PREDICTION OF CAB FARES

1. Abstract:

It is difficult for individuals and organizations to estimate taxi trip fares using various dynamic conditions such as time and day, which affect the traffic conditions and starting location in a big city. Ridesharing is a service that uses travel information to match passengers, thereby reducing the total demand of cars on road. However, the problems with the system are that it is expensive, and not suitable for high capacity and long distances. The taxi service industry has been booming in recent years and is expected to grow significantly in the short term. This growing demand has led many companies to offer taxi rides to their users. However, few companies charge higher fares for the same route. Therefore, the customer unintentionally has to pay a higher price when the price should be lower. The main objective is to estimate travel costs before booking a taxi to ensure transparency and avoid unfair practices. Our system is designed to allow individuals to estimate taxi fares using a variety of dynamic conditions such as weather, taxi availability, taxi size, and distance between two points. Data already available can help create mathematical models that record important trends. In this paper, we define the problems mentioned above methodologically. The proposed system uses information such as ride requests to achieve efficient taxi-request indexing and thus, improving the matching of taxi and passenger. Particularly, it marks indexes on taxis such as the geographical location, source, and destination, while selecting the most optimum route for the travel and satisfying both online and offline ride bookings. Considerable amount of evaluation shows the accuracy of the system proposed, which is quick enough to process the requests in milliseconds. Unlike other services, it makes it easier to be deployed of shared and distributed infrastructure. In our proposed model we have used various machine learning algorithms such as linear regression, decision tree and random forest, gradient boosting and lasso regression

Keywords: Linear regression, decision tree, random forest, gradient boosting, lasso regression

2. Introduction

Ridesharing is a service that uses travel information to match passengers, thereby reducing the total demand of cars on road. However, the problems with the system are that it is expensive, and not suitable for high capacity and long distances. In this paper, we define the problems mentioned above methodologically. The proposed system uses information such as ride requests to achieve efficient taxi-request indexing and thus, improving the matching of taxi and passenger. Particularly, it marks indexes on taxis such as the geographical location, source, and destination, while selecting the most optimum route for the travel and satisfying both online and offline ride bookings. Considerable amount of evaluation shows the accuracy of the system proposed, which is quick enough to process the requests in milliseconds. Unlike other services, it makes it easier to be deployed of shared and distributed infrastructure. One of the major problems of travel, and transportation is solved by taxi ridesharing. According to a recent report, most passengers approved of taxi ridesharing. Particularly, the most common ways passengers use ridesharing are by booking a ride on an App, or by taking ridesharing services offline through a taxi. The proposed system looks for routes while satisfying certain travel conditions. Stability of route here refers to the frequency of rerouting or transferring needs come up for a taxi driver. It could also be viewed as the probability of an unexpected road condition coming up and affecting the travel and time requirements of the driver and the passenger. The system proposed takes origin, as well as destination into consideration. Also, the current systems immediately return a valid taxi as soon as it is found, instead of searching for the most optimum one. Systems like this which work on partial information fail to filter out the irrelevant taxis in the beginning, while also missing the potential best matches with the minimum cost. The current system solves the problems of the HDFS (Hadoop Distributed File System) by implementing RSP (Random Sample Partitioning). The results of HDFS are more prone to error. This further generates another drawback in this system because, HDFS does not store the properties of the data. The proposed system solves the method of the current ridesharing services by enabling longitude and latitude-based ridesharing, considering passengers on same route, updating requests dynamically, indexing a taxi that supports the most optimum route.

3. Literature Survey

Balika J Chelliah, Jai Singh, Devansh Chaturvedi and Avinash Kumar Singh(2021) has brought about methodologies to overcome ridesharing problems such as high cost, high capacity and long distance travel problems. Their taxi fare prediction system uses key feature extraction in artificial intelligence. Their system also resolves the problems of Hadoop Distributed File System(HDFS) by implementing Random Sample Partitioning(RSP). Their model uses Euclidean Formula and Haversine formula to measure the distance between 2 points. The system developed by them is capable of reducing the number of private vehicles on road leading to less traffic, shorter rides, driver's custom schedule manager and reduction in fares for rides.[1]

Christophoros Antoniades, Delara Fadavi, Antoine Foba Amon Jr. (2016) studied the fare and duration prediction of the New York taxi rides. The methodologies used by them to implement the same are linear regression, lasso, random forest and coordinate transformation. Their linear regression model used forward selection. Lasso confirms the best set of covariates to use. Coordinate transform is taken to model the effect of the pickup and drop off locations. Random forest as implemented in their model manages to model the non linearities of traffic and location effect. [2]

The predictive analysis given by Pallab Banerjee , Biresh Kumar, Amarnath Singh , Priyeta Ranjan4 and Kunal Soni, use machine learning algorithms for taxi fare prediction. They have proposed this paper after a detailed comparison of algorithms like regression and classification to get the most accurate value for prediction model. Their study is based on supervised learning. They have enforced 5 major steps in building their model which include: data collection, data visualization, data pre-processing, model selection and performance evaluation. Their model also used feature selection and also uses a correlation matrix. Random forest technique is also used. They finally evaluate their model built on linear regression and random forest based on mean absolute error and mean squared error to measure the accurate difference between the predicted scores and actual rating. Random forest proves to perform better. [3]

In this paper, Shashank H has given a data analysis of Uber and Lyft cab services. His system considers various factors that affect the cab prices such as traffic, weather and peak hours. The model uses methods of logistic and linear regression to predict the cab fares from source to the destination. The model uses supervised learning which helps to train the machine with labelled data that is already tagged with soe predicted values. The model is then tested with some random unknown szets of data and predicts the fare. The project makes use of various tools such as google colaboratory, anaconda 3, sklearn and pandas, numpy, matplotlib.pyplot seaborn python libraries. [4]

The goal of this research analysed by G. Venkat Sai Tarun a and P.Sriramya b is to use a machine learning technique called XGBoost, an algorithm for Predicting Fares for Online Taxi Services before trip traveling through comparison of r-squared and MSE values using the LASSO regression algorithm. The central importance of this study is in predicting prices for online taxi services. Before the journey starts, you can view the price of the journey even before the starting of the trip. It is displayed as a price forecast. Fare forecast shows the fare for the trip. It computes the given value of an attribute. Attribute must be central value that is calculated to show the forecast. Attributes include location, date and time, passenger number and fare. The existing fare amount is updated/changed depending on the program, fares are updated according to weather conditions, day and night, etc. These conditions affect and update fares. XGBoost is mainly used in cache-aware and out-of-core computing, parallel trees. Structural regularization for efficient handling and avoidance of missing data overfitting. The results obtained showed a slightly better accuracy standard for producing a near accurate estimation result. Based on the significance value achieved through SPSS. The Mean Accuracy error of XGBoost was also lower when compared to the Lasso regression Algorithm. Thus, the XGBoost algorithm has slightly better accuracy when compared to the Lasso regression algorithm. [5]

Hyeonjun Hwang, Clifford Winston and Jia Yan measure the benefits of ride hailing services to urban travellers of the San Francisco Bay Area. They estimate the total benefits with and without Uber along different time periods and then consider their difference. They estimate a transport mode choice model of SF travelers under the 2016-2017 transportation market environment (with Uber/Lyft services). Calibration of the choice models to replicate the 2008-2009 market outcomes before the existence of Uber services. They then run counterfactual simulations to estimate benefits without Uber and then compare those benefits with the estimated benefits with Uber available. Google maps API is used for the same purpose. [6]

Trh taxi fare rate classification by Rishabh Upadhyay and Simon Lui predicts taxi fares based on various features and selected meta information associated with each ride. Their approach uses simple and computationally rapid method that is completely automated. It is also based on deep neural networks. Feature extraction is also implemented. The proposed algorithm uses deep learning for classification. Deep neural networks (DNN) which comprises of multiple learning layers is used and for its implementation they use Lasagne. In addition to this, they have also used Stacking classifier to fit the training set to generate the target function. They compare their models using two large datasets- a validation dataset and a test dataset. [7]

ThananutPhiboonbanakit and TeerayutHoranont have analysed the Bangkok city's taxi ride by reforming dares for profitable sustainability using big data driven model. Their proposed model comprises of 6 components such as data, cost-distance algorithm, trip basic statistic report, data analysis, proft prediction and result evaluation and profit recommendation. From the proposed big-data-driven model, the analyzed results are divided into four parts. A taxi survey is first presented. The survey then aims to obtain all information related to taxi businesses and how they provide service to customers. These results are preliminary results, and assist in the following part of this study. The experiment is then presented, conducted on the data and indicate the types of routes that yield more profit. Comparative predictive models are also presented and are used to perform the tasks of extracting insight from the data. Finally, the add-on fare is presented. This is used to recommend a reasonable fare and driving pattern for drivers. This study investigated several taxi problems and improved the methodology of. The primary purpose determines why a taxi driver declines to serve customers in Bangkok using a data-driven approach. Taxi probe data collected over several months in the Bangkok area were used for analysis. They first started with data exploration on raw data, and cleaned unwanted data included outliers. The major technique behind this process is geospatial techniques such as geospatial grids and intersections. They then constructed a big-data-driven model to be a framework for the analysis of this data. From the results, it was discovered that the solution proposed by taxi drivers directly benefits the drivers but does not satisfy customers. This study suggests that the recommendation returned from the proposed model will satisfy both customers and drivers. The number of fares to be added is a balance between the traditional fare and the fee proposed by the driver's so-called optimal solution. The driver can make more profit, and customers do not have to pay an extremely high fare under trafc congestion. Second, it was proven that distance, followed by travel time, speed, and traffic congestion, are crucial factors for determining the trajectory patterns. These patterns have a significant impact on taxi driver profits. As a result, these factors are considered for recommending the add-on fare to the drivers. The recommendation of an add-on fare is based on travel distance and traffic congestion at a specific time. Tese factors are extracted from big data and are effectively predicted by the RF, GBRT, and DT at up to 9.80 RMSE and 0.19 min of computational time. In this study, we demonstrated the practical use of the proposed model through an example of a regular taxi trip. The results show the solution when the trip is involved and not involved under traffic congestion. As a result, the recommendation of an add-on fare can be used to enable taxi drivers to earn more profit. For instance, 7 THB is added for a non-trafc congestion trip and 15 THB is added for the traffic congestion trip. Therefore, drivers do not need to consider whether the trip they received will reduce their income. The big data model is adapted to changes driven by the data and returns the effective solution at the current time of the day. [8]

Qasima Chogale, Sonam Gupta, Ronak Jain, B.W. Balkhande aim to design aapplication program for fare prediction along with real time function in taxi routing and security. It is also used to upload the current location to a social netwoek or even send to a smart phone or phone number via SMS when the user is travelling which therefore contributes towards security. They use various modules such as GPS, social networking websites, SD card, cloud storage and XML for the purpose. [9]

In this work of ,Son Nguyen Van, Nhan Vu Thi Hong, Dung Pham Quang, Hoai Nguyen Xuan4, Behrouz Babaki5, Anton Drieswe, solved the problem of routing taxis through a road network in an offline shared peopleand parcel ride situation. They first imporved the share-a-ride model to predict parcel delivery demands based on historical requests. To this end, they derived a model that predicts the most likely parts of the road network that will receive taxi requests at some period of the day. This model utilizes a spatiotemporal Poisson process. Next, they proposed the OTSF-DP algorithm for the online routing problem. Conventional navigation systems typically provide the driver with the shortest path to the nearest parking location after completing all requests. In contrast, it was suggested that drivers follow the route with highest probability of receiving a new request while traveling to the specified parking location. The OTSF-DP uses the predicted information to guide the routing. However, if such an opportunity is lacking, a new parking location is recommended to the driver. At this location, it is hoped that the taxi will soon receive a new request. The OTSF-DP algorithm aims to minimize the taxi's idle time. Finally, we compared the performances of the proposed and previous algorithms on adapted real datasets. The experimental results showed that our algorithms provided more profits and overall benefits than the existing methods, and also reduced the idle travel distance in most of the considered instances. [10]

4. Proposed methodology

4.1. Architectural Diagram

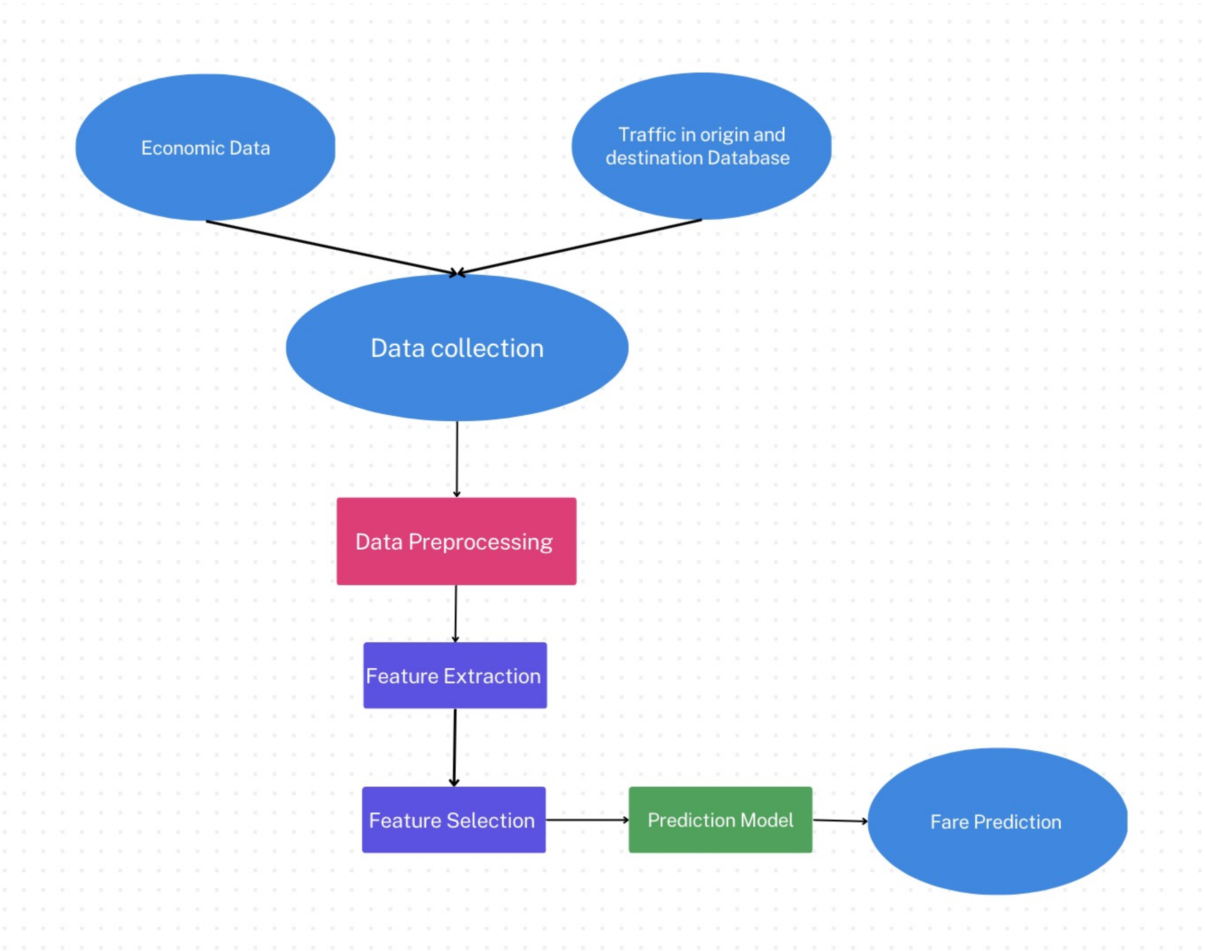


Figure 1

4.2. Module Description

4.2.1. Module 1: Data Evaluation

EDA depends vigorously on perceptions and graphical understandings of information. While measurable displaying gives a "straightforward" low-dimensional portrayal of connections between factors, they by and large require progressed information on factual procedures and numerical standards. Representations and diagrams are commonly significantly more interpretable and simple to create, so you can quickly investigate various parts of a dataset.

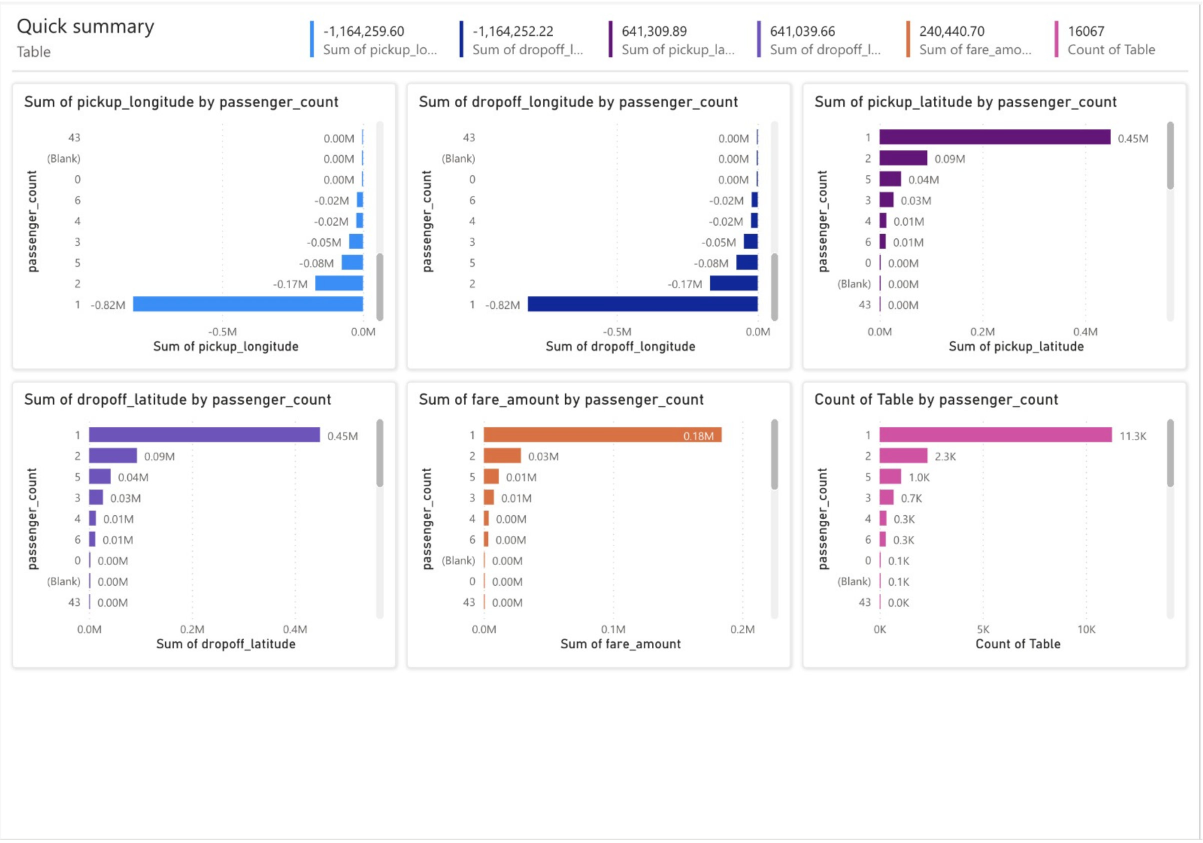


Figure 2

4.2.2. Module 2: Pre-processing

Information pre-handling is a significant advance to set up the information to shape a Taxi Fare Prediction model. There are numerous significant strides in information prepreparing, for example, information cleaning, information change, and highlight determination. Taxi Fare Prediction Data cleaning and change are strategies used to eliminate exceptions and normalize the information so they take a structure that can be effectively used to make a model A Taxi Fare Prediction informational collection may contain many factors (descriptors); be that as it may, a significant number of these factors will contain excess information. To improve on the dimensionality of the model, it is critical to choose just factors that contain novel and significant data

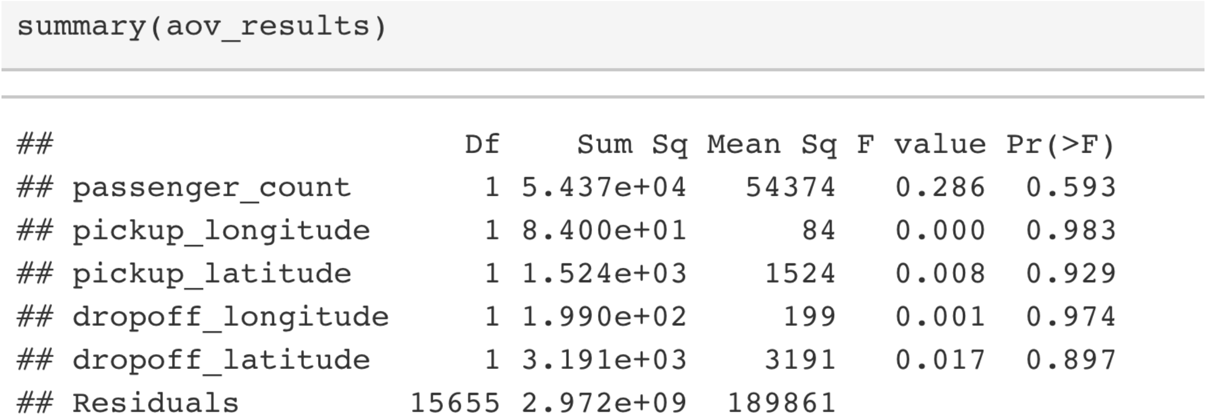
4.2.3. Module 3: Prediction

We are going to use Linear regression, decision tree, random forest, gradient boosting, Lasso regression algorithms to predict the fare amount.

4.3. Data set

The data sets that we used are test_cab_fare and train_cab_fare.

The attributes used are described as follows:

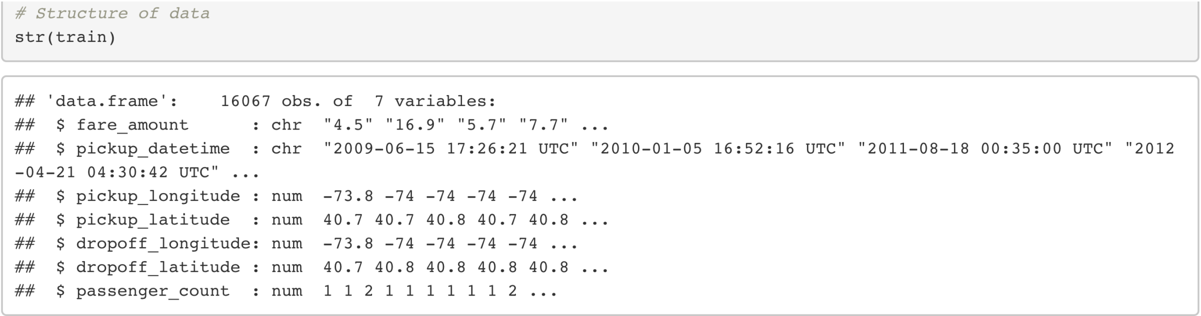


```
summary(aov_results)
```

```
##                       Df    Sum Sq Mean Sq F value Pr(>F)
## passenger_count        1 5.437e+04   54374   0.286  0.593
## pickup_longitude       1 8.400e+01      84   0.000  0.983
## pickup_latitude        1 1.524e+03    1524   0.008  0.929
## dropoff_longitude      1 1.990e+02     199   0.001  0.974
## dropoff_latitude       1 3.191e+03    3191   0.017  0.897
## Residuals          15655 2.972e+09  189861
```

- pickup_datetime - timestamp value indicating when the cab ride started.
- pickup_longitude - float for longitude coordinate of where the cab ride started.
- pickup_latitude - float for latitude coordinate of where the cab ride started.
- dropoff_longitude - float for longitude coordinate of where the cab ride ended.
- dropoff_latitude - float for latitude coordinate of where the cab ride ended.
- passenger_count - an integer indicating the number of passengers in the cab ride.

Results and Discussion:



```
# Structure of data
str(train)
```

```
## 'data.frame':    16067 obs. of  7 variables:
##  $ fare_amount      : chr  "4.5" "16.9" "5.7" "7.7" ...
##  $ pickup_datetime  : chr  "2009-06-15 17:26:21 UTC" "2010-01-05 16:52:16 UTC" "2011-08-18 00:35:00 UTC" "2012
-04-21 04:30:42 UTC" ...
##  $ pickup_longitude : num  -73.8 -74 -74 -74 -74 ...
##  $ pickup_latitude  : num  40.7 40.7 40.8 40.7 40.8 ...
##  $ dropoff_longitude: num  -73.8 -74 -74 -74 -74 ...
##  $ dropoff_latitude : num  40.7 40.8 40.8 40.8 40.8 ...
##  $ passenger_count  : num  1 1 2 1 1 1 1 1 1 2 ...
```

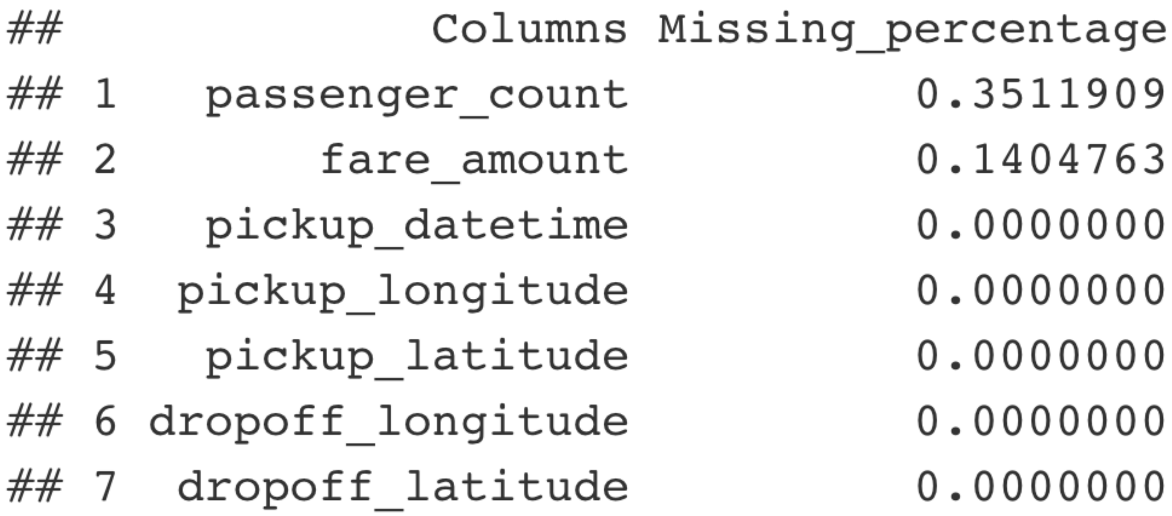
Structure of the data:

Summary of the dataset:

```
summary(train)
```

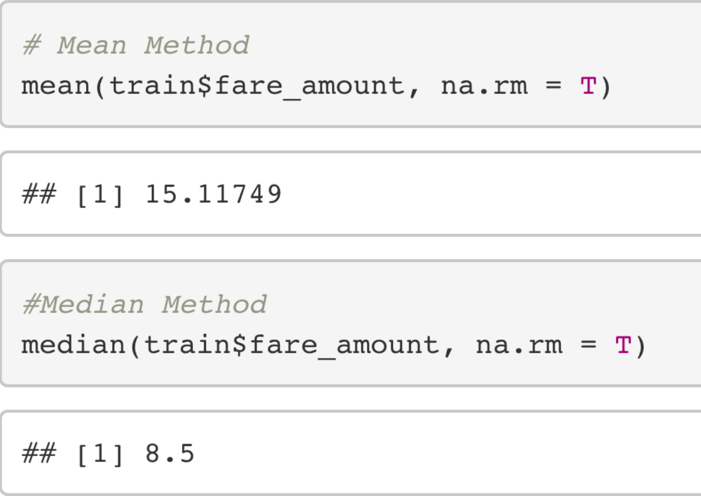
```
##   fare_amount        pickup_datetime    pickup_longitude pickup_latitude
##   Length:16067       Length:16067       Min.   :-74.44   Min.   :-74.01
##   Class :character   Class :character   1st Qu.:-73.99   1st Qu.: 40.73
##   Mode  :character   Mode  :character   Median :-73.98   Median : 40.75
##                                         Mean   :-72.46   Mean   : 39.91
##                                         3rd Qu.:-73.97   3rd Qu.: 40.77
##                                         Max.   : 40.77   Max.   :401.08
##
##   dropoff_longitude dropoff_latitude passenger_count
##   Min.   :-74.43    Min.   :-74.01   Min.   :   0.000
##   1st Qu.:-73.99    1st Qu.: 40.73   1st Qu.:   1.000
##   Median :-73.98    Median : 40.75   Median :   1.000
##   Mean   :-72.46    Mean   : 39.90   Mean   :   2.625
##   3rd Qu.:-73.96    3rd Qu.: 40.77   3rd Qu.:   2.000
##   Max.   : 40.80    Max.   : 41.37   Max.   :5345.000
##                                      NA's   :55
```

Below are the missing value percentage for each variable:



```
##             Columns Missing_percentage
## 1   passenger_count          0.3511909
## 2       fare_amount          0.1404763
## 3   pickup_datetime          0.0000000
## 4  pickup_longitude          0.0000000
## 5   pickup_latitude          0.0000000
## 6 dropoff_longitude          0.0000000
## 7  dropoff_latitude          0.0000000
```

We have to impute values for each variable

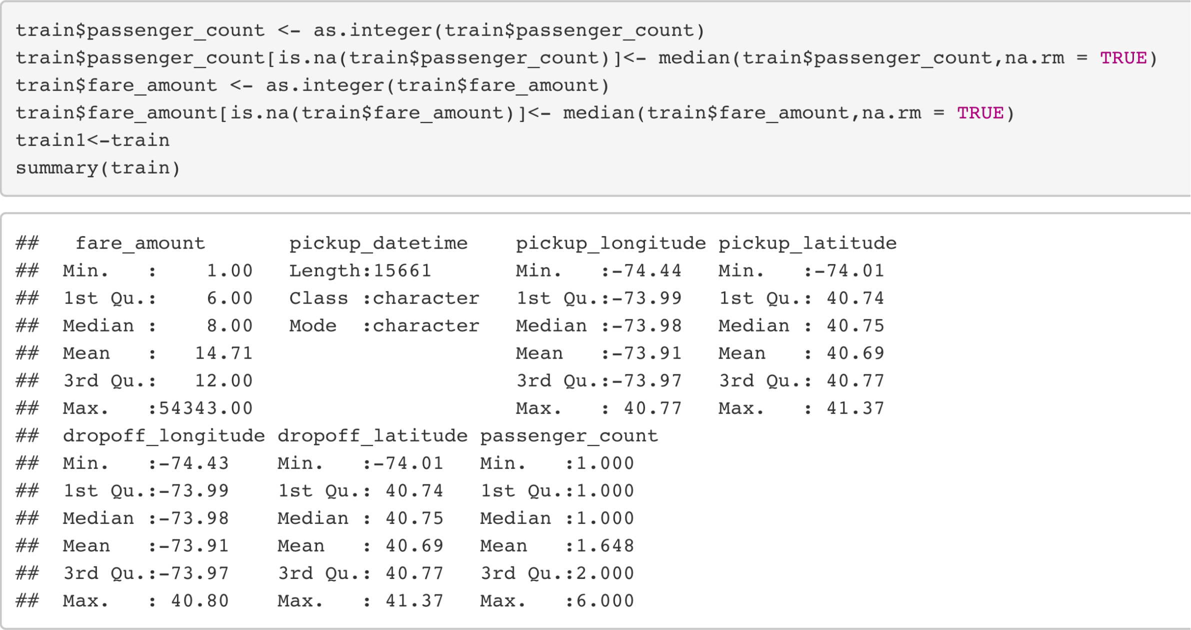


```
# Mean Method
mean(train$fare_amount, na.rm = T)
```

```
## [1] 15.11749
```

```
#Median Method
median(train$fare_amount, na.rm = T)
```

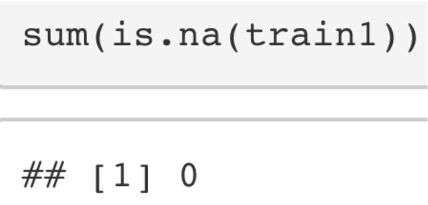
```
## [1] 8.5
```



```
train$passenger_count <- as.integer(train$passenger_count)
train$passenger_count[is.na(train$passenger_count)]<- median(train$passenger_count,na.rm = TRUE)
train$fare_amount <- as.integer(train$fare_amount)
train$fare_amount[is.na(train$fare_amount)]<- median(train$fare_amount,na.rm = TRUE)
train1<-train
summary(train)
```

```
##   fare_amount       pickup_datetime    pickup_longitude pickup_latitude
##  Min.   :    1.00   Length:15661       Min.   :-74.44   Min.   :-74.01
##  1st Qu.:    6.00   Class :character   1st Qu.:-73.99   1st Qu.: 40.74
##  Median :    8.00   Mode  :character   Median :-73.98   Median : 40.75
##  Mean   :   14.71                      Mean   :-73.91   Mean   : 40.69
##  3rd Qu.:   12.00                      3rd Qu.:-73.97   3rd Qu.: 40.77
##  Max.   :54343.00                      Max.   : 40.77   Max.   : 41.37
##  dropoff_longitude dropoff_latitude passenger_count
##  Min.   :-74.43    Min.   :-74.01   Min.   :1.000
##  1st Qu.:-73.99    1st Qu.: 40.74   1st Qu.:1.000
##  Median :-73.98    Median : 40.75   Median :1.000
##  Mean   :-73.91    Mean   : 40.69   Mean   :1.648
##  3rd Qu.:-73.97    3rd Qu.: 40.77   3rd Qu.:2.000
##  Max.   : 40.80    Max.   : 41.37   Max.   :6.000
```

Now all the Na values are imputed.



```
sum(is.na(train1))
```

```
## [1] 0
```

Feature Engineering for timestamp variable:

We will derive new features from pickup_datetime variable

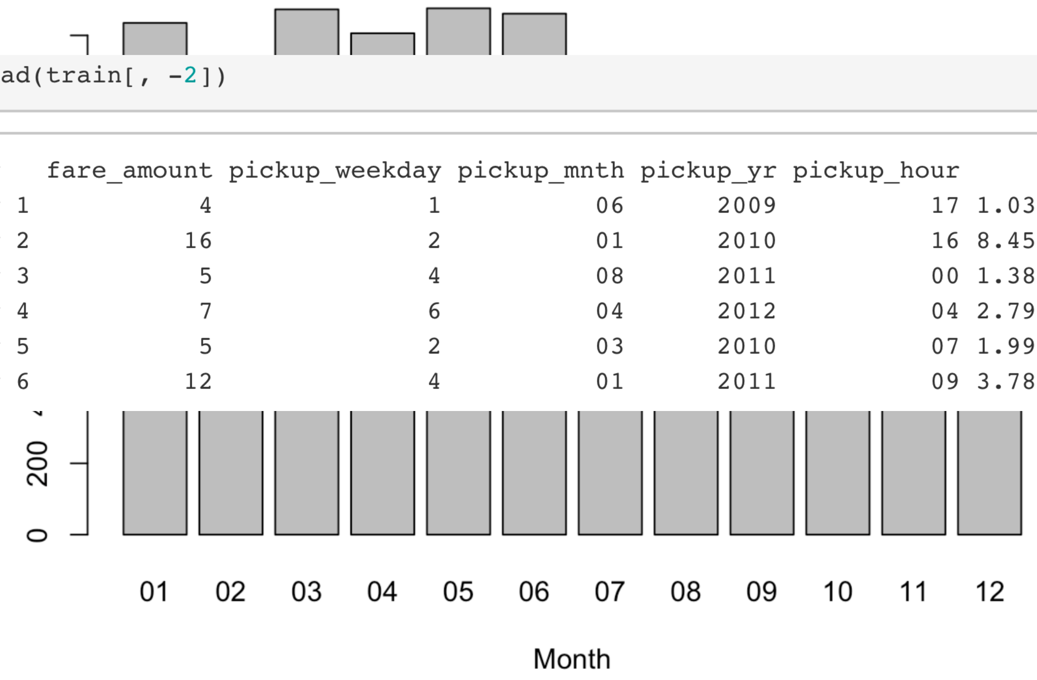
The new features will be year,month,day_of_week,hour

And we convert pickup_datetime from factor to date time.

```
str(train)
```

```
## 'data.frame':    15660 obs. of  7 variables:
##  $ fare_amount    : num  4 16 5 7 5 12 7 16 8 8 ...
##  $ passenger_count: int  1 1 2 1 1 1 1 1 1 2 ...
##  $ pickup_weekday : Factor w/ 7 levels "1","2","3","4",..: 1 2 4 6 2 4 2 3 1 3 ...
##  $ pickup_mnth    : Factor w/ 12 levels "01","02","03",..: 6 1 8 4 3 1 11 1 12 9 ...
##  $ pickup_yr      : Factor w/ 7 levels "2009","2010",..: 1 2 3 4 2 3 4 4 4 1 ...
##  $ pickup_hour    : Factor w/ 24 levels "00","01","02",..: 18 17 1 5 8 10 21 18 14 2 ...
##  $ dist           : num  1.03 8.45 1.39 2.8 2 ...
```

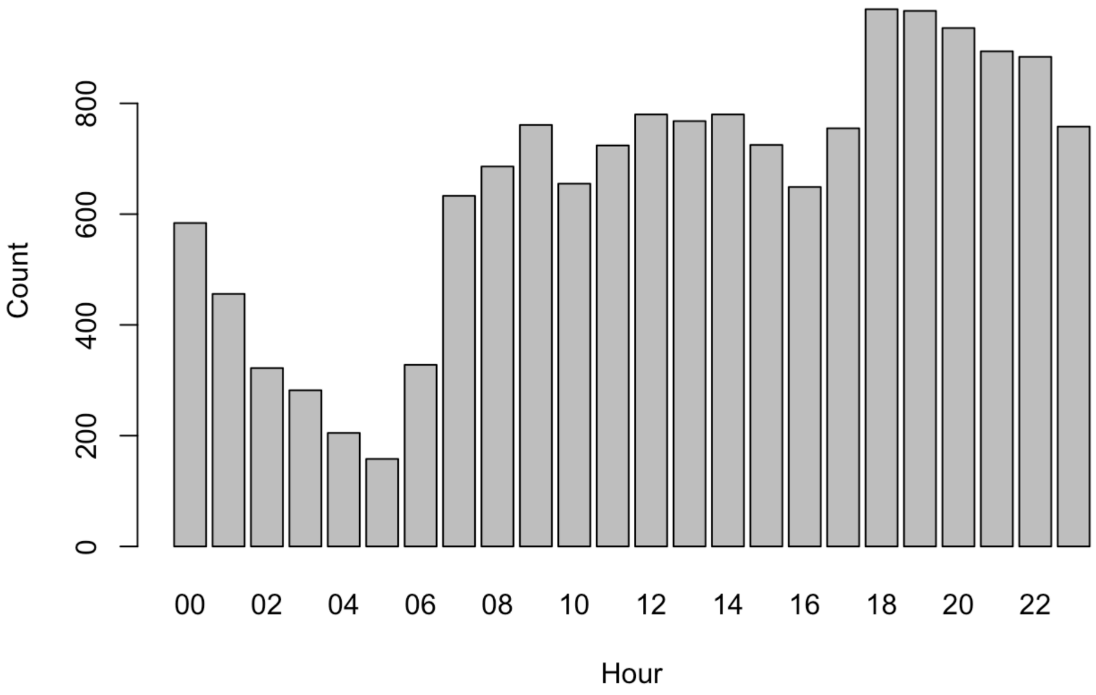
```
barplot(table(train$pickup_mnth),xlab="Month",
ylab="Count")
```



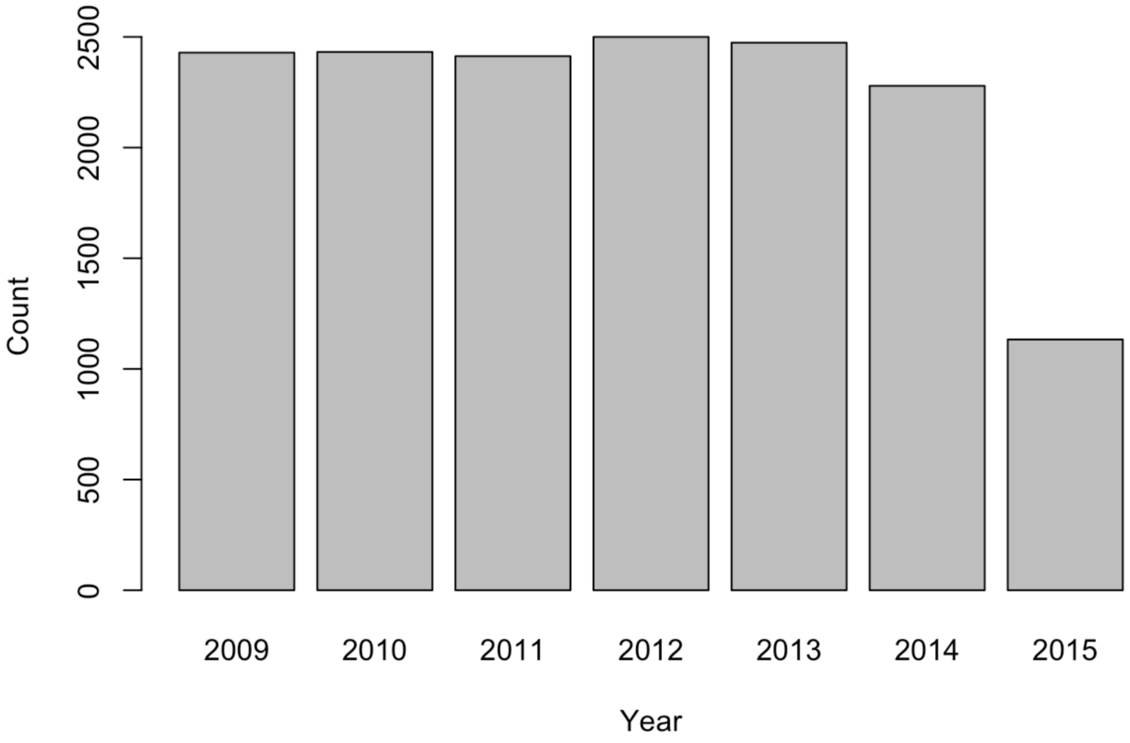
```
head(train[, -2])
```

```
##   fare_amount pickup_weekday pickup_mnth pickup_yr pickup_hour     dist
## 1           4              1          06      2009          17 1.030764
## 2          16              2          01      2010          16 8.450134
## 3           5              4          08      2011          00 1.389525
## 4           7              6          04      2012          04 2.799270
## 5           5              2          03      2010          07 1.999157
## 6          12              4          01      2011          09 3.787239
```

```
barplot(table(train$pickup_hour),xlab="Hour",
ylab="Count")
```



```
barplot(table(train$pickup_yr),xlab="Year",
ylab="Count")
```

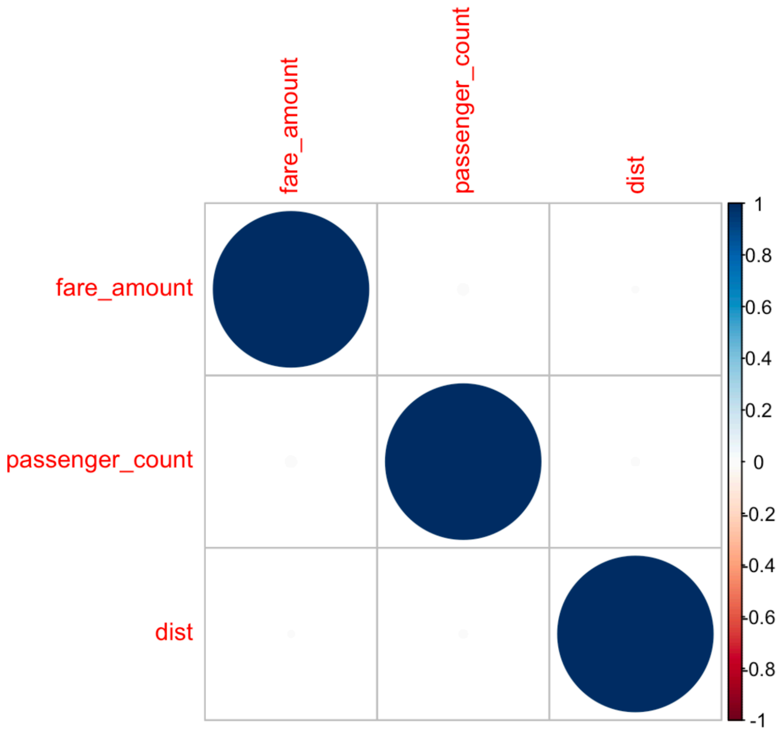


Correlation matrix:

- The correlation coefficient between fare_amount and fare_amount is 1, which is expected since a variable is perfectly correlated with itself.
- The correlation coefficient between fare_amount and passenger_count is -0.004277385, which indicates a very weak negative correlation between the two variables. This suggests that there is no meaningful relationship between the fare amount and the number of passengers in the taxi.
- The correlation coefficient between fare_amount and dist is 0.001274211, which indicates a very weak positive correlation between the two variables. This suggests that there is no meaningful relationship between the fare amount and the distance travelled in the taxi.
- The correlation coefficient between passenger_count and passenger_count is 1, which is expected since a variable is perfectly correlated with itself.
- The correlation coefficient between passenger_count and dist is 0.002100382, which indicates a very weak positive correlation between the two variables. This suggests that there is no meaningful relationship between the number of passengers in the taxi and the distance travelled.
- The correlation coefficient between dist and dist is 1, which is expected since a variable is perfectly correlated with itself.

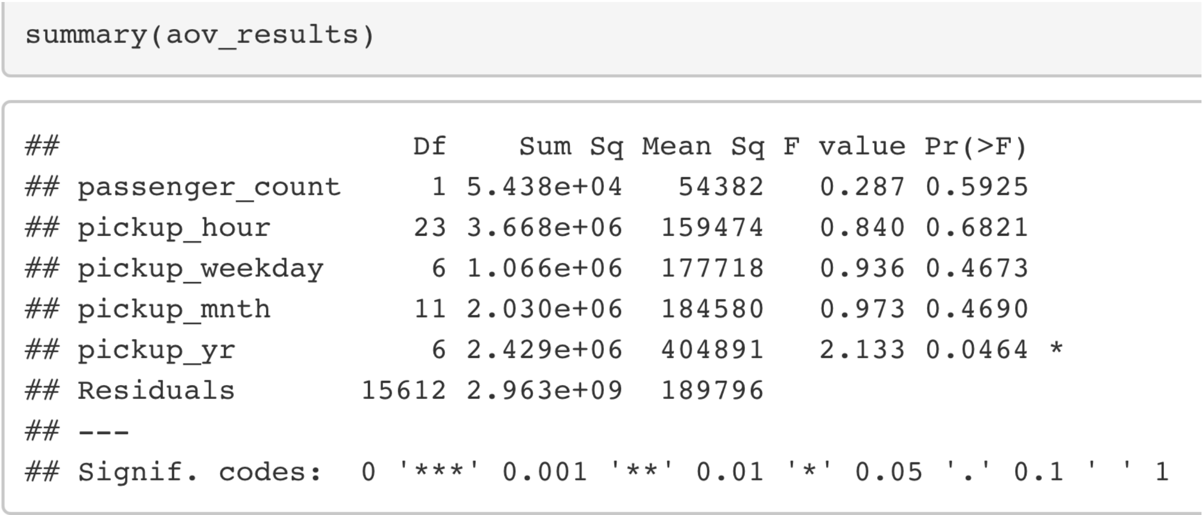
Inference: Based on the correlation matrix, there is no strong correlation between the fare_amount, passenger_count, and dist variables. This suggests that these variables are likely independent of each other and may not have a significant impact on each other's values. However, it's important to note that correlation does not imply causation, and other factors not included in the dataset may also be influencing the variables. Therefore, further analysis and modeling are required to fully understand the relationships between these variables.

```
# Create a correlation matrix plot
corrplot(correlation_matrix, method = "circle")
```



Anova results:

- The passenger_count variable has a p-value of 0.5925, which is larger than the significance level of 0.05. Therefore, we fail to reject the null hypothesis and conclude that there is no significant relationship between passenger_count and fare_amount.
- The pickup_hour, pickup_weekday, and pickup_mnth variables have p-values greater than the significance level of 0.05, indicating that there is no significant evidence to suggest that these variables have a significant impact on the fare_amount.
- The pickup_yr variable has a p-value of 0.0464, which is smaller than the significance level of 0.05. Therefore, we reject the null hypothesis and conclude that there is a significant relationship between pickup_yr and fare_amount. However, it's important to note that a p-value slightly below the significance level does not necessarily mean that the relationship is practically significant or useful for prediction purposes. Further analysis and modeling may be necessary to fully understand the relationship between pickup_yr and fare_amount.



```
summary(aov_results)
```

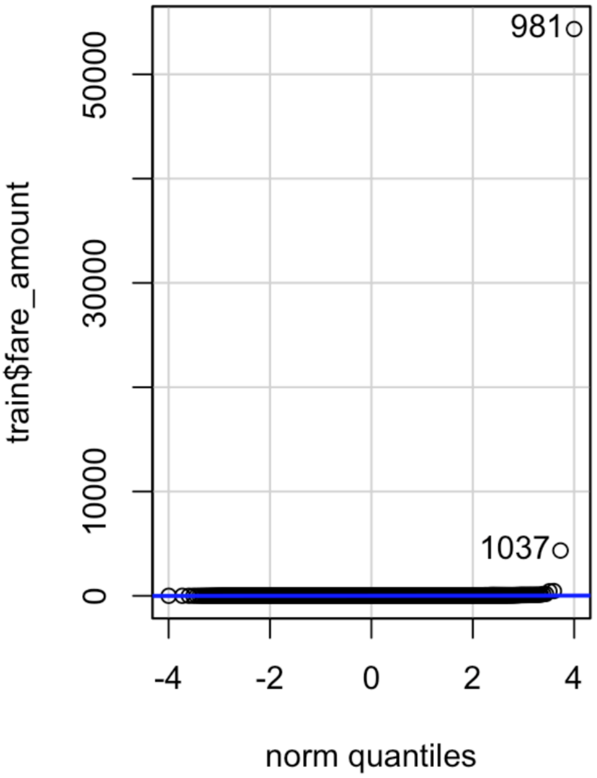
```
##                    Df    Sum Sq Mean Sq F value Pr(>F)  
## passenger_count     1 5.438e+04   54382   0.287 0.5925  
## pickup_hour        23 3.668e+06  159474   0.840 0.6821  
## pickup_weekday      6 1.066e+06  177718   0.936 0.4673  
## pickup_mnth        11 2.030e+06  184580   0.973 0.4690  
## pickup_yr           6 2.429e+06  404891   2.133 0.0464 *
## Residuals       15612 2.963e+09  189796                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov_results)
```

```
##                       Df    Sum Sq Mean Sq F value Pr(>F)
## passenger_count        1 5.437e+04   54374   0.286  0.593
## pickup_longitude       1 8.400e+01      84   0.000  0.983
## pickup_latitude        1 1.524e+03    1524   0.008  0.929
## dropoff_longitude      1 1.990e+02     199   0.001  0.974
## dropoff_latitude       1 3.191e+03    3191   0.017  0.897
## Residuals          15655 2.972e+09  189861
```

- The p-values for all the predictor variables (passenger_count, pickup_longitude, pickup_latitude, dropoff_longitude, and dropoff_latitude) are greater than the significance level of 0.05, indicating that none of these variables are statistically significant in explaining the variance in the response variable.
- The residual mean square (MSE) is 189861, which is a measure of the unexplained variance in the response variable.
- Therefore, we can conclude that none of the predictor variables are significant in predicting the response variable, and the model is not a good fit for the data.

qqPlot(train$fare_amount)



Inference: A straight line parallel to the x-axis in a QQ plot is an indication of normality of the data. It suggests that the data follows a normal distribution, which is an important assumption for many statistical tests and modeling techniques. However, it is important to note that a QQ plot is just a visual tool and other statistical tests should also be performed to confirm the normality of the data.

Algorithms used:

Decision Tree

Random forest

Gradient Boosting

Lasso Regression model

From metrics of these algorithms we can infer that,

Based on the RMSE and R-squared values, the random forest model seems to be the best performing model with an RMSE of 17.36495 and an R-squared value of 0.02855797. The decision tree model has a higher RMSE and a very low R-squared value, indicating poor performance. The gradient boosting model has a slightly better R-squared value than the decision tree model but still lower than the random forest model. The lasso regression model's RMSE value is lower than the other models, but since it is a regression model, we cannot compare its R-squared value with the other models.

Therefore, we can infer that the random forest model is the best model among the ones evaluated in terms of performance.

5. *Conclusion and future work:*

Considering what is and what is not accounted for in the models built in this study, their predicting results are fairly accurate. To further improve the prediction accuracy, more variabilities need to be considered and modeled. Although the rides in hour and average speed in hour work as proxies for traffic, more modeling on the effect of location is needed. These quantities could be calculated for different areas to further model local effects of traffic. Also, modeling traffic and the effect of location in between pickup and dropoff points should be considered as well as difference in drivers' speed. These further steps could be taken both by analyzing larger sets of the data to infer relationships and effects of location and traffic at different times, as well as aggregation with other datasets, as data on traffic, speed limitations, etc.

References:

[1] Chelliah, Balika J. "Taxi fare prediction system using key feature extraction in artificial intelligence." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.6 (2021): 3803-3808.

[2] Antoniades, Christophoros, Delara Fadavi, and A. F. J. Amon. "Fare and duration prediction: A study of New York city taxi rides." *Unpublished student paper* (2016): 104.

[3] Banerjee, Pallab, et al. "Predictive analysis of taxi fare using machine learning." *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol* (2020): 373-378.

[4] Shashank, H. "Data Analysis of Uber and Lyft Cab Services."

[5] Taruna, G. Venkat Sai, and P. Sriramyab. "Analyzing Ola Data for Precise Price Prediction Using XGBoost Technique Comparing with LASSO Regression." (2022).

[6] Hwang, Hyeonjun, Clifford Winston, and Jia Yan. "Measuring the Benefits of Ride-hailing Services to Urban Travelers: The Case of the San Francisco Bay Area." (2020).

[7] Upadhyay, Rishabh, and Simon Lui. "Taxi fare rate classification using deep networks." *EncontroPortuguês de InteligênciaArtificialAt, Portugal* (2017).

[8] Phiboonbanakit, Thananut, and TeerayutHoranont. "Analyzing Bangkok city taxi ride: reforming fares for profit sustainability using big data driven model." *Journal of Big Data* 8 (2021): 1-27.

[9] Liu, Ce, and Qiang Qu. "Trip fare estimation study from taxi routing behaviors and localizing traces." *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015.

[10] guyen Van, Son, et al. "Novel online routing algorithms for smart people-parcel taxi sharing services." *ETRI Journal* 44.2 (2022): 220-231.

[11] https://rpubs.com/AnkitRaj/TaxiFareAnalysis