Food Delivery Time Prediction

About the Dataset

Food delivery is a service where a restaurant, store, or independent food-delivery company brings food to a customer. Orders are usually placed through a restaurant or grocer's website or mobile app, or via a food ordering company. The delivered items can include entrees, sides, drinks, desserts, or grocery items, and are typically packaged in boxes or bags. Delivery personnel usually drive cars, but in larger cities where homes and restaurants are closer together, they may use bikes or motorized scooters.

Data given is in the train and test csv files. I have combined the data to perform the data cleaning on it.

```
#join train n test to clean the data
df = pd.concat([df_test.assign(indic="test"), df_train.assign(indic="train")])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 56992 entries, 0 to 45592
Data columns (total 21 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   ID                           56992 non-null  object 
 1   Delivery_person_ID           56992 non-null  object 
 2   Delivery_person_Age          54647 non-null  object 
 3   Delivery_person_Ratings      54577 non-null  object 
 4   Restaurant_latitude          56992 non-null  float64
 5   Restaurant_longitude         56992 non-null  float64
 6   Delivery_location_latitude   56992 non-null  float64
 7   Delivery_location_longitude  56992 non-null  float64
 8   Order_Date                   56992 non-null  object 
 9   Time_Orderd                  54817 non-null  object 
 10  Time_Order_picked            56992 non-null  object 
 11  Weatherconditions            56218 non-null  object 
 12  Road_traffic_density         56237 non-null  object 
 13  Vehicle_condition            56992 non-null  int64  
 14  Type_of_order                56992 non-null  object 
 15  Type_of_vehicle              56992 non-null  object 
 16  multiple_deliveries          55761 non-null  object 
 17  Festival                     56699 non-null  object 
 18  City                         55468 non-null  object 
 19  indic                        56992 non-null  object 
 20  Time_taken(min)              45593 non-null  object 
```

Link to Dataset: https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset

Aim

To predict the delivery time (in minutes) for a food delivery app using regression models applied to the Food Delivery Dataset from Kaggle.

Data Cleaning

1. Removed the string part from **Weatherconditions** & **Time_taken(min)**

2. Changed datatype of quantitative data to float and Date data to datetime

3. Creating new variables from Geographic data

    1. Calculated **Distance** from geographic columns 'Restaurant_latitude','Restaurant_longitude','Delivery_location_latitude','Delivery_location_longitude'
    2. Calculated **order_preparation_time** from order_placed and order_picked

4. Imputed missing data with sklearn's SimpleImputer method

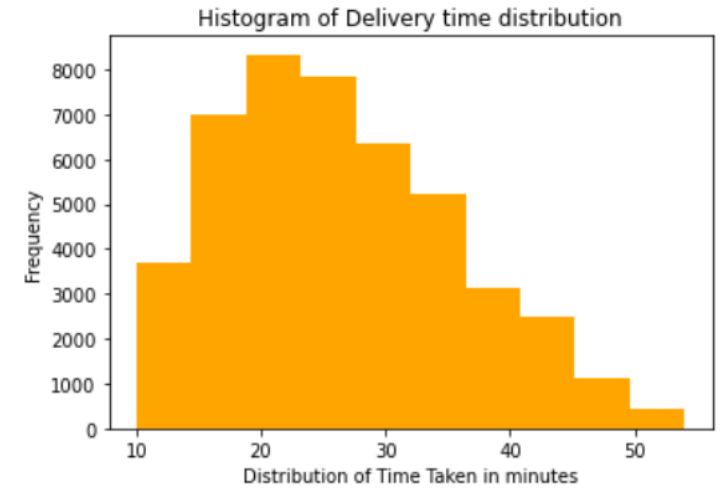
    1. Imputed categorical data with Mode

    2. Imputed Numerical data with Median

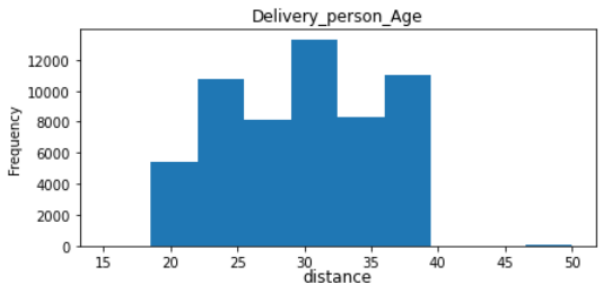
Continuous	Discrete	Categorical
Delivery_person_Age	Vehicle_condition	Weatherconditions
Delivery_person_Ratings	order_preparation_time	Road_traffic_density
distance	multiple_deliveries	Type_of_order
		Type_of_vehicle
		Festival
		City

```
Data columns (total 25 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   ID                           56992 non-null  object
 1   Delivery_person_ID           56992 non-null  object
 2   Delivery_person_Age          54647 non-null  float64
 3   Delivery_person_Ratings      54577 non-null  float64
 4   Restaurant_latitude          56992 non-null  float64
 5   Restaurant_longitude         56992 non-null  float64
 6   Delivery_location_latitude   56992 non-null  float64
 7   Delivery_location_longitude  56992 non-null  float64
 8   Order_Date                   56992 non-null  datetime64[ns]
 9   Time_Orderd                  54817 non-null  timedelta64[ns]
 10  Time_Order_picked            56992 non-null  timedelta64[ns]
 11  Weatherconditions            56218 non-null  object
 12  Road_traffic_density         56237 non-null  object
 13  Vehicle_condition            56992 non-null  float64
 14  Type_of_order                56992 non-null  object
 15  Type_of_vehicle              56992 non-null  object
 16  multiple_deliveries          55761 non-null  object
 17  Festival                     56699 non-null  object
 18  City                         55468 non-null  object
 19  indic                        56992 non-null  object
 20  Time_taken(min)              45593 non-null  float64
 21  distance                     56992 non-null  float64
 22  Time_Order_picked_formatted  56992 non-null  datetime64[ns]
 23  Time_Ordered_formatted       54817 non-null  datetime64[ns]
 24  order_preparation_time       54817 non-null  float64
```

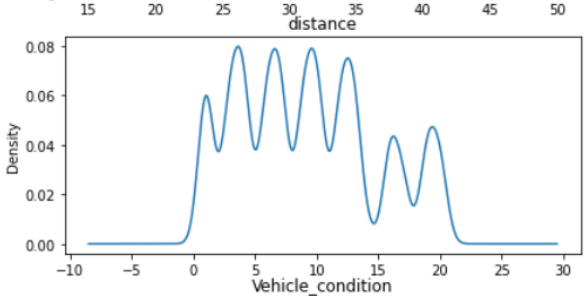
Exploratory Data Analysis

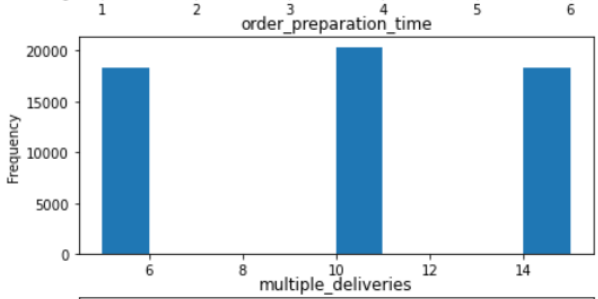


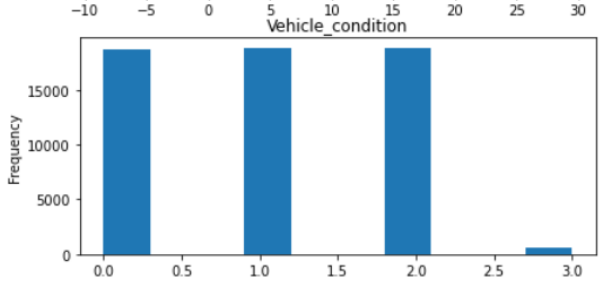
The distribution increases rapidly, peaks between 20 and 30 minutes, and then gradually declines.

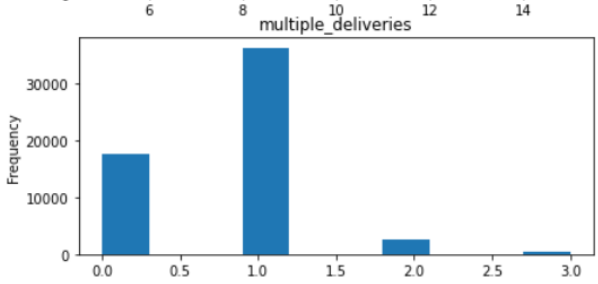

Delivery_person_Age
Frequency
12000
10000
8000
6000
4000
2000
0
15
20
25
30
35
40
45
50

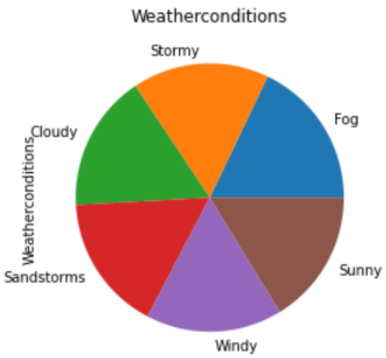

Delivery_person_Ratings
Frequency
40000
30000
20000
10000
0
1
2
3
4
5
6

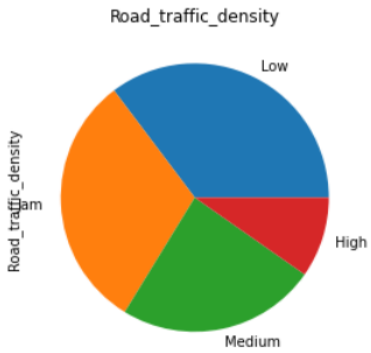

distance
Density
0.08
0.06
0.04
0.02
0.00
−10
−5
0
5
10
15
20
25
30

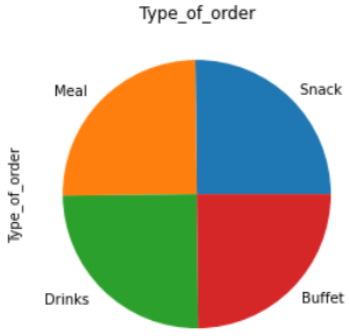

order_preparation_time
Frequency
20000
15000
10000
5000
0
6
8
10
12
14

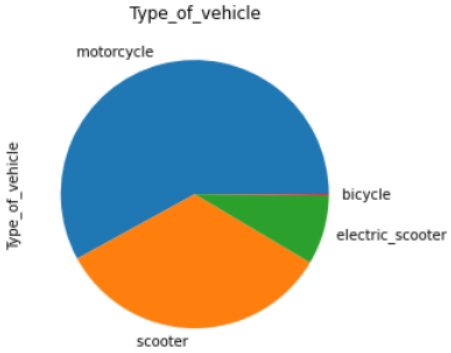

Vehicle_condition
Frequency
15000
10000
5000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0


multiple_deliveries
Frequency
30000
20000
10000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Weatherconditions
Stormy
Fog
Cloudy
Weatherconditions
Sandstorms
Sunny
Windy


Road_traffic_density
Low
Road_traffic_density
Jam
High
Medium

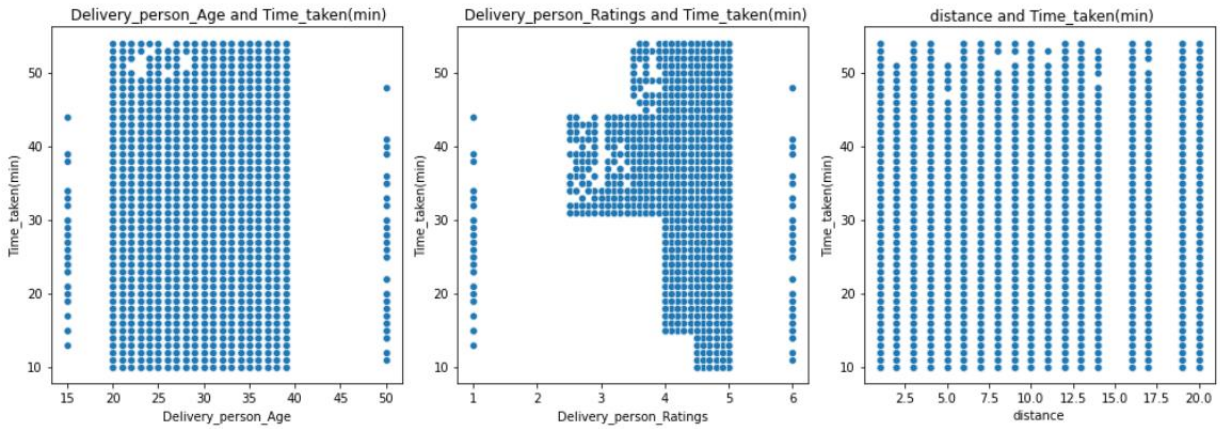

Type_of_order
Meal
Snack
Type_of_order
Drinks
Buffet


Type_of_vehicle
motorcycle
Type_of_vehicle
bicycle
electric_scooter
scooter

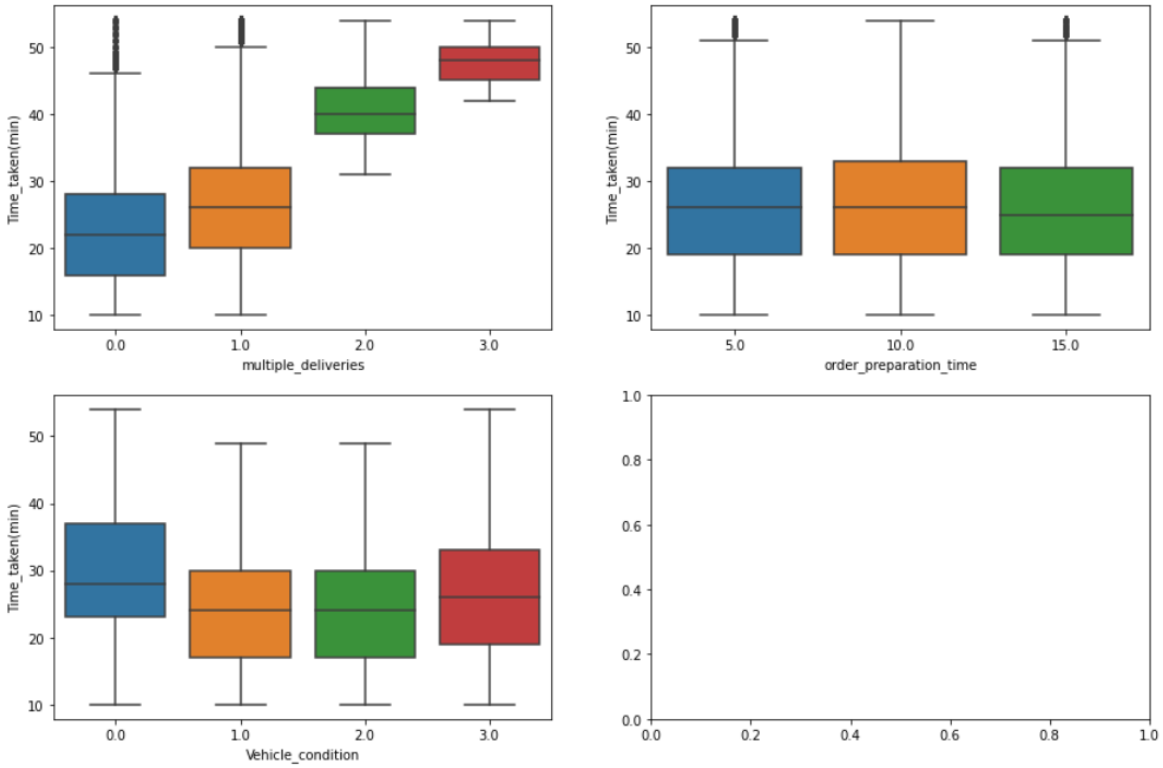
From the graphs, we can observe that the majority of data points occur under the conditions of road traffic jams, the use of motorcycles, metropolitan city locations, and non-festival days.

Visualizing Target Variable (Time_taken(min)) with Continuous, Discrete and Categorical Data.

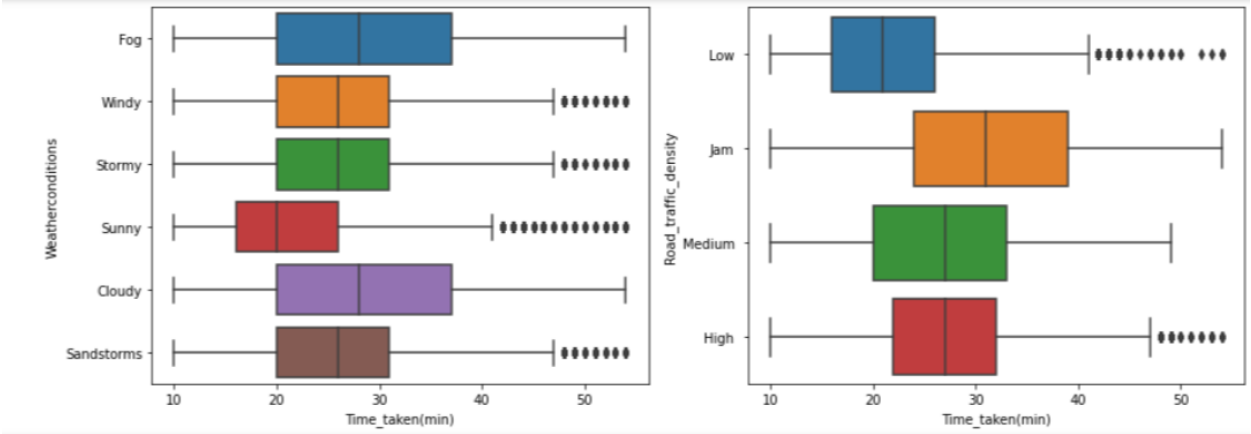
1. Relationship between timetaken and Continuous data

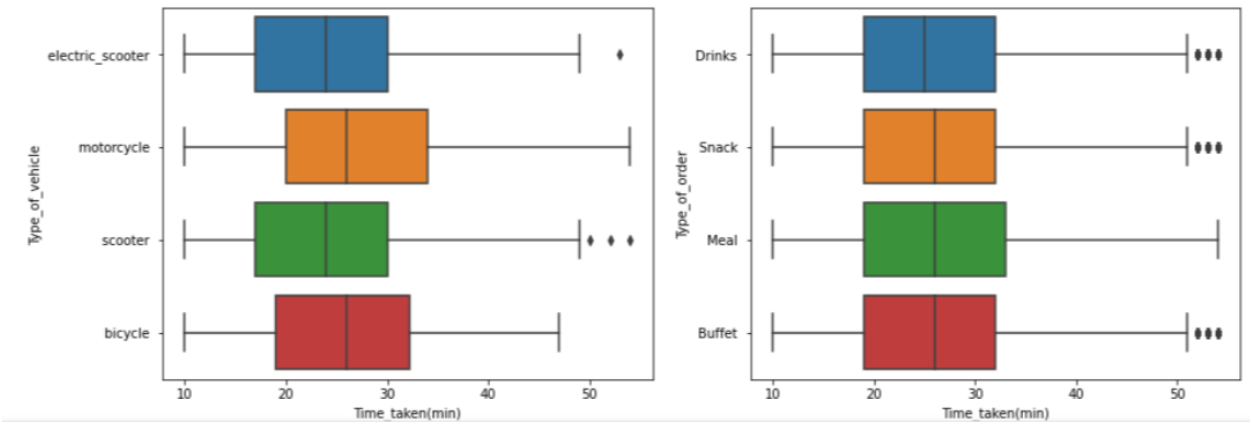


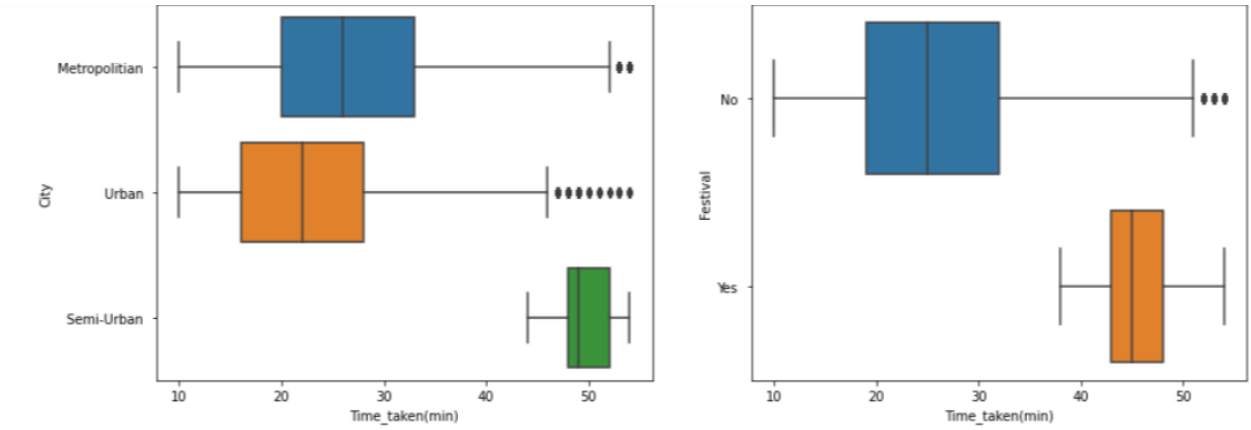
2. Relationship between timetaken and Continuous data



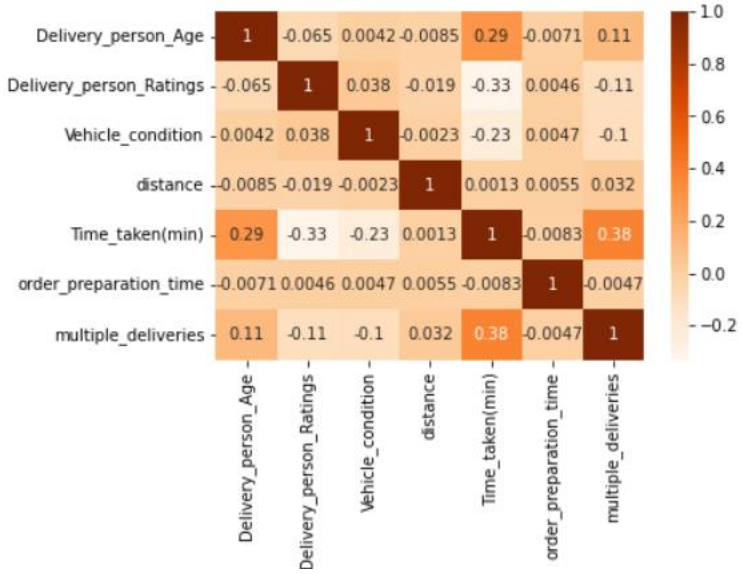
3. Relationship between time taken and Categorical variables







4. Correlation Heat Map



Upon analyzing the variables, there appears to be limited variation in delivery time based on the continuous variables. The graphs indicate that delivery time is primarily influenced by the following features:

1. Multiple Deliveries
2. Weather Condition
3. Road Traffic
4. City
5. Festival
6. Type of Vehicle

Additionally, features such as Vehicle Condition and Order Preparation Time exhibit similar means but different distributions. Overall, the data does not show any strong linear

correlation with the target variable, suggesting that other factors or non-linear relationships might be at play.

5. Converting categorical data to binary using OneHotEncoder

```
Data columns (total 24 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Delivery_person_Age     45593 non-null  float64
 1   Delivery_person_Ratings 45593 non-null  float64
 2   Vehicle_condition       45593 non-null  float64
 3   distance                45593 non-null  float64
 4   Time_taken(min)         45593 non-null  float64
 5   order_preparation_time  45593 non-null  float64
 6   multiple_deliveries     45593 non-null  float64
 7   Fog                     45593 non-null  float64
 8   Sandstorms              45593 non-null  float64
 9   Stormy                  45593 non-null  float64
 10  Sunny                   45593 non-null  float64
 11  Windy                   45593 non-null  float64
 12  Jam                     45593 non-null  float64
 13  Low                     45593 non-null  float64
 14  Medium                  45593 non-null  float64
 15  Drinks                  45593 non-null  float64
 16  Meal                    45593 non-null  float64
 17  Snack                   45593 non-null  float64
 18  electric_scooter        45593 non-null  float64
 19  motorcycle              45593 non-null  float64
 20  scooter                 45593 non-null  float64
 21  festival                45593 non-null  float64
 22  Semi-Urban              45593 non-null  float64
 23  Urban                   45593 non-null  float64
dtypes: float64(24)
```

Splitting Data Back to train and test data

```
test, train = df_transformed[df_transformed["indic"].eq("test")], df_transformed[df_transformed["indic"].eq("train")]
```

Feature Selection:

Utilizing all 24 features may result in overfitting and inaccurate predictions. Moreover, it can be computationally intensive. Therefore, effective feature selection is crucial. I will employ three methods for this purpose:

1. **Random Forest Importance**: Features with higher importance scores are considered more influential in predicting the target variable.
   Features:
   delivery_person_ratings,delivery_person_age,multiple,deliveries,Vehicle_condition,low,sunny,distance

2. **Lasso Regression**: Features with non-zero coefficients after regularization (lasso shrinkage) are retained.
   Features: multiple deliveries,jam,festival,semi-urban,delivery perosn age,fog
3. **Fisher's Score**: Features with higher scores are more likely to be relevant for predicting the target variable.
   Features: Delivery_person_Age ,multiple_deliveries ,Low ,Delivery_person_Ratings ,Vehicle_condition ,Sunny,Urban ,Jam
4. **Mutual Information**: Features with higher mutual information scores have a stronger relationship with the target variable.
   Features: Delivery_person_Age ,Delivery_person_Ratings, multiple_deliveries, Jam , Low

Based on the above I have selected 7 features: Delivery_person_Age, Delivery_person_Ratings,Vehicle_condition,multiple_deliveries,Jam,Low,festival

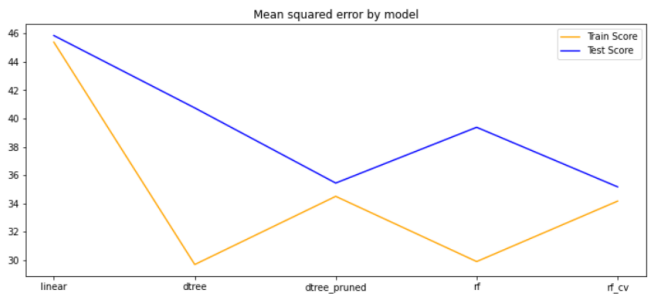
Building Models

1. **Multiple Linear Regression**
   R-squared is used to assess the goodness-of-fit of a regression model. Higher R-squared values indicate a better fit, meaning that the model's predictions are closer to the actual observed values.
   R-squared : 0.4826304013277636

2. **Decision Trees**
   Tree Depth 24
   Terminal Nodes 4894
3. **Random Forest**

Comparing Models MSE

```
Mean Squared Error (MSE) by Model:
          Method  Train MSE   Test MSE
0         linear  45.394865  45.860002
1          dtree  29.686422  40.746305
2  dtree_pruned  34.500937  35.439211
3             rf  29.894493  39.379055
4          rf_cv  34.164175  35.169980
```



The Mean Squared Error (MSE) values provide insights into the predictive performance of each model for estimating delivery time. Here's what we can infer from the MSE results:

Linear Regression(linear) : Train MSE:45.39, Test MSE: 45.86

Interpretation: The linear regression model exhibits relatively high MSE values, suggesting it may struggle to capture the complexities in the data adequately. The higher test MSE compared to train MSE indicates potential overfitting.

Decision Tree (dtree): Train MSE: 29.69, Test MSE: 40.75

Interpretation: The decision tree model shows a significant drop in train MSE, indicating good fit to the training data. However, the higher test MSE suggests overfitting, meaning the model may not generalize well to new data.

Pruned Decision Tree(dtree_pruned) : Train MSE: 34.50, Test MSE: 35.44

Interpretation: Pruning the decision tree reduces overfitting compared to the unpruned version, as evidenced by the closer train and test MSE values. This suggests improved generalization ability while maintaining a competitive performance level.

Random Forest(rf): Train MSE: 29.89, Test MSE: 39.38

Interpretation: The random forest model performs similarly to the decision tree on the train set but exhibits better generalization on the test set. However, there is still some indication of overfitting, as the test MSE is higher than the train MSE.

Cross-Validated Random Forest(rf_cv): Train MSE: 34.16, Test MSE: 35.17

Interpretation: Cross-validation with random forest demonstrates consistent performance between train and test MSE, indicating improved generalization and reduced overfitting compared to non-cross-validated models.

RMSE=35.17≈5.93 ≈ 6

Overall Conclusion:

The Random Forest model achieved the lowest test MSE of 35.17 after hyperparameter tuning, indicating an average prediction error of approximately 6 minutes in delivery time. Such a margin of error can significantly impact operational efficiency and service quality for a delivery firm, highlighting the importance of further optimizing the model to reduce this discrepancy.