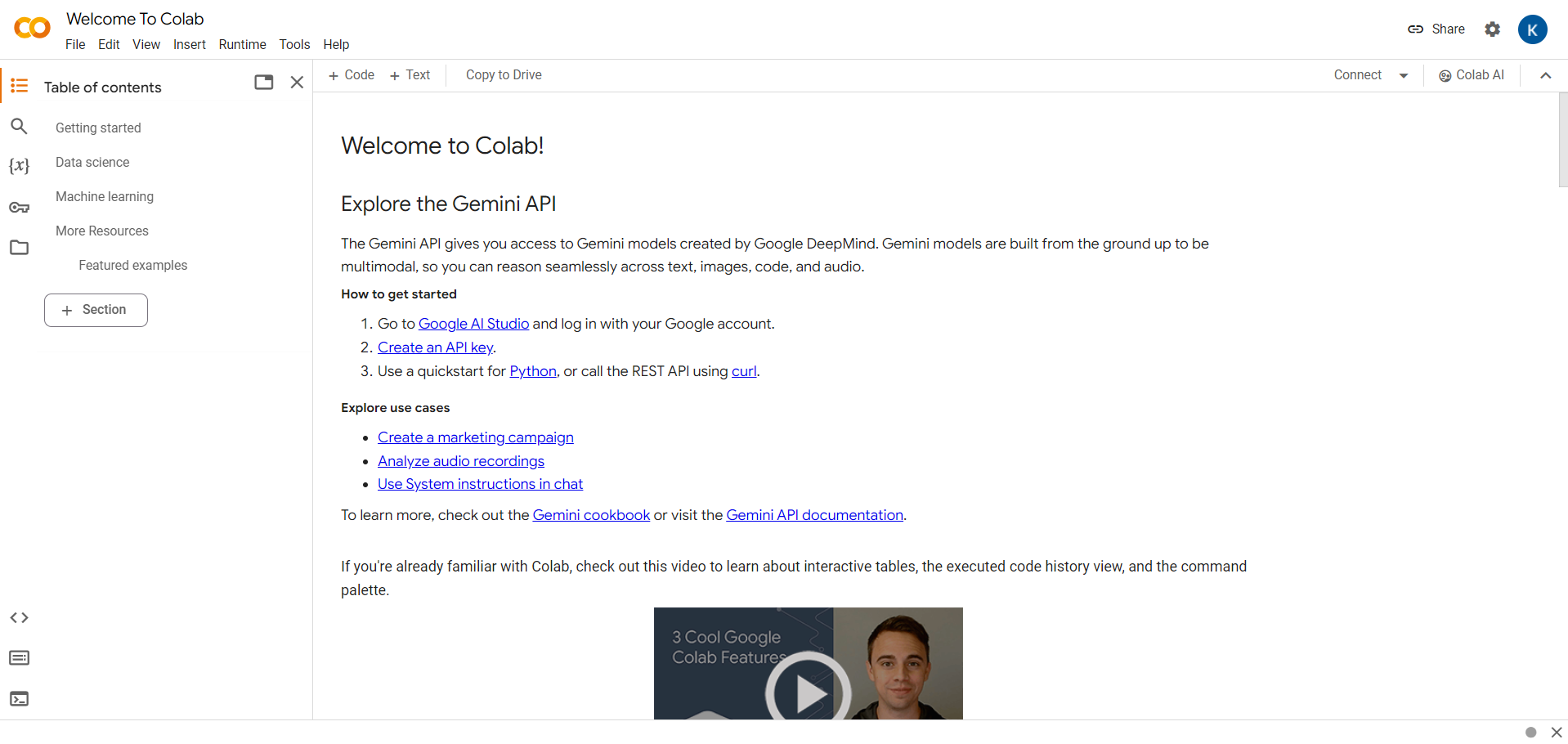
By following this detailed and systematic approach to data cleaning, we prepared a dataset that was robust, reliable, and ready for in-depth exploratory data analysis. This cleaned dataset laid a solid foundation for uncovering meaningful insights into the factors influencing delivery times in online delivery services.

# CHAPTER 3

## System Architecture and Implementation

### Google Colab

Google Colaboratory, commonly known as Google Colab, is a free online cloud-based Jupyter notebook environment tailored for training machine learn- ing and deep learning models. This article explores the functionalities, benefits, and features of Google Colab, elucidating its significance in the realm of data science and machine learning.



**Figure 3.1:** Google Colab

#### 3.1.1What is Google Colab?

Google Colab offers a cloud-based environment accessible via any web browser, eliminating the need for local software installation. Users can leverage its computing resources, including CPUs, GPUs, and TPUs, facilitating efficient model training and execution.

### Benefits of Google Colab

**Accessibility**: Users can access Google Colab from any location with internet connectivity, streamlining collaboration and workflow.

**Power**: The platform provides access to potent computing resources like

GPUs and TPUs, enabling swift and effective model training.

**Collaboration**: Google Colab simplifies collaborative efforts by allowing real-time editing and sharing of notebooks among team members.

**Education**: It serves as an invaluable educational tool for learning about

machine learning and data science, offering a plethora of tutorials and resources.

#### Why Choose Google Colab?

Google Colab stands out as an ideal choice for students, data scientists, researchers, and enthusiasts due to its:

**Ease of Use**: With no setup requirements, users can swiftly start coding

after creating an account.

**Affordability**: The platform is largely free to use, with paid plans available for more demanding tasks.

**Flexibility**: Users can seamlessly train models, process data, create visu-

alizations, and collaborate with others, making it a versatile tool for various applications.

#### Notebook in Google Colab

In Google Colab, a notebook serves as a web-based environment for code creation and execution. Notebooks offer several advantages, including real-time code execution and visualization, support for markdown for documentation,

and collaboration features, making them indispensable for data scientists and machine learning practitioners.

#### Google Colab Features

Google Colab boasts several features that enhance its usability and effec- tiveness:

**Free Access to GPUs and TPUs**: Users can leverage powerful computing

resources without any additional cost.

**Web-based Interface**: The intuitive and user-friendly interface eliminates the need for local software installation.

**Collaboration Tools**: Multiple users can collaborate on the same notebook

simultaneously, streamlining teamwork.

**Markdown Support**: Notebooks support markdown, enabling users to include formatted text, equations, and images alongside their code.

**Pre-installed Libraries**: Google Colab comes pre-installed with popular

libraries and tools for machine learning and deep learning, such as TensorFlow and PyTorch, saving time on setup and configuration.

Google Colab emerges as a versatile and indispensable tool for machine

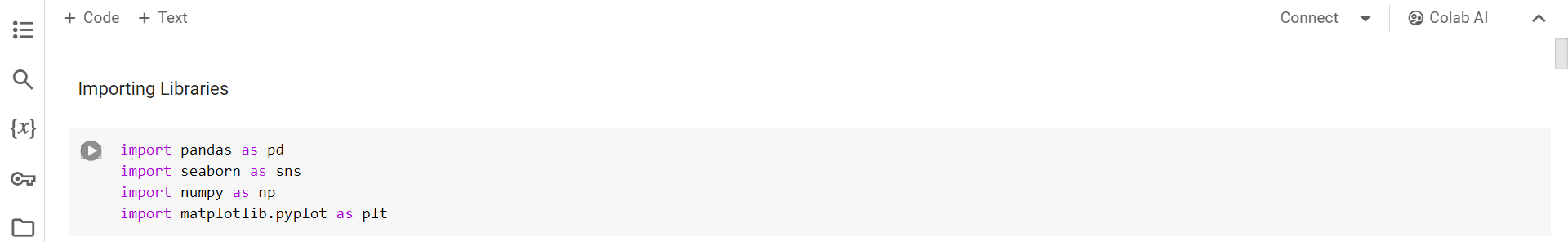
learning and data science tasks, offering accessibility, power, and flexibility. Its user-friendly interface, collaborative features, and integration with powerful computing resources make it an invaluable asset for individuals and teams alike, driving innovation and progress in the field of machine learning and beyond.

### Code Snippets

#### Importing libraries and Data loading

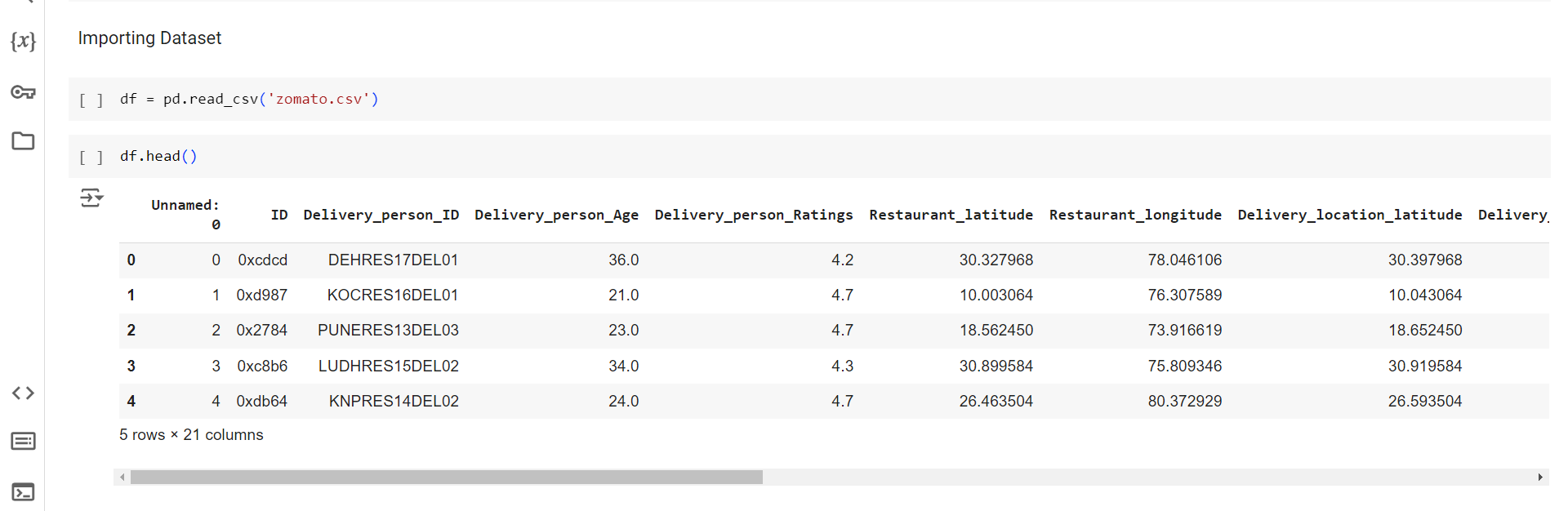
To begin our project, we first import the necessary libraries for data analysis and visualization. We import pandas as pd and numpy as np for data handling and numerical operations, respectively. We use Matplotlib and Seaborn for data visualization.

Next, we load the dataset into our code using the read csv function from pandas, assuming the dataset is stored in a CSV file named 'zomato.csv’. We assign the loaded dataset to a variable named 'df'.

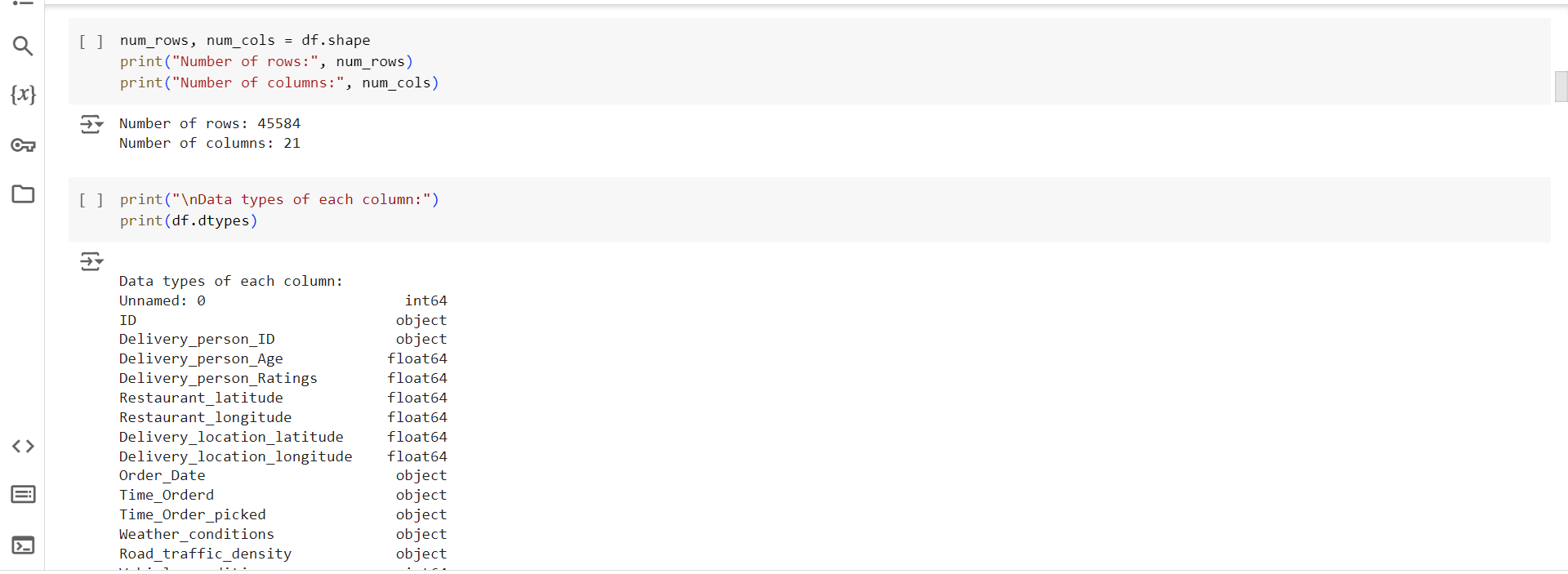


**Figure 3.2:** importing libraries and Dataset loading

To ensure that the dataset has been loaded successfully, we display the first few rows of the dataset using the head() function. This allows us to inspect the structure and content of the dataset, confirming that it has been imported correctly and is ready for further processing.

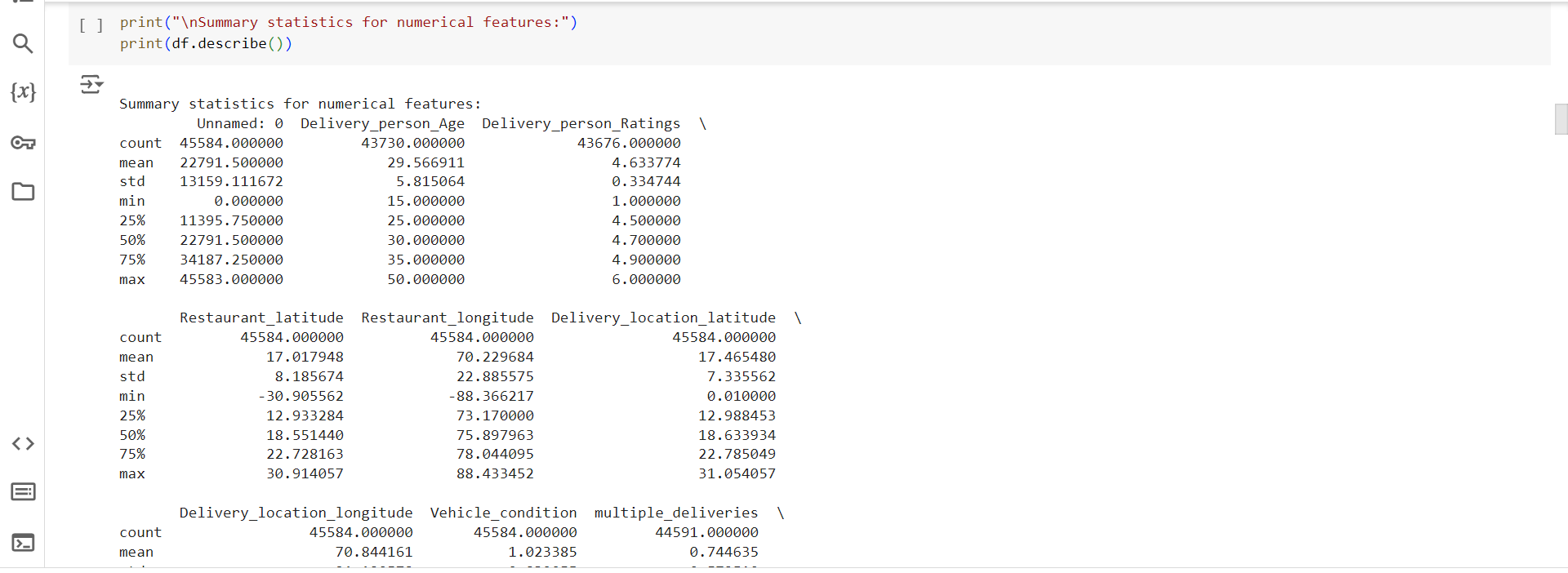


**Figure 3.3:** Staring cells of our Data set



**Figure 3.4:**  Identifying number of rows and column

This code snippet prints the data types of each column in a DataFrame (df) and then calculates and prints the number of rows and columns in the DataFrame.



**Figure 3.5:**  Statistics for Numerical features

This code snippet prints out summary statistics for numerical features in a DataFrame (df). It provides key statistical measures such as count, mean, standard deviation, minimum, maximum, and quartile values for each numerical column in the DataFrame.

#### Data cleaning and preprocessing

The data cleaning process for our dataset was carried out in a structured manner to ensure quality and reliability for subsequent analysis.

**1.** **Removing Unnecessary Columns:**

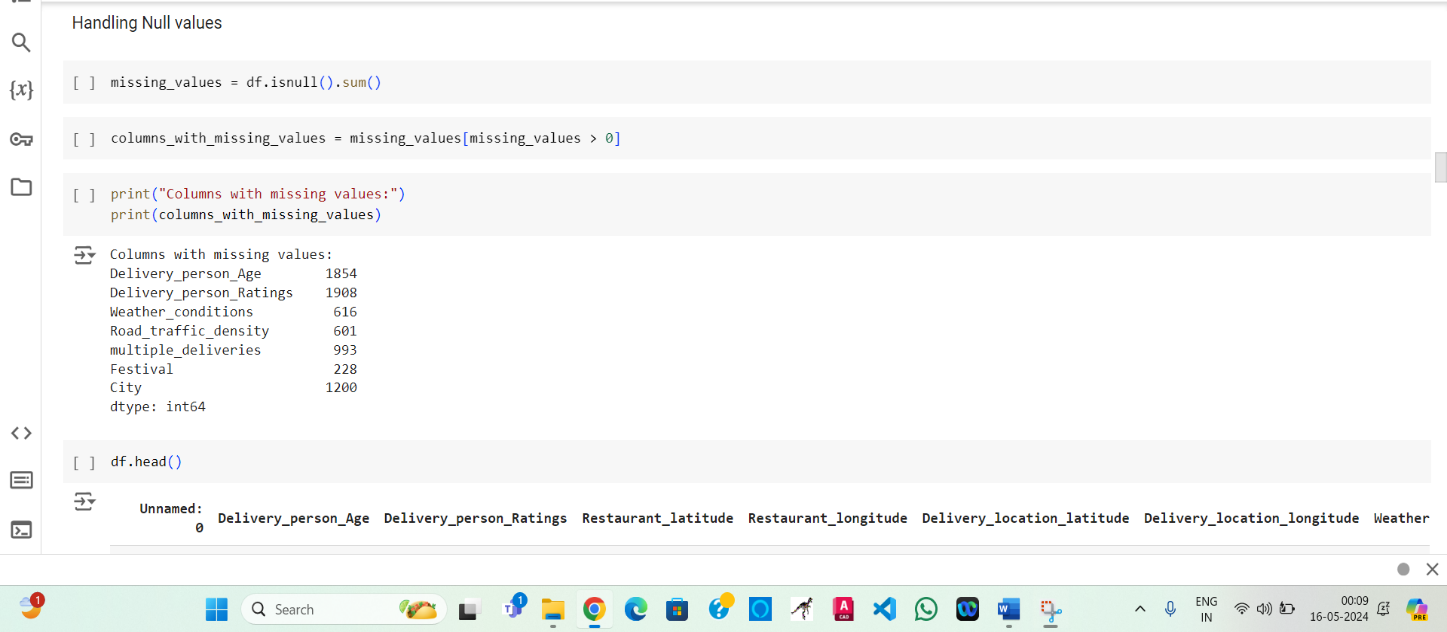
We began by reviewing the dataset to understand each column's significance. Columns deemed irrelevant or redundant were removed using the Pandas drop function, streamlining the dataset.

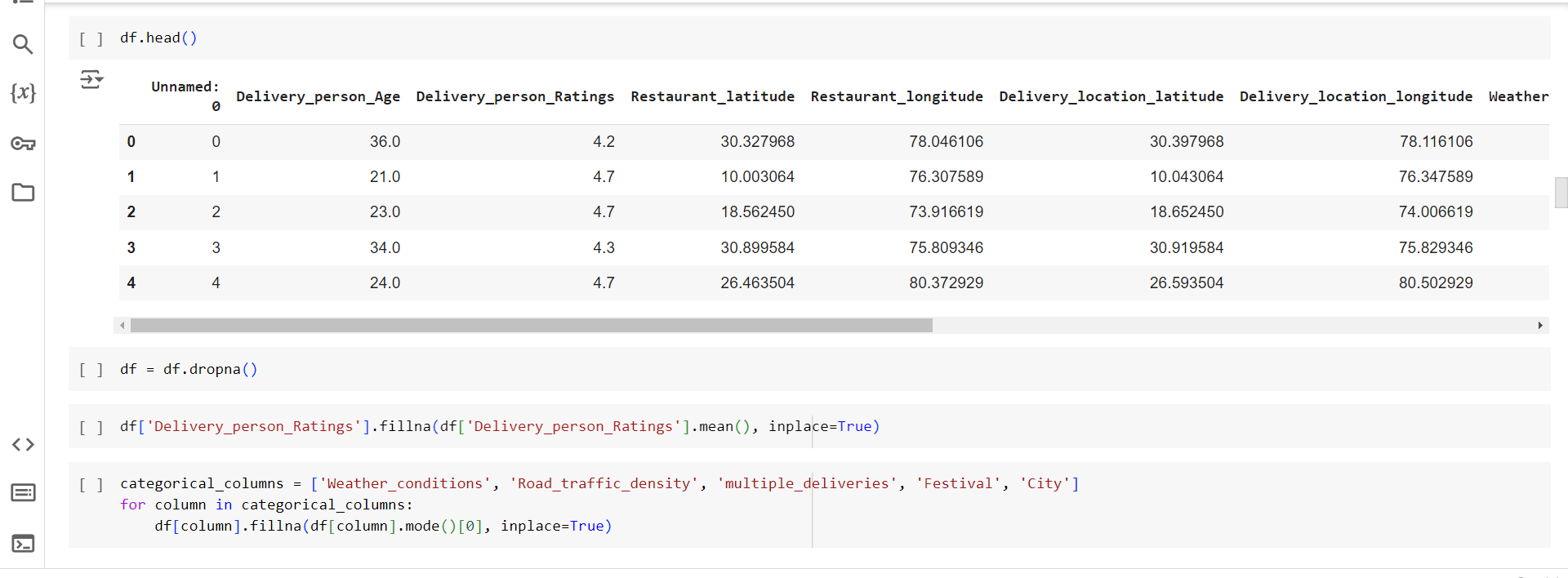


**Figure 3.6:**  Removing Unnecessary Columns

**2. Addressing Missing Values:**

We identified columns with missing data using Pandas’ isnull functions. For columns with a few missing values, we used the fillna function to fill them with the mean or mode. Columns with significant missing data and not critical to the analysis were dropped using the dropna method, maintaining data integrity.

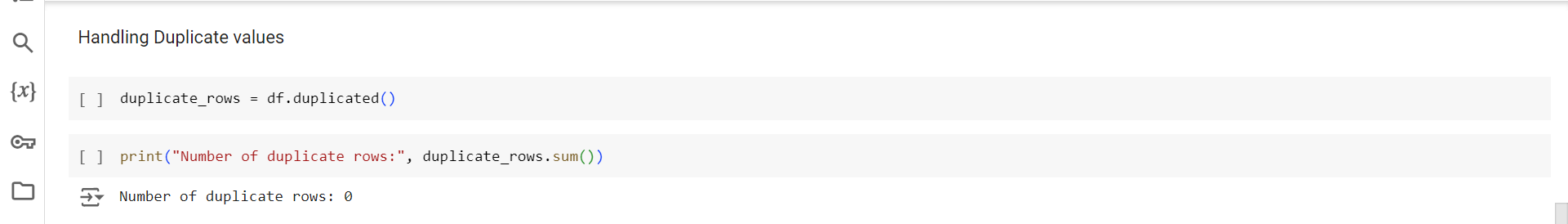




**Figure 3.7:**  Addressing Missing Values

**3. Handling Duplicate Values:**

We used Pandas’ duplicated function to identify duplicate rows. These duplicates were removed with the drop\_duplicates method, ensuring unique entries and avoiding skewed analysis results.



**Figure 3.8:**  Handling Duplicate Values

**4. Addressing Data Types:**

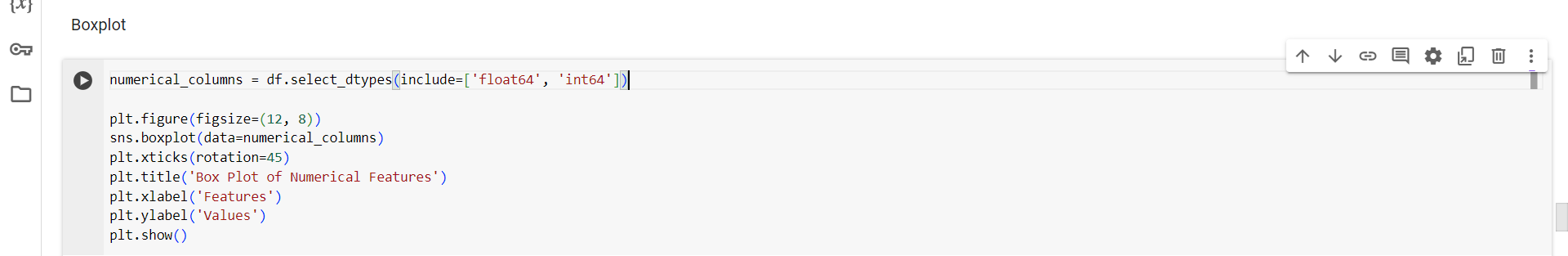
We reviewed the data types using Pandas’ dtypes attribute and converted columns as needed with the astype method. For instance, delivery times stored as strings were converted to numerical types or datetime objects, enabling accurate computations and comparisons.

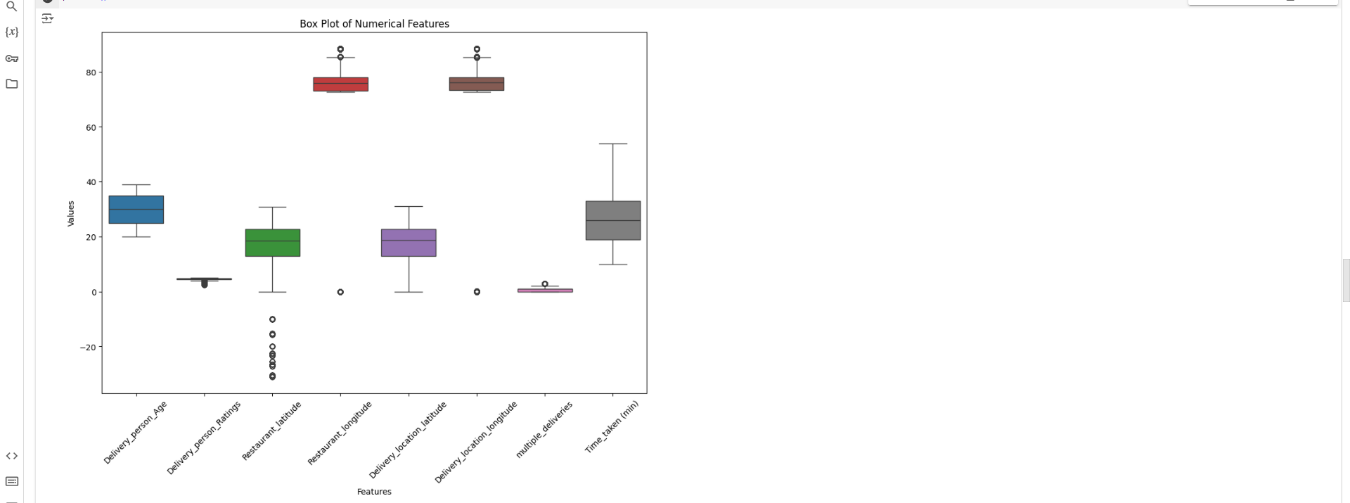


**Figure 3.9:**  Addressing Data Types

**5. Handling Outliers:**

Outliers were detected using statistical methods. Depending on the context, outliers were either removed or capped to a specified limit using Pandas and NumPy functions, preventing undue influence on the analysis.

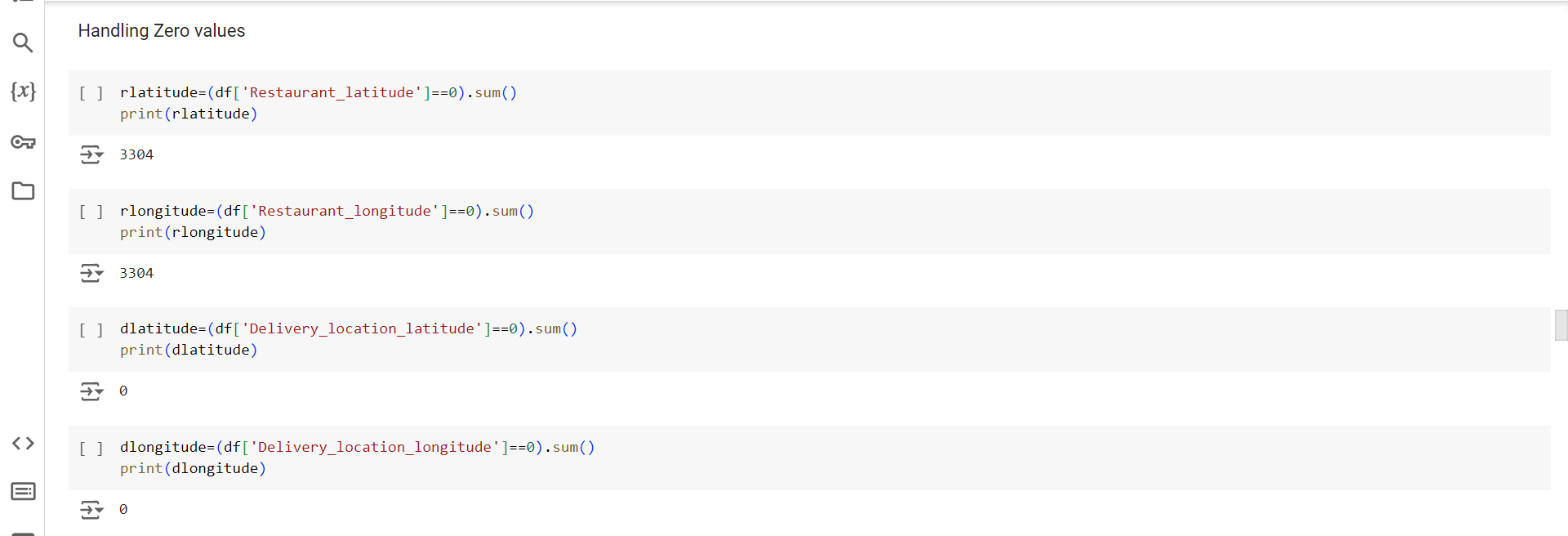




**Figure 3.10:**  Handling Outliers

**6. Handling Zero Values:**

We identified zero values in columns where zeros were not logically valid, such as the distance between the restaurant and delivery location. These zero values were replaced with the column mean using the replace method, ensuring logical consistency and validity.



**Figure 3.11**: Handling Zero Values

#### Feature Engineering

#### In our project, we engaged in feature engineering to enhance the dataset with a new variable: the distance between the order delivery point and the restaurant location. This distance was calculated by leveraging the latitude and longitude coordinates of both locations. We implemented the distcalculate function, which utilizes the Haversine formula—an algorithm for computing distances between two points on a sphere. Initially, the function computes the differences in latitude and longitude, converting these values from degrees to radians. Subsequently, it applies the Haversine formula to determine the distance based on spherical trigonometry principles. This involves computing the central angle between the points and subsequently obtaining the distance by multiplying the Earth's radius by the central angle. Incorporating this function into our data processing pipeline allowed us to generate a crucial feature, 'distance', which directly impacts delivery time. This engineered feature enriches our dataset, offering valuable insights into delivery performance and aiding in predictive modeling and decision-making processes within the online delivery services domain.

#### 

#### Figure 3.12: Feature Engineering

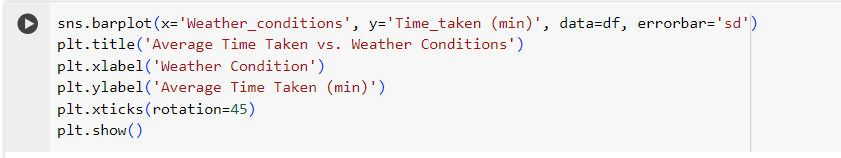
#### Exploratory Data Analysis

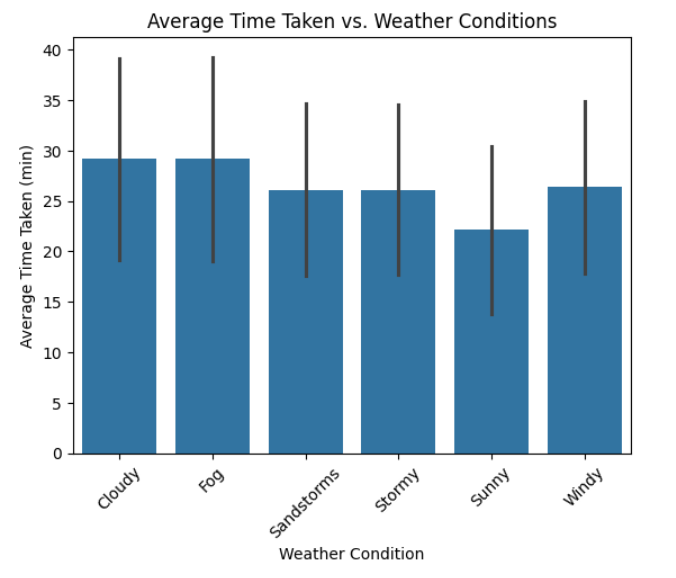
Exploratory Data Analysis (EDA) is a crucial preliminary step in data analysis, focusing on understanding the dataset's structure, identifying patterns, and uncovering relationships between variables. It involves visualizing data, summarizing key features, and detecting potential anomalies. EDA serves as a foundation for further analysis and model building.

1. **Bar Plot Analysis:**

Using bar plots to visualize the relationship between categorical features and the Time\_taken (min), allows for a clear comparison of the average time taken across different categories of each feature. This approach provides insights into how each categorical attribute may influence the delivery time. Let's delve into each feature:

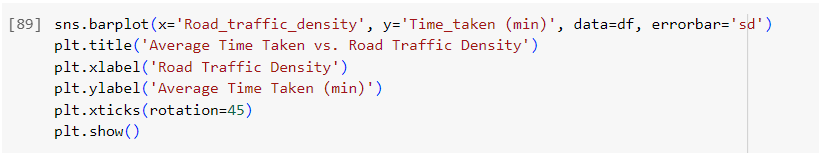
1. **Weather\_conditions**: By plotting a bar plot of 'Weather\_conditions' against 'Time\_taken (min),' we can observe how different weather conditions impact delivery times. For example, we might find that delivery times are longer during rainy or adverse weather conditions compared to sunny days.

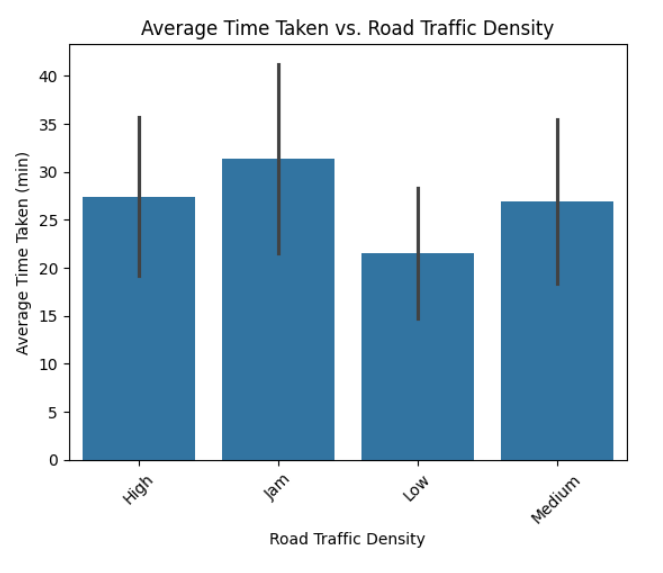




**Figure 3.13**: Average Time Taken vs. Weather Conditions

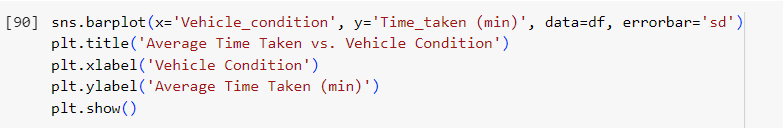
1. **Road\_traffic\_density**: A bar plot of 'Road\_traffic\_density' versus 'Time\_taken (min)' can reveal how traffic density affects delivery times. Higher traffic density might lead to longer delivery times due to congestion or delays, while lower traffic density could result in quicker deliveries.

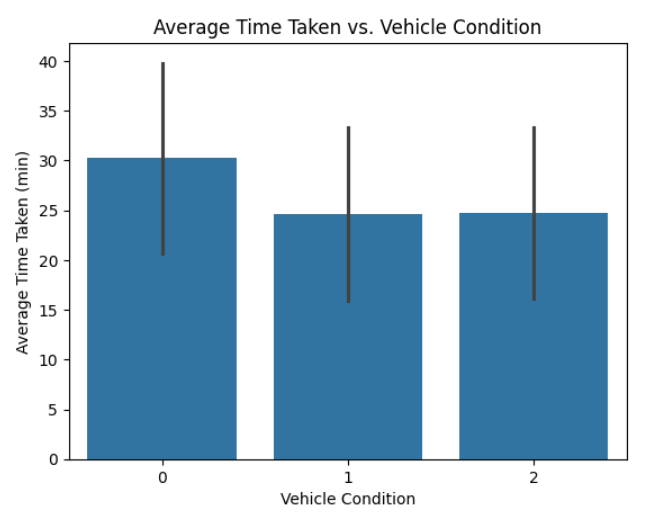




**Figure 3.14**: Average Time Taken vs. Road Traffic Density

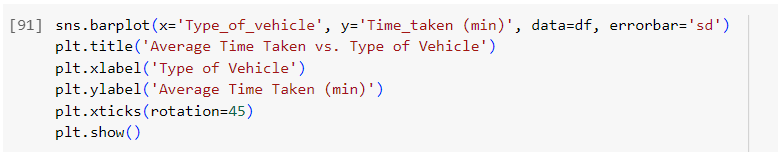
1. **Vehicle\_condition**: Analyzing the relationship between 'Vehicle\_condition' and 'Time\_taken (min)' through a bar plot helps understand if the condition of the delivery vehicle influences delivery times. For instance, well-maintained vehicles might lead to faster deliveries compared to those in poor condition.

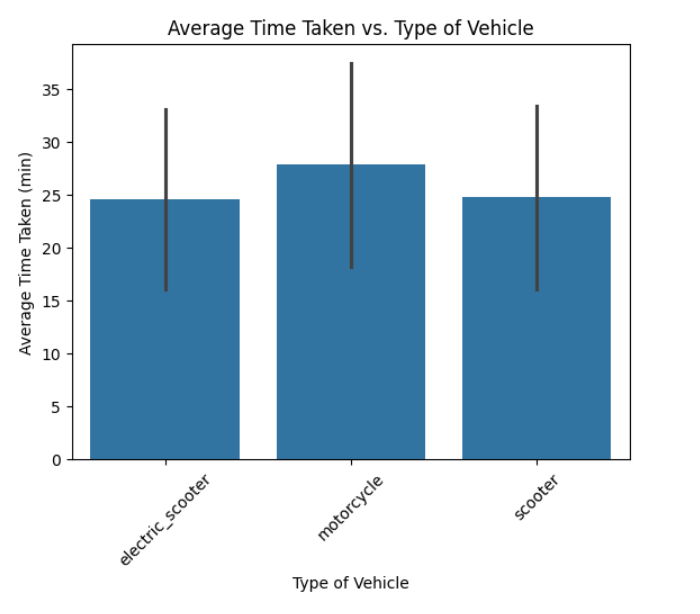




**Figure 3.15**: Average Time Taken vs. Vehicle Condition

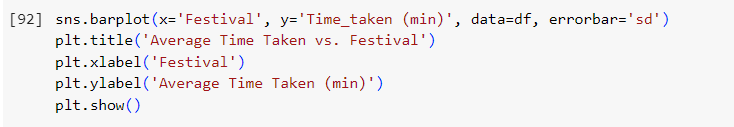
1. **Type\_of\_vehicle**: A bar plot of 'Type\_of\_vehicle' against 'Time\_taken (min)' provides insights into how different types of vehicles impact delivery times. For example, deliveries made using bikes might be faster than those made using cars due to their ability to navigate through traffic more efficiently.

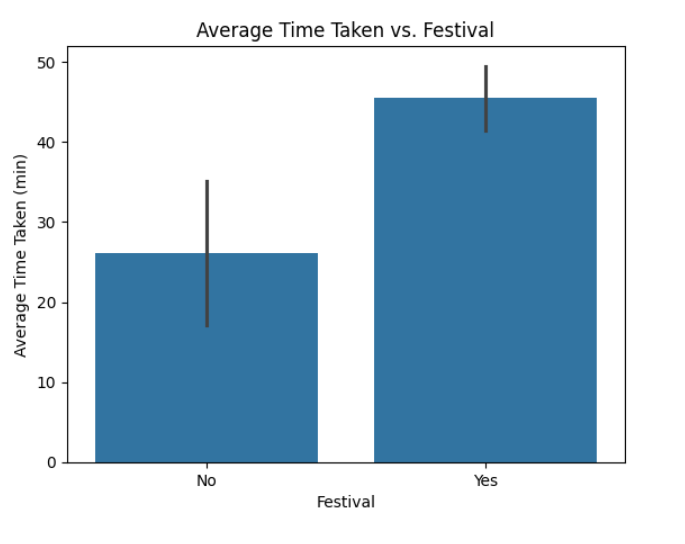




**Figure 3.16**: Average Time Taken vs. Type of Vehicle

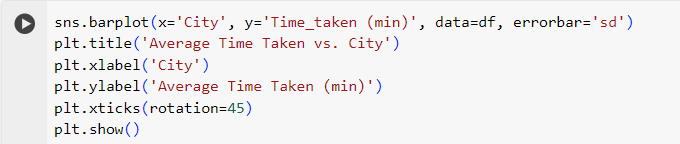
1. **Festival**: Plotting 'Festival' versus 'Time\_taken (min)' using a bar plot helps assess whether delivery times are affected during festive seasons or holidays. Longer delivery times during festivals may be attributed to increased orders or traffic congestion.

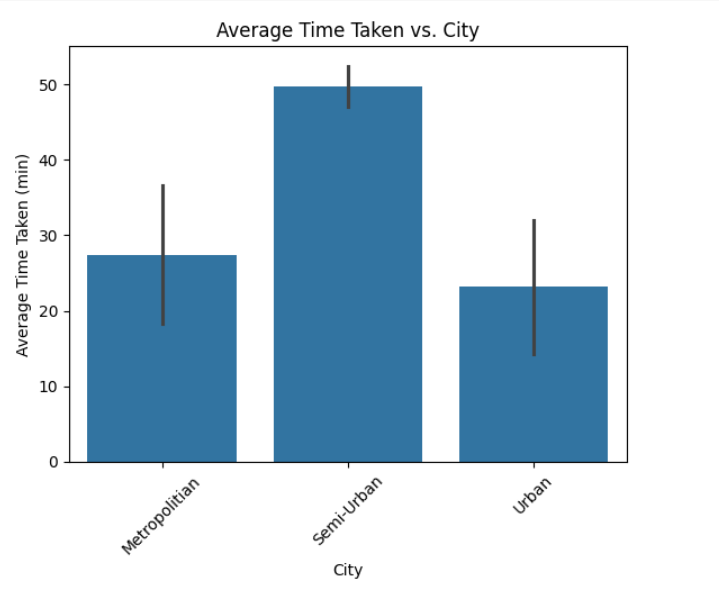




**Figure 3.17**: Average Time Taken vs. Festival

1. **City**: A bar plot of 'City' against 'Time\_taken (min)' reveals any differences in delivery times across different cities. Variations in infrastructure, traffic conditions, or distance between delivery locations within each city can impact delivery times.



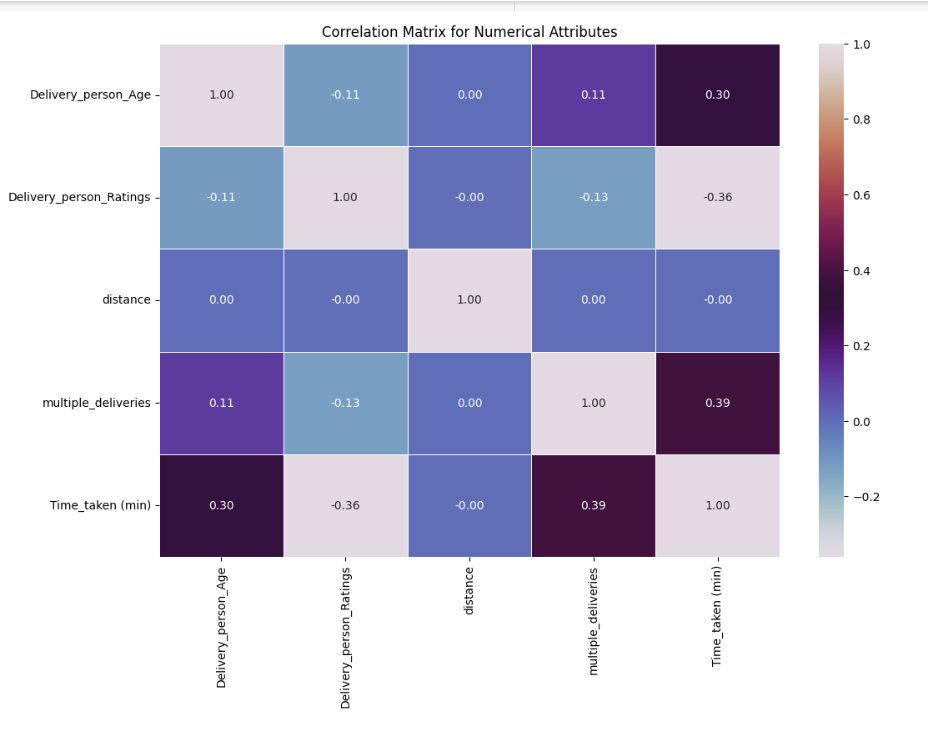


**Figure 3.18**: Average Time Taken vs. City

1. **Correlation Matrix Analysis:**

A correlation matrix serves as a tool to quantify the linear relationship between pairs of numerical variables. In the provided code, this matrix is computed specifically for a subset of numerical attributes, comprising 'Delivery\_person\_Age', 'Delivery\_person\_Ratings', 'distance', 'multiple\_deliveries', and 'Time\_taken (min)'. Visualizing this correlation matrix using a heatmap, each cell contains an annotation displaying the correlation coefficient between respective attribute pairs. High positive or negative correlation coefficients within the matrix indicate strong linear relationships between attributes, while coefficients close to zero suggest weak or negligible correlation. This analytical tool is instrumental in identifying potential multicollinearity issues, where attributes are highly correlated, and aids in making informed feature selection decisions for subsequent modeling tasks. By leveraging the correlation matrix, analysts gain valuable insights into the interplay between numerical variables, guiding the refinement of predictive models and enhancing their interpretability.





**Figure 3.19**: Correlation Matrix

#### 3.3.5Conclusion

Based on the correlation matrix analysis, several insights emerge regarding the relationships between various attributes and delivery time:

1. **Delivery Person Age**: There exists a moderate positive correlation (+0.30) between the age of the delivery person and the time taken for delivery. This suggests that older delivery personnel may take slightly longer to complete deliveries compared to younger counterparts. Possible reasons could include differences in energy levels, efficiency, or familiarity with delivery routes.
2. **Delivery Person Ratings**: A moderate negative correlation (-0.36) is observed between delivery person ratings and delivery time. This implies that delivery personnel with higher ratings tend to complete deliveries more quickly than those with lower ratings. Higher ratings may indicate greater proficiency, reliability, and effectiveness in handling delivery tasks.
3. **Multiple Deliveries**: There is a moderate positive correlation (+0.39) between the number of multiple deliveries and delivery time. This suggests that when a delivery person handles multiple deliveries in a single trip, the overall delivery time tends to increase. This could be due to the added complexity and time required to navigate between multiple delivery locations efficiently.

Based on the analysis of the bar plots:

1. **Urban vs. Semi-Urban vs. Metropolitan Areas**: The comparison across urbanization levels reveals distinct patterns in delivery times. Urban areas exhibit the most variable delivery times, approximately 20 minutes, due to better infrastructure and closer delivery points. Metropolitan areas experience slightly longer delivery times, around 30 minutes, with the most variability likely due to traffic congestion and larger delivery areas. Conversely, Semi-Urban areas demonstrate the most consistent delivery times, averaging around 50 minutes, owing to less developed infrastructure and greater distances between delivery points. Overall, the findings suggest a progression of delivery efficiency from Urban to Semi-Urban areas, influenced by varying levels of infrastructure and distance considerations.
2. **Festival vs. Non-Festival Periods**: The analysis of delivery times during festival and non-festival periods underscores significant fluctuations in service efficiency. During festivals, the average delivery time is approximately 45 minutes, with moderate variability indicating some consistency despite increased demand and traffic. Conversely, non-festival periods witness a substantial increase in average delivery time to around 25 minutes, with larger variability suggesting slower and less predictable deliveries. These observations point to the impact of seasonal variations on logistical operations, with festivals posing challenges to timely delivery services.
3. **Road Traffic Density**: The examination of road traffic density reveals a direct correlation with delivery efficiency. As traffic density increases, the average delivery time also rises. Low traffic density areas witness the shortest average delivery time, indicating quicker deliveries. In contrast, higher traffic density areas, ranging from medium to very high, exhibit progressively longer and more variable delivery times, reflective of the impediments posed by congestion and traffic delays. This finding underscores the pivotal role of traffic conditions in shaping delivery timelines and highlights the importance of route optimization strategies in mitigating delivery delays.

These findings provide valuable insights for optimizing online delivery services:

1. **Optimizing Delivery Routes**: Understanding the impact of delivery person age and ratings on delivery time can help in optimizing delivery routes and assigning deliveries to personnel based on their characteristics. For instance, younger delivery personnel may be assigned to routes with shorter distances or higher demand areas, while those with higher ratings can handle time-sensitive orders.
2. **Efficiency Improvement**: Strategies aimed at improving delivery efficiency, such as providing training or incentives to delivery personnel with lower ratings, or optimizing the allocation of multiple deliveries to minimize overall delivery time, can be implemented to enhance service quality and customer satisfaction.
3. **Data-Driven Decision Making**: These insights underscore the importance of data-driven decision-making in online delivery services. By continuously analyzing and leveraging data on delivery operations, companies can identify areas for improvement, refine their strategies, and stay competitive in the rapidly evolving market landscape.

#### References

#### [1] On-Time Last-Mile Delivery: Order Assignment with Travel-Time Predictors

#### Sheng Liu, Long He, Z. Shen ,2020, Management Sciences

#### [2] Data-driven optimization for last-mile delivery

#### Hongrui Chu, Wensi Zhang, P. Bai, Yahong Chen,2021, Complex & Intelligent Systems

#### [3] Factor Analysis on Online Food Delivery Service in the New Normal

#### Alexis M Bagot,Linette H Maalihan,Ma Nicole Pattalitan,Saira M Moro,Ginell Joseph De Roxas,Jen Ysabelle Bagui,Noelah Mae Dimayuga Borbon,2022,Asian pacific journel of Academic Research in Buisness Administration

#### 

#### [4] A review on delivery routing problem and its approaches

#### S.N. Ismail, Ku Ruhana Ku-Mahamud ,Syariza Abdul Rahman,2017, Journal of Theoretical and Applied Information Technology

#### [5] Optimisation of takeaway delivery routes considering the mutual satisfactions of merchants and customers, Ren Teng , Xu Hong-bo , Jin Kang-ning , Luo Tian-yu , Wang Ling , Xing Li-ning ,2021, Computers & Industrial Engineering