Sharif University of Technology

Phase One of the Transportation Planning Course Project

Authors:

Amirhossein Pourkarimi

Mahshad Ebrahimi

Ali Bozorgmehry

Course Instructor:

Professor Hasan Naeibi

Winter 2024

Request 1 – Data Preprocessing

1. In this section, columns that were irrelevant to the analysis were removed from the dataset.
2. In this section, for rows where tips had been paid, the payment method was set to "credit card. We assumed that the value 1 in the "Payment Type" column represented credit card payments, and those rows were categorized accordingly.
3. A new column named "Dispatch Type" was added for the trip type, containing binary values (0 and 1) to indicate whether each trip belonged to this category or not. For unclear payment types, the value 6 was assigned, categorizing them as a distinct new type.
4. All rows with empty VendorID values were removed from the dataset as they did not provide any meaningful information.
5. The dataset exclusively contains data from the year 2021, with data from other years excluded from the analysis.

In addition to the tasks mentioned above, further steps were carried out to clean and prepare the dataset. These actions will be explained in more detail in the following sections.

6. Removal of inconsistent or noisy data: Certain columns in this dataset contained inconsistent and noisy data that were logically invalid. For instance, rows with zero or negative values in the "Total Payment Amount" column were deleted. Similarly, rows with trip distances of zero or negative values were removed. Additionally, the "Tip Amount" column included negative values, which were also eliminated. These steps ensured that the resulting dataset adhered to logical standards and was compatible for analysis.
7. The dataset was examined for duplicate rows, and any duplicate entries were removed to maintain the integrity and uniqueness of the data.
8. After completing all tasks from the first section, the percentage of missing or incomplete data was evaluated. Referring to the chart provided, three columns "Trip Type," "Payment Type," and "Passenger Count"—showed a missing data percentage exceeding 5%. When the percentage of missing data exceeds 5%, it needs to be filled using suitable algorithms. In this case, the Random Forest algorithm was utilized to predict and fill in missing rows for these three columns.

It is worth noting that none of these three columns influenced one another in the prediction process, ensuring unbiased filling of the missing values.

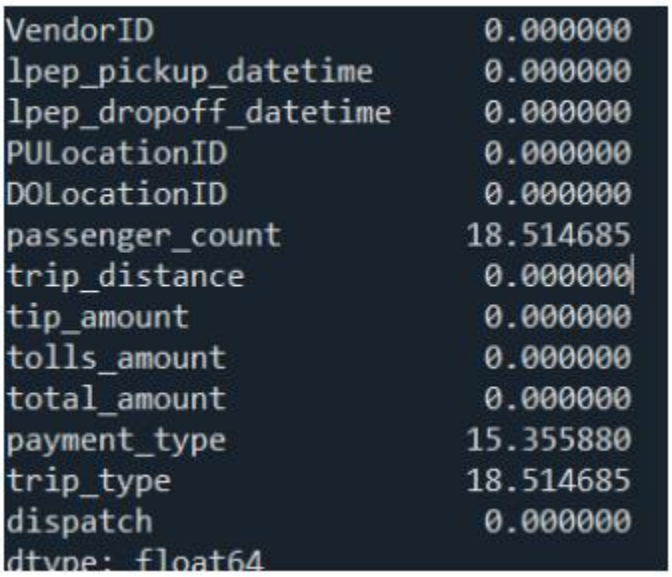


Figure 1 The amount of missing data in each column.

Second requirement – Data description and interpretation

This heat map shows the distribution of trip numbers based on weekdays and hours of the day. During workdays (Monday to Friday), most trips occur during early morning hours (6–9) and evening hours (16–19), indicating commuting to workplaces or schools and returning from them. In contrast, during weekends (Saturday and Sunday), trips are more dispersed and occur mostly during midday hours, likely related to recreational activities or shopping. Additionally, across all days, the number of trips between midnight and early morning (0–5) is significantly lower. These patterns reflect the impact of people's daily schedules on trip behavior.

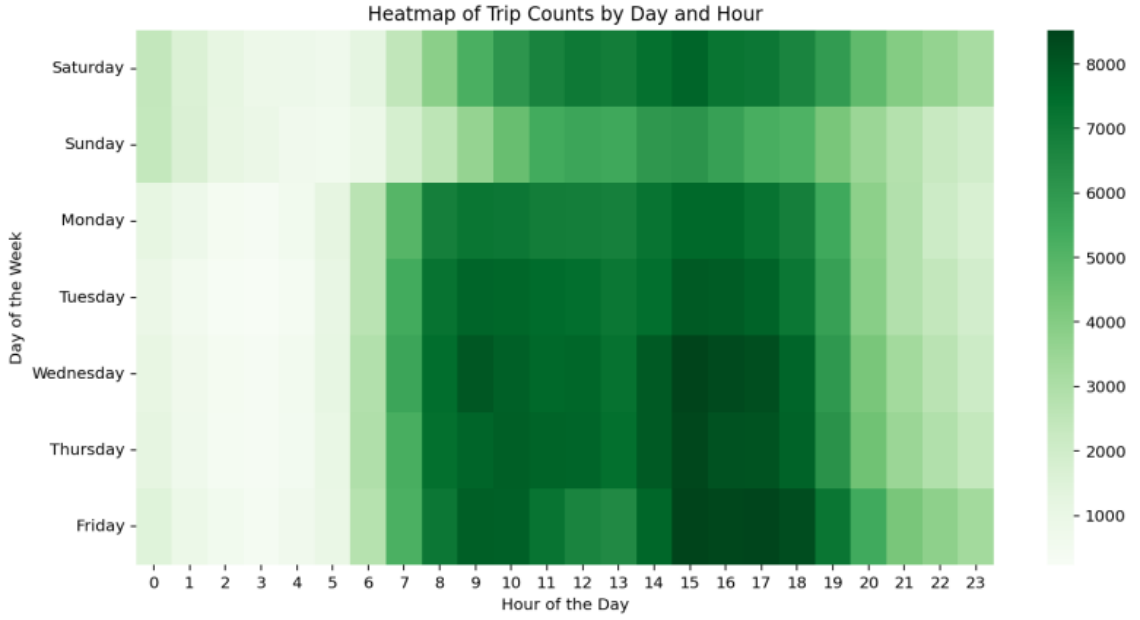


Figure 2 Heatmap of weekdays and hours of the day.

In this section, charts have been created to analyze passenger payment trends over time, trip type trends, trends in the use of green taxis in different areas of the city, and usage trends during different time intervals throughout the day. Each will be examined further below.

This chart represents the number of different payment types for the taxi company over the time period from January 2021 to January 2022. Payments are divided into six distinct types, which demonstrate significant fluctuations during this period. The highest number of payments corresponds to Type 1, which saw a notable increase at the beginning of 2021 and peaked during the spring months. After a decline in summer, this payment type rose again during autumn. Type 2 also gradually

increased, reaching its highest level in the second half of 2021. Other payment types showed fewer changes and remained relatively stable. These fluctuations may indicate seasonal changes in taxi demand, the impact of specific events, and shifts in customer behavior. This payment trend was analyzed across various dates and months.

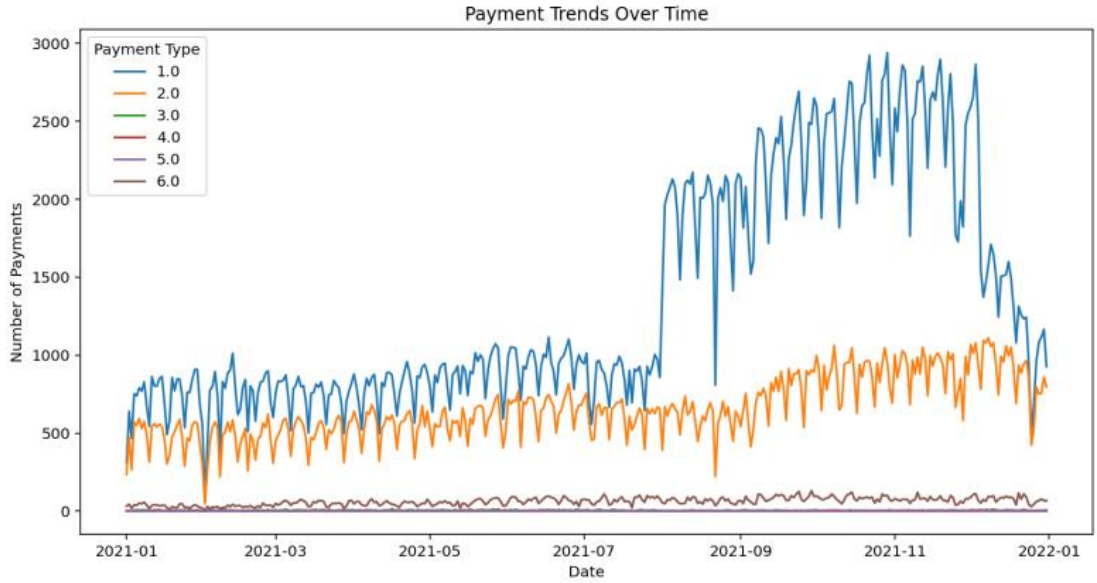


Figure 3 Chart of payment trends across different dates.

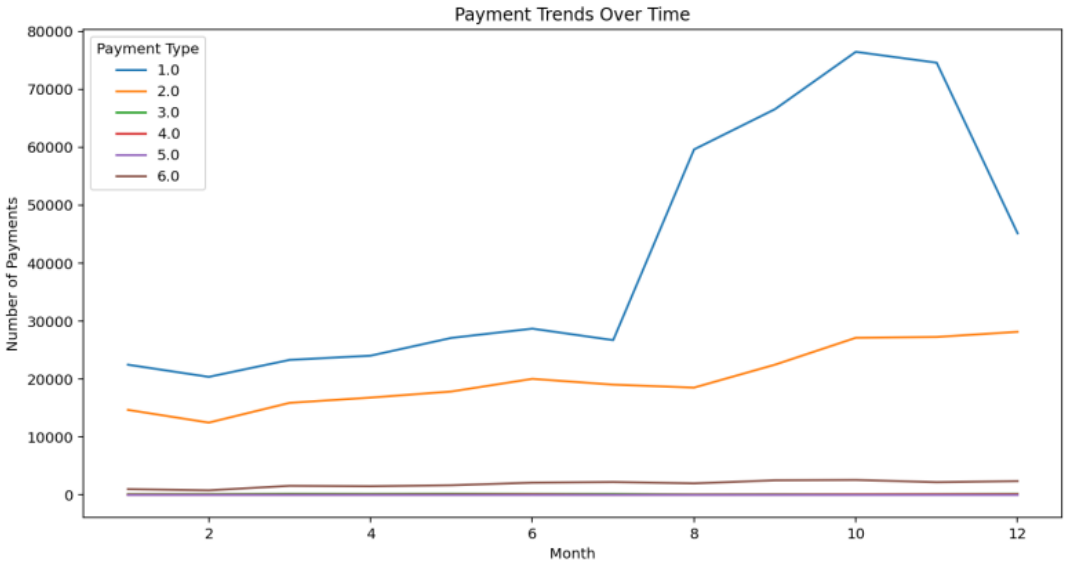


Figure 4 Chart of payment trends across different months.

The horizontal axis represents the date, and the vertical axis shows the number of trips. Three different types of trips are displayed using blue, orange, and green colors. The number of Type 1 trips (orange line) is consistently higher than the other types and has sharply increased since the middle of 2021. Type 0 trips (blue line) are fewer than Type 1 trips, and after a significant increase in the middle of the year, they decrease at the end of the year. Type 2 trips (green line) have the lowest numbers and demonstrate a relatively stable trend throughout the year. This chart indicates that Type 1 trips have had the most fluctuations and increases throughout the year, while the other types of trips have remained more stable. The sudden spike in Type 0 trips in the tenth month may result from a specific event or change. Due to the low numbers and stability, Type 2 trips may require optimization and modifications. The trip trends are analyzed daily and monthly.

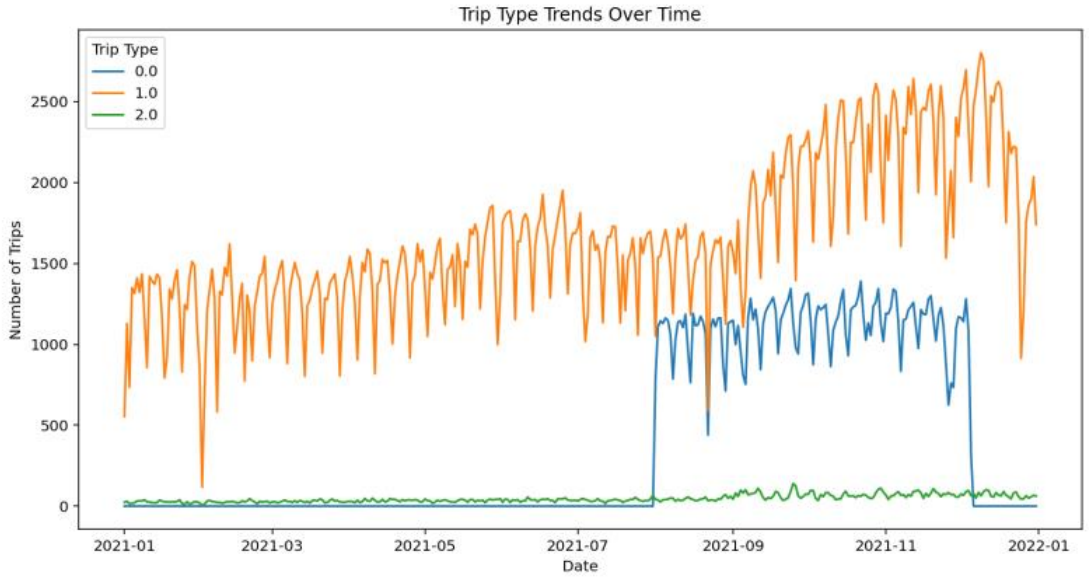


Figure 5 Chart of trip trends across different dates.

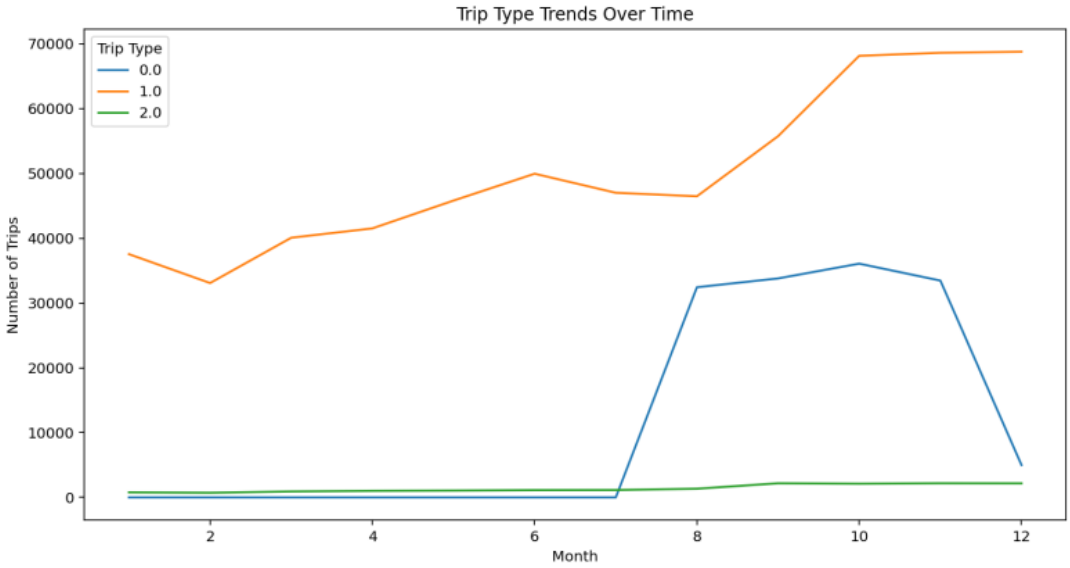


Figure 6 Chart of trip trends across different months.

The highest number of pickups is reported in the Manhattan area, with nearly 400,000 pickups. This indicates a high demand for taxis in this densely populated and busy region. Queens and Brooklyn follow with approximately 150,000 and 130,000 pickups, reflecting a moderate need for taxis in these areas. The Bronx and Staten Island have lower pickup counts, while EWR has the least number of pickups. This information can aid in more optimal resource allocation and better planning for services in high-demand areas like Manhattan. The analysis of drop-offs, or in other words, destinations, follows a similar pattern.

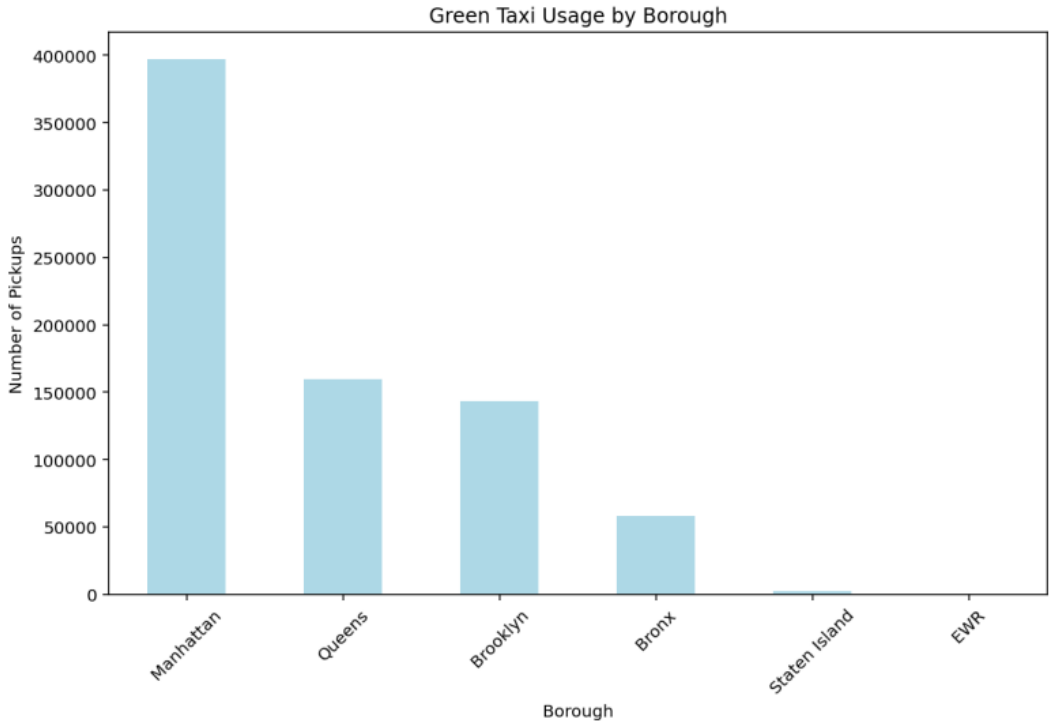


Figure 7 Chart of trip origins in different regions.

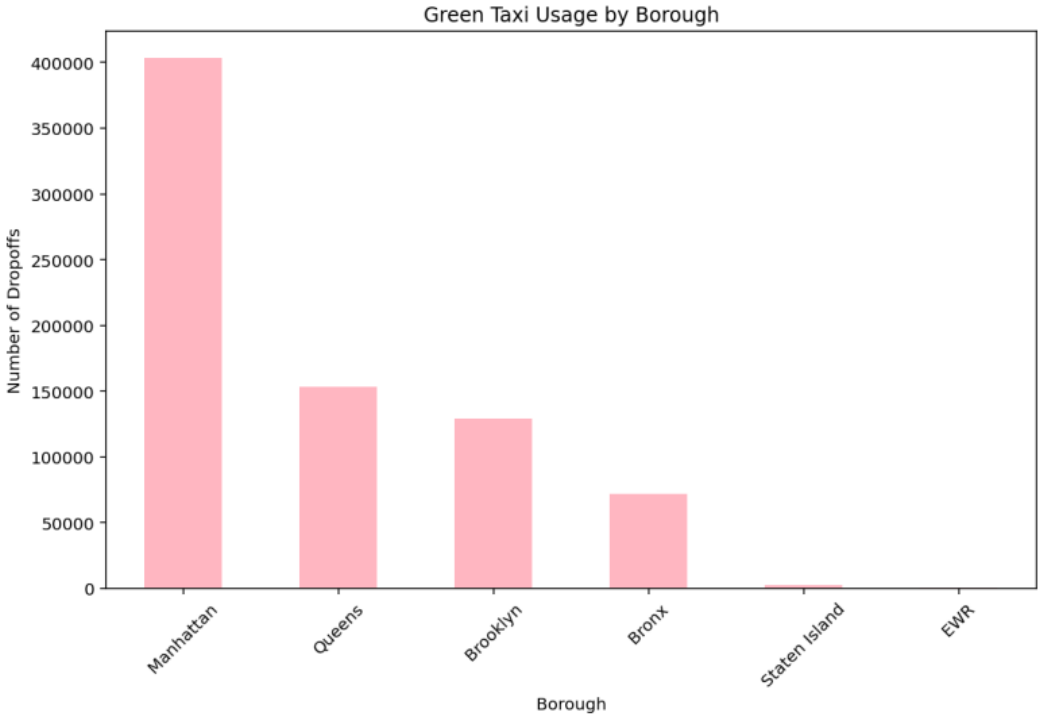


Figure 8 Chart of trip destinations in different regions.

This line chart illustrates the trend of green taxi usage across various regions of New York from January 2021 to January 2022. Manhattan, with significant increases and notable fluctuations, has the highest number of pickups (trip origins) and consistently sees the most usage of green taxis. Queens and Brooklyn also show an upward trend but have fewer pickups compared to Manhattan. The Bronx, EWR, and Staten Island demonstrate more stable pickup numbers, experiencing fewer variations over time. These patterns indicate that the demand for green taxis in Manhattan is considerably higher, with the impact of seasonal fluctuations and other factors being more pronounced in this area, while other regions show less variability.

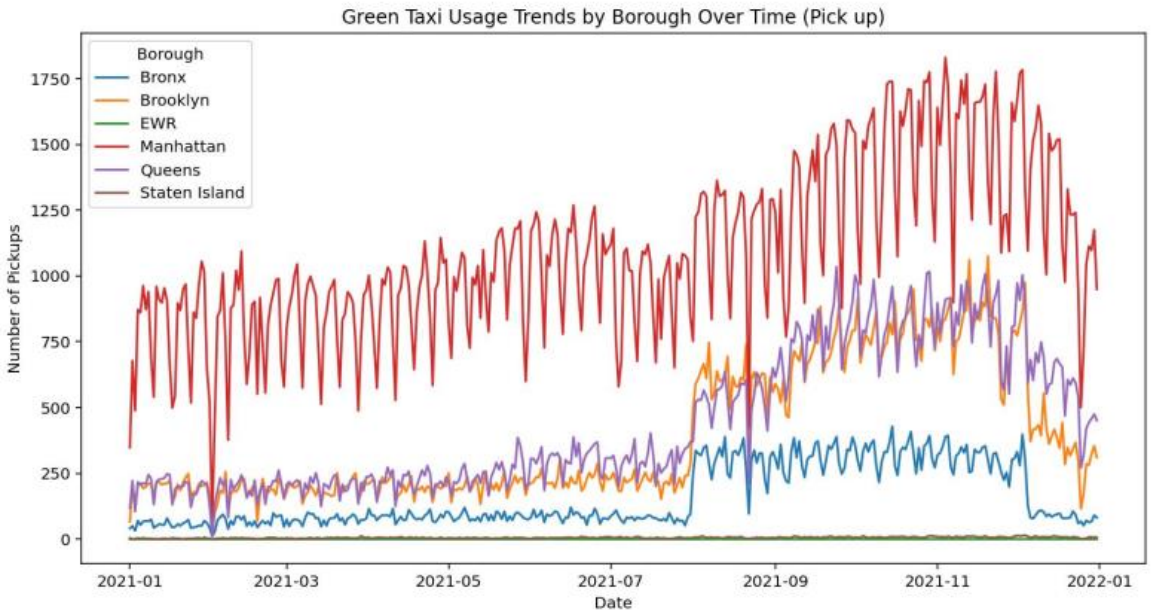


Figure 9 Chart of trip origins across different dates.

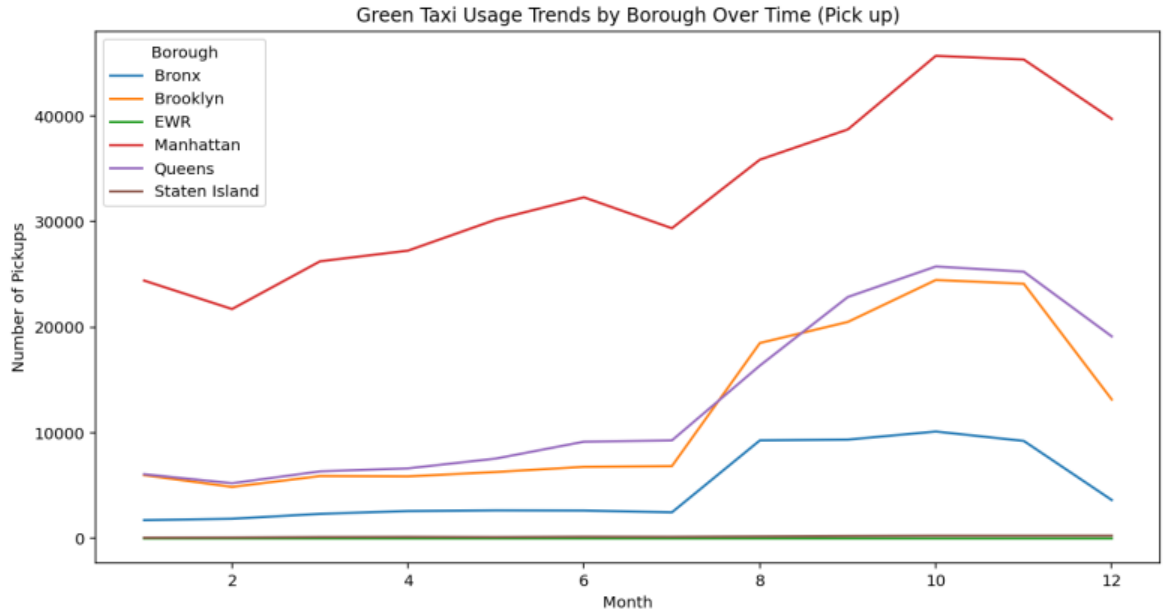


Figure 10 Chart of trip origins across different months.

These patterns, similar to the interpretation of the pickup section, indicate that the demand for green taxis in Manhattan is significantly higher and is influenced by seasonal fluctuations and other factors. In contrast, other regions show fewer variations, suggesting that the use of green taxis in these areas is more stable.

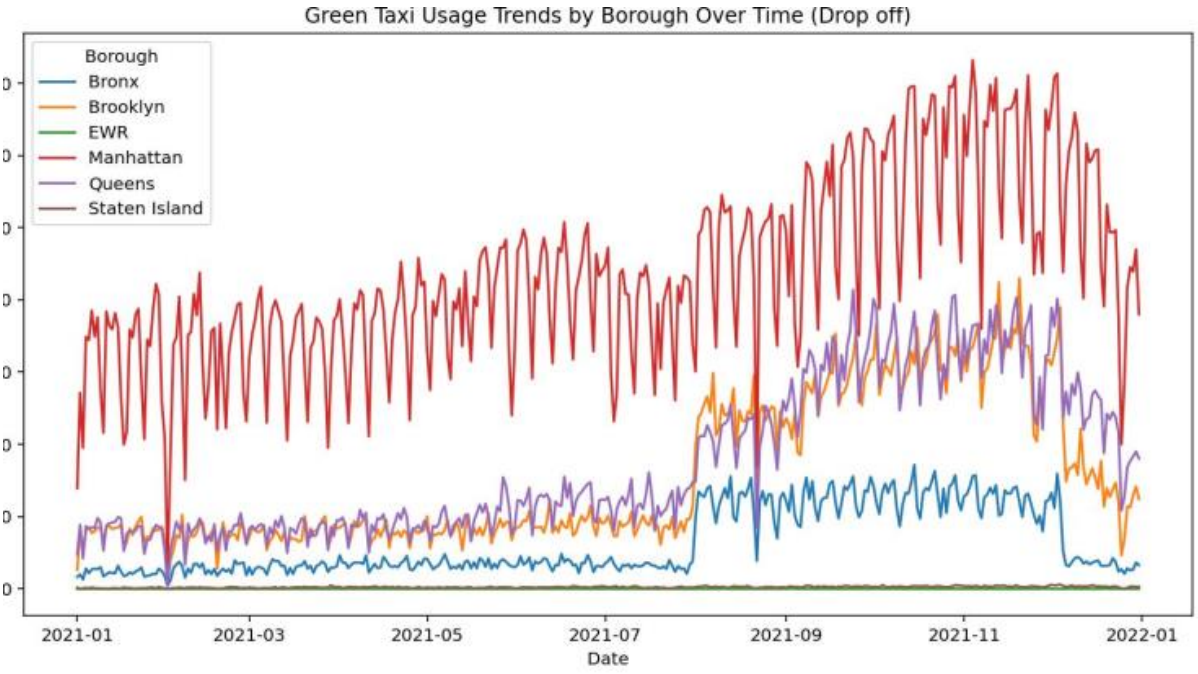


Figure 11 Chart of trip destinations across different dates.

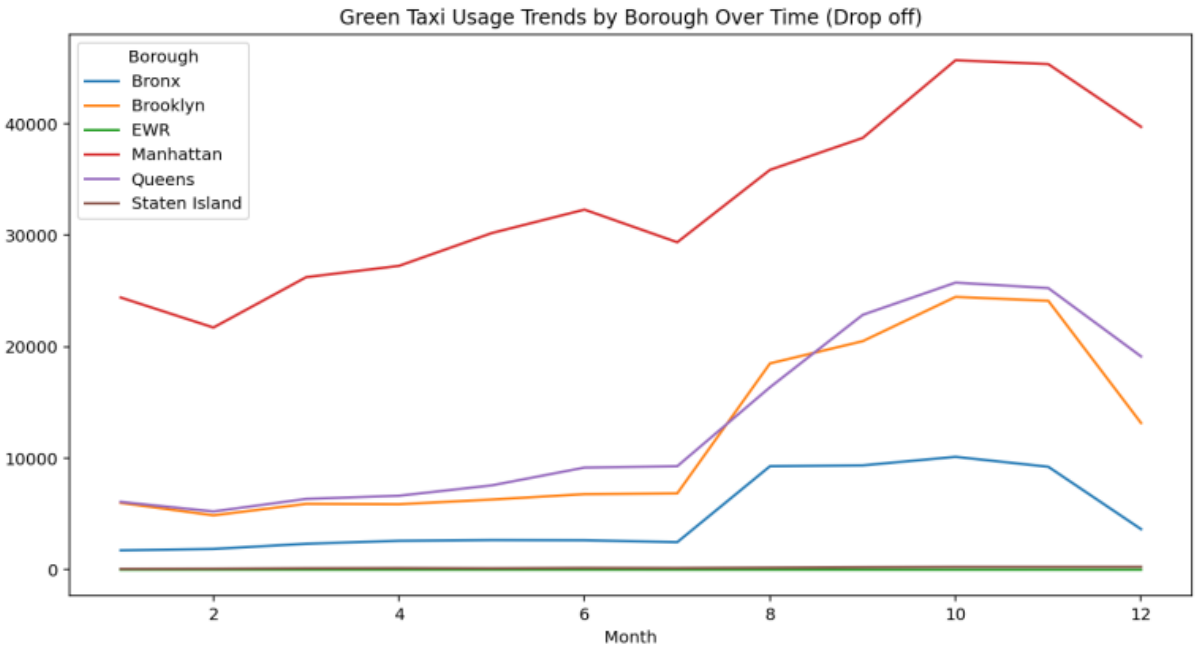


Figure 12 Chart of trip destinations across different months.

This chart illustrates the number of taxi trips during different hours of the day. The horizontal axis represents the hours of the day (from 0 to 23), while the vertical axis shows the number of trips. The highest number of trips is observed between 14:00 and 18:00, with each hour exceeding 50,000 trips. Conversely, the lowest number of trips occurs between 2:00 and 5:00 AM, with fewer than 10,000 trips. This chart indicates that taxi usage gradually increases throughout the day, peaking in the afternoon and then declining at night.

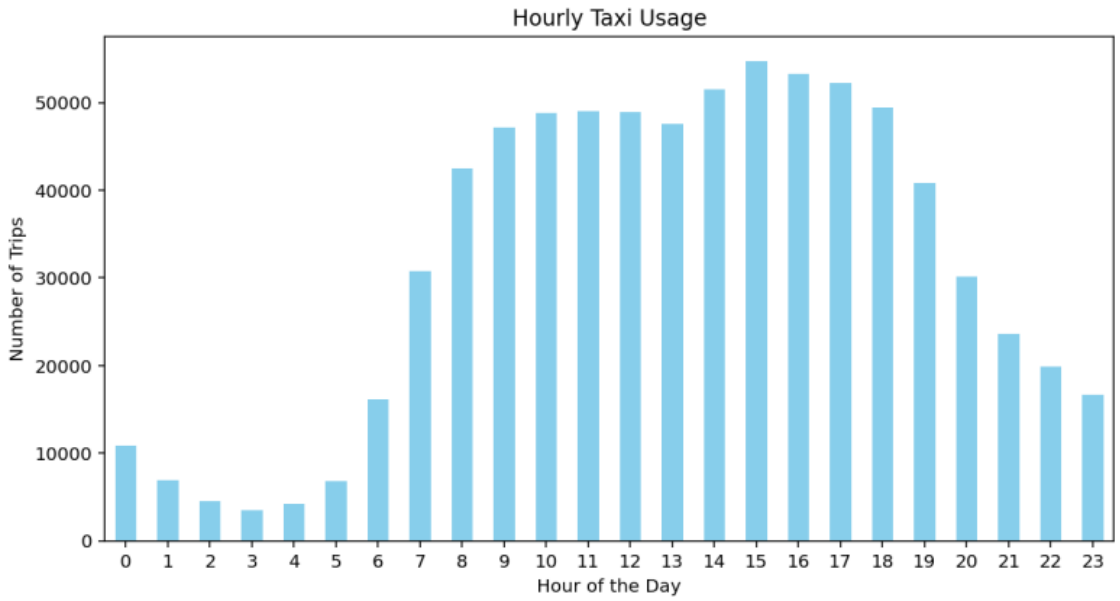


Figure 13 Chart of trip trends during different hours of the day.

Request Three – Detailed Data Analysis

In this section, before addressing the requests, we examined the outliers and removed them from our dataset to increase the accuracy of the models and prevent their adverse effects. We analyzed the outliers for "trip_duration," "trip_distance," and "total_amount," as these three columns warranted this examination due to their nature. The remaining columns did not require such scrutiny because of their nature.

The "trip_duration" column is a new column that we added to the dataset to calculate the duration of each trip.

To examine outliers, we used two methods: Z-Score (if the columns follow a normal distribution) and the interquartile range (IQR). For all three columns, we performed outlier detection and removal, and we have plotted the number of outliers, the count of deleted entries, and the related charts. You can find these results in the Python code.

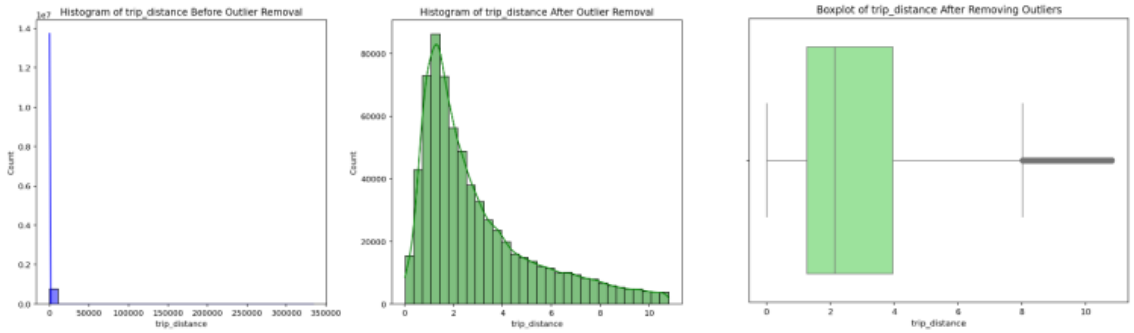


Figure 14 Charts of the outlier data for trip distance before and after their removal.

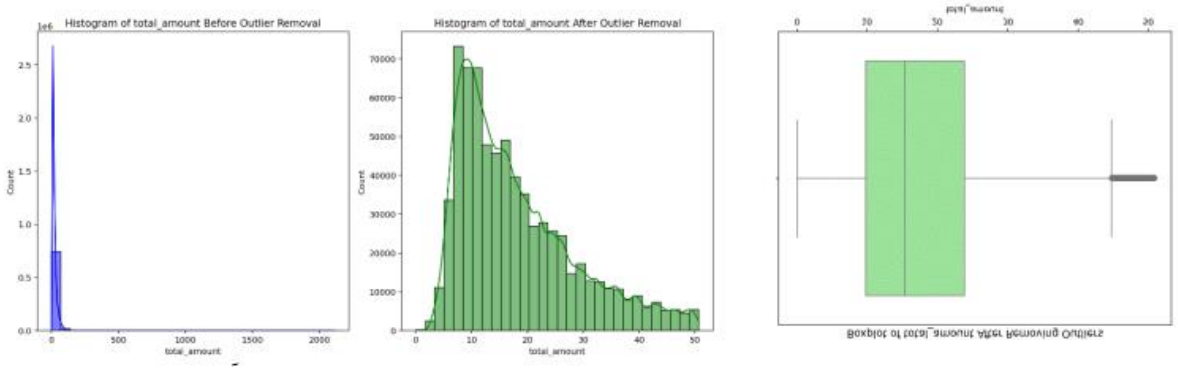


Figure 15 Charts of the outlier data for total amount paid before and after their removal.

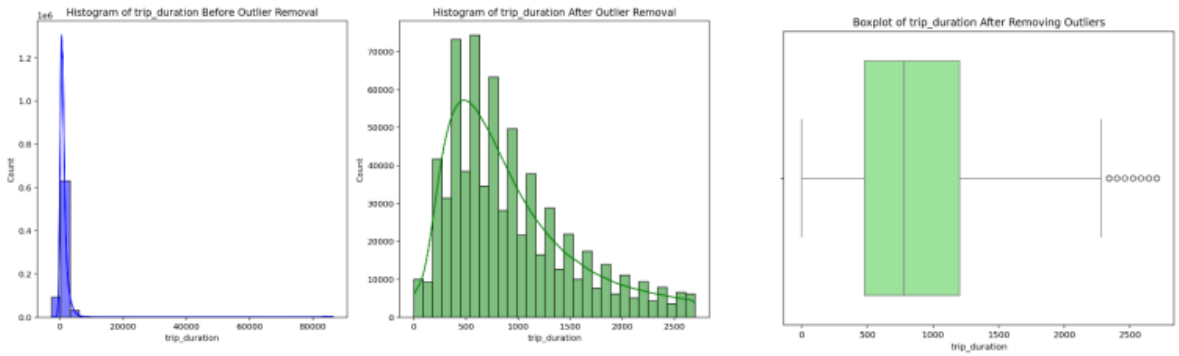


Figure 16 Charts of the outlier data for trip duration before and after their removal.

In this section, a correlation matrix of the various columns has been plotted. It should be noted that the days of the week, time, month, and dispatch are not included in this matrix. The pickup time (origin of the passenger) and the drop-off time (arrival at the destination) have a strong relationship with a correlation coefficient of 1.00. Similarly, the number of passengers and the number of pickups have a strong positive correlation of 0.89. Additionally, the trip distance and trip duration show a high correlation.

Conversely, the trip type and total amount have a moderate negative correlation of -0.28, indicating a lesser influence of trip type on the total amount. The payment type and pickup time show little correlation, and the number of passengers also has minimal impact on the gratuity amount. This matrix is very useful for identifying important patterns and relationships between variables in the data.

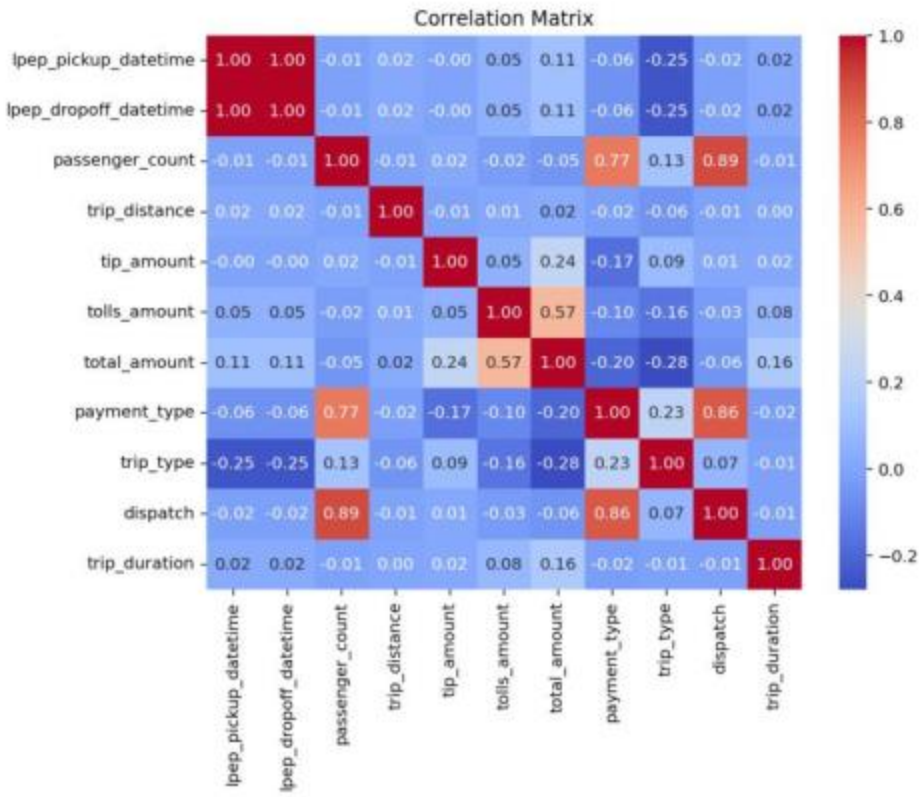


Figure 17 Correlation matrix.

We used the methods mentioned in the question, such as the chi-squared test, random forest importance, backward, and forward selection, to predict the price from the columns (these columns can be viewed and examined in the Python code) that contain continuous values. This approach is due to the fact that the final price also has a continuous value.

To evaluate and select the best features, we employed the Random Forest Regressor because of the presence of continuous values. We organized the output according to the importance of each feature, as shown in the diagram below, from high to low.

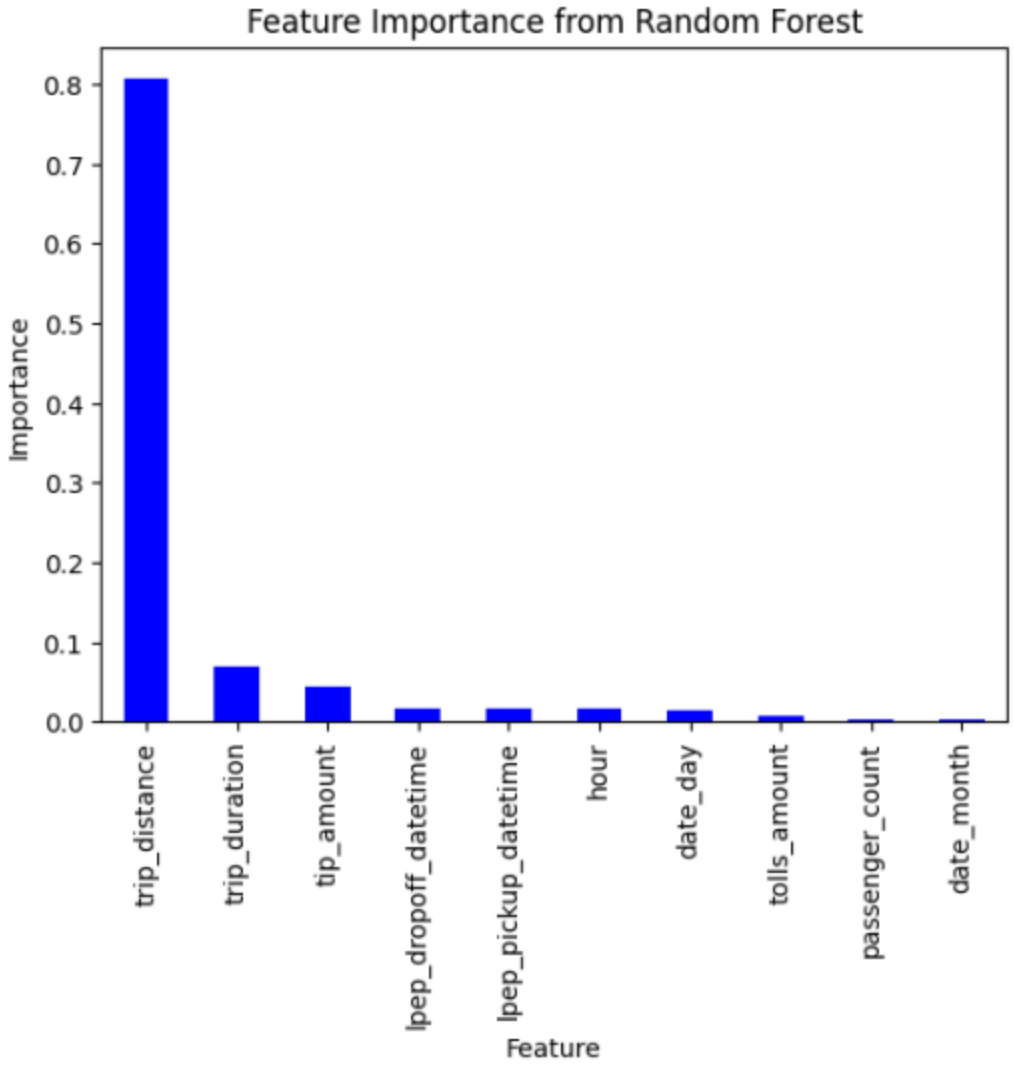


Figure 18 Random Forest method.

Since the chi-squared test pertains to continuous values and our columns of interest all contain continuous values, we will bin these values into intervals and record the number of observations within each interval. In this section, the outputs are then arranged from high to low, as illustrated in the diagram below.

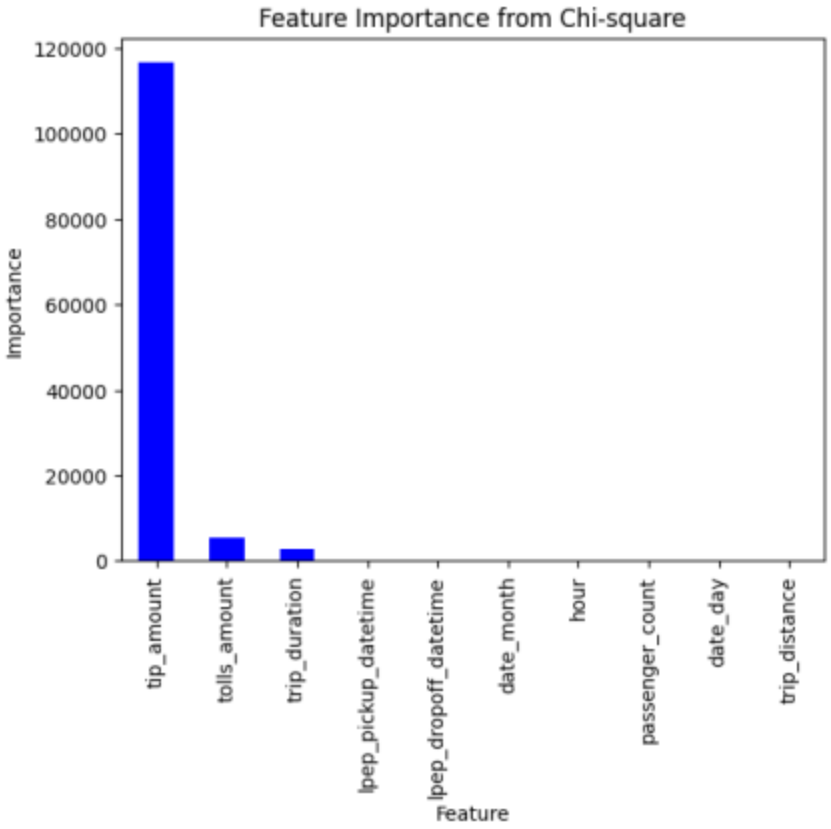


Figure 19 Chi-squared method.

For the forward method, we use linear regression to sequentially add each feature and assess their impact. Essentially, we select the features or columns that provide us with a better outcome. By incrementally incorporating these columns, we can evaluate which combination of features leads to the most significant improvement in the predictive model's performance.

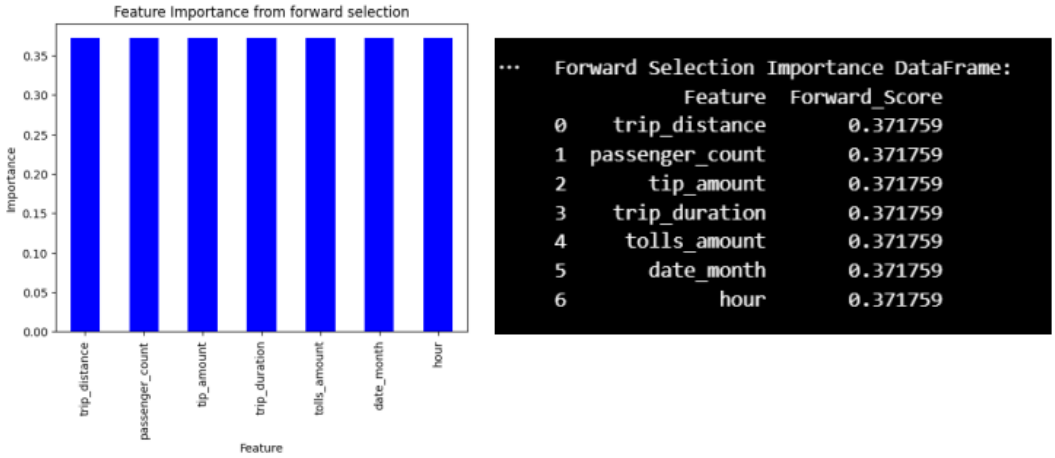
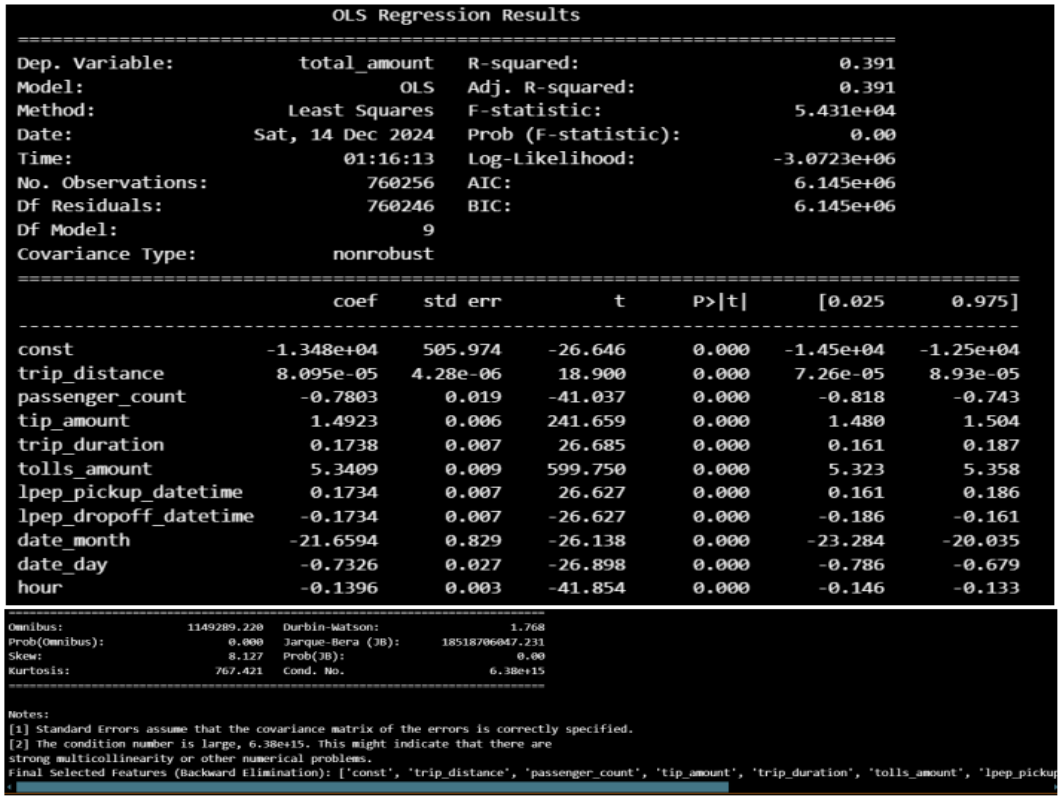


Figure 20 Forward method.

This image illustrates that, according to this method, the importance of all seven variables is the same and equal. The abbreviation for this method is Sequential Feature Selector (SFS).

The backward method is the opposite of the previous approach. Initially, all features are included, and we examine the columns in a loop. We continue to remove features until there is no further change in the accuracy of our model. This technique allows us to identify and retain only the most significant features that contribute to model performance.



OLS Regression Results

Dep. Variable:	total_amount	R-squared:	0.391
Model:	OLS	Adj. R-squared:	0.391
Method:	Least Squares	F-statistic:	5.431e+04
Date:	Sat, 14 Dec 2024	Prob (F-statistic):	0.00
Time:	01:16:13	Log-Likelihood:	-3.0723e+06
No. Observations:	760256	AIC:	6.145e+06
Df Residuals:	760246	BIC:	6.145e+06
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.348e+04	505.974	-26.646	0.000	-1.45e+04	-1.25e+04
trip_distance	8.095e-05	4.28e-06	18.900	0.000	7.26e-05	8.93e-05
passenger_count	-0.7803	0.019	-41.037	0.000	-0.818	-0.743
tip_amount	1.4923	0.006	241.659	0.000	1.480	1.504
trip_duration	0.1738	0.007	26.685	0.000	0.161	0.187
tolls_amount	5.3409	0.009	599.750	0.000	5.323	5.358
lpep_pickup_datetime	0.1734	0.007	26.627	0.000	0.161	0.186
lpep_dropoff_datetime	-0.1734	0.007	-26.627	0.000	-0.186	-0.161
date_month	-21.6594	0.829	-26.138	0.000	-23.284	-20.035
date_day	-0.7326	0.027	-26.898	0.000	-0.786	-0.679
hour	-0.1396	0.003	-41.854	0.000	-0.146	-0.133

Omnibus:	1149289.220	Durbin-Watson:	1.768
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18518706047.231
Skew:	8.127	Prob(JB):	0.00
Kurtosis:	767.421	Cond. No.	6.38e+15

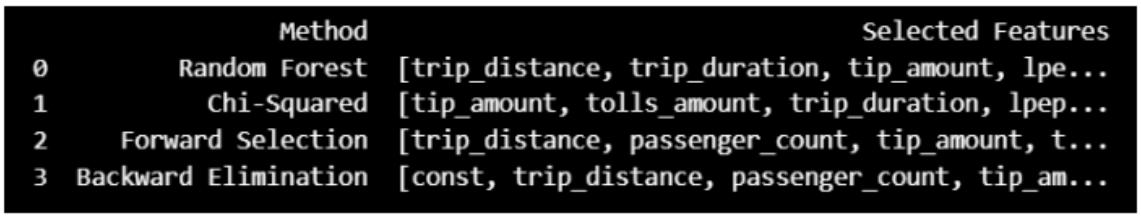
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.38e+15. This might indicate that there are
strong multicollinearity or other numerical problems.
Final Selected Features (Backward Elimination): ['const', 'trip_distance', 'passenger_count', 'tip_amount', 'trip_duration', 'tolls_amount', 'lpep_pickup

Figure 21 Backward method.

Result:

Final Selected Features (Backward Elimination): ['const', 'trip_distance', 'passenger_count', 'tip_amount', 'trip_duration', 'tolls_amount', 'lpep_pickup_datetime', 'lpep_dropoff_datetime', 'date_month', 'date_day', 'hour']

Ultimately, we have evaluated all these methods together.



	Method	Selected Features
0	Random Forest	[trip_distance, trip_duration, tip_amount, lpe...
1	Chi-Squared	[tip_amount, tolls_amount, trip_duration, lpep...
2	Forward Selection	[trip_distance, passenger_count, tip_amount, t...
3	Backward Elimination	[const, trip_distance, passenger_count, tip_am...

Figure 22 Comparison of methods.

Results of different methods:

Random Forest:

Selected Features: ['trip_distance', 'trip_duration', 'tip_amount', 'lpep_dropoff_datetime', 'lpep_pickup_datetime', 'hour', 'date_day', 'tolls_amount', 'passenger_count', 'date_month']

Chi-Squared:

Selected Features: ['tip_amount', 'tolls_amount', 'trip_duration', 'lpep_pickup_datetime', 'lpep_dropoff_datetime', 'date_month', 'hour', 'passenger_count', 'date_day', 'trip_distance']

Forward Selection:

Selected Features: ['trip_distance', 'passenger_count', 'tip_amount', 'trip_duration', 'tolls_amount', 'date_month', 'hour']

Backward Elimination:

Selected Features: ['const', 'trip_distance', 'passenger_count', 'tip_amount', 'trip_duration', 'tolls_amount', 'lpep_pickup_datetime', 'lpep_dropoff_datetime', 'date_month', 'date_day', 'hour']

In this section, we focused on predicting the payment or non-payment of tips (in a classified manner). Initially, we created the target variable is_tip_paid by converting the values of the tip_amount column to 0 and 1. Then, the data was processed and scaled by selecting key features such as travel distance, number of passengers, and duration of the trip. We split the data into two parts: 70% for training and 30% for testing, and trained five different Decision Tree models. The last three models, namely Logistic Regression, KNN, XGBoost, and Random Forest, were used solely for comparison with the main models. Finally, the performance of the models was assessed using criteria such as the confusion matrix and classification reports in the relevant results images.

The Decision Tree model has a satisfactory performance due to its simple structure, but it may be prone to overfitting, especially with large and complex datasets. The results of this model are often unbalanced, and metrics such as accuracy or recall can vary depending on the data distribution. In contrast, the Random Forest model, which is a combination of several decision trees, shows greater stability and has better generalization on test data due to the use of bagging techniques. The XGBoost model generally achieves the best performance among these models. This model, using gradient boosting, has the ability to identify complex patterns; however, it requires more execution time compared to Random Forest or Logistic Regression. Regarding simpler models, such as Logistic Regression, this method is suitable for recognizing linear patterns but may perform less effectively in identifying complex data. KNN can also exhibit variable performance depending on the number of neighbors selected and is sensitive to the optimality of its parameters.

Based on the results obtained, we observe that the Random Forest model performs better than the other methods used, considering metrics such as F1 Score, accuracy, and others, as long as computational costs and execution time do not pose limitations.

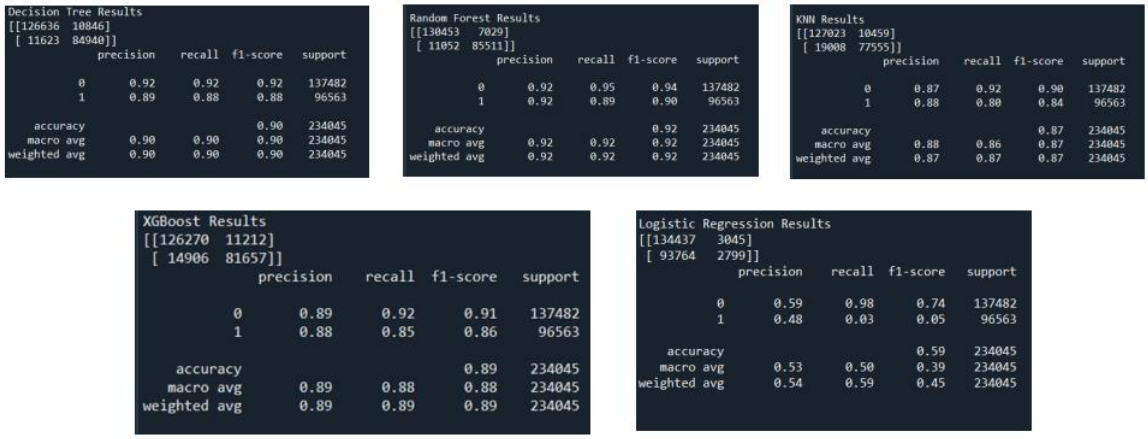


Figure 23 Review of different models.

In this section, the goal is to develop two models to predict the final price of the trip. Since the target variable we aim to predict is numeric, it is preferable to use columns that are also numeric. One of the models should be regression-based, and for this model, we consider linear regression, while the second model should be an enhanced version, for which we choose Random Forest.

Using the continuous variables that we believe have a stronger relationship with the target variable, as specified in the second part, we developed the models, noting that using these variables increases the accuracy of the models. The results of the model solutions are as follows:

Result of linear regression :

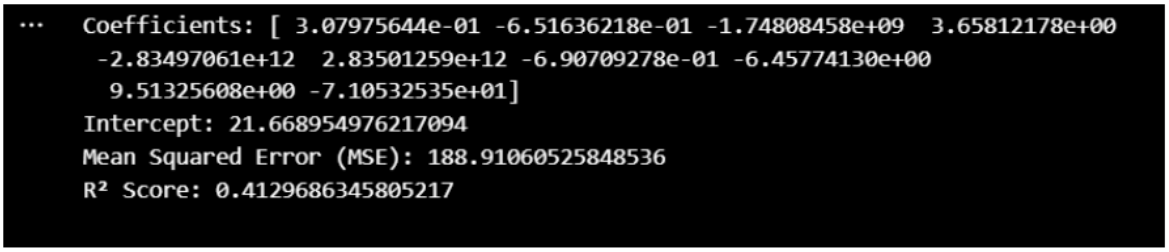


Figure 24 Result of linear regression

Result of Random Forest :

Figure 25 Result of Random Forest

It is also worth mentioning that initially, for more innovation, we wanted to use the concept of Ensemble Learning for Random Forest, using two RandomForrestRegressor models for prediction. In this way, one of the models would make the prediction using scaled, numeric variables, and the other model would first convert the categorical variables into numbers by appropriate binning and then use those variables to determine the predicted final price with RandomForrestRegressor. Then, using the concept of Majority vote in Ensemble Learning by averaging the two predicted values, we would predict a more accurate value for the final trip price. However, time constraints did not allow us to fully add the code for this part to our final code, although, with a full explanation of the logic behind this initiative, we hope you understand our efforts.

It might be asked why we do not use RandomForestClassifier for categorical variables in this described method and why we even convert these variables to numerical values and then use RandomForrestRegressor again. The reason for this is that the variable we want to predict is a numerical and continuous variable, and if we use RandomForestClassifier, the accuracy in predicting the target variable will be lower, and therefore we use two RandomForrestRegressors. Also, in addition to these, we must keep in mind that to increase the accuracy of the model and ensure it is not illogical, it is better to input the numerical data into the model in a scaled manner.

In this section, we will examine the models that we have developed in the previous sections. In the very simple Linear Regression model, all parameters are specified. Other models such as XGBoost or Decision Tree do not have optimal parameters, and they do not determine their parameters themselves. We use the Taguchi method to check the value of the parameters to create an optimal model. The code for this part is written in Python without using ready-made libraries, for better learning. In general, we used DOE (Design of Experiments) to find the optimal state. For this purpose, we consider an initial state and reach the optimal state with a small number of iterations.

```
Running Taguchi for Decision Tree Classifier...
Best Decision Tree Params: {'max_depth': 10, 'min_samples_split': 5}, Accuracy: 0.7979063855241514

Running Taguchi for Random Forest Classifier...
Best Random Forest Classifier Params: {'max_depth': None, 'min_samples_split': 2, 'n_estimators':
100}, Accuracy: 0.9224123565980901

Running Taguchi for Random Forest Regressor...
Best Random Forest Regressor Params: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 50},
MSE: 0.32619476103692435

Running Taguchi for XGBoost Classifier...
Best XGBoost Params: {'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 100}, Accuracy:
0.8808348821807772
```

Figure 26 Results from Taguchi.