UNIVERSITY OF NEVADA RENO

IS704: Data Analysis in Information Systems

Predictive Modeling of Trip Prices Using Tree-Based Methods

Angel Carranco Muller

May 2025

Introduction

Taxi prices can vary based on many factors such as time of day, distance traveled, and the pickup or drop-off location. This study aims to explore and model how various factors influence trip prices using tree-based methods.

Methodology

The dataset used for this study contains eleven columns and one thousand records. Please see table 1 for detailed information on the data collected from the samples.

Column Name	Description
Trip_Distance_km	Distance covered during the trip, measured in kilometers
Time_of_Day	Time of day the trip started (Morning, Afternoon, Evening, or Night)
Day_of_Week	Indicates whether the trip took place on a Weekday or Weekend
Passenger_Count	Number of passengers in the taxi during the trip
Traffic_Conditions	Traffic intensity during the trip (Low, Medium, High)
Weather	Weather condition during the trip (Clear, Rain, Snow)
Base_Fair	Initial base fare of the taxi ride before any distance or time charges
Per_Km_Rate	Rate charged per kilometer of the trip
Per_Minute_Rate	Rate charged per minute of the trip duration
Trip_Duration	Total time taken for the trip, measured in minutes
Trip_Price	Cost of the trip in USD

Table 1. Description of all eleven variables of the taxi trip pricing.

Please see Figure 1 for the summary of each variable from the dataset used.

```
Rows: 1000 Columns: 11
-- Column specification ------------------------------------------------------------------------
Delimiter: ","
chr (4): Time_of_Day, Day_of_Week, Traffic_Conditions, Weather
dbl (7): Trip_Distance_km, Passenger_Count, Base_Fare, Per_Km_Rate, Per_Minute_Rate, Trip_Duratio...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> summary(df)
 Trip_Distance_km Time_of_Day        Day_of_Week        Passenger_Count Traffic_Conditions
 Min.   :  1.23   Length:1000        Length:1000        Min.   :1.000   Length:1000
 1st Qu.: 12.63   Class :character   Class :character   1st Qu.:1.250   Class :character
 Median : 25.83   Mode  :character   Mode  :character   Median :2.000   Mode  :character
 Mean   : 27.07                                         Mean   :2.477
 3rd Qu.: 38.41                                         3rd Qu.:3.000
 Max.   :146.07                                         Max.   :4.000
 NA's   :50                                             NA's   :50
   Weather            Base_Fare       Per_Km_Rate     Per_Minute_Rate  Trip_Duration_Minutes
 Length:1000        Min.   :2.010   Min.   :0.500   Min.   :0.1000   Min.   :  5.01
 Class :character   1st Qu.:2.730   1st Qu.:0.860   1st Qu.:0.1900   1st Qu.: 35.88
 Mode  :character   Median :3.520   Median :1.220   Median :0.2900   Median : 61.86
                    Mean   :3.503   Mean   :1.233   Mean   :0.2929   Mean   : 62.12
                    3rd Qu.:4.260   3rd Qu.:1.610   3rd Qu.:0.3900   3rd Qu.: 89.06
                    Max.   :5.000   Max.   :2.000   Max.   :0.5000   Max.   :119.84
                    NA's   :50      NA's   :50      NA's   :50       NA's   :50
   Trip_Price
 Min.   :  6.127
 1st Qu.: 33.743
 Median : 50.075
 Mean   : 56.875
 3rd Qu.: 69.099
 Max.   :332.044
 NA's   :49
```

Figure 1. Summary of all variables in dataset.

From this summary, we can see that there are four categorical variables: *Time_of_Day, Day_of_Week, Traffic_Conditions*, and *Weather*.
The remaining seven variables are numerical variables in this dataset: *Trip_Distance_km, Passenger_Count, Base_Fare, Per_Km_Rate, Per_Minute_Rate, Trip_Duration_Minutes,* and *Trip_Price*.

Because the main goal of this pricing analysis is to predict the final price charged to customers based on other useful variables, "*Trip_Price*" becomes a strong and logical variable for this. This target variable helps with predicting revenue, dynamic pricing and variables analysis.

We discovered that this dataset contains null values as shown in Figure 2.

```
     Trip_Distance_km           Time_of_Day           Day_of_Week       Passenger_Count
                   50                    50                    50                    50
   Traffic_Conditions               Weather             Base_Fare           Per_Km_Rate
                   50                    50                    50                    50
      Per_Minute_Rate Trip_Duration_Minutes            Trip_Price
                   50                    50                    49
```

Figure 2. Count of Null values by variables in the dataset.

The target variable "*Trip_Price*" contains 49 null values, and every other variable contains also null values. To deal with this, we must begin by handling the null values from the target variable because the models built for training in later sections of this document will need to have something to learn from. Not having a target variable will affect the models because it will not know what we are trying to predict for that observation.

After removing the null values from the target variable to be able to train the models on rows where we know the outcome, we continue to handle the other variables. For all categorical variables, the null values will be replaced with *Unknown.* This will allow the model use the unknown category as a symbol of missing data for further analysis. Similarly, all the numerical variables with null values will be replaced with the median of the values of those specific variables. This will now allow outliers influence the representation of the data, and it helps preserve central tendency. After dealing with the null values of the dataset, we have the count of null values in Figure 3.

```
     Trip_Distance_km           Time_of_Day           Day_of_Week       Passenger_Count
                    0                     0                     0                     0
   Traffic_Conditions               Weather             Base_Fare           Per_Km_Rate
                    0                     0                     0                     0
      Per_Minute_Rate Trip_Duration_Minutes            Trip_Price
                    0                     0                     0
```

Figure 3. New count of Null values by variables in the dataset.

When comparing Figure 2 and Figure 3, we can confidently state that null values will not cause any issues while creating and testing tree-based models. This change leaves us with a total of 951 sample data for this analysis.

Data Visualization

Let's explore and understand what the dataset information has for us.

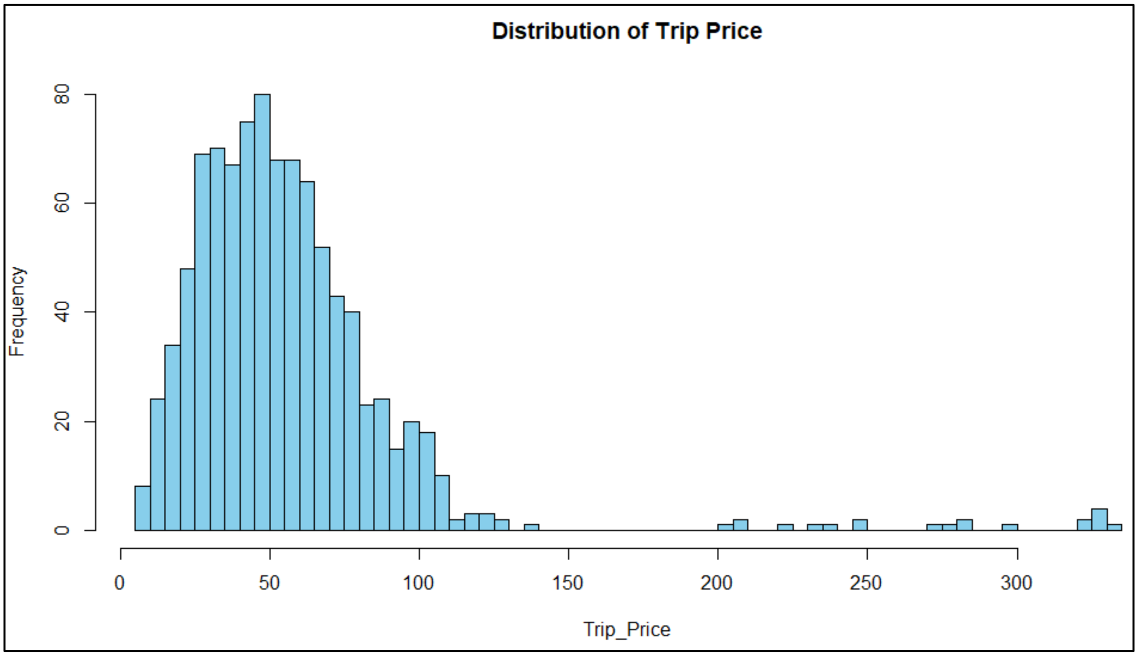

Figure 4. Distribution of Trip Price in the dataset.

Figure 4 demonstrates a roughly normal distribution of trip prices in our dataset, with a few noticeable outliers. It shows that most of the trip's prices range from $6.12 to around $135.00. However, a small number of trips exceeded that range due to other reasons and they are in the right-hand side of the distribution.

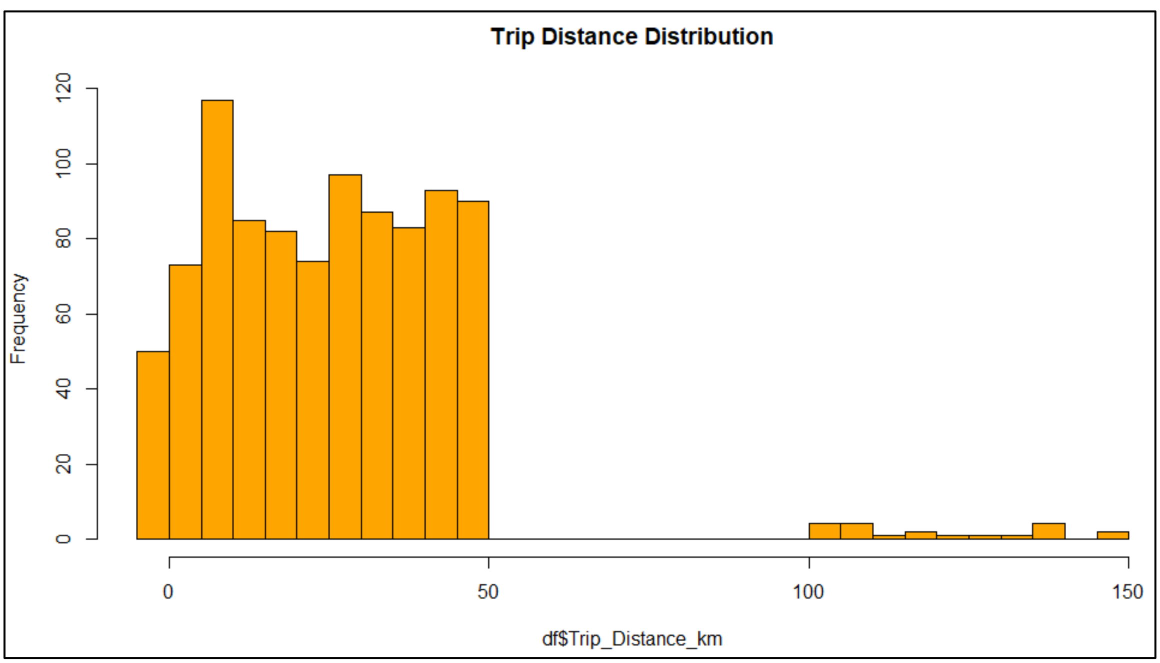

Figure 5. Distribution of Trip Distances in the dataset.

Figure 5 illustrates that the distance traveled by customers when using taxi services is constant, ranging from short trips to 50 kilometers. There are also those few trips that exceeded that range representing longer distance trips. Figure 5 represents good evidence that the distance traveled is a key factor correlated with the price charged per trip.

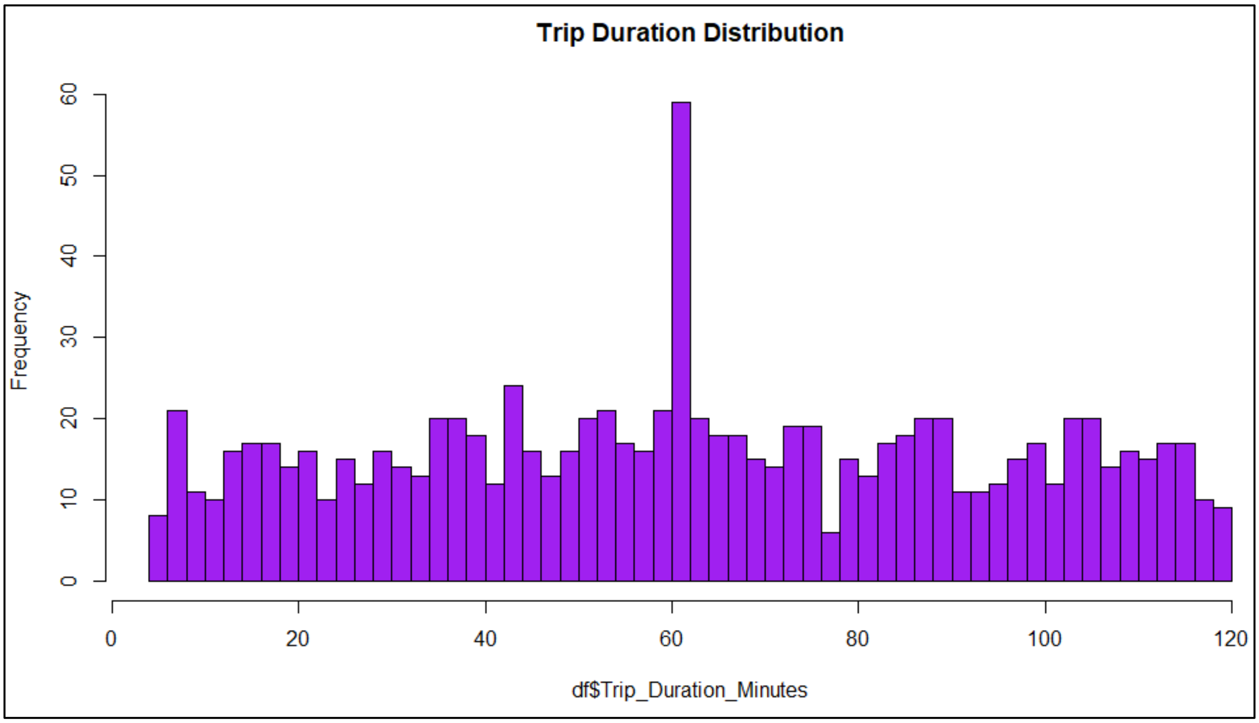

Figure 6. Distribution of Trip Duration in the dataset.

Figure 6 shows a wide range of trip durations in minutes from all the recorded taxi trips. Besides the approximately 60-minute trips which were the most frequent, there are no other notably constant trip duration.

In Figure 7, we can observe the correlation matrix that highlights relationships between numerical variables.

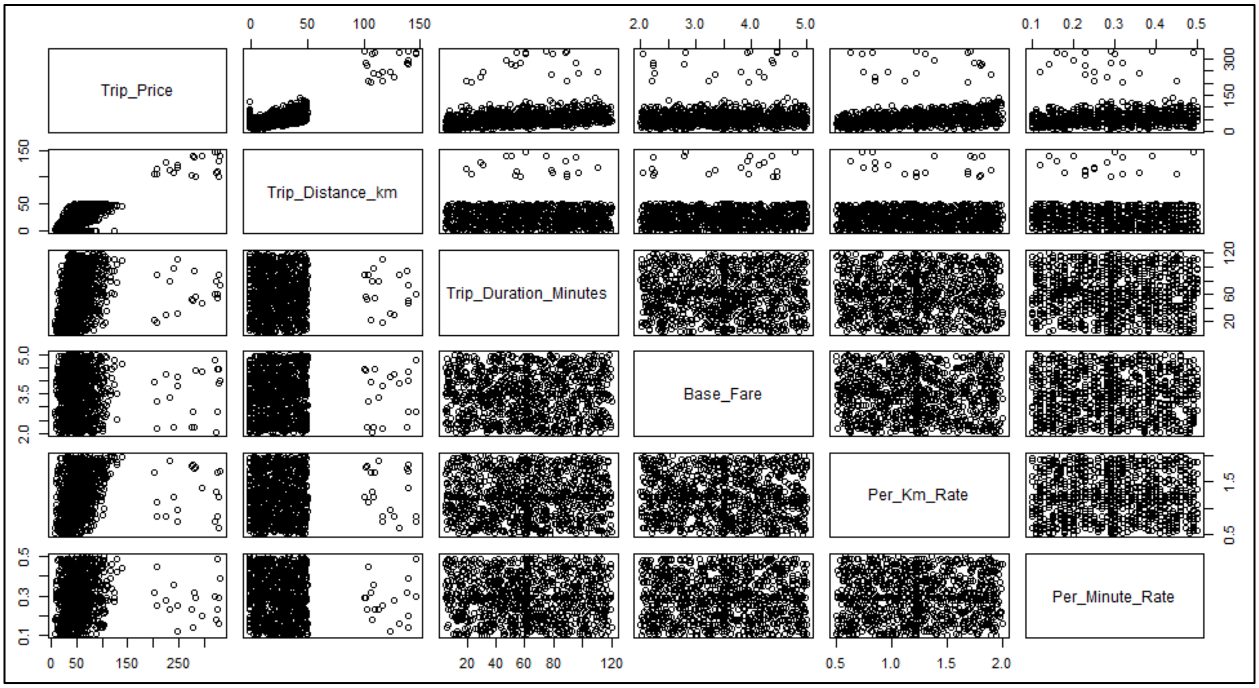

Figure 7. Scatter Plot Matrix of the numerical variables in the dataset.

The variables *Trip_Duration_Minutes, Base_Fare, Per_Km_Rate, and Per_Minute_Rate* do not show a clear relationship with one another. However, each of them shows visible patterns when plotted against *Trip_Distance_km and Trip_Price*, suggesting that they have a stronger relationship with the target variable. While they are not tightly correlated with each other, they may still have an important role in predicting prices.

Baseline Model

The baseline regression tree model is displayed in the following figure.

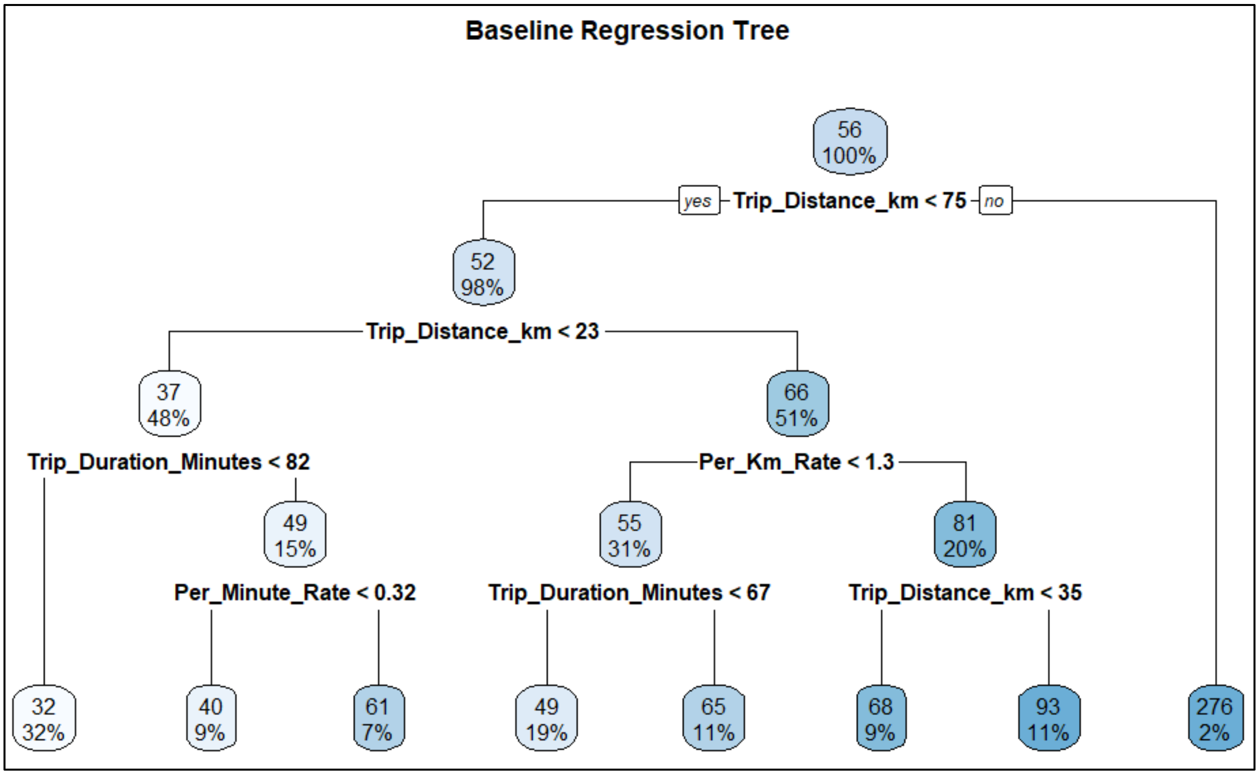

Figure 8. Baseline Decision Tree Model.

Figure 8 shows that this tree model used a subset of variables *(Trip_Distance_Km, Per_Km_Rate, Trip_Duration_Minutes and Per_Minute_Rate)* to build the tree while ignoring the other variables. This demonstrates the tree followed a greedy approach selecting the variables that reduce prediction error at each split. By using this baseline model, we have discovered that the predicted trip price has an average error of approximately $16.19 dollars. Considering that the median trip price is $50.07, we can calculate that the average prediction error is about 32%. This model captures 86% of the variability in trip prices. These may be considered strong results, or weak for some, but there is room for improvement which will be explored in the following sections using more advanced tree-based methods.

Bagging

When building and fitting a bagging model using the same structure as the baseline tree, we observe similar patterns in variable importance.

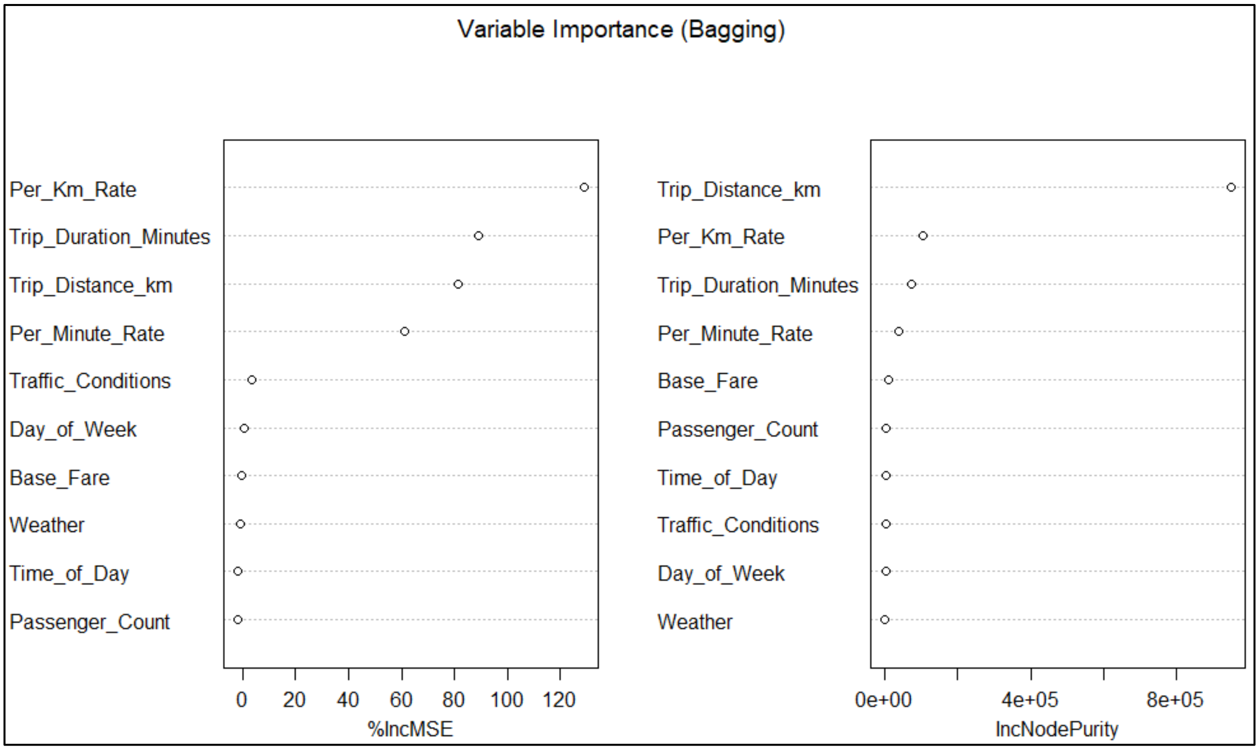

Figure 9. Variable Importance in Bagging Model.

As shown in Figure 9, the variables *Trip_Distance_Km, Per_Km_Rate, Trip_Duration_Minutes* and *Per_Minute_Rate* are the most influential variables in this model. This is shown in the % increase in Mean Squared Error (MSE) bar plot where these variables contribute the most to the predictions accuracy. This aligns with the information observed in Figure 8, where the baseline tree uses the same four variables for its splits. What is interesting is that *Per_Km_Rate* is the most impactful variable, where it was not clear in the baseline model and from figure 7. Additionally, the "IncNodePurity" demonstrates that *Trip_Distance_km* is the most useful and impactful in node splits reducing model error.

Although the baseline tree and bagging model show a similar structure in terms of variable importance, bagging demonstrates to have a higher performance than the baseline tree. The baseline decision tree has an RMSE of 16.19 and R^2 of 0.86, whereas the bagging model achieves a much lower RMSE of 9.32 and a R^2 of 0.95. This demonstrates that Bagging has an obvious improvement in predictive accuracy and reducing variance.

Random Forest Model

The Random Forest model demonstrates an approach a bit different compared to the bagging and baseline models.

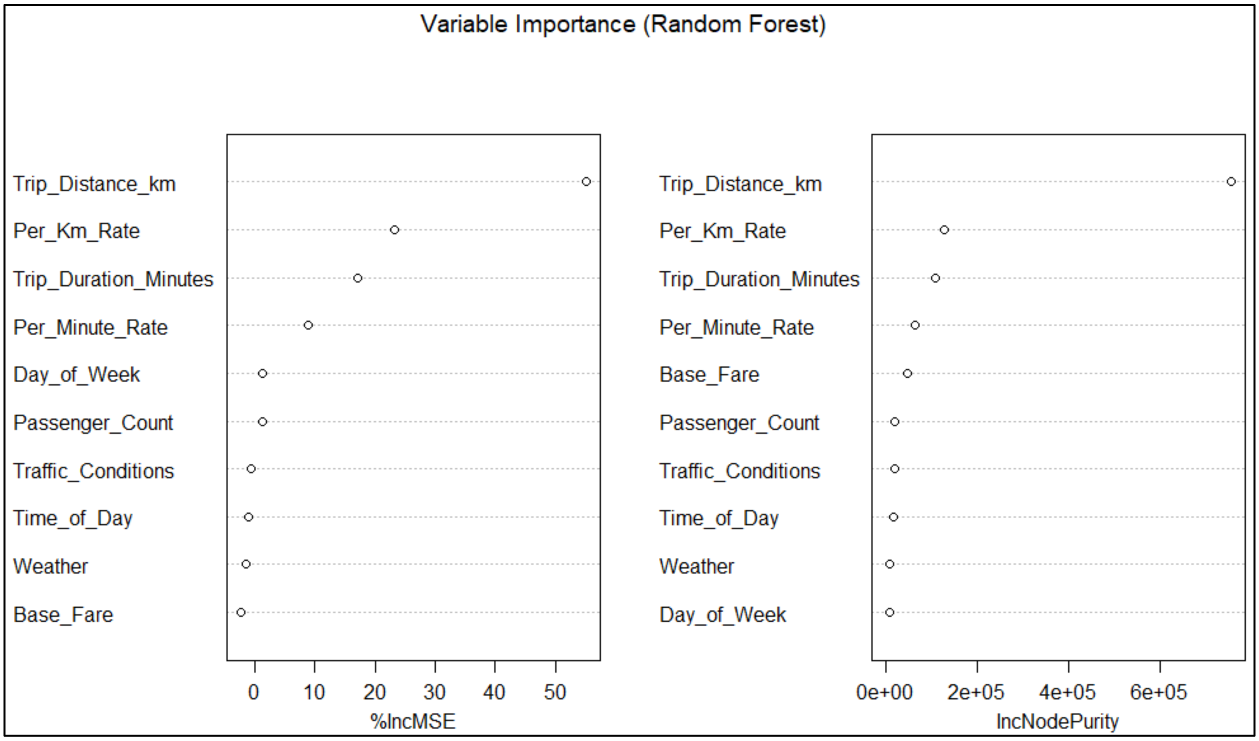

Figure 10. Variable Importance in Random Forest Model.

In figure 10, we can see from the left bar plot that the randomness of this model while growing its trees takes into consideration variables that were excluded from previous models. Variables like *Passenger_Count* and *Day_of_Week* have a higher value and the model depends more on them than *Traffic_Conditions* as preferably used in previous models. With this in mind we can predict that performance will vary from the previous models.

No. of Predictors	RMSE	R^2
Default (3)	15.26	0.9299
2	19.16	0.9253
6	10.16	0.9518
9	9.42	0.9534

Table 2. Performance results from two Random Forest models and different number of predictors.

In table 2 we can observe performance metrics with different number of variables randomly sampled at each split from four different Random Forest models. It is clear that the random forest models using two and three predictors have the higher RMSE and R^2 values hence they perform worse. The models using six and nine predictors have a better predictive performance with lower RMSE and higher R^2. However, considering that there is a total of 10 predictors, utilizing nine of them means that the model is using almost the whole set at each split. This creates the risk of overfitting because of the reduced tree diversity. Six number of predictors is a better choice because it still offers strong results and it is less likely to overfit.

Boosting Model

The boosting model focuses on sequentially training trees based on errors from previous trees. In this example, we developed two different boosting models with different parameters to observe their performance.

```
> # Summary with variable importance of Model 1
> summary(boost_model)
                                      var     rel.inf
Trip_Distance_km         Trip_Distance_km 79.73858072
Per_Km_Rate                   Per_Km_Rate  8.46095432
Trip_Duration_Minutes Trip_Duration_Minutes  5.68728705
Per_Minute_Rate           Per_Minute_Rate  4.07985499
Base_Fare                       Base_Fare  1.03825072
Traffic_Conditions     Traffic_Conditions  0.37962542
Time_of_Day                   Time_of_Day  0.26750363
Day_of_Week                   Day_of_Week  0.13123815
Passenger_Count           Passenger_Count  0.12548003
Weather                           Weather  0.09122496
> # Summary with variable importance of Model 2
> summary(boost_model2)
                                      var    rel.inf
Trip_Distance_km         Trip_Distance_km 79.0758306
Per_Km_Rate                   Per_Km_Rate  8.2519853
Trip_Duration_Minutes Trip_Duration_Minutes  5.8693480
Per_Minute_Rate           Per_Minute_Rate  4.2703063
Base_Fare                       Base_Fare  1.2713459
Traffic_Conditions     Traffic_Conditions  0.4829675
Time_of_Day                   Time_of_Day  0.3737282
Day_of_Week                   Day_of_Week  0.1645475
Passenger_Count           Passenger_Count  0.1288906
Weather                           Weather  0.1110500
```

Figure 11. Summary of two Boosting Models using different parameters.

We can observe in Figure 11 the difference between the two boosting models based on their variable importance. It is clear that *Trip_Distance_km* is by far, with almost 80% of variable importance, the major factor in determining the trip price. Other important factors are *Per_Km_Rate, Trip_Duration_Minutes,* and *Per_Minute_Rate*. Both models also demonstrate that factors like *Weather, Passenger_Count, Days_of_Week*, and others had minimal impact determining the trip price.

Model	# of trees	Shrinkage	RMSE	R^2
Model 1	5000	0.01	10.25	0.9463
Model 2	1000	0.05	10.69	0.9435

Table 3. Performance results from two Boosting models and parameters.

Table 3 shows how each model perform with different number of trees and shrinkage parameters. The first model uses a lower learning rate and more trees than the second model. Model 1 has a lower RMSE and a higher R^2 suggesting that it fits the data more accurately and explains the variance in trip prices better. However, the second model results are not far behind the first model. A faster model that handles larger datasets well, such as the second model, might be a good option to consider.

Bayesian Additive Regression Tree (BART) Model

After building a successful BART model, we received the following results:

RMSE	10.8872
R^2	0.939

Table 4. Performance results of BART Model.

With an RMSE of 10.89, the model's predictions are off by about $10.89 of the trip price. The model also shows a strong result explaining 94% of the variance in trip price. Although the model has good results, previous models used in this analysis have a better performance.

Conclusion

This analysis demonstrates that while simple decision trees offer useful data insight, more advanced tree-based methods as the ones used in this document provide a significant improvement in predictive accuracy. Throughout this document, we used multiple models experimenting with different parameter settings that demonstrate such improvements.

Model	RMSE	R^2
Baseline	16.19	0.8596
Bagging	9.32	0.9538
Random Forest	10.16	0.9518
Boosting	10.25	0.9463
BART	10.88	0.9396

Table 5. Performance Results of all Models used in this Analysis.

Table 5 shows the results for the best performing models of each tree-based method. The advanced models reduce RMSE by nearly $6 compared to the baseline model. This improvement highlights the importance of tuning and selecting the right model for your specific analysis.

While all methods provided consistent and good results, the bagging method achieved the best performance with lowest RMSE and highest R^2. It provides the most accurate and stable predictions for this dataset. By reducing variance, bagging improves performance while minimizing the risk of overfitting.

Resources

Den_Kuznetz. (2024). Taxi Price Regression. *Kaggle.com*.
https://doi.org/10188831/d2ebd685a60b7e9d48eb0a74f93ffde4