### **Capstone Project 2: New York City Taxi Fare Prediction**

### 

### **Milestone Report**

**Table of Contents**

|  |  |
| --- | --- |
| 1. Business Problem | 2 |
| 1. Business Objective | 2 |
| 1. Data | 2 |
| 1. Methodology | 2 |
| 1. Data wrangling | 3 |
| 1. Exploratory Data Analysis (EDA) | 4 |
| 1. Feature engineering | 5 |
| 1. Model evaluation | 11 |
| 1. Model validation and analysis | 11 |
| 1. Conclusions | 13 |

1. **Business Problem**:

In this problem, we predict the fare amount (inclusive of tolls) for a taxi ride in New York City, given the pickup and dropoff locations. The price estimation based on the distance between the pickup and dropoff locations results in an RMSE of $5-$8. The proposed Machine Learning model should obtain better results than the price estimation based on the distance between two points.

1. **Business Objective**:

Build a machine learning model that gives an error rate at least less than 5.0 RMSE to predict New York City taxi fare based on two given points.

1. **Data**:

The data contains three files, and the training data file has eight features that provide the details of yellow taxi rides in New York City.

Data Source: <https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/data>

Features:

* key: ID variable that provides a unique string to identify each row.
* fare\_amount: Target variable that provides the amount of the taxi ride in dollars.
* pickup\_datetime: It provides the start of the taxi ride as a timestamp.
* pickup\_longitude: Longitude coordinate of the pickup location.
* Pickup\_latitude: Latitude coordinate of the pickup location.
* dropoff\_longitude: Longitude coordinate of the dropoff location.
* dropoff\_latitude: Latitude coordinate of the dropoff location.
* passenger\_count: The number of passengers in the taxi ride.

1. **Methodology**:

The objective of this project is to accurately predict the taxi fare based on the pickup and dropoff location. The New York City Taxi fare prediction is a supervised regression problem, and we explore the data to choose the right regression model. This problem demonstrates the importance of feature engineering, where most of the features need to be discovered based on business knowledge to generate accurate predictions. In this project, we will follow these steps:

* Data wrangling
* Exploratory Data Analysis (EDA)
* Feature engineering
* Model evaluation
* Model validation and analysis

1. **Data wrangling:**

The analytical algorithms generally cannot use the raw input data, which is messy and incomplete. Data wrangling is a set of activities that prepare the raw data by cleaning them and make them usable for the analytical algorithms.

* 1. **Missing value treatment:** Missing data can lead to wrong predictions. In this data set, few rows have missing values, and we have removed them.
  2. **Outlier Detection and Treatment:** In this sub-section, we will describe the variables which contain outliers and discuss the methods to treat them.
     1. **Fare Amount:** The fare amount variable contains outliers. It contains negatives, zeroes, and values less than the minimum fare amount, which is highly unlikely events. This variable shows few entries of fare amount greater than $100, which are likely to be outliers. Therefore, in this project, we have considered the fare amount between $2.5, which is the minimum fare amount, and $100. Fig.1 illustrates the distribution of the fare amount variable.

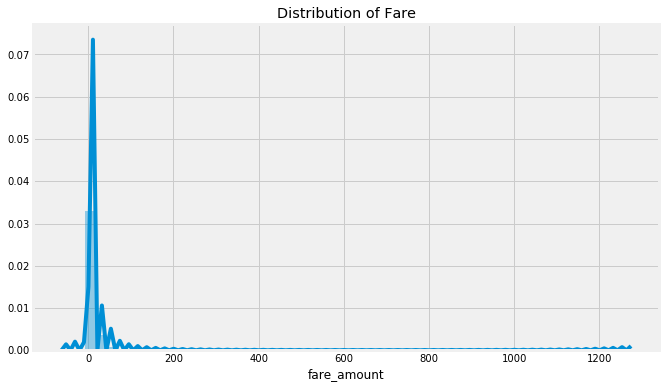


Fig. 1. Distribution of Fare Amount.

* + 1. **Passenger Counts:** We assume that a taxi is a vehicle whose body type is a sedan. Therefore, the maximum number of passengers can be 5, and we also remove the records where the passenger count is 0. Fig. 2 shows the number of passenger counts per trip.

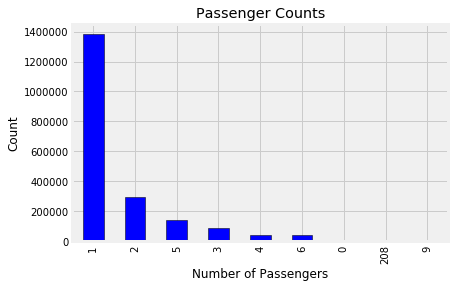


Fig. 2. Passenger counts per trip.

* + 1. **Location Data:** The location data contains two variables latitude and longitude. We have found the 2.5% and 97.5% percentile values of the latitude and longitude columns to remove the outliers.

1. **Exploratory Data Analysis (EDA):** We will use visualization methods to find trends, anomalies, patterns, or relationships within the data. EDA helps us to learn more about our data.
   1. **Fare Amount:** Fig. 3 describes the ECDF plot of the fare amount.

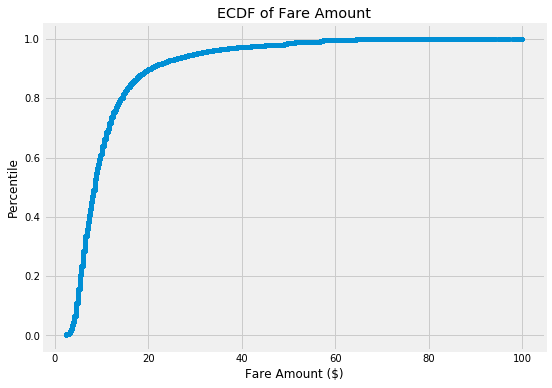


Fig. 3. ECDF of the fare amount.

From the ECDF plot, we can observe that most of the rides are below $20.

* 1. **Relationship between the number of passengers and fare:** Fig. 4 shows that single passengers are the most frequent travelers and Fig. 5 reveals the taxi receives the highest fares when they carry single passengers.

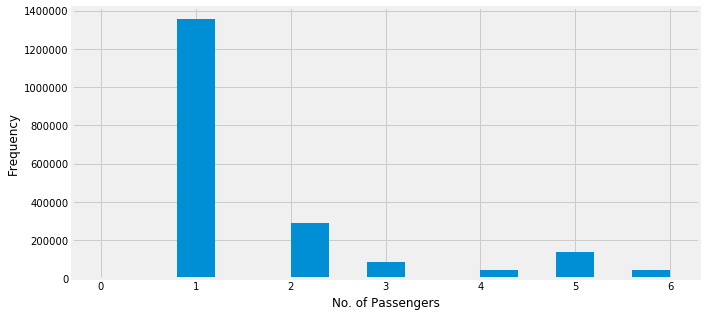


Fig. 4. Frequency of taxi rides based on passenger count.

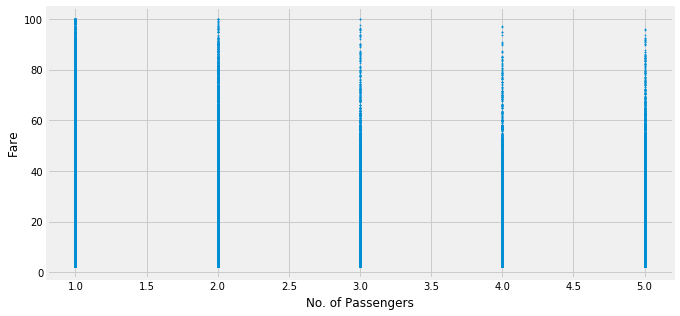


Fig. 5. Fare amount received based on the passenger count.

1. **Feature engineering:**

We create new features from the input data to build an accurate machine learning model. The first feature we create is distance. We compute the distance from the pickup and dropoff latitudes and longitudes values by using the Haversine formula. We add this new distance feature in our data.

We have analyzed that the 'passenger\_count' variable does not have an impact on the 'fare\_amount' when the distance is constant. For instance, both index 960389 and 95391 shows the same fare of $8.0 for a traveling distance of 1.290km while the number of passengers differs. Therefore, we remove the 'passenger\_count' column from our machine learning model.

We create the new features, ‘year’, ‘month’, ‘date’, ‘hour’, and ‘day\_of\_week’ from the ‘pickup\_datetime’ variable and then remove the ‘pickup\_datetime’ variable.

We have created a new feature 'dt' from the 'hour' variable. The new variable 'dt' consists of four values, 'Morning', 'Afternoon', 'Evening', or 'Night'. From 6 to 11 am is ‘Morning’. 'Afternoon' represents the value between 12 noon to 5 pm. 'Evening' represents the hour value between 6 pm to 11 pm, and 'Night' represents the rest of the hour values. Later, we remove the 'hour' variable.

We have added the month name as a new variable 'mn' based on the numeric value from the 'month' variable and then deleted the 'month'.

We have calculated the relative distances from the pickup and dropoff latitude and longitude values and added as new columns.

Now, we will explore and analyze the new features.

7.1. **Effect of the date of pickup on the frequency of ride:**

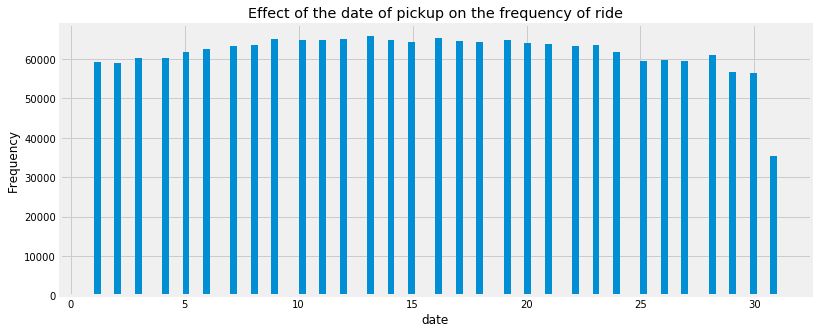


Fig. 6. Effect of the date of pickup on the frequency of ride.

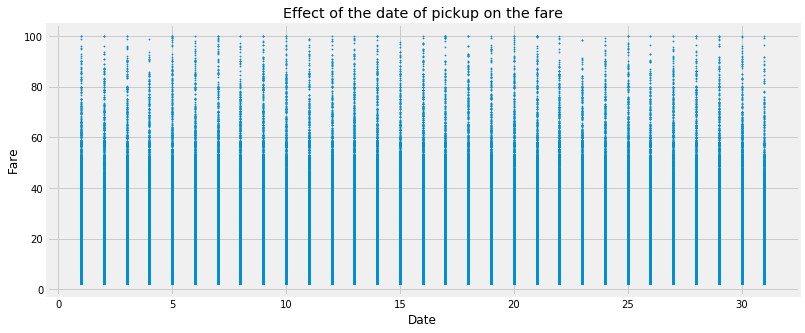


Fig. 7. Effect of the date of pickup on the fare of a ride.

Fig. 6 and 7 show that the frequency of ride and the fare are not affected by the date.

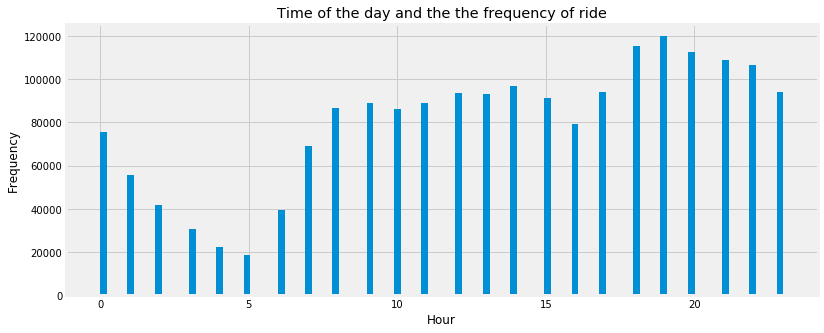


Fig. 8. Time of the day on the frequency of ride.

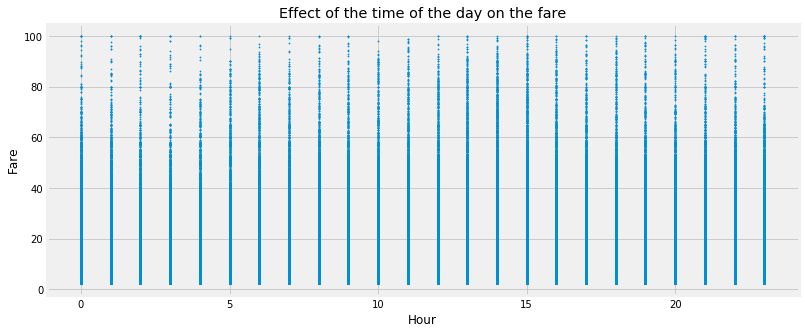


Fig. 9. Effect of the time of the day on the fare.

The time of the day affects the frequency of the rides. Morning 5 sees the lowest and evening 7 being the highest. However, the fare is high between 5 AM to 9 AM and 1 PM to 4 PM. The following figures show the time of the day with the new feature 'dt'.

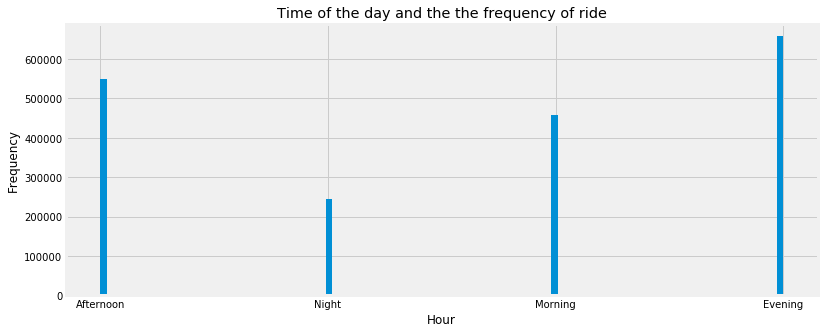


Fig. 10. Time of the day with variable ‘dt’ on the frequency of ride.

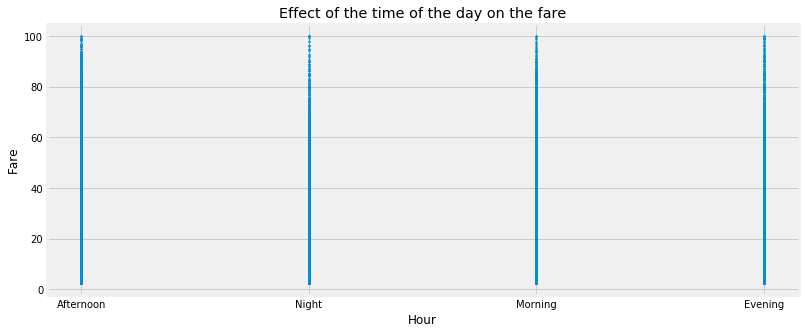


Fig. 11. Time of the day with variable ‘dt’ on the fare.

Rides increase in the evening and decrease at night.

7.2. **Effect of the day of the week on the fare:** The week does not seem to have much of an influence on the number of taxi rides. However, the frequency of rides increases on Thursday and decreases on Sunday and Monday.

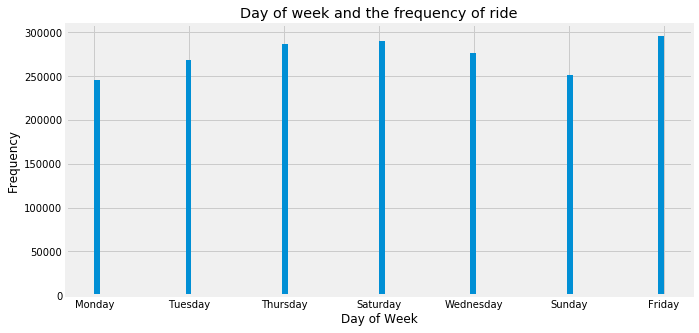


Fig. 12. Day of the week and the frequency of ride.

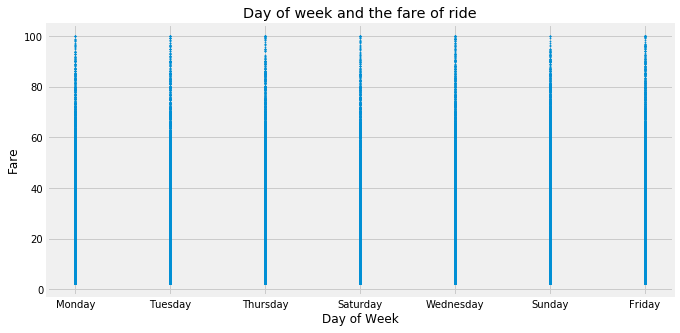


Fig. 13. Day of the week and the fare of ride.

7.3. **Effect of the month on the fare:** The first half of the year observed more rides than the second half of the year.

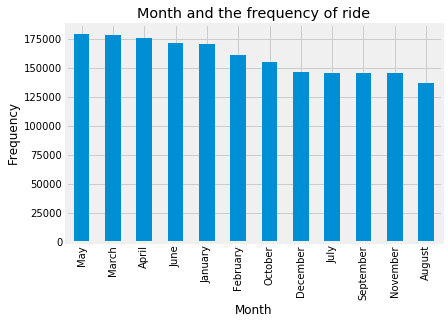


Fig. 14. Month and frequency of ride.

7.4. **Effect of the distance on the fare:** We will delete all records where distance is 0 km, but the fare amount is not $0.

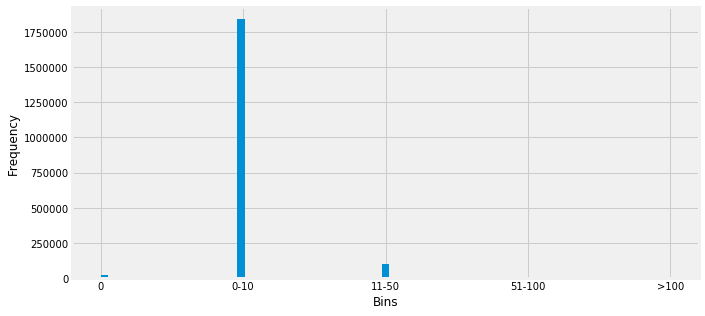


Fig. 15. Effect of the distance on the fare

7.5. **Encoding categorical variables:** We have encoded the following variables with one-hot encoding.

* **'date':** The date variable does not show any change in the fare. Therefore, we have encoded it.
* **'year':** The year variable does not show any difference in the fare based on the increase or decrease in the year.
* **'passenger\_count':** The fare remains unaffected by the number of passengers riding in a taxi.
* The object type variables, such as the month, week, and time of the day are also encoded.

7.6. **Feature selection:**

Finally, we have deleted the following features from our model, which are: 'month', 'hour', 'year', 'date', 'pickup\_datetime', 'passenger\_count', 'pickup\_longitude', 'pickup\_latitude', 'dropoff\_longitude', 'dropoff\_latitude', 'distance'.

We have found out that the relative distance of the latitude and longitude values produces a better model than the one with the distance feature. Therefore, we have removed the feature distance in favor of the relative distance.

1. **Model evaluation:**

To evaluate our model, we will use two metrics, root mean squared error and mean absolute percentage error. Root mean squared error (RMSE), which measures the difference between the predictions of a model, and the corresponding ground truth. Mean absolute percentage error (MAPE), which gives the average percentage error of the predictions.

1. **Model validation and analysis:**

We have used multicollinearity analysis to remove the variables which are highly linearly correlated with another variable. The resultant multicollinearity analysis removed the six features, wk\_Friday, dt\_Afternoon, mn\_April, dd\_1, yr\_2009, pc\_1.

We have tuned the parameters of the light GBM model using GridSearchCV, and the resultant values are, 'learning\_rate': 0.1, 'max\_depth': 10, 'num\_leaves': 31.

From our analysis, we observe that LightGBM produces the most accurate result. Therefore, we will use LightGBM to build the proposed model.

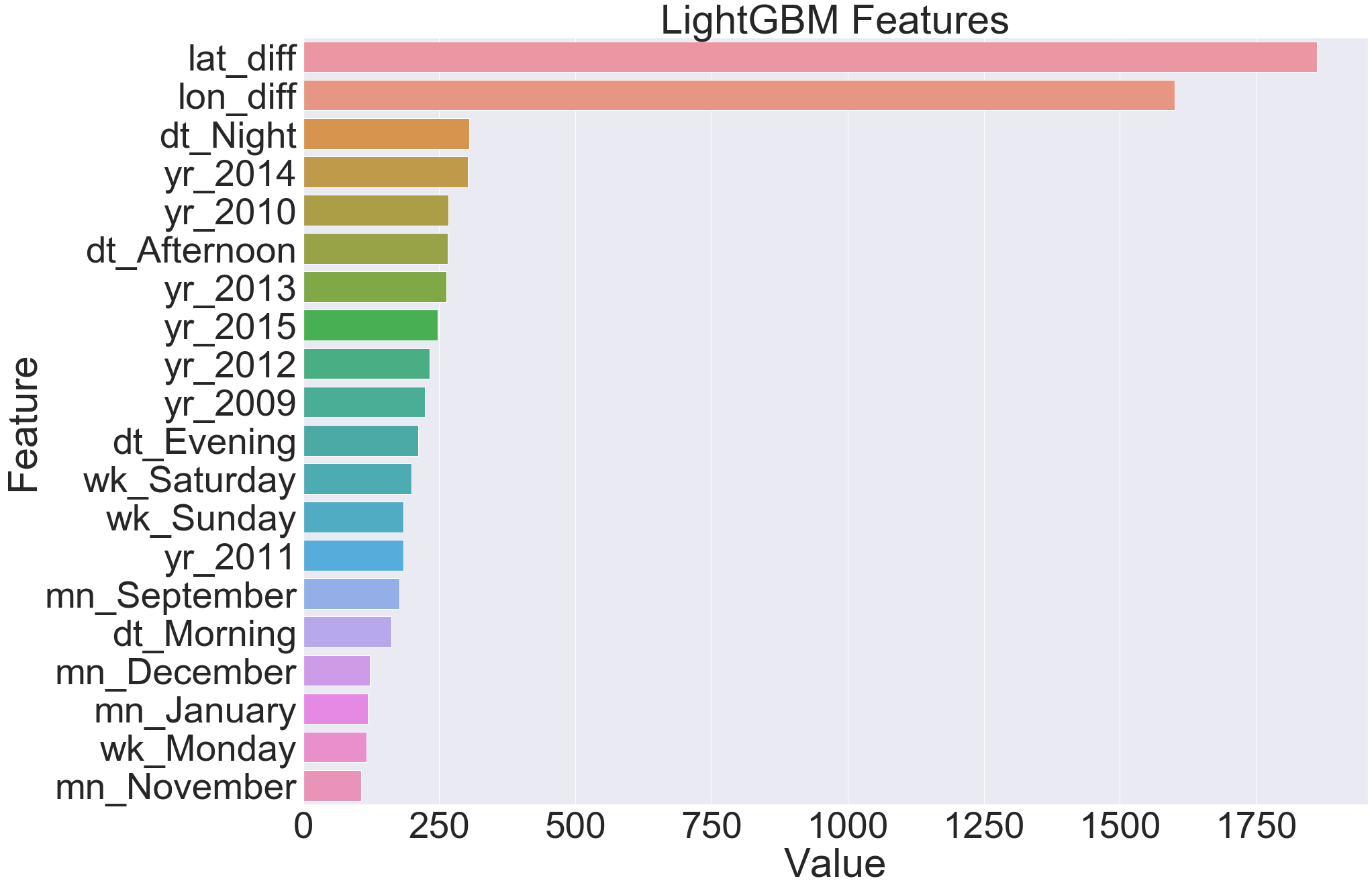


Fig. 16. LightGBM feature importance

The feature importance evaluated by the LightGBM model illustrates that the most important ones are the lat\_diff and lon\_diff variables. We have created these two variables from the pickup and dropoff latitude and longitude data.

We have used the multicollinearity analysis to remove the correlated variables from our input data. Thus, the proposed model works more efficiently.

We have also used parameter tuning on the final LightGBM model. Hence, the resultant model used both multicollinearity analysis and parameter tuning to produce more efficient results.

We have presented the RMSE and MAPE scores of the various machine learning algorithms under different experimental scenarios in Table I.

**Table I: rmse and mape scores under different experimental scenarios**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **RMSE** | **MAPE** |
| Linear Regression | 5.07 | 26.01 |
| Light Gradient Boosting | 3.67 | 18.97 |
| Random Forest | 3.75 | 19.98 |
| XGBoost with Multicollinearity | 8.58 | 31.36 |
| Random Forest with Multicollinearity | 3.74 | 19.07 |
| Light Gradient Boosting with Multicollinearity | 3.67 | 18.93 |
| **Light Gradient Boosting with Multicollinearity and Parameter tuning** | **3.65** | **18.67** |

We can observe from the experiments that Light Gradient Boosting with Multicollinearity and Parameter tuning is the most accurate machine model. It achieves 3.65 and 18.67 RMSE and MAPE scores.

1. **Conclusions:**

In this project, we have achieved a MAPE score of 18.67%, which means that we can predict the taxi fare from the pickup and dropoff location with 18.67% errors. This project demonstrates the importance of feature engineering. Here, we have created most of the features of the existing data. We have used the multicollinearity analysis to increase the accuracy of our model by removing collinear variables. Finally, we have fine-tuned the parameters of the LightGBM algorithm using GridSearchCV to improve our model. The feature importance evaluated by the LightGBM model illustrates that the most important ones are the lat\_diff and lon\_diff variables. We have created these two variables from the pickup and dropoff latitude and longitude data.