CAPSTONE PROJECT **REPORT**

Analysis and Prediction of Time Taken for Food Delivery

PGPDSE – FT Bangalore April 2023

Mentored by:

Mr. SAI SOURAB REDDY P

Submitted by:

Mr. MAHESH PATIL

Contents:

1.Abstract

In the fast-paced landscape of the modern food industry, efficient and reliable delivery services play a pivotal role in customer satisfaction. This study delves into the analysis and prediction of the time taken for food delivery, aiming to enhance the overall service quality and meet customer expectations.

The research leverages a comprehensive dataset comprising historical delivery records, encompassing variables such as distance, time of day, day of the week, and external factors like weather conditions and traffic patterns. Through rigorous data analysis techniques, including statistical modeling and machine learning algorithms, we uncover patterns and correlations that impact delivery times.

The predictive models developed in this study aim to provide accurate estimates of delivery times, empowering both customers and service providers with valuable insights. By understanding the factors influencing delivery speed, businesses can optimize their operations, allocate resources more effectively, and improve overall service efficiency.

The findings of this research contribute to the growing field of food delivery logistics, shedding light on the intricacies of the delivery process. Moreover, the predictive models developed offer practical applications for businesses seeking to streamline their operations and enhance customer satisfaction in the competitive food delivery market. As the demand for online food delivery continues to rise, this study provides a valuable framework for understanding and improving the temporal aspects of this essential service.

1.1 Industry Review - Current practices, Background Research

Recent changes in consumer preferences and technological advancements have caused a considerable development and shift in the meal delivery industry.

Consumers increasingly anticipate receiving their favorite restaurant meals faster, more reliably, more conveniently. Food delivery services are continuously inventing and streamlining their processes to match these demands.

Accurately estimating how long it will take a delivery worker to arrive at a customer's location is a crucial component of food delivery services. In order to guarantee client satisfaction and effective resource management, this Estimated Time of Arrival (ETA) projection is essential. Conventional approaches to ETA calculation might not always take into account variables that change in real time, like traffic, weather, and the performance of the delivery person.

1. **<u>Key Objectives:</u>**

Improving Data-Driven Decision Making, Customer Satisfaction, Operational Efficiency, and ETA Accuracy.

The business problem formulation:

i. Understanding Business Problems:

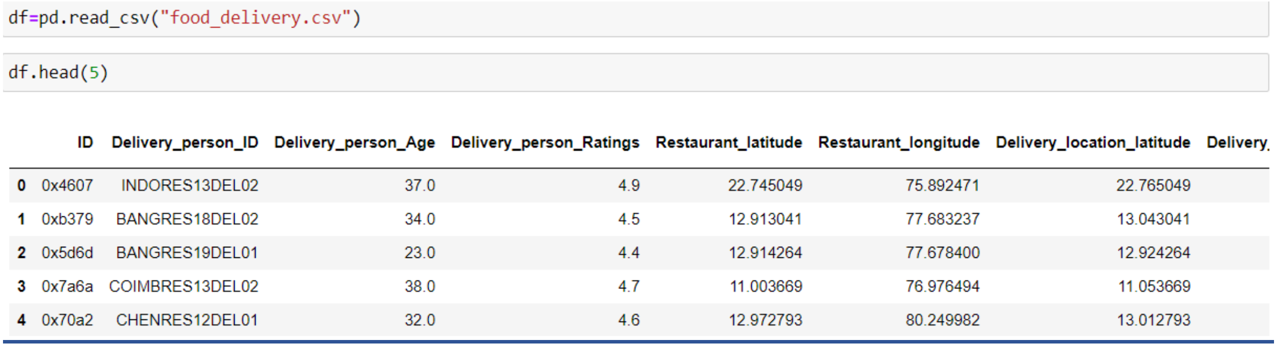
It's difficult for a food delivery service to estimate how long a delivery person will take to finish a delivery. The company wants to solve this problem by using a machine learning model that can determine the estimated time of arrival (ETA) for delivery staff while accounting for a variety of factors that affect delivery timings.

ii. Business Goal :

In general, there is interest in applying machine learning and datamining approaches in a variety of industries due to recent advancements in these fields. In this sense, the e-commerce industry is not any different. The application of machine learning techniques may result in increased corporate profitability overall.

2.Data set and Domain

The Dataset which we choose for our Capstone Project is “Analysis and Prediction of Time Taken for Food Delivery” belongs to the domain supply chain department.

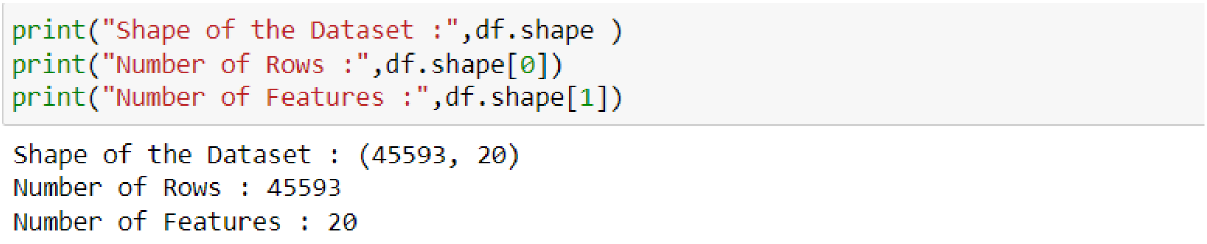


```
df=pd.read_csv("food_delivery.csv")
```

```
df.head(5)
```

	ID	Delivery_person_ID	Delivery_person_Age	Delivery_person_Ratings	Restaurant_latitude	Restaurant_longitude	Delivery_location_latitude	Delivery
0	0x4607	INDORES13DEL02	37.0	4.9	22.745049	75.892471	22.765049	
1	0xb379	BANGRES18DEL02	34.0	4.5	12.913041	77.683237	13.043041	
2	0x5d6d	BANGRES19DEL01	23.0	4.4	12.914264	77.678400	12.924264	
3	0x7a6a	COIMBRES13DEL02	38.0	4.7	11.003669	76.976494	11.053669	
4	0x70a2	CHENRES12DEL01	32.0	4.6	12.972793	80.249982	13.012793	

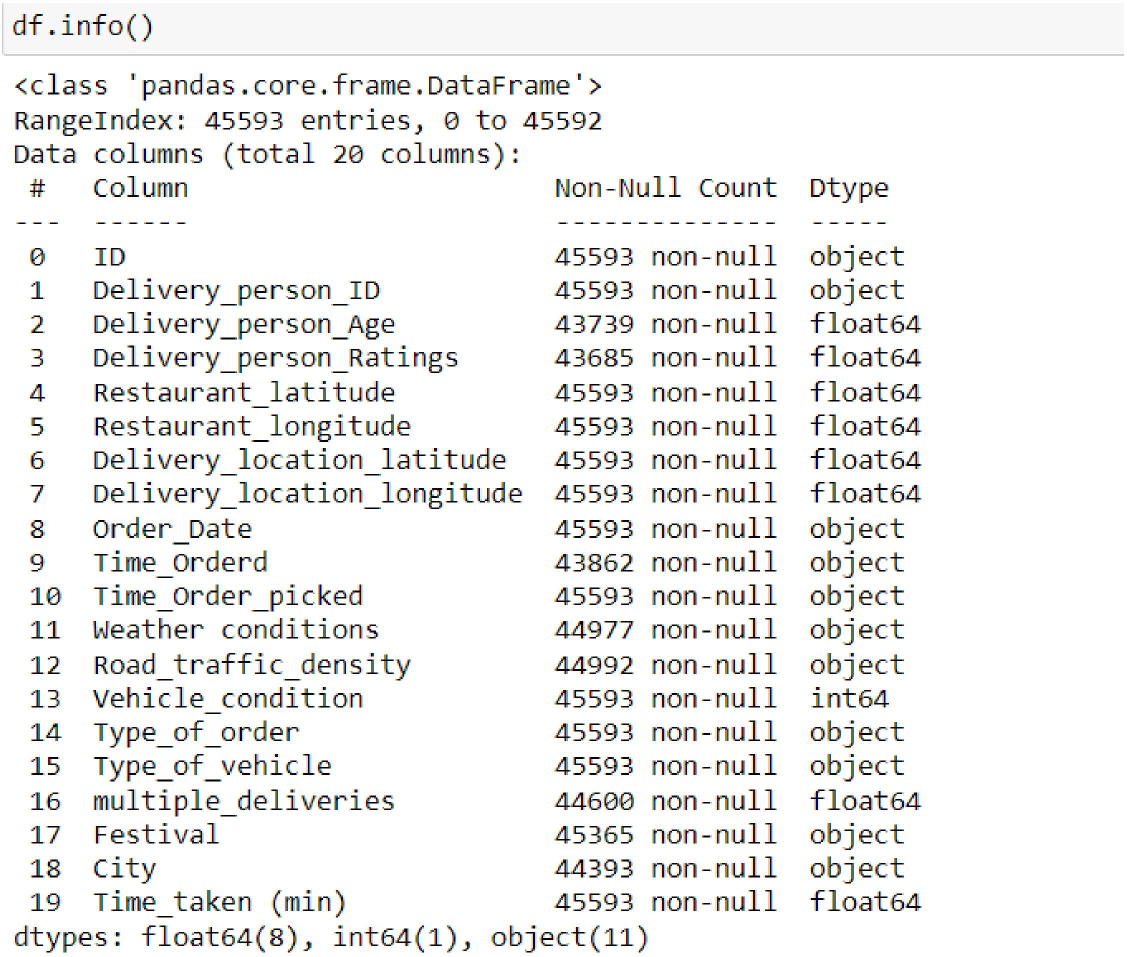
a. Basic Statistics of Dataset



```
print("Shape of the Dataset :",df.shape )
print("Number of Rows :",df.shape[0])
print("Number of Features :",df.shape[1])
```

```
Shape of the Dataset : (45593, 20)
Number of Rows : 45593
Number of Features : 20
```

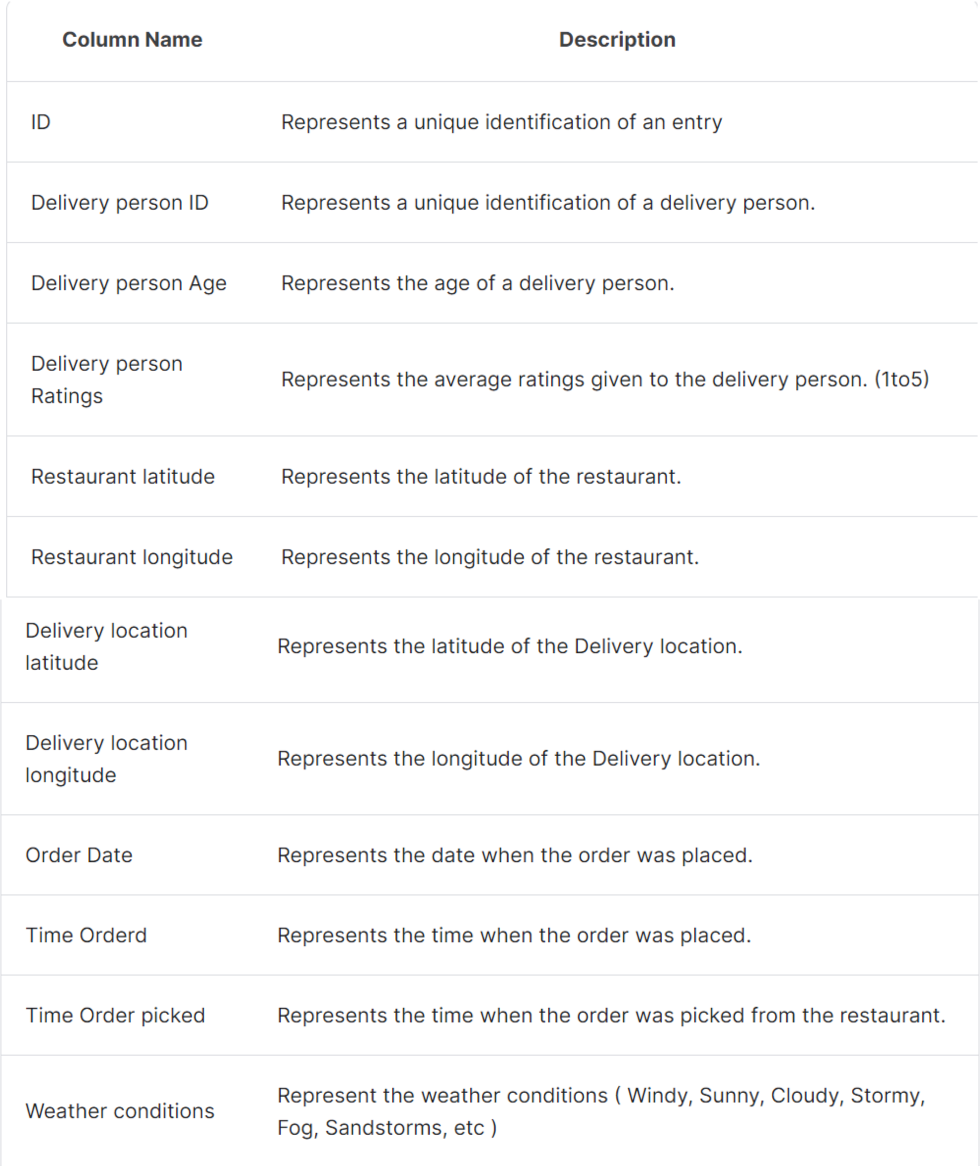
b.Data Dictionary



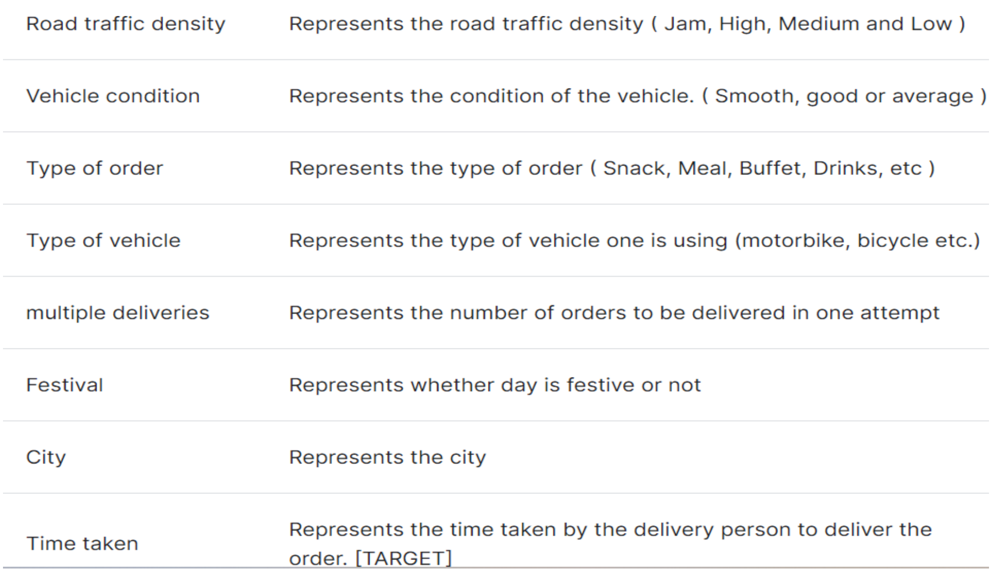
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45593 entries, 0 to 45592
Data columns (total 20 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   ID                           45593 non-null  object 
 1   Delivery_person_ID           45593 non-null  object 
 2   Delivery_person_Age          43739 non-null  float64
 3   Delivery_person_Ratings      43685 non-null  float64
 4   Restaurant_latitude          45593 non-null  float64
 5   Restaurant_longitude         45593 non-null  float64
 6   Delivery_location_latitude   45593 non-null  float64
 7   Delivery_location_longitude  45593 non-null  float64
 8   Order_Date                   45593 non-null  object 
 9   Time_Orderd                  43862 non-null  object 
 10  Time_Order_picked            45593 non-null  object 
 11  Weather conditions           44977 non-null  object 
 12  Road_traffic_density         44992 non-null  object 
 13  Vehicle_condition            45593 non-null  int64  
 14  Type_of_order                45593 non-null  object 
 15  Type_of_vehicle              45593 non-null  object 
 16  multiple_deliveries          44600 non-null  float64
 17  Festival                     45365 non-null  object 
 18  City                         44393 non-null  object 
 19  Time_taken (min)             45593 non-null  float64
dtypes: float64(8), int64(1), object(11)
```

2.3.Feature description



Column Name	Description
ID	Represents a unique identification of an entry
Delivery person ID	Represents a unique identification of a delivery person.
Delivery person Age	Represents the age of a delivery person.
Delivery person Ratings	Represents the average ratings given to the delivery person. (1to5)
Restaurant latitude	Represents the latitude of the restaurant.
Restaurant longitude	Represents the longitude of the restaurant.
Delivery location latitude	Represents the latitude of the Delivery location.
Delivery location longitude	Represents the longitude of the Delivery location.
Order Date	Represents the date when the order was placed.
Time Orderd	Represents the time when the order was placed.
Time Order picked	Represents the time when the order was picked from the restaurant.
Weather conditions	Represent the weather conditions (Windy, Sunny, Cloudy, Stormy, Fog, Sandstorms, etc)



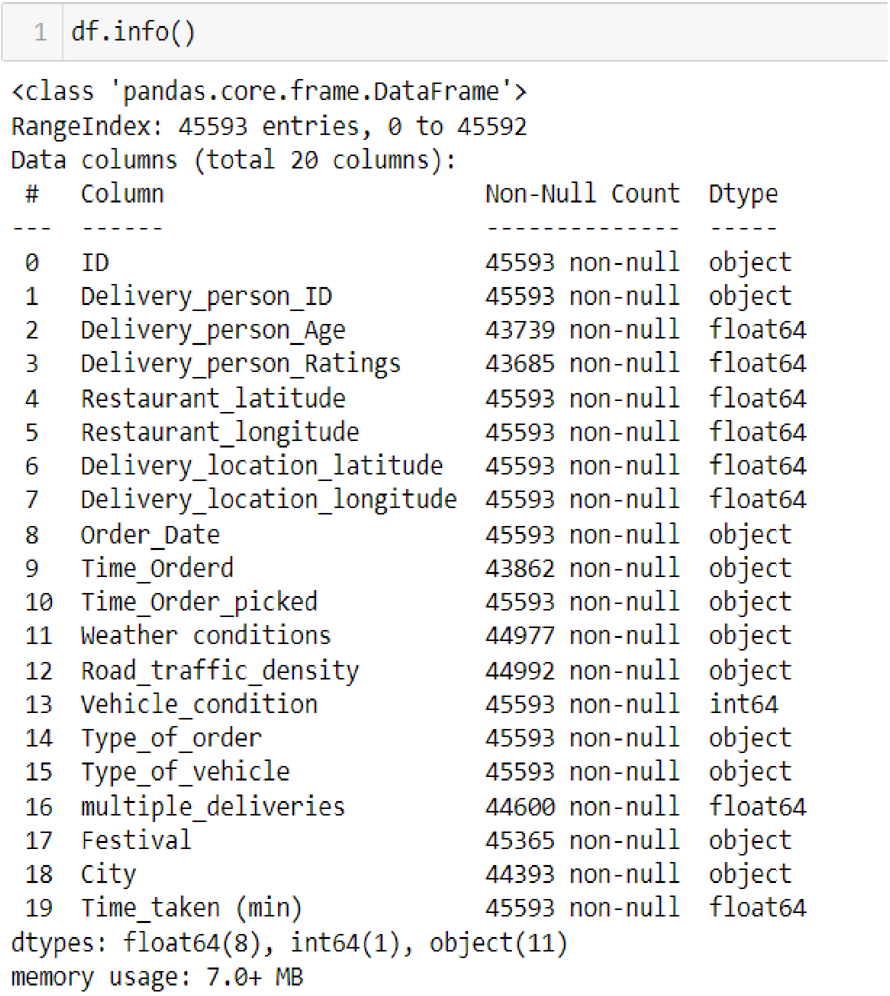
Road traffic density	Represents the road traffic density (Jam, High, Medium and Low)
Vehicle condition	Represents the condition of the vehicle. (Smooth, good or average)
Type of order	Represents the type of order (Snack, Meal, Buffet, Drinks, etc)
Type of vehicle	Represents the type of vehicle one is using (motorbike, bicycle etc.)
multiple deliveries	Represents the number of orders to be delivered in one attempt
Festival	Represents whether day is festive or not
City	Represents the city
Time taken	Represents the time taken by the delivery person to deliver the order. [TARGET]

b.Variable categorization (count of numeric and categorical)

- Numerical variables count – 9
- Categorical variables count – 11

4. Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)

Now, we know the shape of the data and also, regarding type of the features that are present in the dataset. Let's get into the information regarding the dataset. There is function called info() in pandas which gives the detail about our dataset.



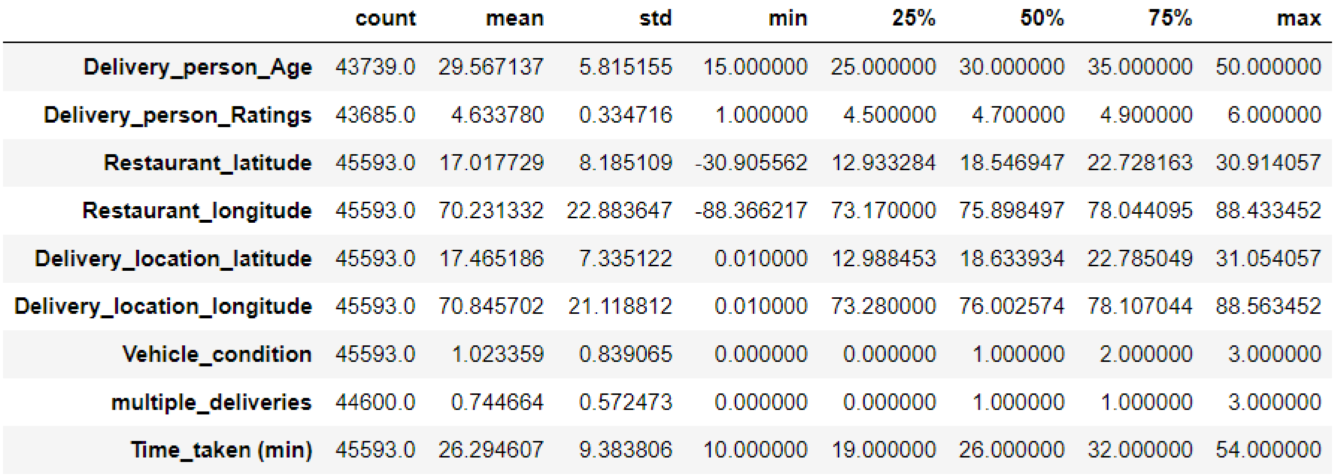
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45593 entries, 0 to 45592
Data columns (total 20 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   ID                           45593 non-null  object 
 1   Delivery_person_ID           45593 non-null  object 
 2   Delivery_person_Age          43739 non-null  float64
 3   Delivery_person_Ratings      43685 non-null  float64
 4   Restaurant_latitude          45593 non-null  float64
 5   Restaurant_longitude         45593 non-null  float64
 6   Delivery_location_latitude   45593 non-null  float64
 7   Delivery_location_longitude  45593 non-null  float64
 8   Order_Date                   45593 non-null  object 
 9   Time_Orderd                  43862 non-null  object 
 10  Time_Order_picked            45593 non-null  object 
 11  Weather conditions           44977 non-null  object 
 12  Road_traffic_density         44992 non-null  object 
 13  Vehicle_condition            45593 non-null  int64  
 14  Type_of_order                45593 non-null  object 
 15  Type_of_vehicle              45593 non-null  object 
 16  multiple_deliveries          44600 non-null  float64
 17  Festival                     45365 non-null  object 
 18  City                         44393 non-null  object 
 19  Time_taken (min)             45593 non-null  float64
dtypes: float64(8), int64(1), object(11)
memory usage: 7.0+ MB
```

From the info() function *we* get to know that,

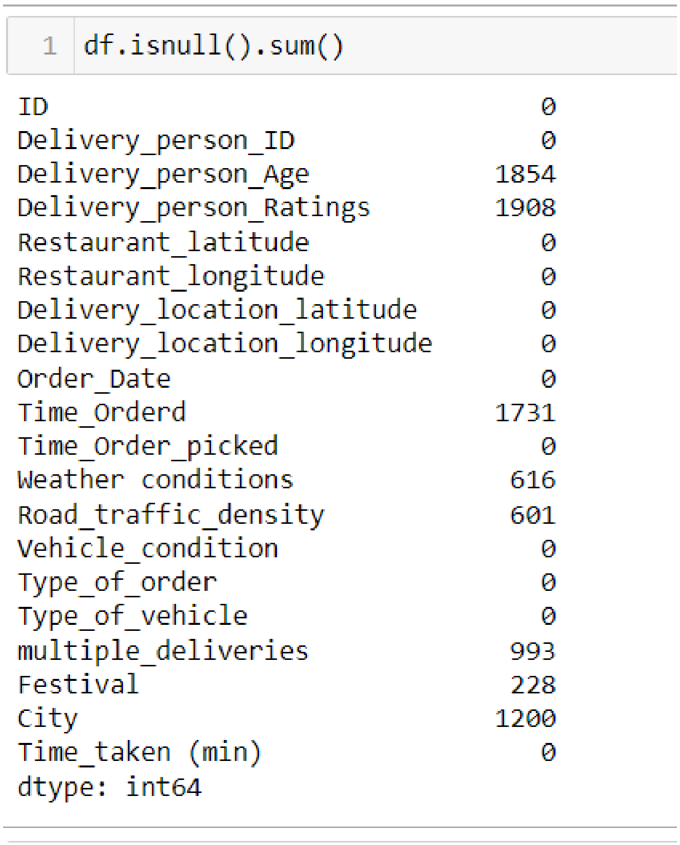
- There are null values/missing values present in the dataset.
- We have range of index from 0 to
- Total memory usage is 7.0+ MB
- This dataset contains totally 8 floating datatypes, 1 int datatype and 11 object datatypes.

4.1 Statistics about the numerical features in the dataset:



	count	mean	std	min	25%	50%	75%	max
Delivery_person_Age	43739.0	29.567137	5.815155	15.000000	25.000000	30.000000	35.000000	50.000000
Delivery_person_Ratings	43685.0	4.633780	0.334716	1.000000	4.500000	4.700000	4.900000	6.000000
Restaurant_latitude	45593.0	17.017729	8.185109	-30.905562	12.933284	18.546947	22.728163	30.914057
Restaurant_longitude	45593.0	70.231332	22.883647	-88.366217	73.170000	75.898497	78.044095	88.433452
Delivery_location_latitude	45593.0	17.465186	7.335122	0.010000	12.988453	18.633934	22.785049	31.054057
Delivery_location_longitude	45593.0	70.845702	21.118812	0.010000	73.280000	76.002574	78.107044	88.563452
Vehicle_condition	45593.0	1.023359	0.839065	0.000000	0.000000	1.000000	2.000000	3.000000
multiple_deliveries	44600.0	0.744664	0.572473	0.000000	0.000000	1.000000	1.000000	3.000000
Time_taken (min)	45593.0	26.294607	9.383806	10.000000	19.000000	26.000000	32.000000	54.000000

- There are Outliers in the features
- Through five-point summary we get to know the count, mean, std, min, max values.
- We get to know that there are missing values by looking at the count column.



```
df.isnull().sum()
```

```
ID                              0
Delivery_person_ID              0
Delivery_person_Age          1854
Delivery_person_Ratings      1908
Restaurant_latitude             0
Restaurant_longitude            0
Delivery_location_latitude      0
Delivery_location_longitude     0
Order_Date                      0
Time_Orderd                  1731
Time_Order_picked               0
Weather conditions            616
Road_traffic_density          601
Vehicle_condition               0
Type_of_order                   0
Type_of_vehicle                 0
multiple_deliveries           993
Festival                      228
City                         1200
Time_taken (min)                0
dtype: int64
```

- From these, we know that totally 8 variables having null values.
- These null values are treated by mean, median and mode imputation methods.

5.Project Justification Project Statement, Complexity involved, Project Outcome:

5.1 Project Statement :

Analysis and Prediction of Time Taken for Food Delivery by the delivery boy based on different conditions like weather, traffic, vehicle condition etc.

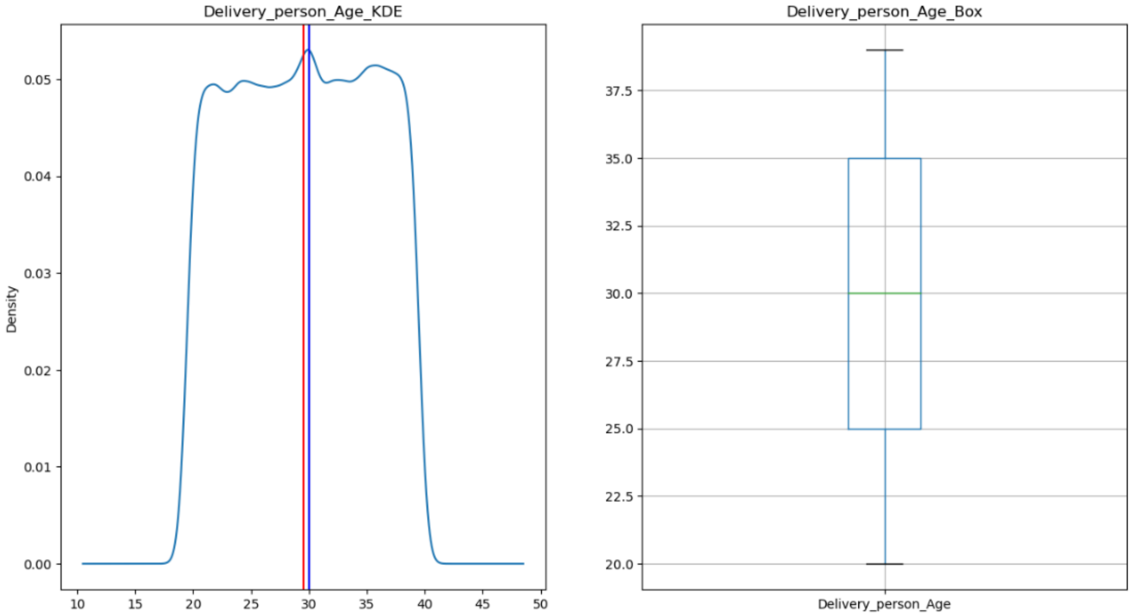
5.2 Complexity Involved :

Since, our data is not having any linear relationship with my target variable(time_taken(min)). There is non-linear relationship with my target. Hence we have to fit non-linear complex models to get better predictions. While building the complex models like decision tree, random forest, xgboost etc. Tuning of hyperparameters in this models plays a major role in getting good results and accurate predictions. This process can be achieved by using GridSearchCV, RandomisedSearch.

6. Data Exploration (EDA):

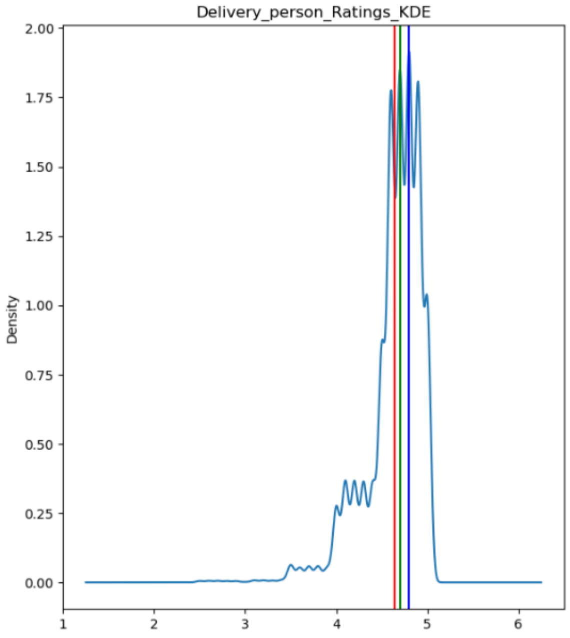
Univariate Analysis for numerical features:

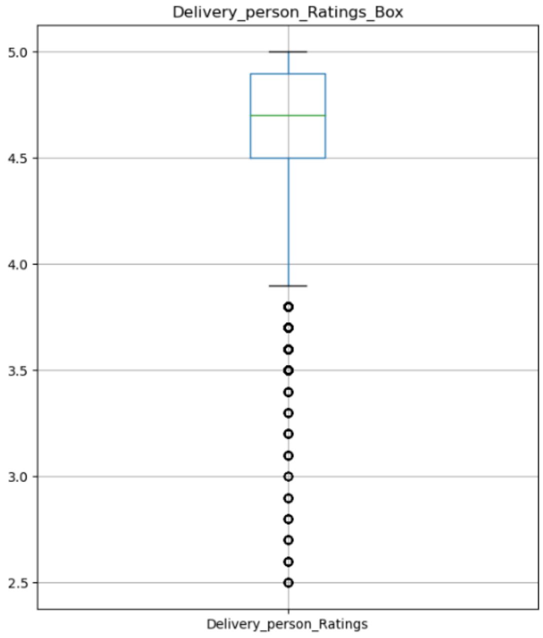
1.‘Delivery_person_Age’:



- The mean and median age of delivery persons is around 30 years old, which suggests that most of delivery persons are relatively young.
- The negative skewness (-0.0157) indicates that the distribution of delivery person ages is slightly skewed to the left, which means that there may be a few older delivery persons, but most are younger.
- The kurtosis value of –1.1999 suggests that the distribustion of delivery person ages is platykurtic, which means that it has fewer outliers than a normal distribution.

2. ‘Delivery_person_Ratings’:



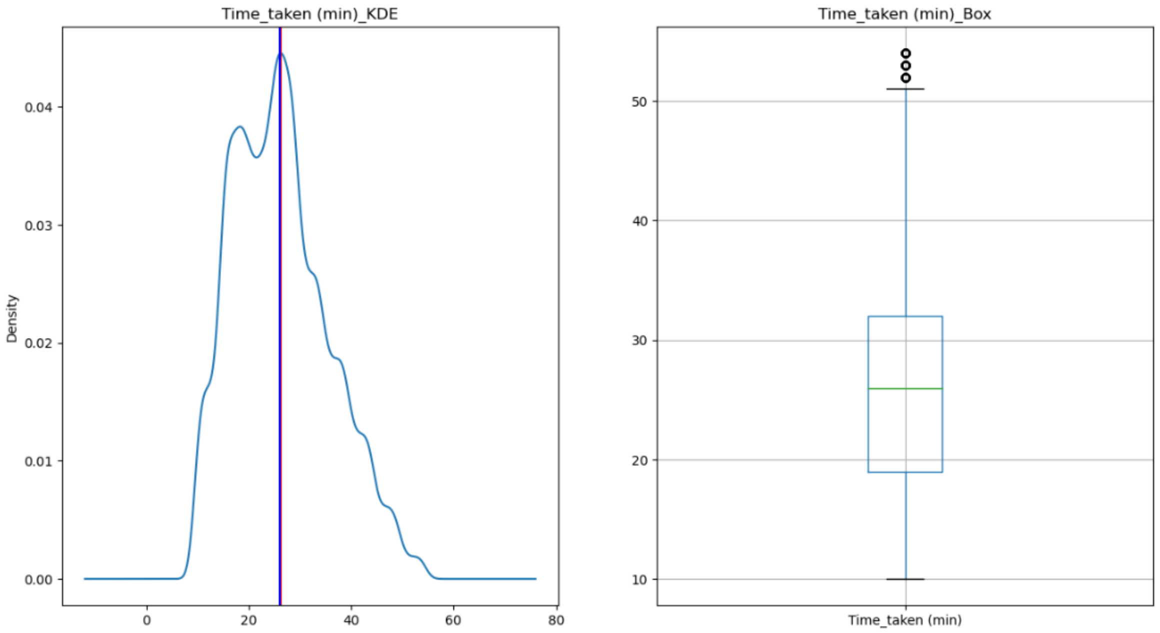


- The mean rating of delivery persons is 4.64, which suggests that delivery persons generally receive high ratings from customers.

- The negative skewness (-1.80) indicates that the distribution of delivery person ratings is skewed to the left, which means that there may be a few delivery persons who receive lower ratings, but most receive higher ratings.

- The positive kurtosis value of 5.18 suggests that the distribution of delivery person ratings is leptokurtic, which means that it has more outliers than a normal distribution. This could indicate that there are a few delivery persons who consistently receive either very high or very low ratings.

2. <u>‘Time_taken (min)’:</u>

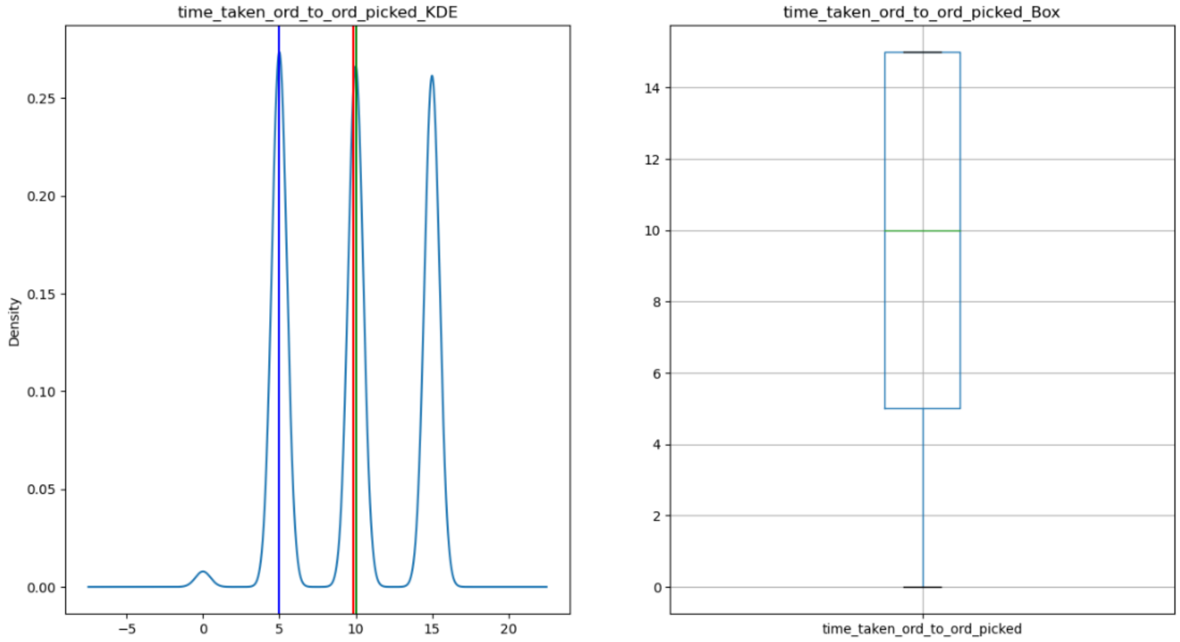


- The mean time taken for deliveries is around 26 minutes, which suggests that customers generally receive their orders within a reasonable amount of time.

- The positive skewness (0.48) indicates that the distribution of delivery times is slightly skewed to the right, which means that there may be a few deliveries that take longer than average, but most are completed within a shorter amount of time.

- The negative kurtosis value of -0.31 suggests that the distribution of delivery times is platykurtic, which means that it has fewer outliers than a normal distribution. This could indicate that there are fewer extremely long or short delivery times than would be expected in a normal distribution.

4. ‘time_taken_ord_to_ord_picked’:

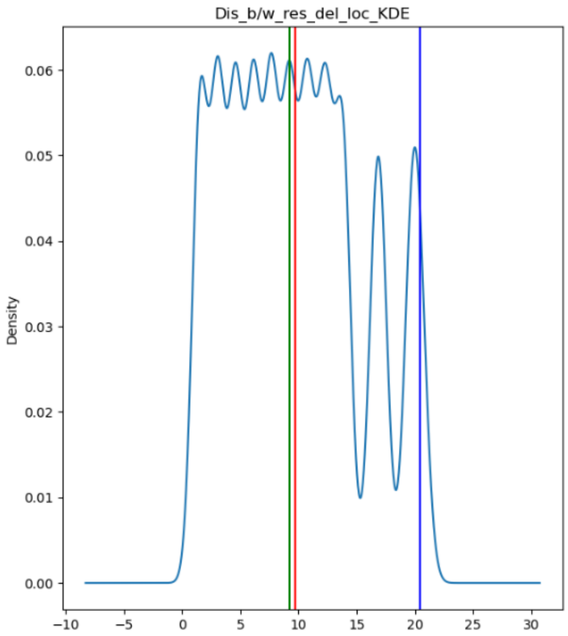


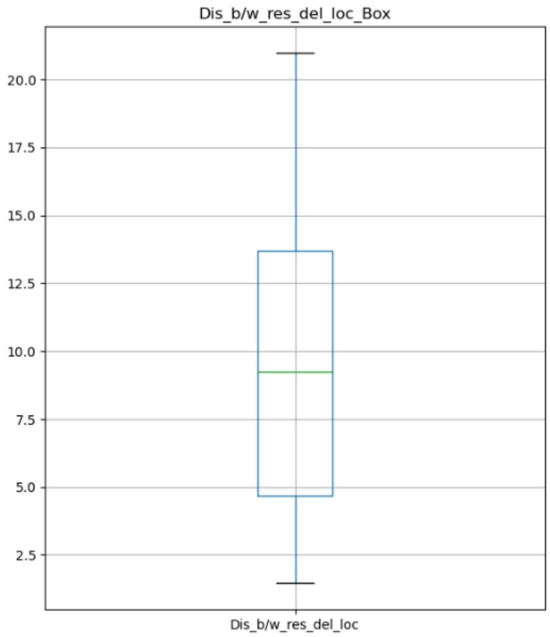
- The mean time taken from order placement to order pickup is around 9.83 minutes, which suggests that orders are generally picked up by delivery persons within a reasonable amount of time.

- The negative kurtosis value of -1.34 suggests that the distribution of time taken from order placement to order pickup is platykurtic, which means that it has fewer outliers than a normal distribution. This could indicate that there are fewer extremely long or short times than what would be expected in a normal distribution.

- The negative skewness (-0.04) indicates that the distribution of time taken from order placement to order pickup is slightly skewed to the left, which means that there may be a few orders that take longer than average to be picked up, but most are picked up within a shorter amount of time.

5. ‘Dis_b/w_res_del_loc’:





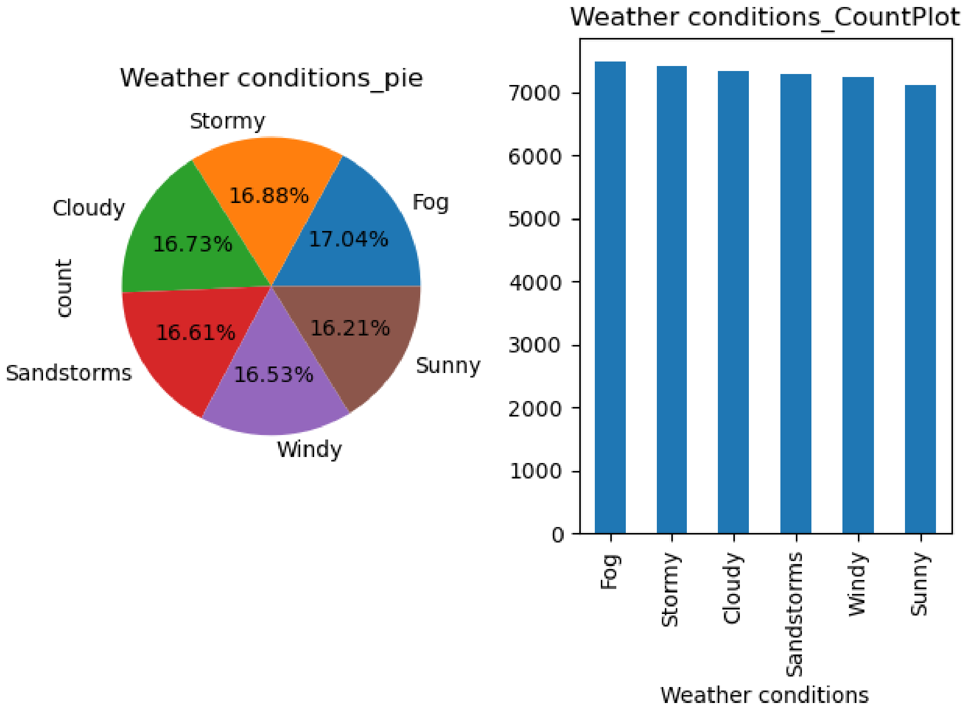
- The mean distance between the restaurant and the delivery location is approximately 9.73 units, indicating that, on average, deliveries are made within a relatively short distance.

- The positive skewness (0.32) suggests that the distribution of distances between the restaurant and the delivery location is slightly skewed to the right. This indicates that there may be a few deliveries that require longer distances, but most deliveries are made within a shorter range.

- The negative kurtosis value of -0.90 suggests that the distribution of distances between the restaurant and the delivery location is platykurtic, meaning it has fewer outliers than a normal distribution. This implies that there are fewer extremely long or short distances than what would be expected in a normal distribution.

Univariate Analysis for Categorical features:

1. ‘Weather conditions’:



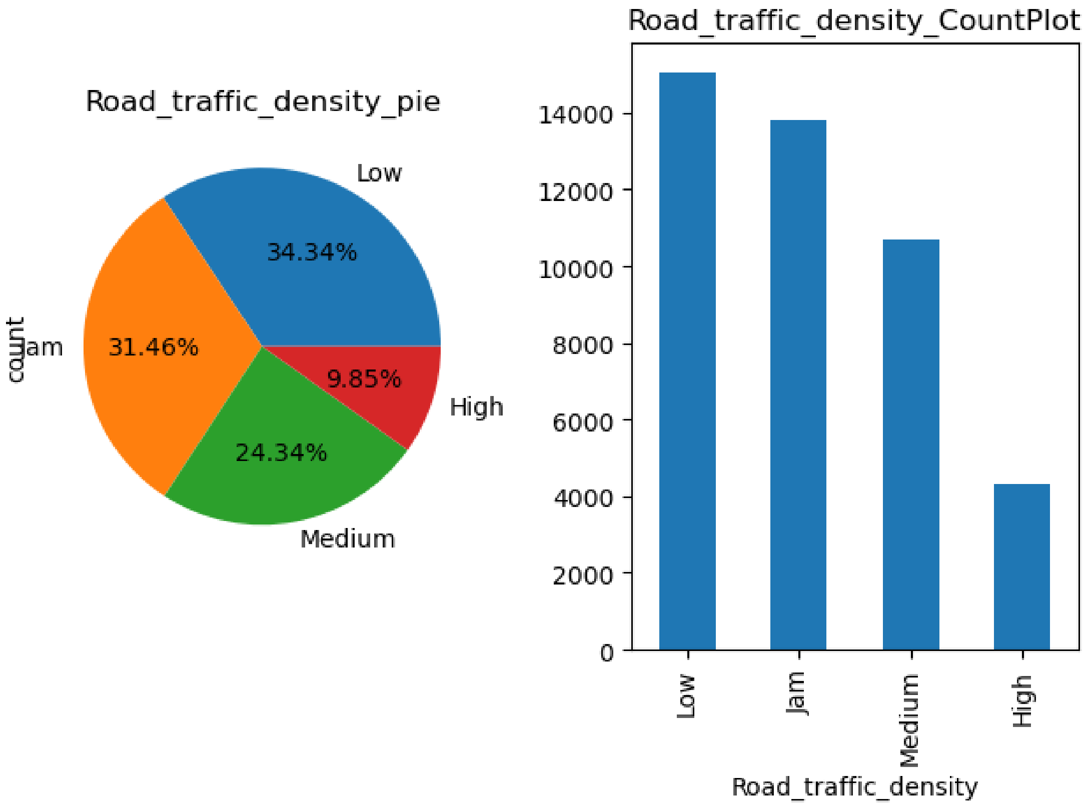
- The most common weather condition observed is "Fog", with a count of 7476 occurrences. This suggests that foggy weather conditions occur more frequently compared to other weather conditions in the dataset.

- The count of all subcategories shows that "Fog", "Stormy", "Cloudy", "Sandstorms", "Windy", and "Sunny" are the weather conditions included in the dataset. These categories represent different types of weather conditions.

- The counts of the different weather conditions indicate that "Fog" has the highest occurrence, followed by "Stormy", "Cloudy", "Sandstorms", "Windy", and "Sunny". This suggests that foggy and stormy weather conditions are relatively more common compared to other weather conditions in the dataset.

greatlearning
Learning for Life

2. ‘Road_Traffic_density’:

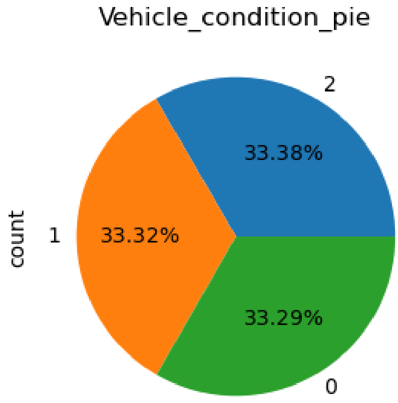


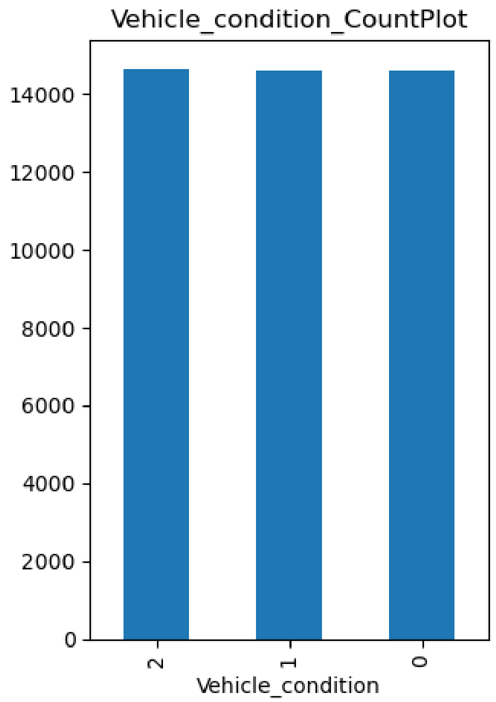
- The most common road traffic density observed is "Low", with a count of 15062 occurrences. This suggests that low traffic density is more frequently observed compared to other traffic densities in the dataset.

- The count of all subcategories shows that "High", "Jam", "Low", and "Medium" are the road traffic densities included in the dataset. These categories represent different levels of traffic density.

- The counts of the different road traffic densities indicate that "Low" has the highest occurrence, followed by "Jam", "Medium", and "High". This suggests that low and jammed traffic densities are relatively more common compared to medium and high traffic densities in the dataset.

3. ‘Vehicle_condition’:



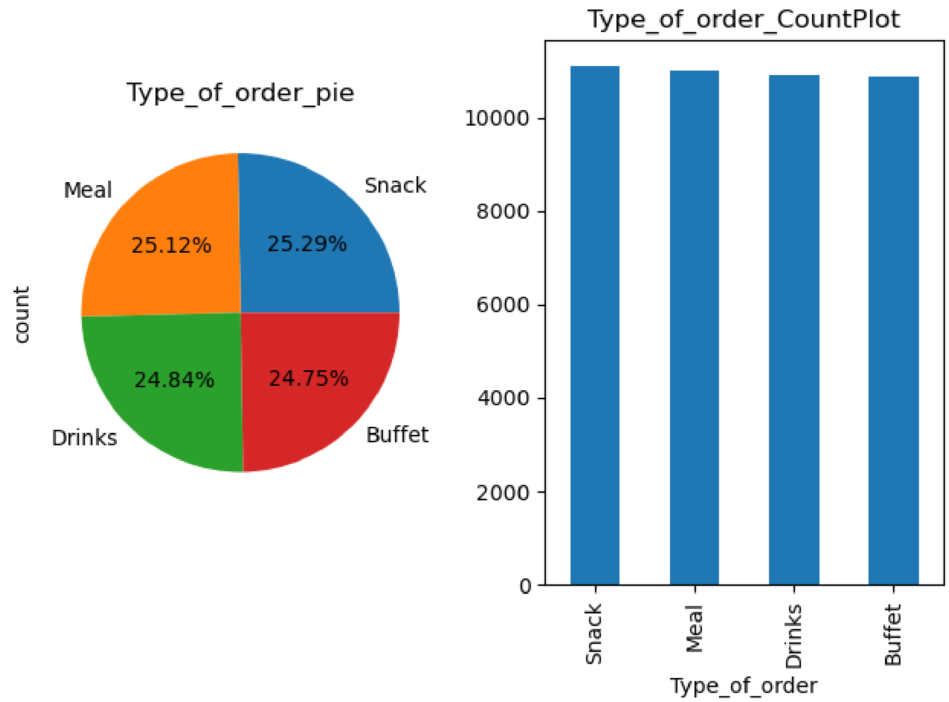


- The most common vehicle condition observed is "2", with a count of 14642 occurrences. This suggests that vehicles in condition "2" are more frequently observed compared to other vehicle conditions in the dataset.

- The count of all subcategories shows that "2", "1", and "0" are the vehicle conditions included in the dataset. These categories represent different levels or states of vehicle condition.

- The counts of the different vehicle conditions indicate that "2" has the highest occurrence, followed by "1" and "0". This suggests that vehicles in condition "2" are relatively more common compared to vehicles in conditions "1" and "0" in the dataset.

4. ‘Type_of_order’:



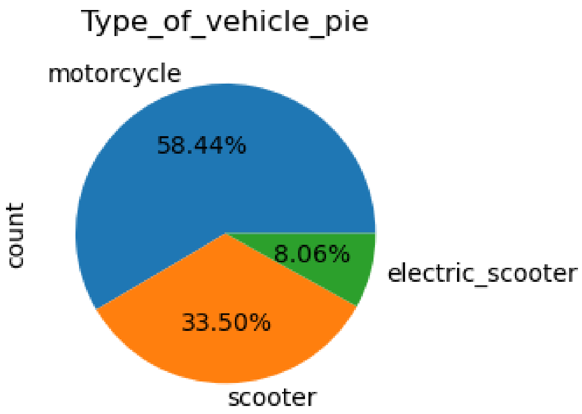
- The most common type of order observed is "Snack", with a count of 11091 occurrences. This suggests that snack orders are more frequently observed compared to other types of orders in the dataset.

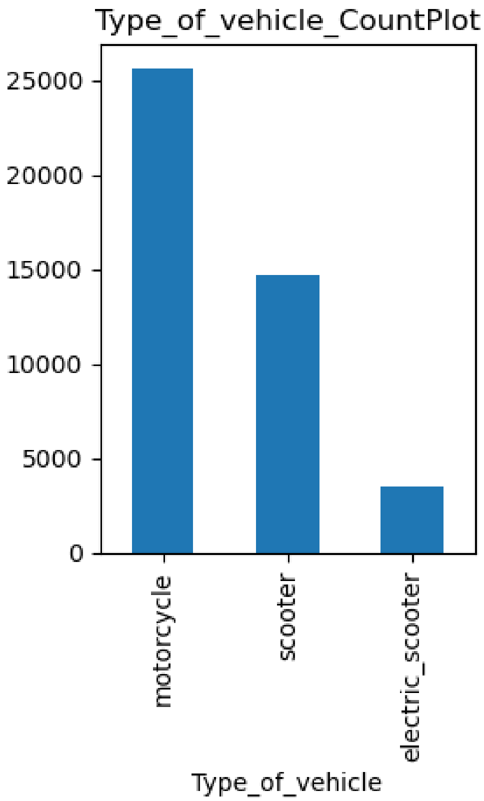
- The count of all subcategories shows that "Snack", "Meal", "Drinks", and "Buffet" are the types of orders included in the dataset. These categories represent different types of food or meal options.

- The counts of the different types of orders indicate that "Snack" has the highest occurrence, followed by "Meal", "Drinks", and "Buffet". This suggests that snack orders are relatively more common compared to meal, drink, and buffet orders in the dataset.

greatlearning
Learning for Life

5. ‘type_of_vehicle’:



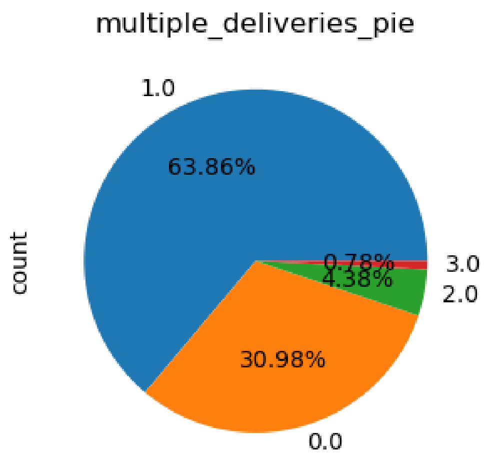


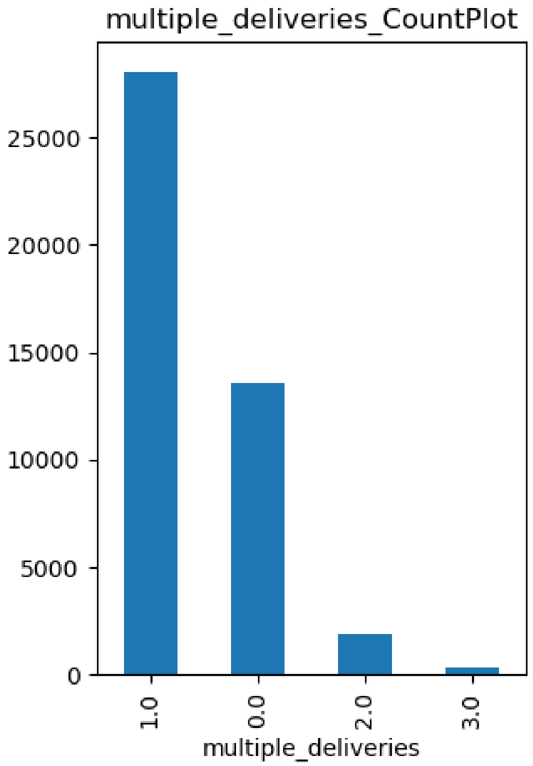
- The most common type of vehicle observed is "motorcycle", with a count of 25633 occurrences. This suggests that motorcycles are more frequently used as delivery vehicles compared to other types of vehicles in the dataset.

- The count of all subcategories shows that "motorcycle", "scooter", and "electric_scooter" are the types of vehicles included in the dataset. These categories represent different types of delivery vehicles.

- The counts of the different types of vehicles indicate that "motorcycle" has the highest occurrence, followed by "scooter" and "electric_scooter". This suggests that motorcycles are relatively more common compared to scooters and electric scooters as delivery vehicles in the dataset.

6.'Multiple deliveries':



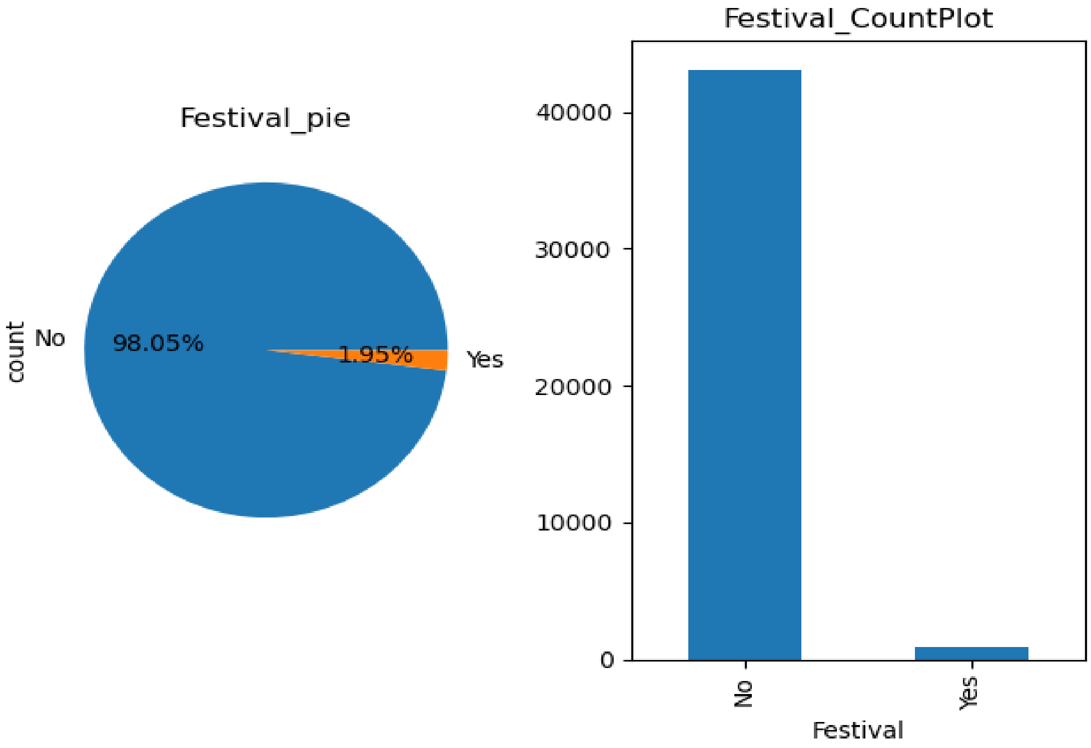


- The most common number of multiple deliveries observed is "1.0", with a count of 28011 occurrences. This suggests that most deliveries involve only one package or order.

- The count of all subcategories shows that "0.0", "1.0", "2.0", and "3.0" are the number of multiple deliveries included in the dataset. These categories represent different numbers of packages or orders delivered in a single trip.

- The counts of the different numbers of multiple deliveries indicate that "1.0" has the highest occurrence, followed by "0.0", "2.0", and "3.0". This suggests that most deliveries involve only one package or order, while a smaller number of deliveries involve multiple packages or orders in a single trip.

6. ‘Festival’:



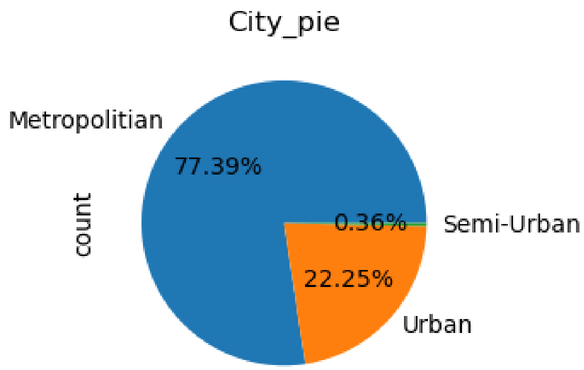
- The most common festival status observed is "No", with a count of 43005 occurrences. This suggests that most orders are placed during non-festival periods.

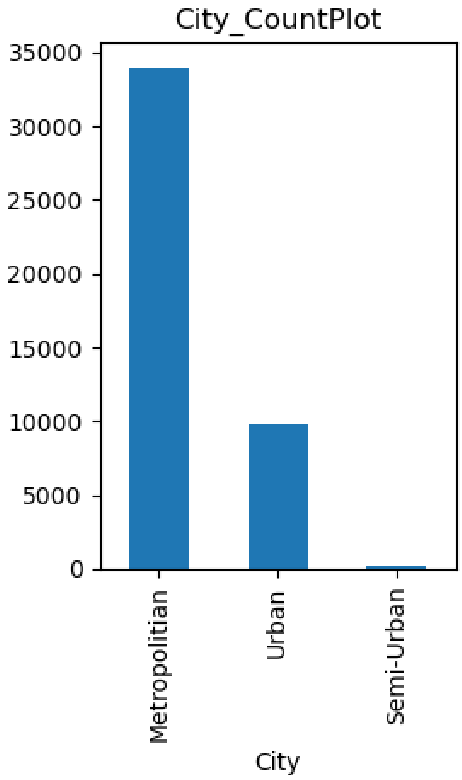
- The count of all subcategories shows that "No" and "Yes" are the festival statuses included in the dataset. These categories represent whether the order was placed during a festival or not.

- The counts of the different festival statuses indicate that "No" has the highest occurrence, followed by "Yes". This suggests that orders placed during non-festival periods are more common compared to orders placed during festivals in the dataset.

greatlearning
Learning for Life

7. ‘City’:



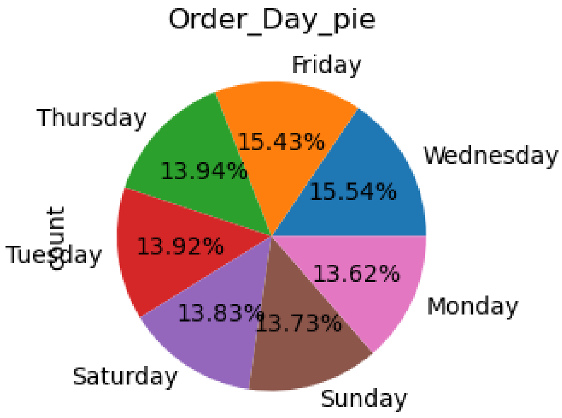


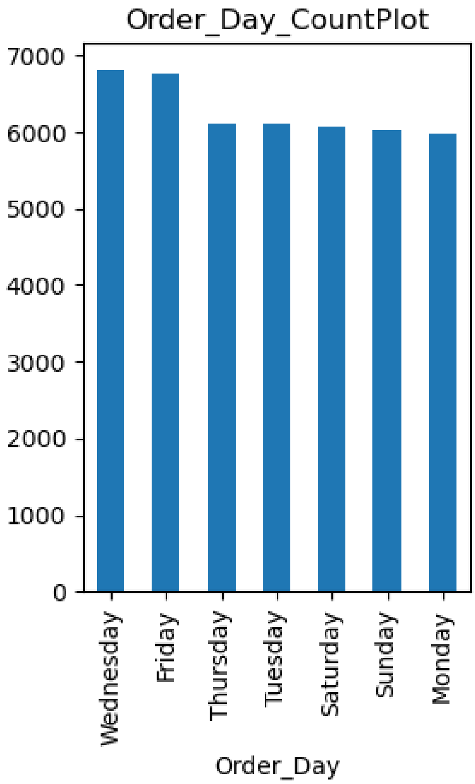
- The most common city type observed is "Metropolitian", with a count of 33946 occurrences. This suggests that a majority of the data points in the dataset correspond to metropolitan cities.

- The count of all subcategories shows that "Metropolitian", "Urban", and "Semi-Urban" are the city types included in the dataset. These categories represent different types of cities based on their level of urbanization.

- The counts of the different city types indicate that "Metropolitian" has the highest occurrence, followed by "Urban" and "Semi-Urban". This suggests that metropolitan cities are relatively more common compared to urban and semi-urban cities in the dataset.

8. Order_Day:





- The most common order day observed is "Wednesday", with a count of 6816 occurrences. This suggests that Wednesdays have the highest number of orders compared to other days of the week in the dataset.

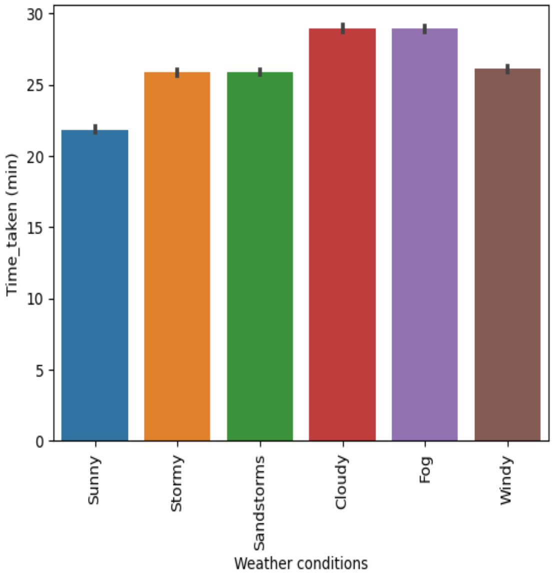
- The count of all subcategories shows that "Wednesday", "Friday", "Thursday", "Tuesday", "Saturday", "Sunday", and "Monday" are the order days included in the dataset. These categories represent different days of the week.

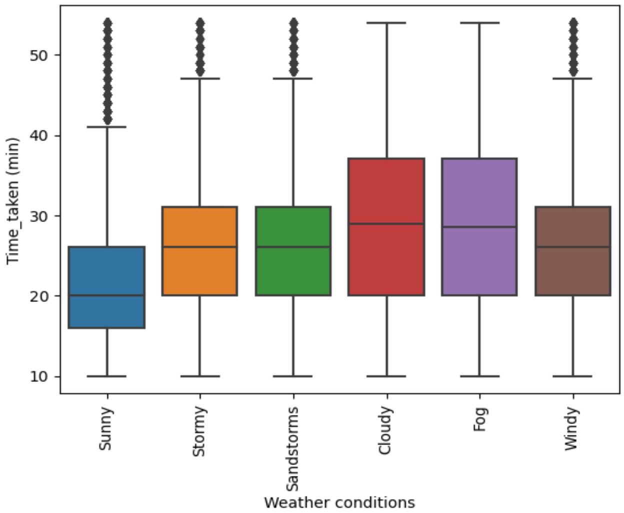
- The counts of the different order days indicate that "Wednesday" has the highest occurrence, followed by "Friday", "Thursday", "Tuesday", "Saturday", "Sunday", and "Monday". This suggests that Wednesdays and Fridays tend to have a higher number of orders compared to other days of the week in the dataset.

Bivariate Analysis:

- **Categorical Features:**

1. Weather conditions:





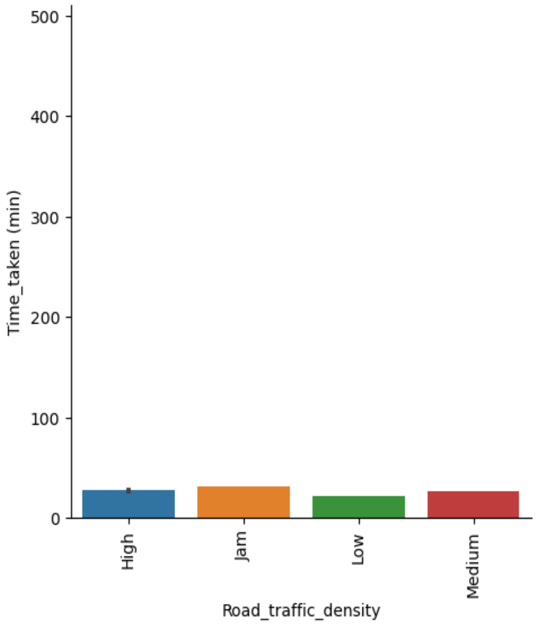
- As we can see when we are plotting weather conditions against our target column (time_taken(min)) we can see that there is no much variation in its categories.

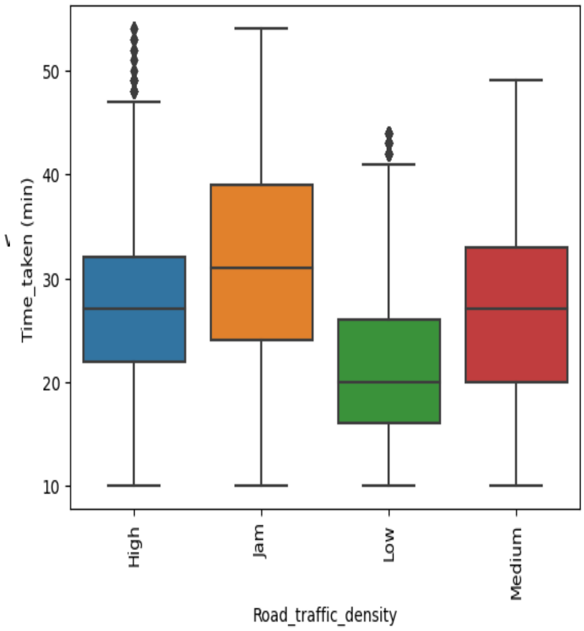
- Both the barplot and boxplot are not showing much variances among the groups.

- That means no matter what the weather condition is the time taken is constant. (Little bit faster in the conditions of cloudy and foggy) .

greatlearning
Learning for Life

2. Road_traffic:

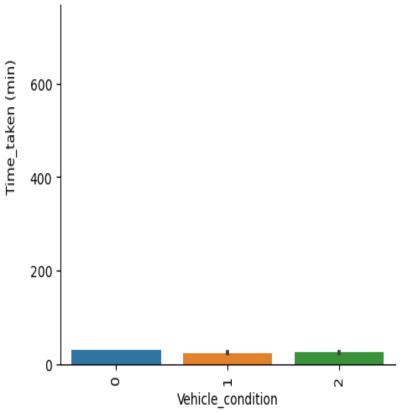


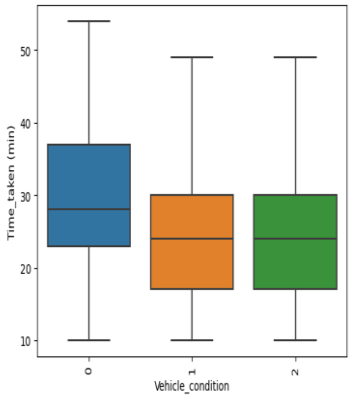


- As we are plotting Road_traffic_density column against our target (time_taken(min)), we can see that when the traffic density is low the time taken to deliver is less.

- And when the traffic is jammed the time taken is little bit more when compared to other traffic situations.

3.Vehicle_condition:

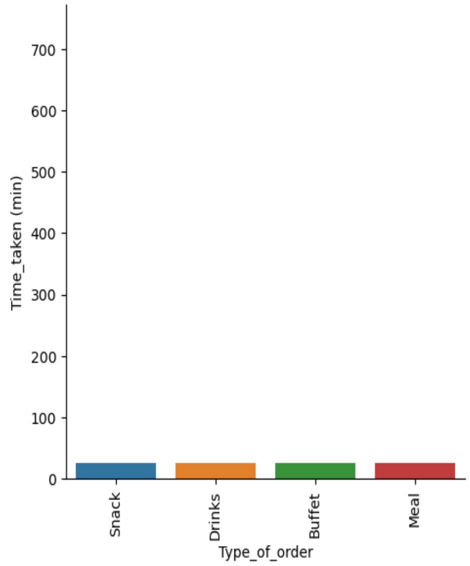


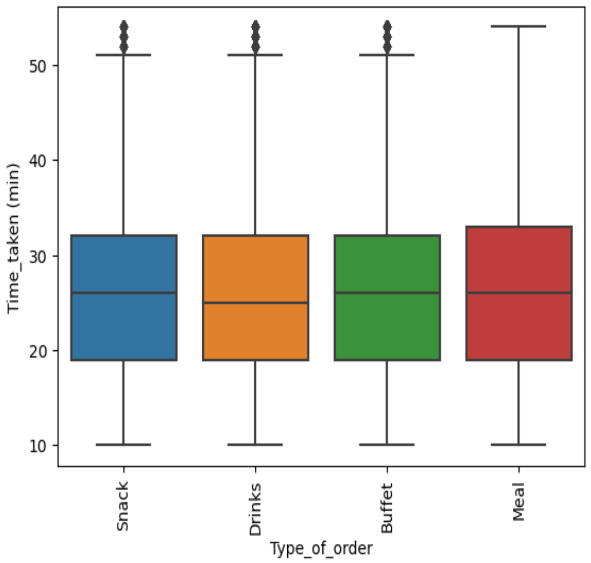


- From the above plots which were plotted vehicle condition against our target column(time_taken(min)) we can infer that when the vehicle condition is bad, more time is taken.

- And for the average condition vehicle and god conditioned vehicle the time taken is same to deliver there is no much variance in it.

4.Type_of_order:



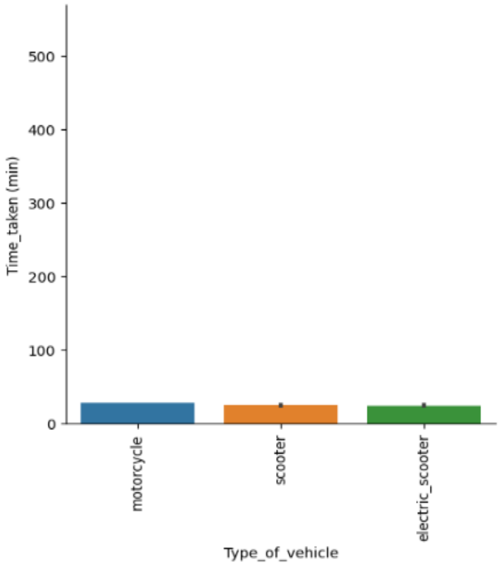


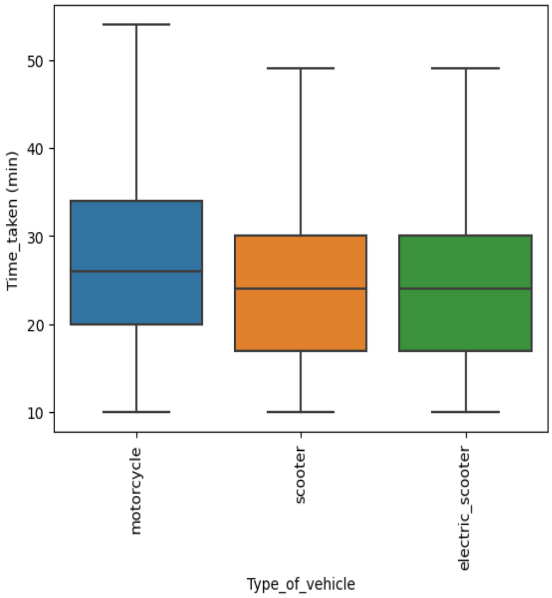
- The above plots are of type_of_order(food) against time_taken(min) (target column).

- As we can see that, no matter what the type of food is, there’s no much variance in the delivery time.

- All most all the orders are taking around 50 min of delivery time.

5.Type_of_vehicle:





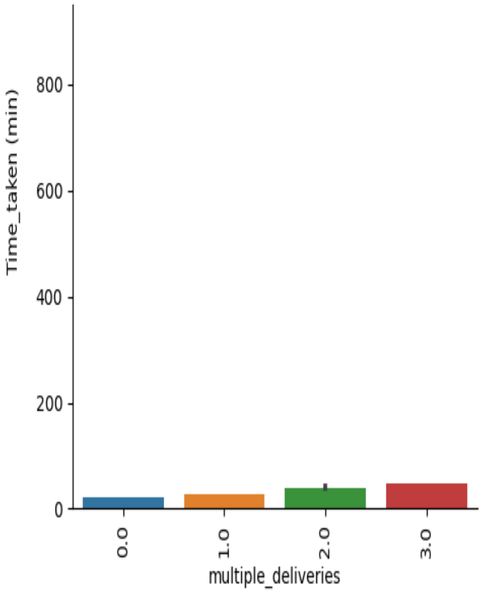
- In the above graphs we plotted type_of_vehicle column against our target column time_taken(min).

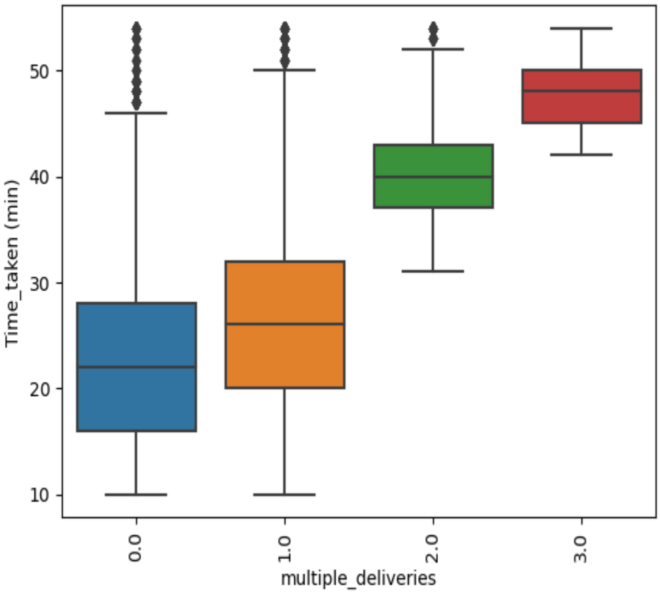
- We can observe that if the vehicle is of motorcycle type, it is taking much time to deliver.

- While the scooter and electric scooter types are delivering faster than compared to motor cycle.

- And both of the scooter types are taking the same time to deliver.

6.Multiple_deliveries:

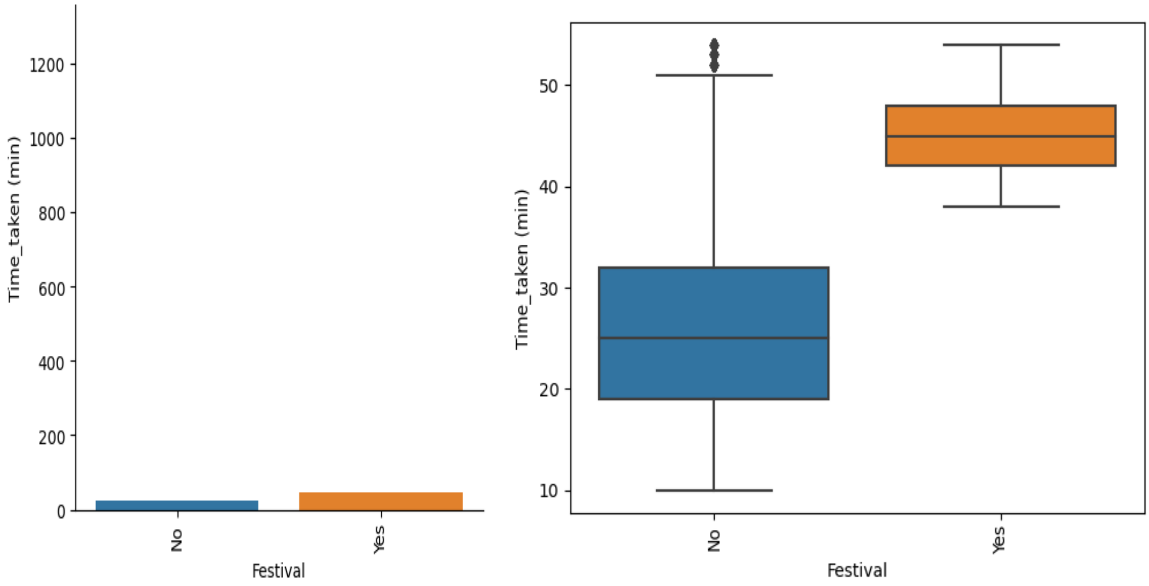




- In the above plots we plotted multiple_deliveries column against our target column(time_taken(min). Here, we can see that there is much variance in the time depending on the number of deliveries.

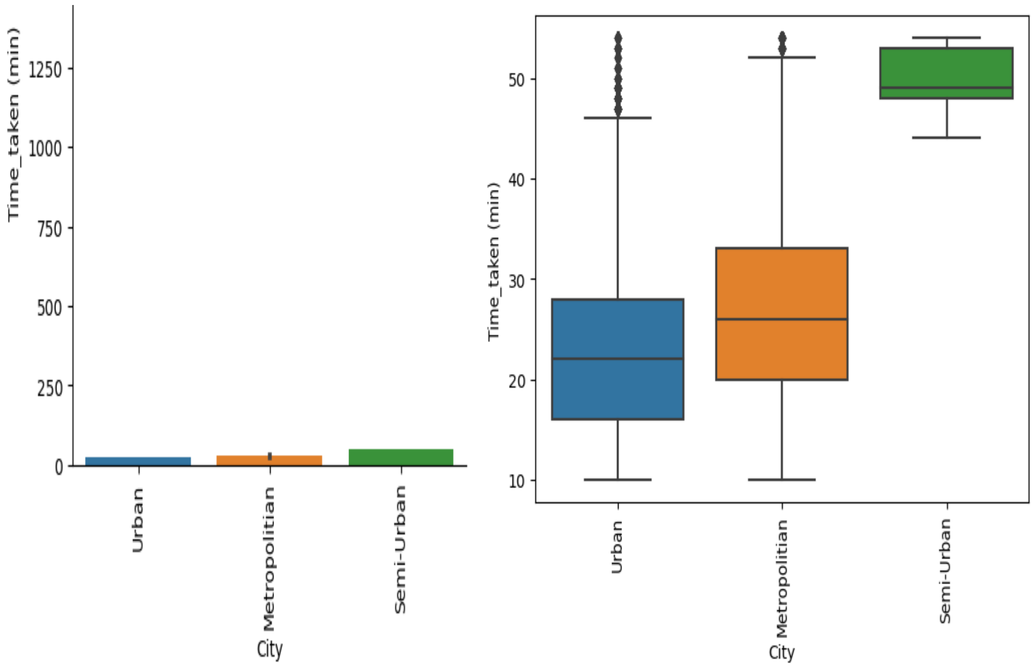
- With the increase in the number of deliveries, the time taken is also increasing gradually.

7.Festival:



- Here, we plotted festival column against our target column (time_taken(min)) .
- If there is a festival on a particular day (ordered day) then the delivery person is taking more time to deliver the product. (As on the traffic will be more on the festival day).
- If it is not a festival day the time taken to deliver is less.

8.City:

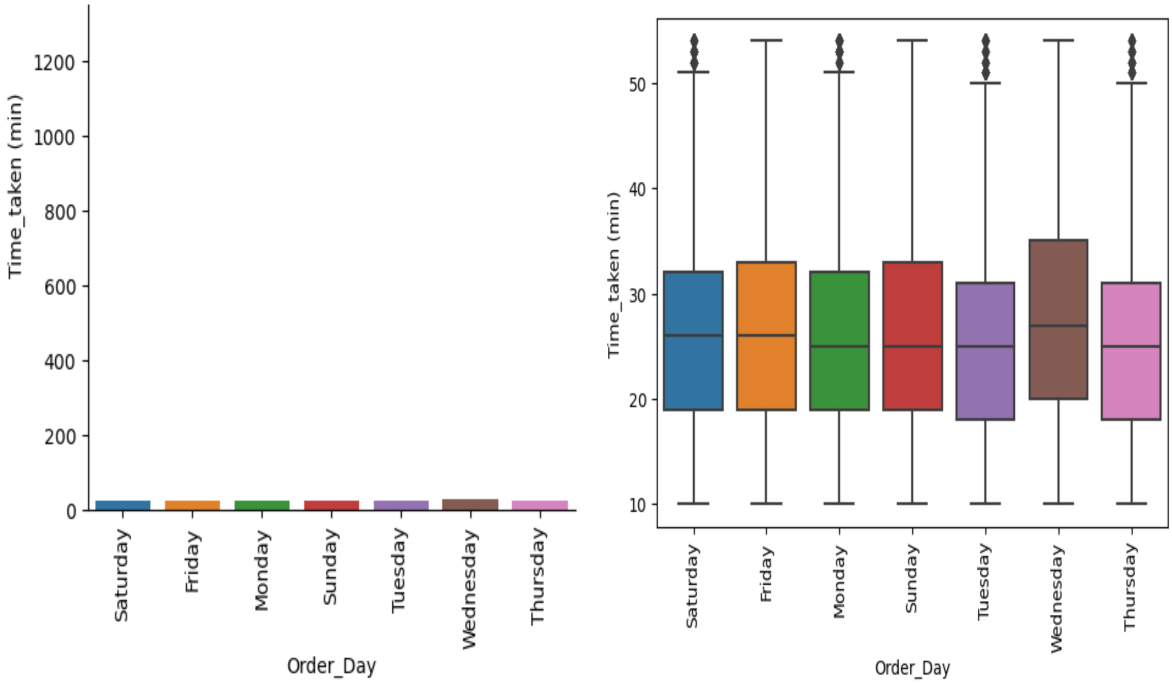


- Here, in the above plots we plotted city column against our target column (time_taken(min)).

- From the above plots we can infer that, semi_urban_cities type are much time taken places to deliver the order.

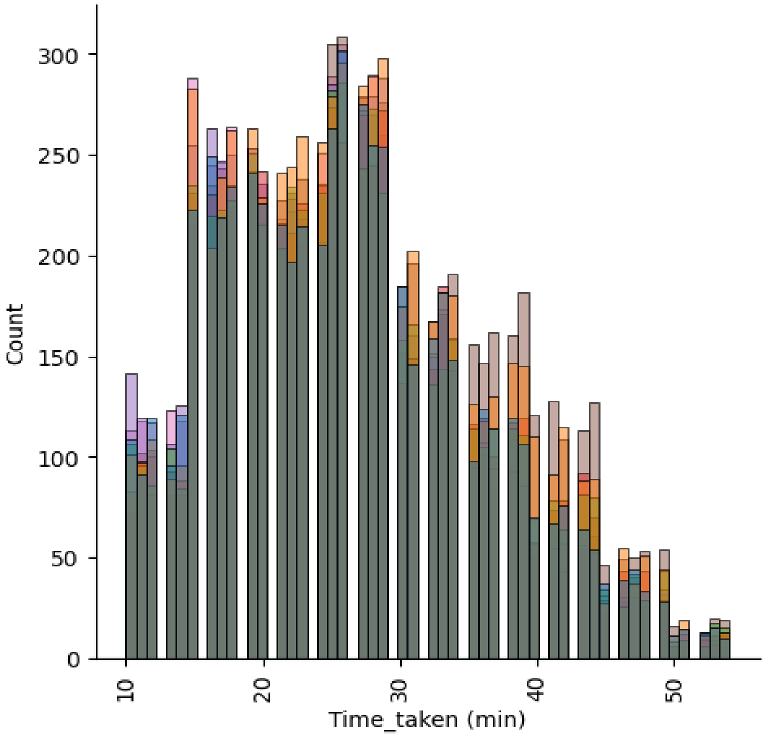
- Urban cities can avail the order in the less time

9.Order Day:



- As we plotted order day column against our dependent variable (time_taken(min)).
- We can observe that, there's no much variance in the time based on the orders.Comparing to other days Wednesday is taking a little more time to deliver the order.
- Almost in all the days, the order is delivered around 26 min.

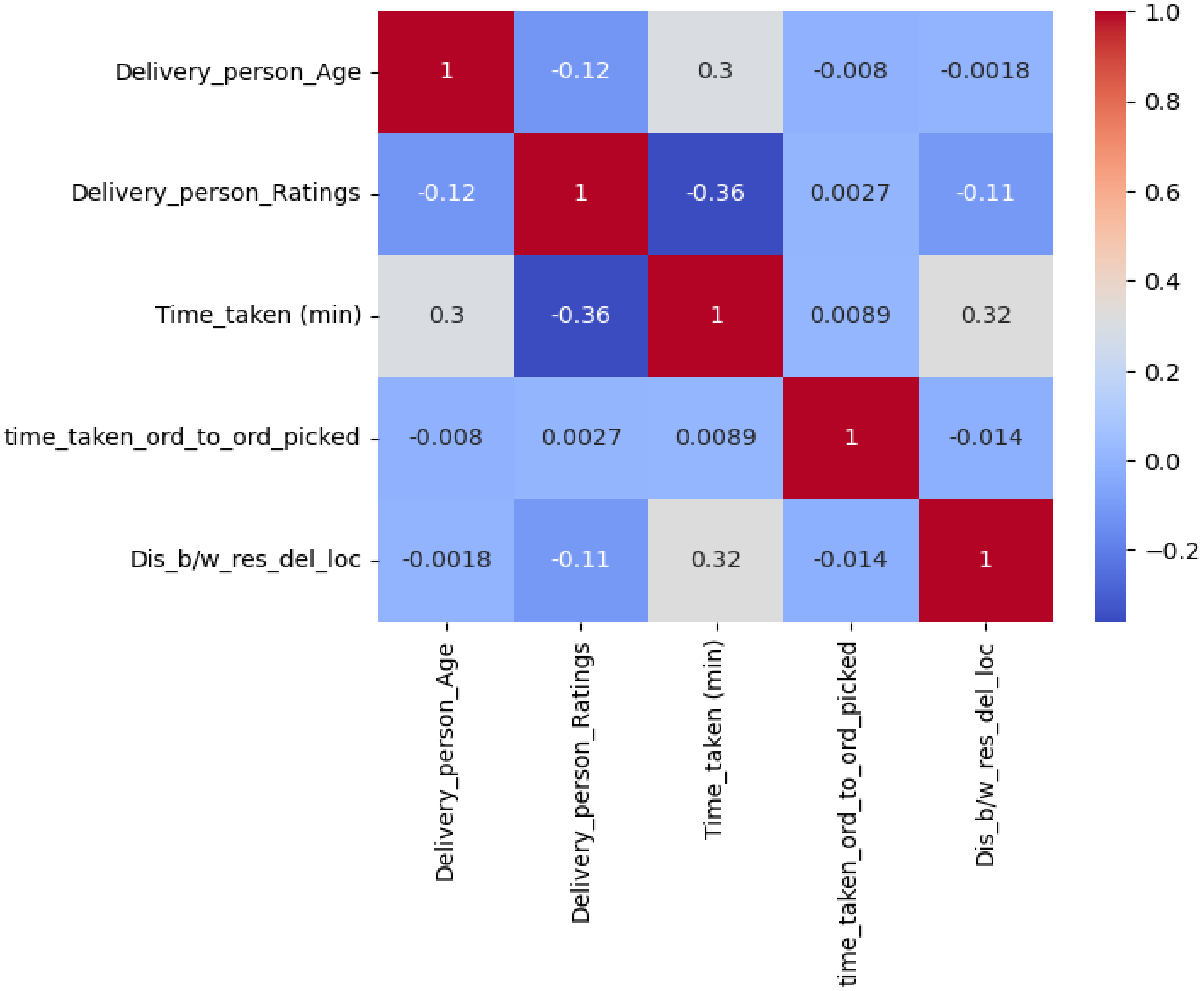
10.Time_Taken(Min):



- Here, we count plotted our taget column (time_taken(min)) .
- From the above plot we can infer that most of the orders taking 15min to 30min to deliver.
- And very less orders are taking more than 50 min.

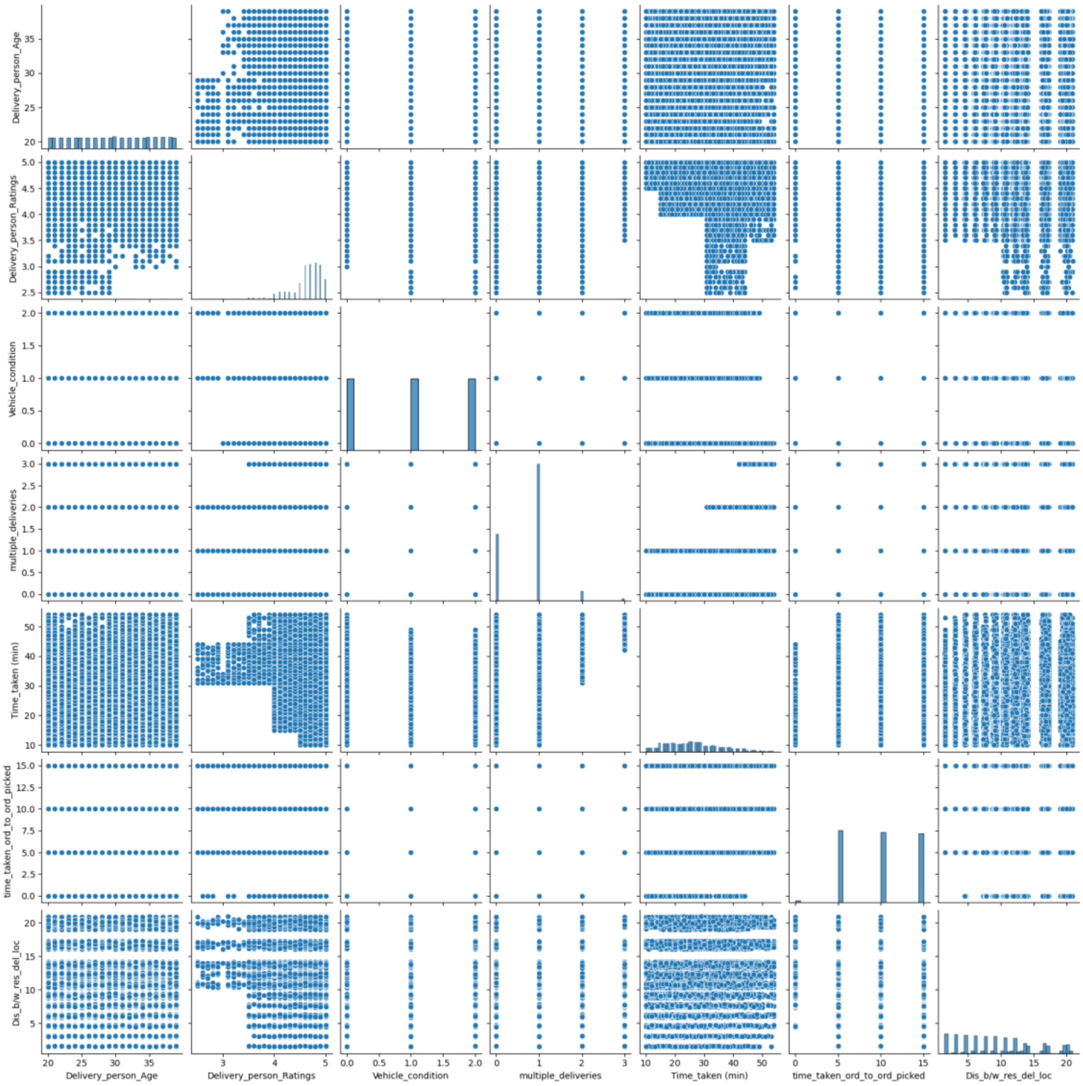
Multivariate Analysis:

- Heatmap for finding the correlation of the numerical features.



- From the above heatmap we can see that the columns time_taken_ord_to_ord_picked and time_taken(min) (target column) are correlating strongly.
- There's no much correlation between the variables.
- Hence, there is no multicollinearity between the variables.

Pair plot :



- From the above pairplot , we can infer that none of the variables are having linear relationships in between them.

- May be they are having complex relations. So, we have to build some complex models (like decision tree, random forest, adaboost, xgboost etc.) so that we can get some good accuracy and good predictions.

Statistical significance of variable:

- The numerical features were checked for normal distribution using Jarque Bera test. The features which were normally distributed were tested using ttest_ind. The features which were not normally distributed were tested using manwhitneyu test.
- For categorical features, since 3 or more categories are present in each feature, ANOVA test is used.
- The results are as follows:

Significant Numerical Featues:

Delivery_person_Age
Delivery_person_Ratings
time_taken_ord_to_ord_picked
Dis_b/w_res_del_loc

Significant Categorical Featues:

Weather conditions	multiple_deliveries
Road_traffic_density	Festival
Vehicle_condition	City
Type_of_order	Order_Day
Type_of_vehicle	

The variable Type_of_order seems to be insignificant as it’s p-value greater than the significance level based on the statistical tests.

Features and respective P values:

Features	P value
Delivery_person_Age	0.00
Delivery_person_Ratings	0.00
time_taken_ord_to_ord_picked	0.00
Dis_b/w_res_del_loc	0.00
multiple_deliveries	0.00
Festival	0.00
City	0.00

Features	P value
Weather conditions	0.00
Road_traffic_density	0.00
Vehicle_condition	2.15322025e-29
Type_of_order	0.377346839
Type_of_vehicle	8.97326862e-265
Order_Day	2.99845787e-196

Here almost every Numerical features seems significant, where as in categorical Type_of_order seems insignificant.

The numerical features were checked for normal distribution using Jarque Bera test. All the features are Normally Distributed.

The features which were normally distributed were tested using t_test_ind.

The categorical features were tested using ANOVA Oneway test.

9.Feature Engineering:

Based on domain knowledge we have selected and transformed the most relevant variables from data .

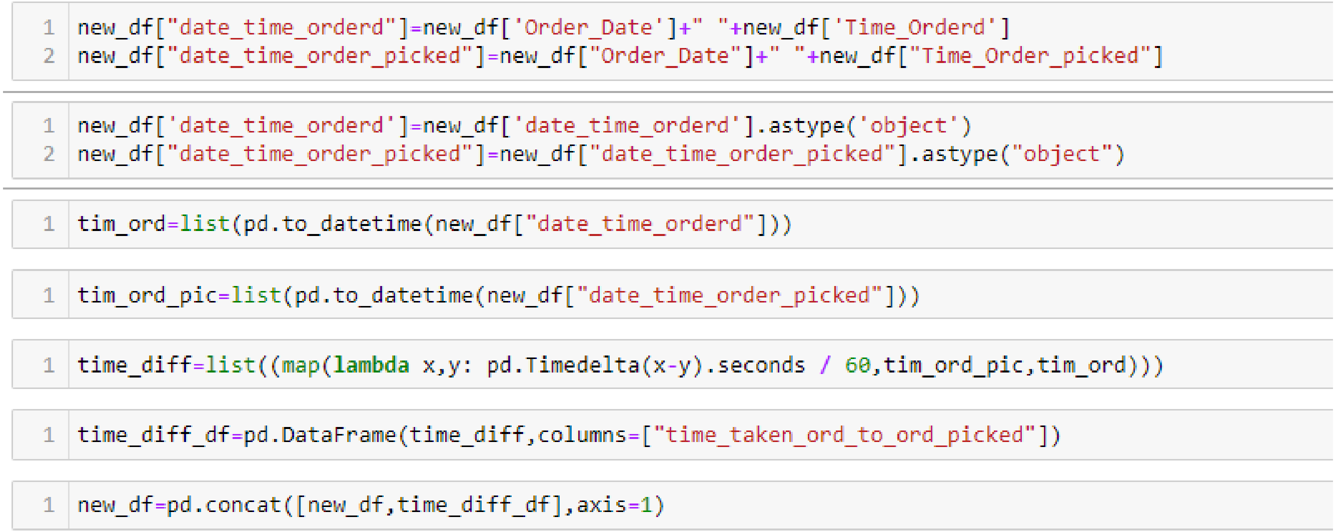
Meticulously chosed the three features based on domain knowledge and statistical analysis for optimal model relevance.

The three new features introduced to our dataset are :

- time_diff_df
- distance_df
- order day

1.Time Calculation:

Calculating the time taken difference from Order placed to Order picked up by the delivery person with the respective columns.



```
new_df["date_time_orderd"]=new_df['Order_Date']+" "+new_df['Time_Orderd']
new_df["date_time_order_picked"]=new_df["Order_Date"]+" "+new_df["Time_Order_picked"]
```

```
new_df['date_time_orderd']=new_df['date_time_orderd'].astype('object')
new_df["date_time_order_picked"]=new_df["date_time_order_picked"].astype("object")
```

```
tim_ord=list(pd.to_datetime(new_df["date_time_orderd"]))
```

```
tim_ord_pic=list(pd.to_datetime(new_df["date_time_order_picked"]))
```

```
time_diff=list((map(lambda x,y: pd.Timedelta(x-y).seconds / 60,tim_ord_pic,tim_ord)))
```

```
time_diff_df=pd.DataFrame(time_diff,columns=["time_taken_ord_to_ord_picked"])
```

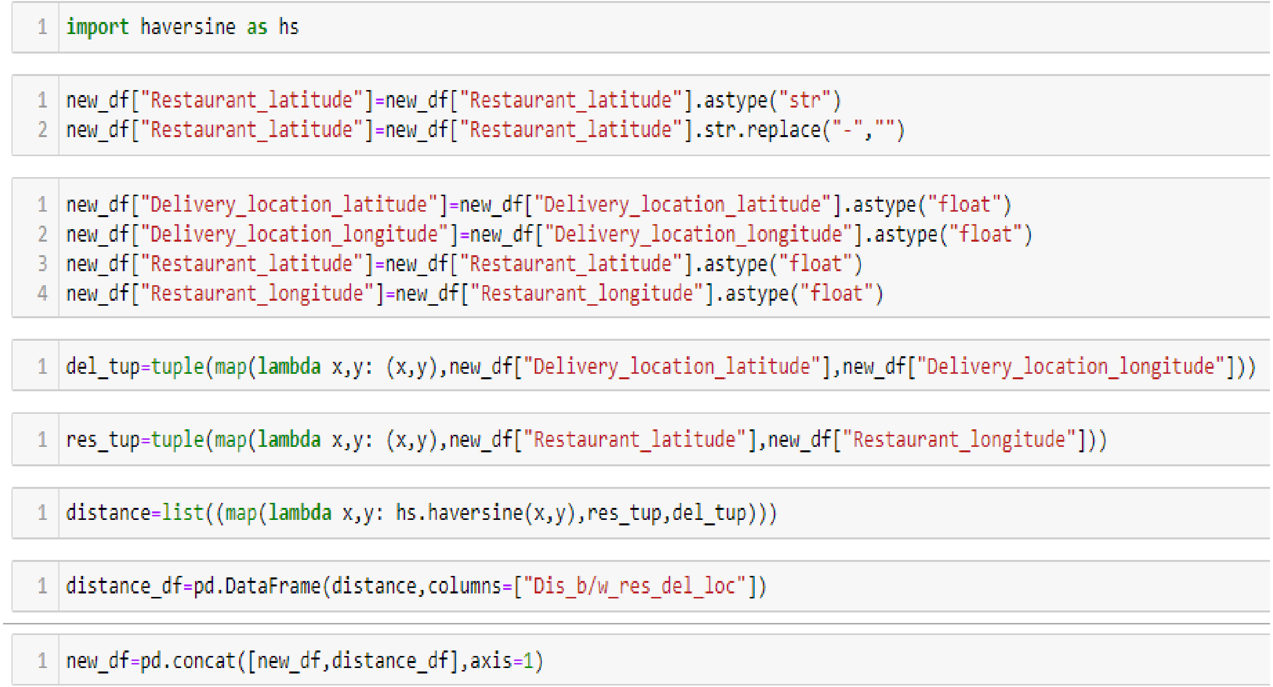
```
new_df=pd.concat([new_df,time_diff_df],axis=1)
```

Figure : 01

In dataset the new column is added that is “time_diff_df” based on feature engineering method where time is calculated by combining the time ordered by the customer and order picked up by the delivery person.

2.Distance Calculation:

Calculating the distance between the restaurant and delivery location. We have latitude and longitude coordinates of restaurant and delivery location, using Haversine function we calculated the respective distance.



```
import haversine as hs
```

```
new_df["Restaurant_latitude"]=new_df["Restaurant_latitude"].astype("str")
new_df["Restaurant_latitude"]=new_df["Restaurant_latitude"].str.replace("-","")
```

```
new_df["Delivery_location_latitude"]=new_df["Delivery_location_latitude"].astype("float")
new_df["Delivery_location_longitude"]=new_df["Delivery_location_longitude"].astype("float")
new_df["Restaurant_latitude"]=new_df["Restaurant_latitude"].astype("float")
new_df["Restaurant_longitude"]=new_df["Restaurant_longitude"].astype("float")
```

```
del_tup=tuple(map(lambda x,y: (x,y),new_df["Delivery_location_latitude"],new_df["Delivery_location_longitude"]))
```

```
res_tup=tuple(map(lambda x,y: (x,y),new_df["Restaurant_latitude"],new_df["Restaurant_longitude"]))
```

```
distance=list((map(lambda x,y: hs.haversine(x,y),res_tup,del_tup)))
```

```
distance_df=pd.DataFrame(distance,columns=["Dis_b/w_res_del_loc"])
```

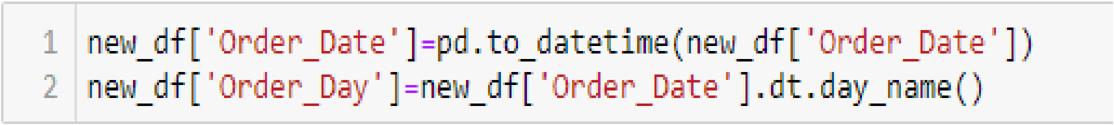
```
new_df=pd.concat([new_df,distance_df],axis=1)
```

Figure:02

The Haversine formula is a mathematical formula which is commanly used in geospatial calculations to calculate the distance between two points on the surface of a sphere, given their latitude and longitude coordinates. It is commonly used in navigation and geolocation applications.So,for our dataset we have used this method to calculate the distance.

3.**Week day Calculation:**

From Order date column we have changed datatype to datetime and created the respective week day column as Order day.



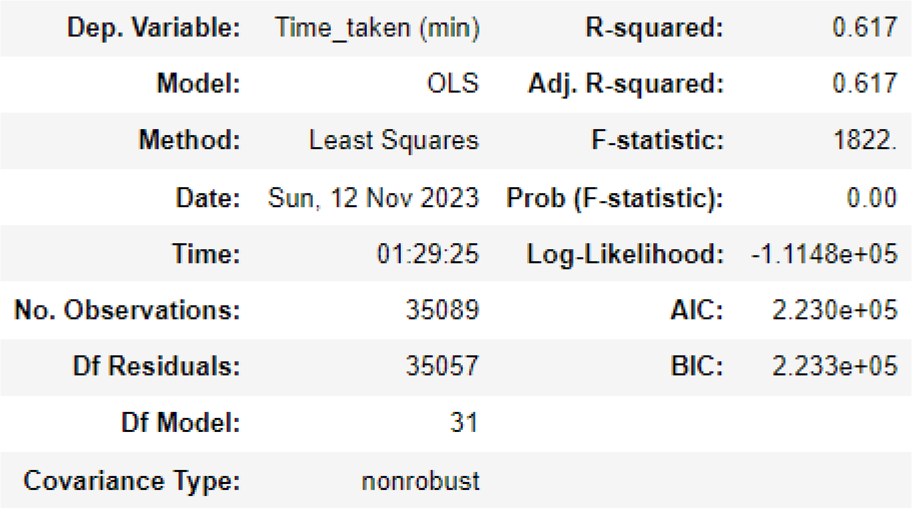
```
new_df['Order_Date']=pd.to_datetime(new_df['Order_Date'])
new_df['Order_Day']=new_df['Order_Date'].dt.day_name()
```

for the column ‘order_date’ we wanted to store date time information so we have used datetime datatype and for column ‘order_day’ we need to use the name of the day so we have used day_name() function.

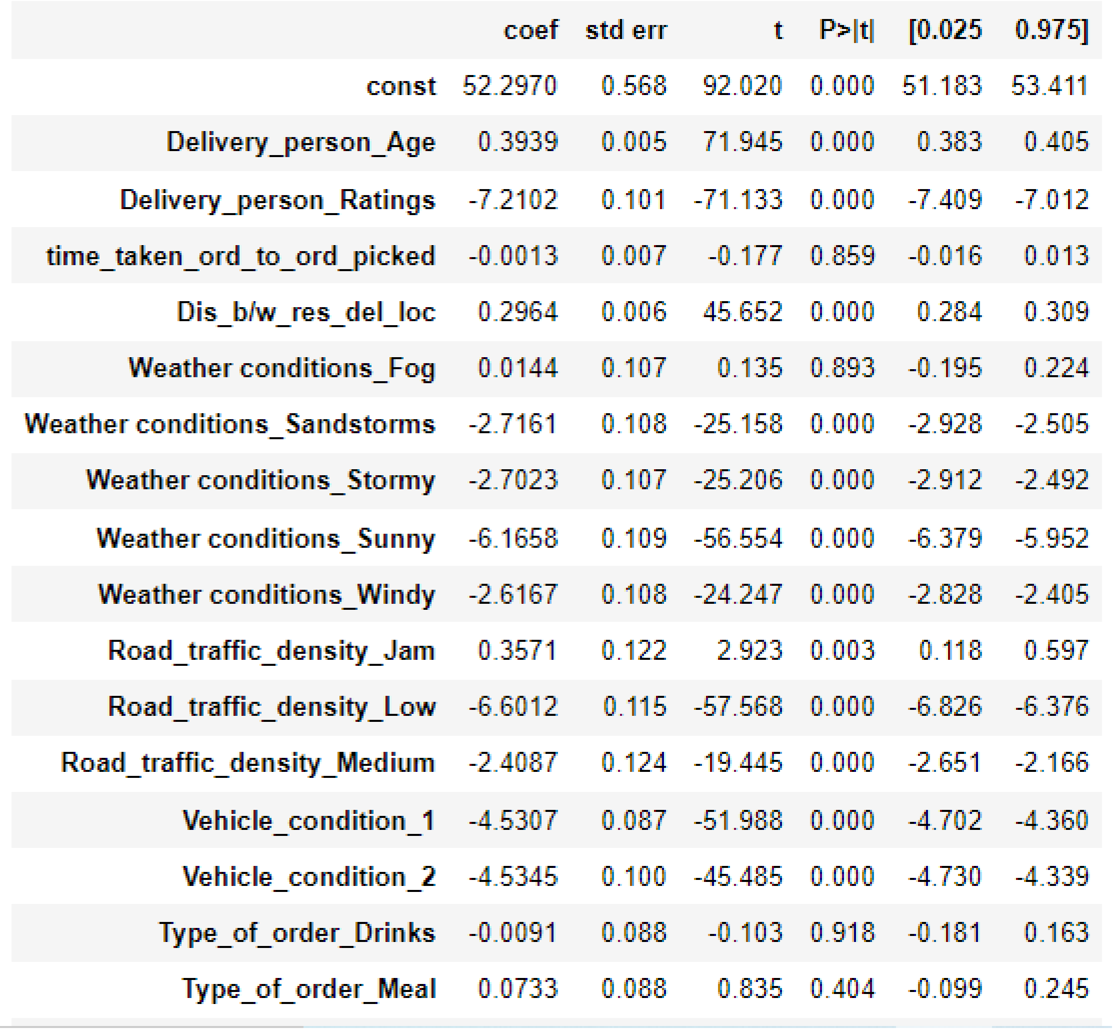
10.Base Model with raw data:

Base model was done on the data using OLS Regression model and SK learn Linear Regression model.

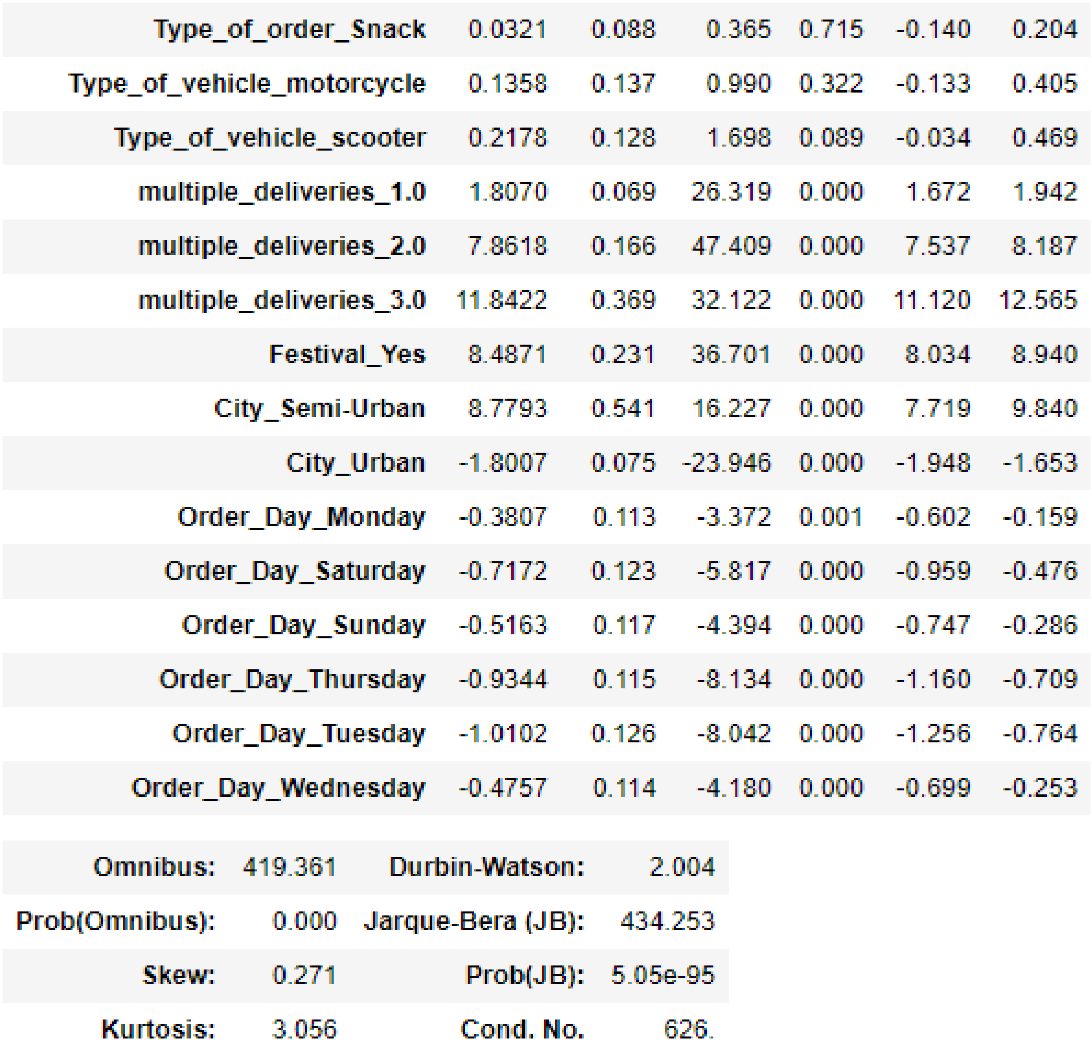
OLS Regression Model Performance:



Dep. Variable:	Time_taken (min)	**R-squared:**	0.617
Model:	OLS	**Adj. R-squared:**	0.617
Method:	Least Squares	**F-statistic:**	1822.
Date:	Sun, 12 Nov 2023	**Prob (F-statistic):**	0.00
Time:	01:29:25	**Log-Likelihood:**	-1.1148e+05
No. Observations:	35089	**AIC:**	2.230e+05
Df Residuals:	35057	**BIC:**	2.233e+05
Df Model:	31		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
const	52.2970	0.568	92.020	0.000	51.183	53.411
Delivery_person_Age	0.3939	0.005	71.945	0.000	0.383	0.405
Delivery_person_Ratings	-7.2102	0.101	-71.133	0.000	-7.409	-7.012
time_taken_ord_to_ord_picked	-0.0013	0.007	-0.177	0.859	-0.016	0.013
Dis_b/w_res_del_loc	0.2964	0.006	45.652	0.000	0.284	0.309
Weather conditions_Fog	0.0144	0.107	0.135	0.893	-0.195	0.224
Weather conditions_Sandstorms	-2.7161	0.108	-25.158	0.000	-2.928	-2.505
Weather conditions_Stormy	-2.7023	0.107	-25.206	0.000	-2.912	-2.492
Weather conditions_Sunny	-6.1658	0.109	-56.554	0.000	-6.379	-5.952
Weather conditions_Windy	-2.6167	0.108	-24.247	0.000	-2.828	-2.405
Road_traffic_density_Jam	0.3571	0.122	2.923	0.003	0.118	0.597
Road_traffic_density_Low	-6.6012	0.115	-57.568	0.000	-6.826	-6.376
Road_traffic_density_Medium	-2.4087	0.124	-19.445	0.000	-2.651	-2.166
Vehicle_condition_1	-4.5307	0.087	-51.988	0.000	-4.702	-4.360
Vehicle_condition_2	-4.5345	0.100	-45.485	0.000	-4.730	-4.339
Type_of_order_Drinks	-0.0091	0.088	-0.103	0.918	-0.181	0.163
Type_of_order_Meal	0.0733	0.088	0.835	0.404	-0.099	0.245

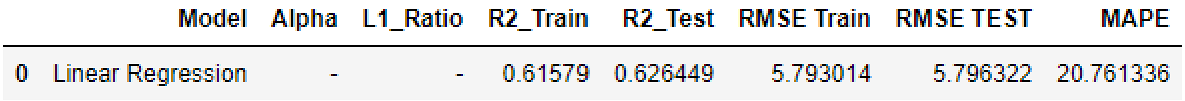


Type_of_order_Snack	0.0321	0.088	0.365	0.715	-0.140	0.204
Type_of_vehicle_motorcycle	0.1358	0.137	0.990	0.322	-0.133	0.405
Type_of_vehicle_scooter	0.2178	0.128	1.698	0.089	-0.034	0.469
multiple_deliveries_1.0	1.8070	0.069	26.319	0.000	1.672	1.942
multiple_deliveries_2.0	7.8618	0.166	47.409	0.000	7.537	8.187
multiple_deliveries_3.0	11.8422	0.369	32.122	0.000	11.120	12.565
Festival_Yes	8.4871	0.231	36.701	0.000	8.034	8.940
City_Semi-Urban	8.7793	0.541	16.227	0.000	7.719	9.840
City_Urban	-1.8007	0.075	-23.946	0.000	-1.948	-1.653
Order_Day_Monday	-0.3807	0.113	-3.372	0.001	-0.602	-0.159
Order_Day_Saturday	-0.7172	0.123	-5.817	0.000	-0.959	-0.476
Order_Day_Sunday	-0.5163	0.117	-4.394	0.000	-0.747	-0.286
Order_Day_Thursday	-0.9344	0.115	-8.134	0.000	-1.160	-0.709
Order_Day_Tuesday	-1.0102	0.126	-8.042	0.000	-1.256	-0.764
Order_Day_Wednesday	-0.4757	0.114	-4.180	0.000	-0.699	-0.253

Omnibus:	419.361	Durbin-Watson:	2.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	434.253
Skew:	0.271	Prob(JB):	5.05e-95
Kurtosis:	3.056	Cond. No.	626.

The R square value we got is low in this model. According to the F stat value we can say at least one feature is significant in building the model. With P stat value we can see the significant and insignificant features, P stat value of more than 0.05 are insignificant features. Here condition number is below 1000 only so there is no multicollinearity. Here Durbin Watson value is 2.004 and we can say that there is no auto correlation.

SK Learn Regression Model Performance:



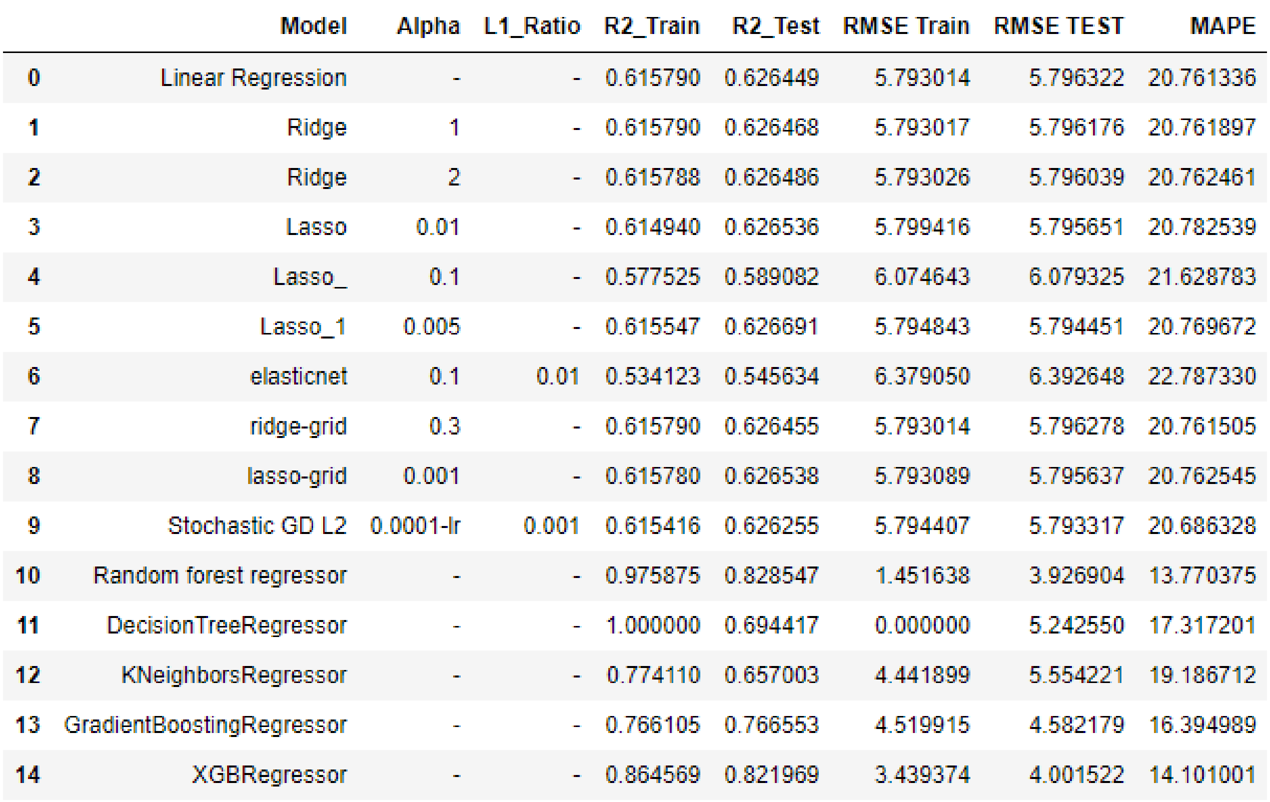
	Model	Alpha	L1_Ratio	R2_Train	R2_Test	RMSE Train	RMSE TEST	MAPE
0	Linear Regression	-	-	0.61579	0.626449	5.793014	5.796322	20.761336

R square value is low in the model. We will try to improve the R square and try to minimize the RMSE value also. As we can see linear regression models are not giving good scores we should try with nonlinear models to get better R square value and minimum error.

11.Final model

As our target variable is a numeric, the models that we considered for modelling are Random Forest Regressor, Gradient Boost Regressor, XG Boost Model, KNeighbors Regressor. We have treated and cleaned the data, so that many models can be applied and check the performance of each model. We have also tried Regularization with linear regression model.

We run the before mentioned models on data and hence evaluated the performance of each model which can be seen below clearly.

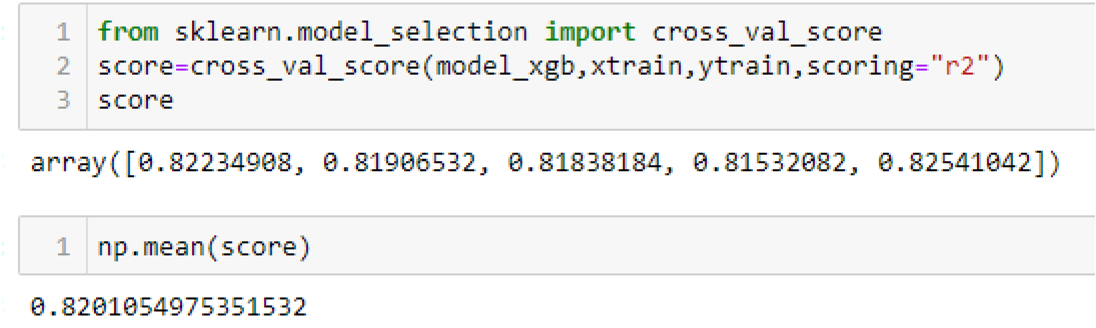


	Model	Alpha	L1_Ratio	R2_Train	R2_Test	RMSE Train	RMSE TEST	MAPE
0	Linear Regression	-	-	0.615790	0.626449	5.793014	5.796322	20.761336
1	Ridge	1	-	0.615790	0.626468	5.793017	5.796176	20.761897
2	Ridge	2	-	0.615788	0.626486	5.793026	5.796039	20.762461
3	Lasso	0.01	-	0.614940	0.626536	5.799416	5.795651	20.782539
4	Lasso_	0.1	-	0.577525	0.589082	6.074643	6.079325	21.628783
5	Lasso_1	0.005	-	0.615547	0.626691	5.794843	5.794451	20.769672
6	elasticnet	0.1	0.01	0.534123	0.545634	6.379050	6.392648	22.787330
7	ridge-grid	0.3	-	0.615790	0.626455	5.793014	5.796278	20.761505
8	lasso-grid	0.001	-	0.615780	0.626538	5.793089	5.795637	20.762545
9	Stochastic GD L2	0.0001-lr	0.001	0.615416	0.626255	5.794407	5.793317	20.686328
10	Random forest regressor	-	-	0.975875	0.828547	1.451638	3.926904	13.770375
11	DecisionTreeRegressor	-	-	1.000000	0.694417	0.000000	5.242550	17.317201
12	KNeighborsRegressor	-	-	0.774110	0.657003	4.441899	5.554221	19.186712
13	GradientBoostingRegressor	-	-	0.766105	0.766553	4.519915	4.582179	16.394989
14	XGBRegressor	-	-	0.864569	0.821969	3.439374	4.001522	14.101001

From the above table, We can see that XGBRegressor gives us the best Rsquare value. Its RMSE values are also minimum when compared to the other models. So we can choose XGBRegressor as the final model and we can implement cross validation, check feature importances and do hyper parameter tunning to increase the Rsquare value and minimize error further.

12. Cross-validation on selected model:

We have performed cross validation on the selected model to check the performance and cross validate on the basis of R square. We observed that the cross validation R square value is between the train and test R square for the model.



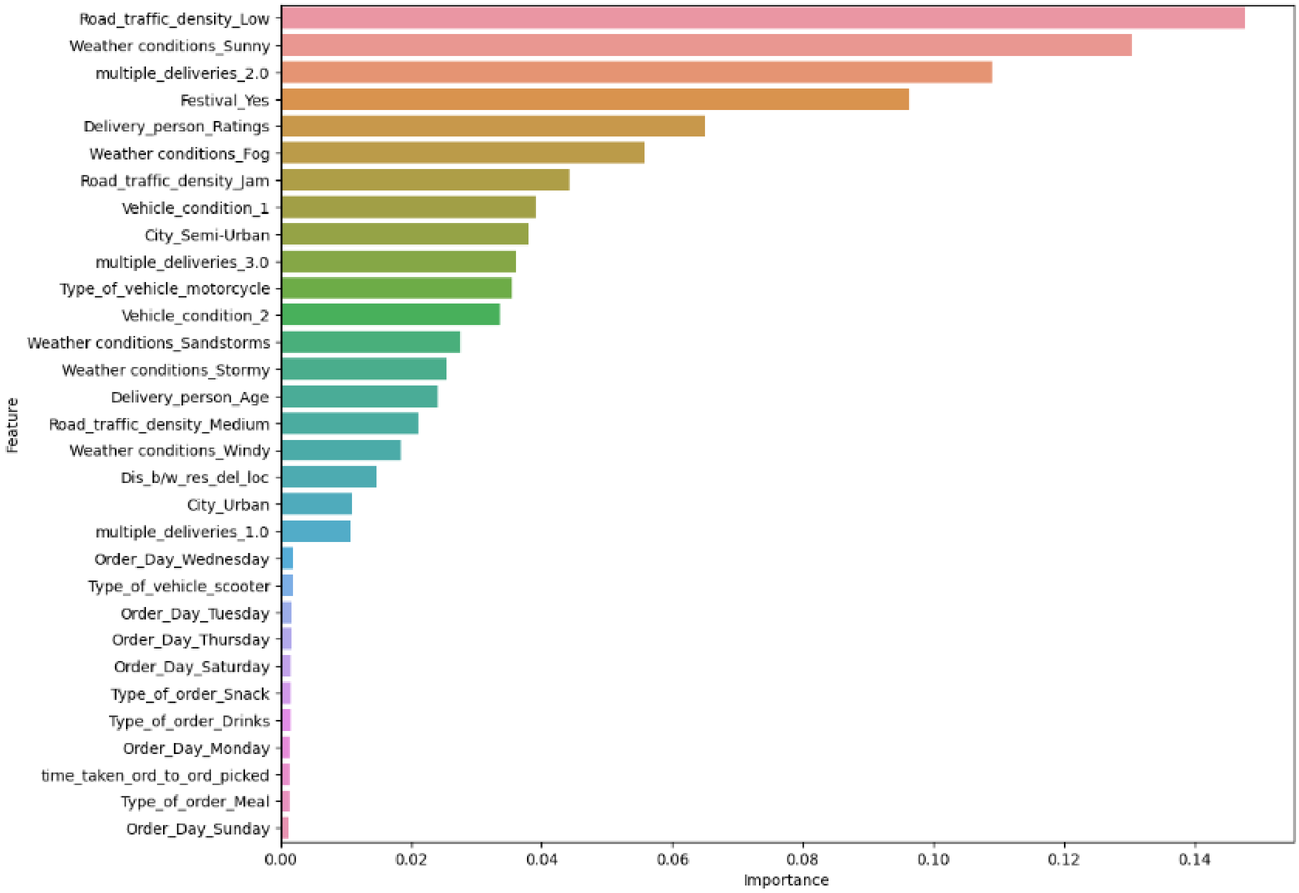
```
from sklearn.model_selection import cross_val_score
score=cross_val_score(model_xgb,xtrain,ytrain,scoring="r2")
score
```

```
array([0.82234908, 0.81906532, 0.81838184, 0.81532082, 0.82541042])
```

```
np.mean(score)
```

```
0.8201054975351532
```

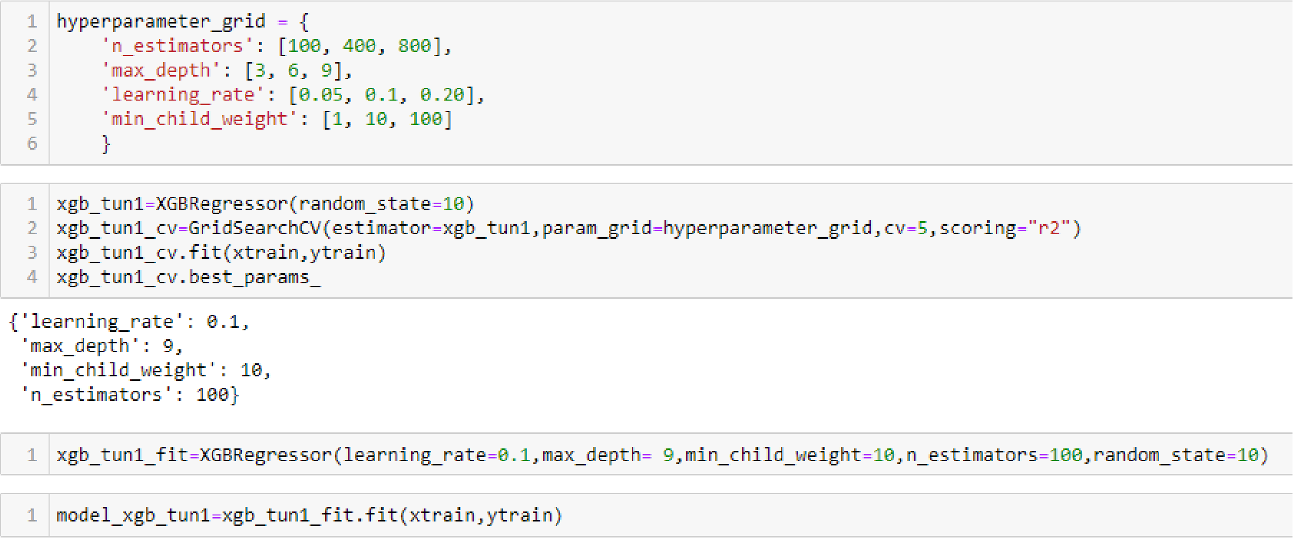
13.Feature Importance:



Top five important features are Road traffic density, Weather conditions, Multiple deliveries, Festival, Delivery person ratings.

14.Hyper parameter tuning:

We did hyper parameter tuning for the final selected model. We can see that the R square value has improved.



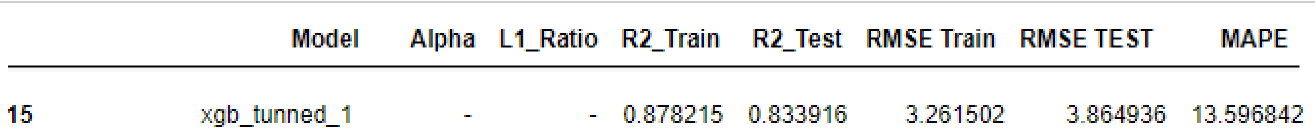
```
hyperparameter_grid = {
    'n_estimators': [100, 400, 800],
    'max_depth': [3, 6, 9],
    'learning_rate': [0.05, 0.1, 0.20],
    'min_child_weight': [1, 10, 100]
    }
```

```
xgb_tun1=XGBRegressor(random_state=10)
xgb_tun1_cv=GridSearchCV(estimator=xgb_tun1,param_grid=hyperparameter_grid,cv=5,scoring="r2")
xgb_tun1_cv.fit(xtrain,ytrain)
xgb_tun1_cv.best_params_
```

```
{'learning_rate': 0.1,
 'max_depth': 9,
 'min_child_weight': 10,
 'n_estimators': 100}
```

```
xgb_tun1_fit=XGBRegressor(learning_rate=0.1,max_depth= 9,min_child_weight=10,n_estimators=100,random_state=10)
```

```
model_xgb_tun1=xgb_tun1_fit.fit(xtrain,ytrain)
```



	Model	Alpha	L1_Ratio	R2_Train	R2_Test	RMSE Train	RMSE TEST	MAPE
15	xgb_tunned_1	-	-	0.878215	0.833916	3.261502	3.864936	13.596842

15. Conclusion:

From all the model we have tried on our data we have selected XGB Regressor model which gives us the best values. After selecting the model, we tuned it using hyper parameters. It gives the best R square value for train and test as 0.87 and 0.83. RMSE of train and test values are 3.26 and 3.86 which is the minimum.

16. Business Interpretation:

We can see the top features contributing to the model as follows:

Road Traffic Density:

Road traffic density is a critical feature as it directly influences the time taken for the delivery person to reach the destination. High traffic density can lead to delays, affecting the overall delivery time. Monitoring and incorporating this feature into the prediction model allows the system to adjust estimates based on real-time traffic conditions. This helps in setting accurate expectations for customers and optimizing delivery routes to minimize delays.

Weather Conditions:

Weather conditions play a significant role in the delivery process. Adverse weather, such as rain, snow, or storms, can impact transportation speed and increase the likelihood of delays. By considering weather conditions in the prediction model, the system can provide more accurate estimated delivery times during inclement weather. This enhances customer satisfaction by offering realistic delivery expectations, considering the challenges posed by weather.

Multiple Deliveries:

Handling multiple deliveries in a single trip is a common scenario in food delivery services. This feature indicates whether the delivery person has multiple orders to deliver on the same route. Efficiently managing multiple deliveries can lead to time savings, as the delivery person optimizes their route to fulfill all orders promptly. Incorporating this feature helps in predicting delivery times more accurately by accounting for the complexity introduced by multiple stops in a single trip.

Festival:

Festivals and special occasions often result in changes in traffic patterns, increased demand, and potential disruptions in regular operations. By including the festival feature, the model can adapt to the unique challenges posed during festive periods. This allows for more precise delivery time predictions, considering the impact of festivities on factors like traffic and order volume. Accurate predictions during festivals contribute to better customer satisfaction and service reliability.

Delivery Person Ratings:

The rating of the delivery person reflects their performance and reliability. High ratings suggest that the delivery person is efficient and trustworthy, likely to complete the delivery within the estimated time. Including this feature in the prediction model enables the system to account for the skill and reliability of the delivery person. Higher-rated delivery persons may be given more challenging routes or time-sensitive orders, contributing to improved overall service quality and customer satisfaction.

17.Implications:

Customer Communication: The model should facilitate proactive communication with customers. Real-time updates, notifications about delays, and clear communication regarding the reasons for delays (such as traffic or weather) can manage customer expectations and enhance their overall experience.

Continuous Monitoring and Feedback: Regularly monitoring and updating the model based on feedback and performance metrics are crucial. This ensures that the model remains adaptive to changing conditions, customer preferences, and delivery person performance.

Operational Efficiency: Implementing the model should lead to improved operational efficiency by optimizing delivery routes, reducing delays, and enhancing overall service reliability. Regular assessments of the model's impact on operational metrics can guide further refinements.

In summary, the implications revolve around optimizing the delivery process, improving customer communication, and enhancing overall operational efficiency based on the insights provided by the machine learning model. Regular adjustments and updates to the model and operational processes are essential to address evolving challenges in the dynamic environment of food delivery services.

18.Limitations:

Challenges include dependency on real-time and accurate data, potential oversight of dynamic environmental factors, incomplete feature coverage, assumptions about stable traffic patterns, and the risk of overfitting. Addressing these limitations requires continuous monitoring, adapting to evolving conditions, and considering ethical and privacy concerns in the implementation of the model.

19. Reference and Bibliography:

The references can be blogs, articles or even social media news relevant to explain the importance of the projects.

Radadiyamohit(2021)

Food delivery data recorded under different time and conditions

https://towardsdatascience.com/is-the-food-here-yet-f13a7bb0cd20

https://bytes.swiggy.com/predicting-food-delivery-time-at-cart-cda23a84ba63