CAB FARE PREDICTION

KARTIK KOHLI

13th August 2020

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

The project is about a cab company who has done its pilot project and now they are looking to predict the fare for their future transactional cases. As, nowadays there are number of cab companies like Uber, Ola, Meru Cabs etc. And these cab companies deliver services to lakhs of customers daily. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. In this case, earn most revenues.

So, it becomes really important estimate the fare prices accurately.

1.2 Data

The given data attributes are

The ,

Train_cab.csv

Test.csv

The datasets features are comprised of dependant and independant features.

Dependant Features:- fare_amont, which exists only in train data set.

Independent Features:- pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, passenger_count. And this features are common for both the dat sets.

Longitude and Latitude are the geometrical coordinates.

CHAPTER 2: PROCEDURE

According to industry standards, the process of Data Analyzing mainly includes 6 main steps and this process is abbreviated as CRISP DM Process, which is Cross-Industry Process for Data Mining. And the Six main steps of CRISP DM Methodology for developing a model are:

1. Business understanding
2. Data understanding
3. Data preparation/Data Preprocessing
4. Modeling
5. Evaluation
6. Deployment

And in this project, all the above steps are followed to develop the model.

2.1.1 Business Understanding

It is important to understand the idea of business behind the data set. The given data set is asking us to predict fare amount. And it really becomes important for us to predict the fare amount accurately. Else, there might be great loss to the revenue of the firm. Thus, we have to concentrate on making the model most efficient.

2.1.2 Data Understanding

To get the best results, to get the most effective model it is really important to our data very well. Here, the given train data is a CSV file that consists 7 variables and 16067 Observation. A snapshot of the data provided.

fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.84161	40.712278	1
16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1
5.7	2011-08-18 00:35:00 UTC	-73.982738	40.76127	-73.991242	40.750562	2
7.7	2012-04-21 04:30:42 UTC	-73.98713	40.733143	-73.991567	40.758092	1
5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1
12.1	2011-01-06 09:50:45 UTC	-74.000964	40.73163	-73.972892	40.758233	1

Table 2.1 : Train Data

The different variables of the data are:

- fare_amount : fare of the given cab ride.
- pickup_datetime : timestamp value explaining the time of ride start.
- pickup_longitude : a float value explaining longitude location of the ride start.
- pickup_latitude : a float value explaining latitude location of the ride start.
- dropoff_longitude: a float value explaining longitude location of the ride end.
- dropoff_latitude : a float value explaining latitude location of the ride end
- passenger_count : an integer indicating the number of passengers Following table explains how the variables are categorized.

Independent Variables
pickup_datetime
pickup_longitude
pickup_latitude
dropoff_longitude
dropoff_latitude
passenger_count

Dependent Variables
Fare_amount

Table 2.2 : Independent Variables

Table 2.3 : Dependent/Target Variable

From the given train data it is understood that, we have to predict fare amount, and other variables will help me achieve that, here pickup_latitude/longitude, dropoff_latitude/longitude this data are signifying the location of pick up and drop off. It is explaining starting point and end point of the ride.With these variables, we can calculate Distance . Passenger_count is another variable, that explains about how many people or passenger boarded the ride, between the pickup and drop off locations. And pick up date time gives information about the time the passenger is picked up and ride has started. But unlike pick up and drop off locations has start and end details both in given data. The time data has only start details and no time value or time related information of end of ride. As it seems the information of time is incomplete. So with this information we conclude that we use Mape instead of RMSE error Matrix.

Chapter 3

Methodology

3.1.1 Pre Processing

The work starts with data preprocessing, means looking at the data to get insights. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis.

To start this process we will first try and look at all the distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the distributions of the variable by QQ-normality graph or histogram. Histogram is the best chart to represent the data distribution, later the data is normalized or standardized, according to the model needs.

And we check for distribution of the target variable with respect to its independent features.

Data pre processing is the tedious task, we need to focus more on this part to reduce the model complexity. Before feeding the data to model, we must preprocess the data, which has various stages like missing value analysis, impute the missing values, outlier check, normality check, sampling, weather the dataset is imbalanced are not. Then the data is subjected to model training and prediction.

After completing the model prediction, the machine learning prediction system is ready for deployment. Project deployment refers to , preparing the machine learning environment to the front end users.

The Incorrect information.

Exploring the data, which means looking at all the features and knowing their characters. According to our problem statement.

We are analysing the data of cab rental startup, whose features are comprised of fare amount, passenger count and geometrical coordinates like pickup latitude,longitude and drop off longitude and latitude.

So keeping in mind that these variables play an important role in the analysis, we keep checking the minimum and maximum values in this features.

There is one incorrect information in pickup_datetime i.e 43.

There were a lot of incorrect information in this feature, like observation showed up passenger count as 0 and 55 , 5000...and few showed decimal values 1.3

And fare amount ranged from 1 to 400+ and its normal, few showed high as 4000 and 5000...etc , which is not possible.

Latitudes range from -90 to 90.Longitudes range from -180 to 180. So Removing those features, which does not satisfy these ranges. Many showed 0 .

So these residual information must be removed from data set, this is known as data cleaning.

Many observations nearly 400 to 600 has this wrong information, so considering this as **OUTLIERS** in the data and removing it.

Deriving the new features year,day, date, month,min,hour by timestamp variable pickup_datetime.

Deriving the new feature distance from the given coordinates with the help of **haversine function.**

Observation

- There are few missing values, nearly 78 NA values.
- With respect to our problem statement, we observed many incorrect data in our features and deleted/cleaned it.(That is in form of outlier.)
- With timestamp features and coordinates we created new features, like year, month,day,hour,date,distance.
- The above data visualizations shows the characters of the features in their respective domain.
- The features like distance in both the datasets and fare amount in the training set must we normalized with log transformation or any other normalization method.
- The dependent feature is Fare_amount.
- The type of machine learning model required to this problem is a REGRESSION MODEL, which can predict the continuous outcome.

Few of the regression model are

- Linear regression
- Decision tree regression
- Random forest regressor
- Lasso Regression

3.1.2 Feature Engineering.

It is a part of data pre-processing, after cleaning the data.

The term Feature Engineering refers to transforming the features available in the dataset.

Transforming in sense, preparing the features ready to use for model building . This includes

Missing Values.

There are few missing values in train dataset.

Nearly 78 missing values, but these missing values are omitted, as they are in very low composition.

The reason for deleting the missing values is , while cleaning the data we encountered many wrong data points nearly 500 to 600 data points, which doesn't make sense for this particular problem statement.

As the missing values is in very low proportion ,decided to delete it.

Outliers

We are analysing the data of cab rental startup, whose features are comprised of fare amount, passenger count and geometrical coordinates like pickup latitude,longitude and drop off longitude and latitude.

So keeping in mind that these variables play an important role in the analysis, we keep checking the minimum and maximum values in this features.

There were a lot of incorrect information in this feature, like observation showed up passenger count as 0 and 55 , 5000...and few showed decimal values 1.3 ...even if we consider suv or sumo max seater is 6.

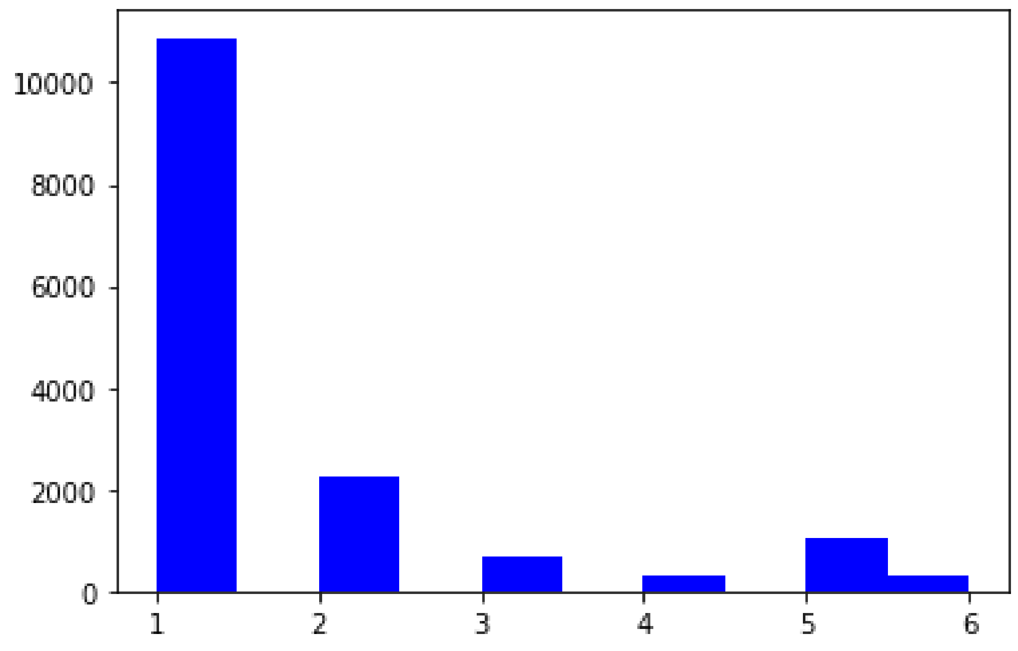
And fare amount ranged from 1 to 400+ and its normal, few showed high as 4000 and 5000...etc , which is not possible.

Latitudes range from -90 to 90.Longitudes range from -180 to 180. So Removing those features, which does not satisfy these ranges. Many showed 0 .

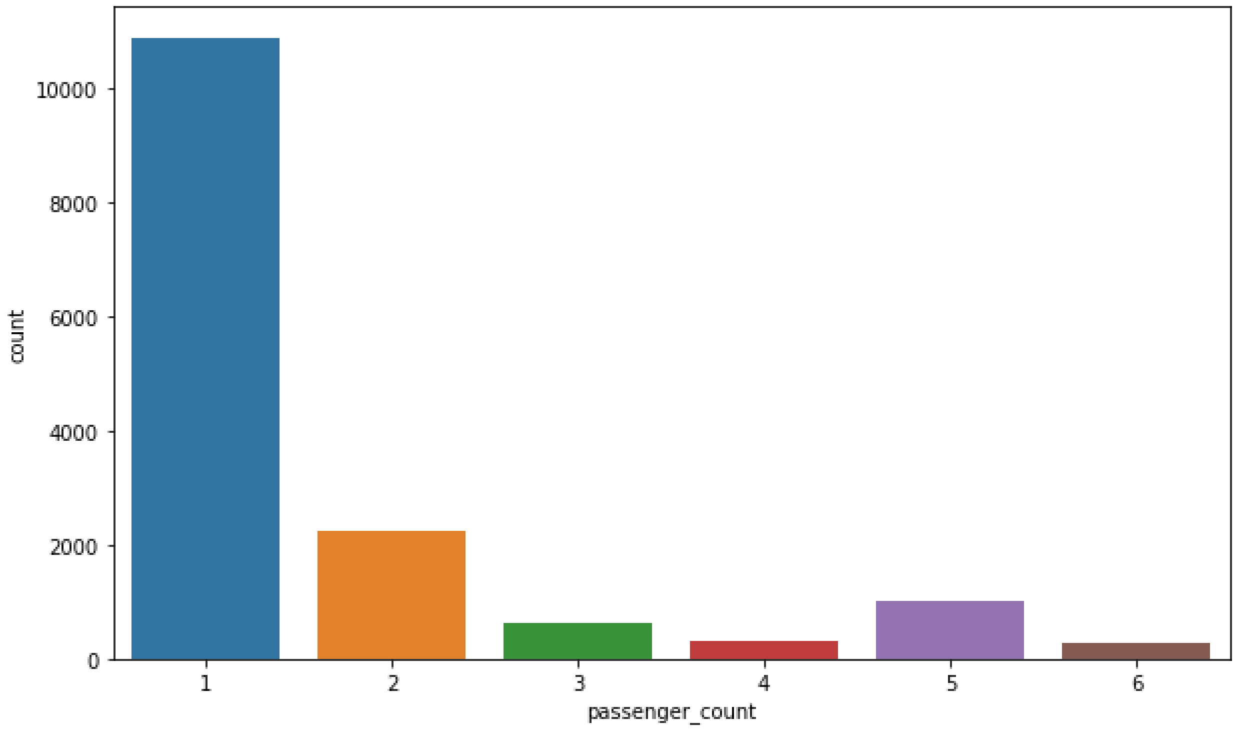
So these residual information must be removed from data set, this is known as data cleaning.

Many observations nearly 400 to 600 has this wrong information, so considering this as OUTLIERS in the data and removing it.

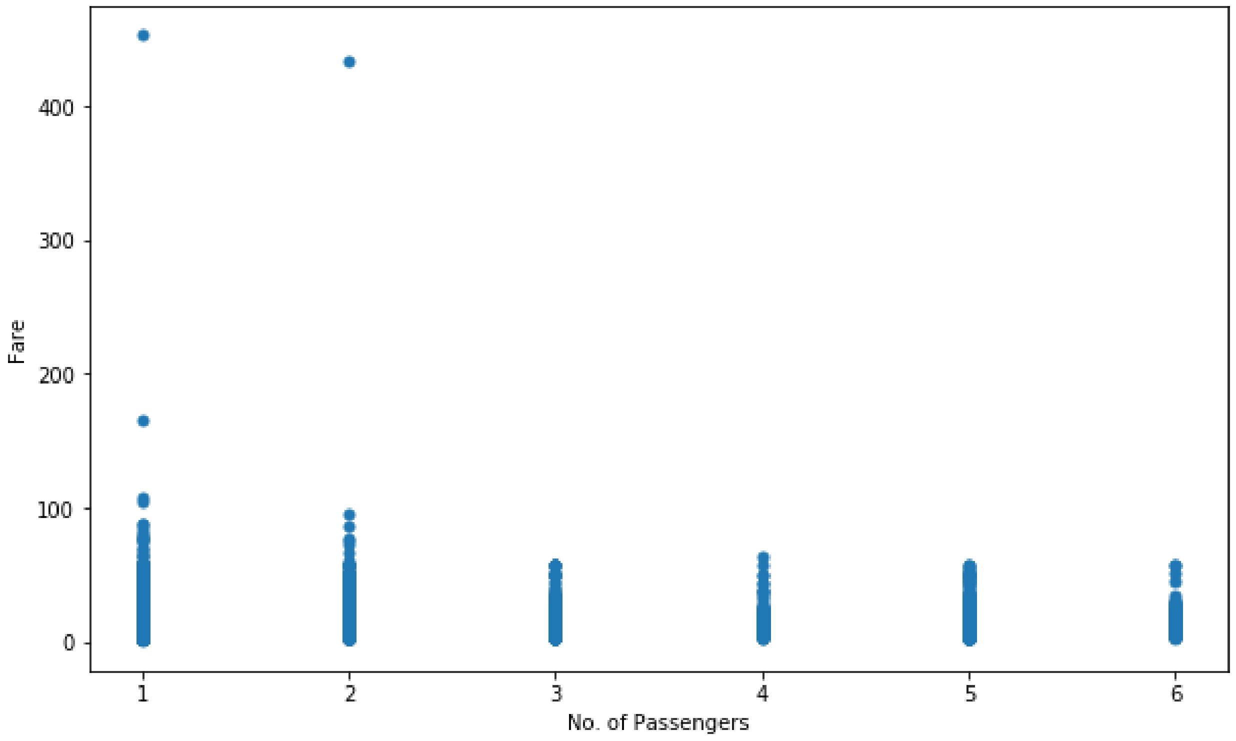
Data Visualization



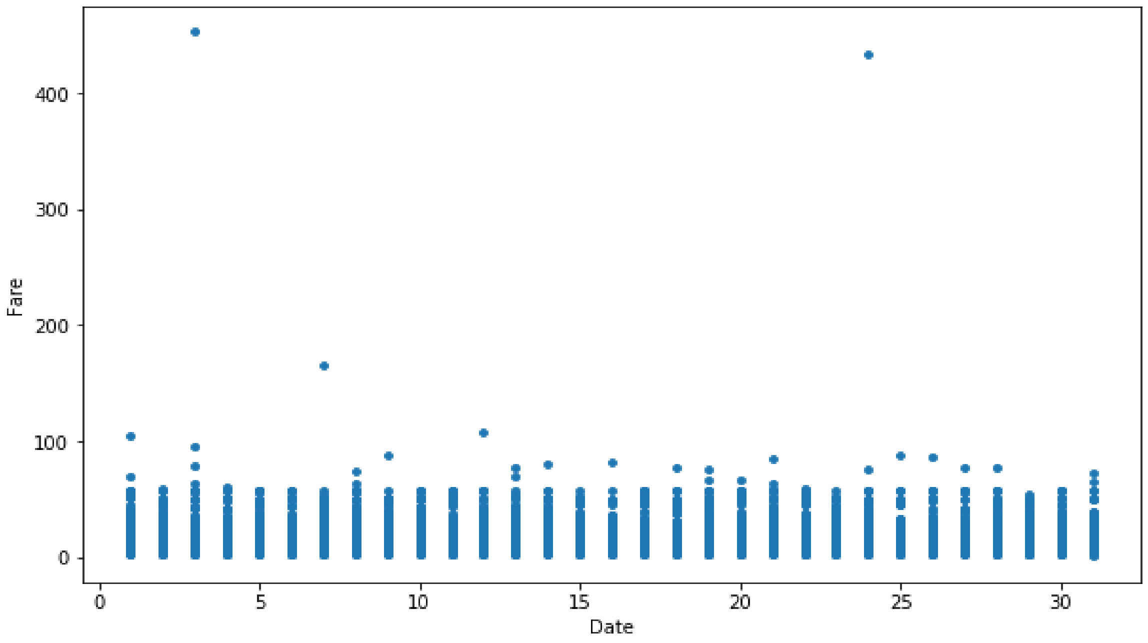
There are many single travelling passengers, followed by 2,5,3,4,6.



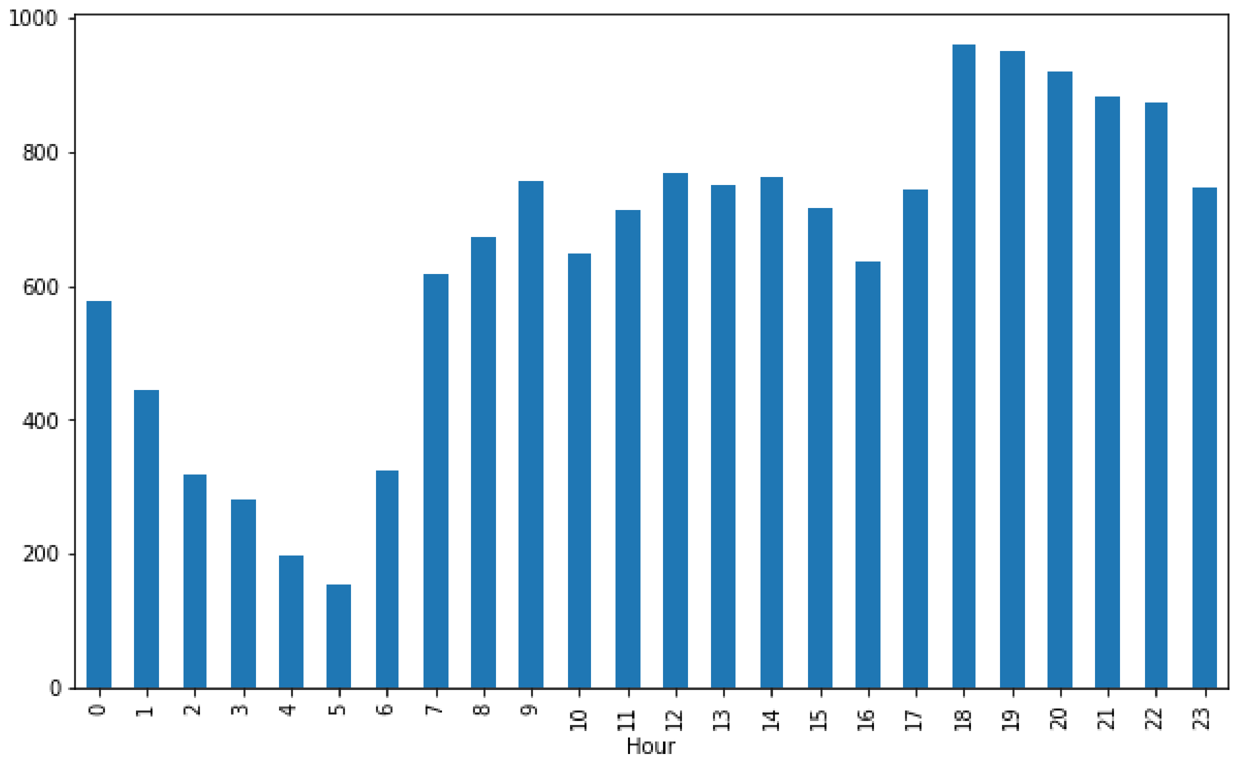
People travelling alone are quite more as compared to group travel



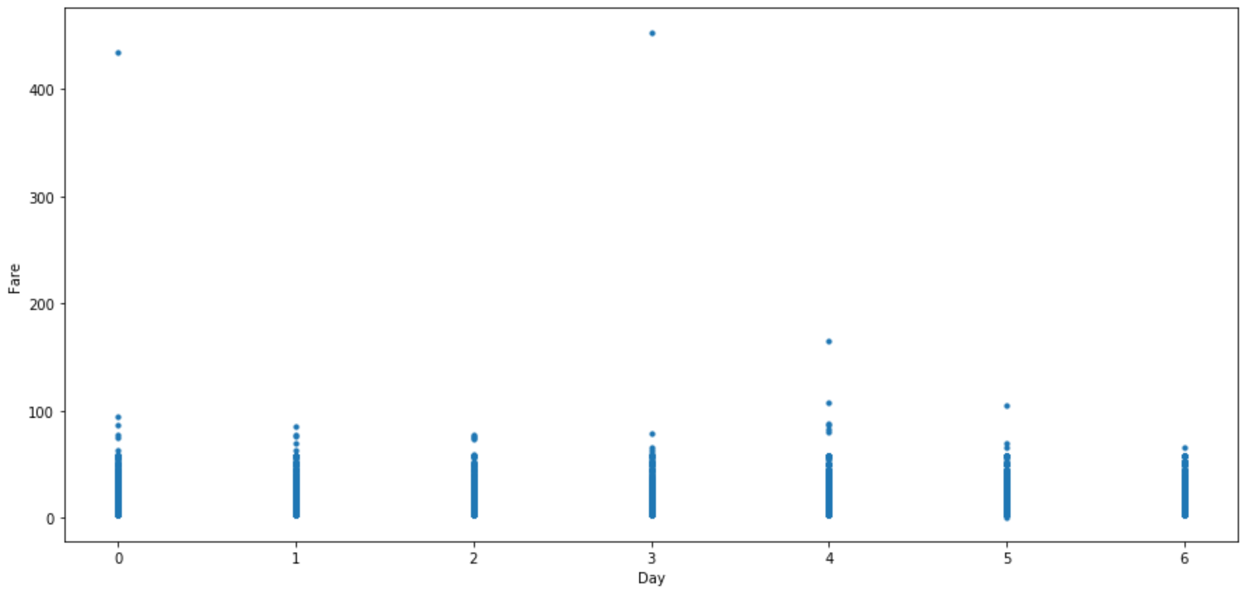
Single travelling passengers contribute a lot to the fare amount.



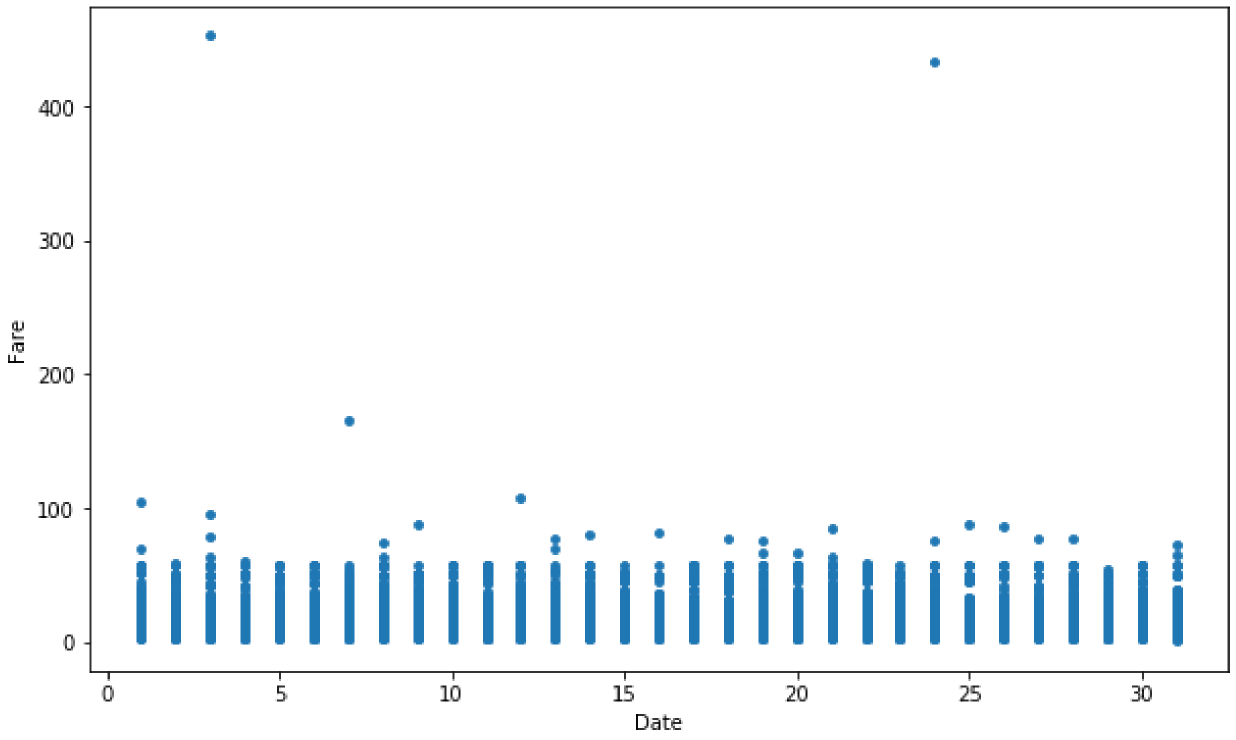
- All dates have high demand of cabs, as it is common.
- Commute is common!!!

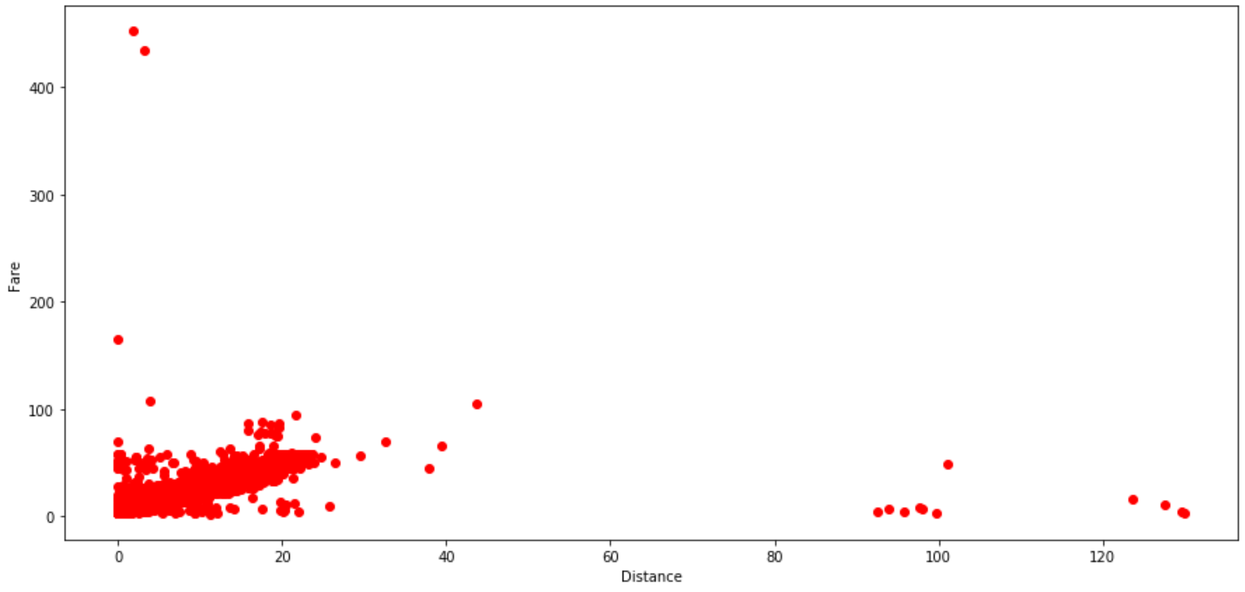


There is a high demand for cabs in the early morning and from night to midnight.



- There is a rise in fare amount on Monday,Thursday and Friday.
- As monday is the starting day of the week and thursday and friday are weekend rush.

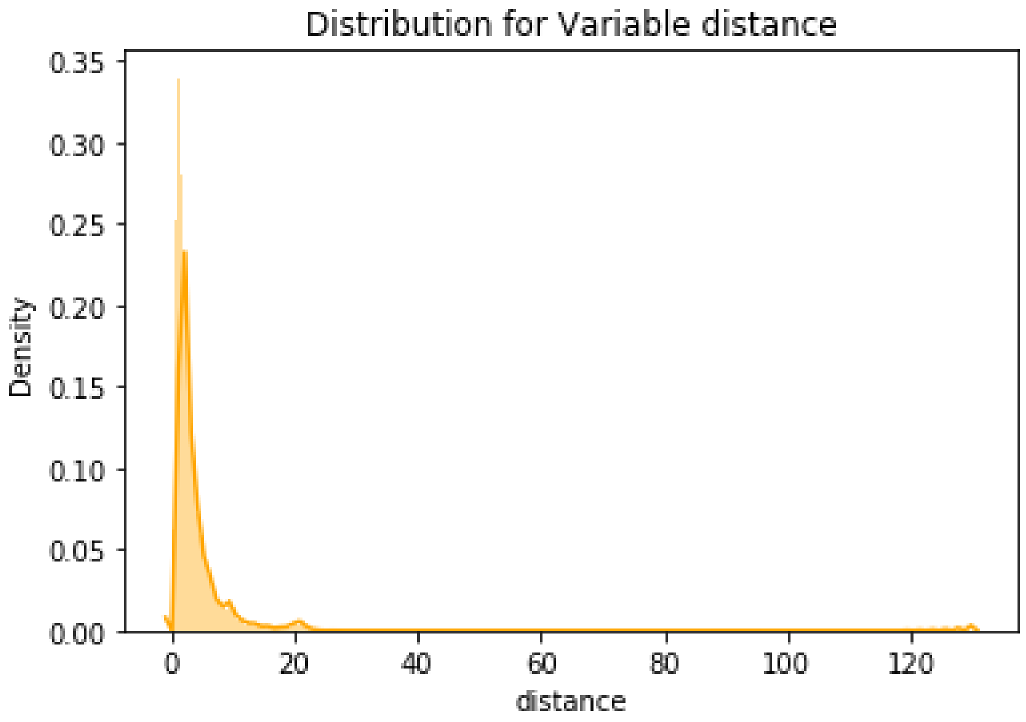


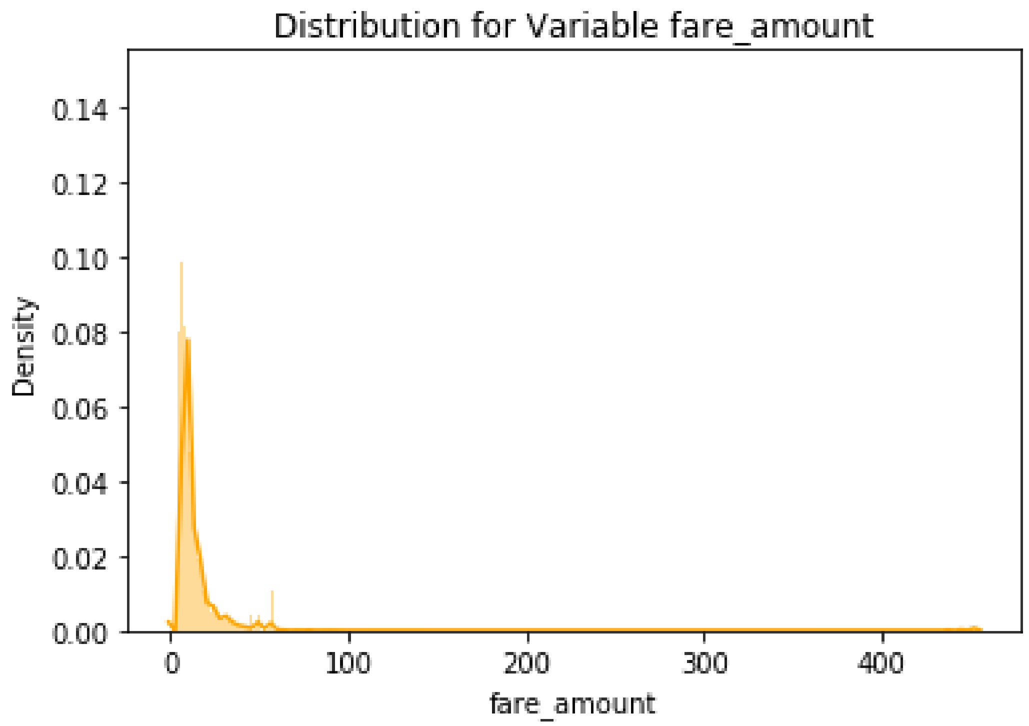


It is quite certain that fare is directly proportional to distance

Distribution of the features.

Let's check the distribution of the features.





The information in fare amount and distance of train dataset are skewed

3.1.3 Feature Scaling

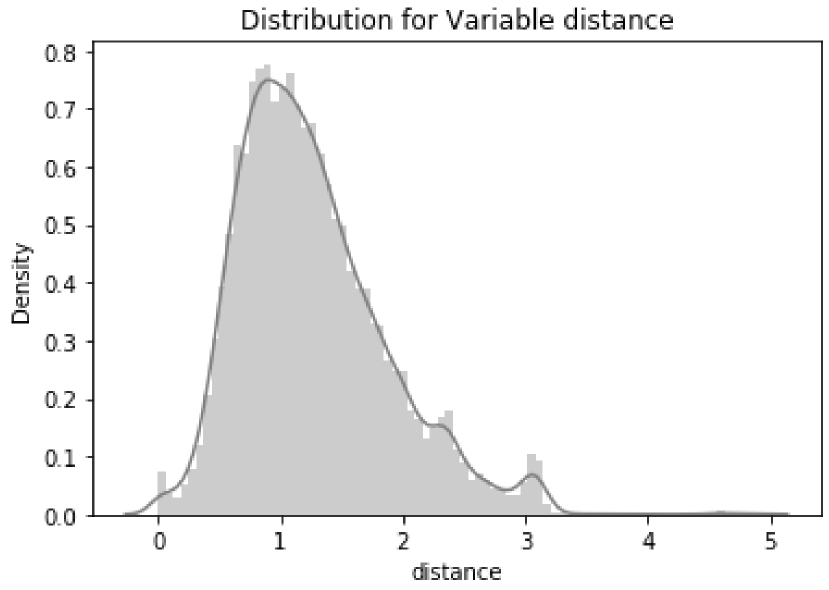
Feature scaling is the process of dealing with skewed data, making the feature to set with in a scale of 0-10, by applying the log transformation.

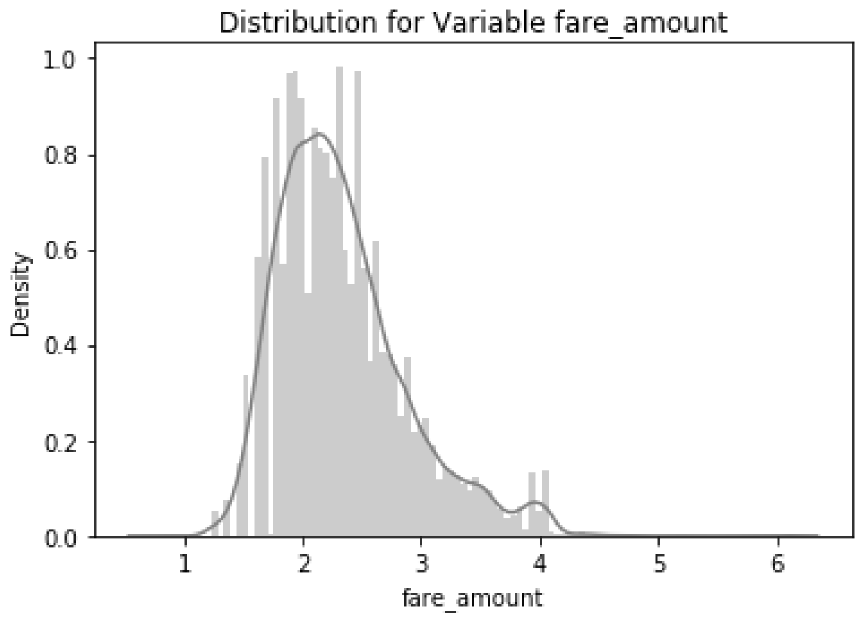
Data Transformation :-

In statistics, data transformation is the application of a deterministic mathematical function to each point in a data set — that is, each data point z_i is replaced with the transformed value $y_i = f(z_i)$, where f is a function. Transforms are usually applied so that the data appear to more closely meet the assumptions of a statistical inference procedure that is to be applied, or to improve the interpretability or appearance of graphs.

Log Transformation:-

The log transformation can be used to make highly skewed distributions less skewed. This can be valuable both for making patterns in the data more interpretable and for helping to meet the assumptions of inferential statistics.





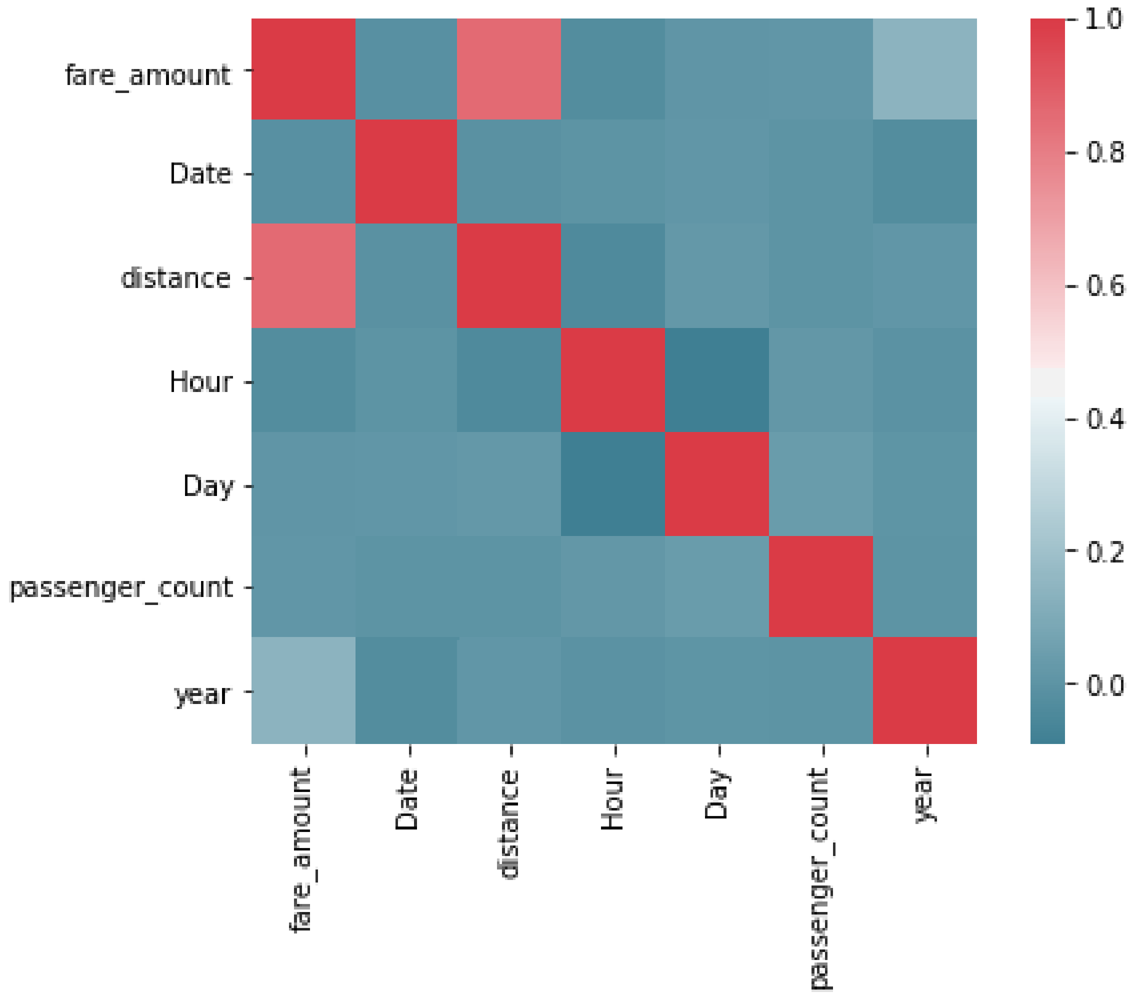
3.1.4 Feature selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction.

For all the numerical variable present in the training data set.

The objective of this selection is to neglect the features with high correlation.

There are several methods of doing that. Below we have used the correlation plot for checking multicollinearity and level 1 regularization (Lasso Regression) Here we use correlation plot.



With this plot, it is difficult to understand the collinearity with in the variables. So using feature selection embedded method.i.e lasso regression.

VIF

```
vif(train[,-1])

      Variables      VIF

1 passenger_count 1.000401

2          month 1.014897

3           year 1.014218

4       distance 1.001237

> vifcor(train[,-1], th = 0.9)

No variable from the 4 input variables has collinearity problem.

The linear correlation coefficients ranges between:

min correlation ( year ~ passenger_count ):  0.01069526

max correlation ( year ~ month ):  -0.1206376

---------- VIFs of the remained variables --------

      Variables      VIF

1 passenger_count 1.000829

2          month 1.015980

3           year 1.015592

4       distance 1.001554
```

What is Regularization?

Regularization is a way to avoid overfitting by penalizing high-valued regression coefficients. In simple terms, it reduces parameters and shrinks (simplifies) the model. This more streamlined, more parsimonious model will likely perform better at predictions. Regularization adds penalties to more complex models and then sorts potential models from least overfit to greatest; The model with the lowest "overfitting" score is usually the best choice for predictive power.

Level 1 regularization(Lasso Regression).

Lasso(Least Absolute Shrinkage and Selection Operator)

L1 regularization adds an L1 penalty equal to the absolute value of the magnitude of coefficients. In other words, it limits the size of the coefficients. L1 can yield sparse models (i.e. models with few coefficients); Some coefficients can become zero and eliminated. Lasso regression uses this method.

Lasso regression is the simple techniques to reduce model complexity and prevent over-fitting which may result from simple linear regression.

Chapter 4

4.1 Data Modeling

4.1.1 Model selection

Based on the target variable or feature we select the model. Here in our case cab rentals data set, the target variable is comprised of continuous distribution, so the model will be the regression model, to predict continuous outcomes.

Regression Model

In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features').

Regression analysis is primarily used for two conceptually distinct purposes. First, regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Second, in some situations regression analysis can be used to infer causal relationships between the independent and dependent variables. Importantly, regressions by themselves only reveal relationships between a dependent variable and a collection of independent variables in a fixed dataset. To use regressions for prediction or to infer causal relationships, respectively, a researcher must carefully justify why existing relationships have predictive power for a new context or why a relationship between two variables has a causal interpretation. The latter is especially important when a researcher hopes to estimate causal relationships using observational data.

4.1.2 Linear Regression model

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Dep. Variable:	fare_amount	**R-squared (uncentered):**	0.986
Model:	OLS	**Adj. R-squared (uncentered):**	0.986
Method:	Least Squares	**F-statistic:**	1.251e+05
Date:	Wed, 12 Aug 2020	**Prob (F-statistic):**	0.00
Time:	00:34:28	**Log-Likelihood:**	-1924.3
No. Observations:	12338	**AIC:**	3863.
Df Residuals:	12331	**BIC:**	3915.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
passenger_count	0.0038	0.002	1.907	0.057	-0.000	0.008
year	0.0007	5.78e-06	114.230	0.000	0.001	0.001
Month	0.0037	0.001	4.981	0.000	0.002	0.005
Date	-0.0002	0.000	-0.684	0.494	-0.001	0.000
Day	-0.0031	0.001	-2.403	0.016	-0.006	-0.001
Hour	0.0010	0.000	2.535	0.011	0.000	0.002
distance	0.7728	0.004	184.465	0.000	0.765	0.781

Omnibus:	6132.979	**Durbin-Watson:**	1.992
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	512981.280
Skew:	1.500	**Prob(JB):**	0.00
Kurtosis:	34.446	**Cond. No.**	3.31e+03

linear_model=lm(fare_amount~.,data=Train)

summary(linear_model)

```
Call:
lm(formula = fare_amount ~ ., data = Train)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5455 -0.1461 -0.0312  0.1094  3.9814 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -78.857805   2.453291 -32.144  < 2e-16 ***
passenger_count   0.004642   0.001789   2.594  0.00949 ** 
mnth              0.005681   0.000661   8.594  < 2e-16 ***
yr                0.039853   0.001219  32.685  < 2e-16 ***
distance          0.770444   0.003721 207.033  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2668 on 13872 degrees of freedom
Multiple R-squared:  0.7615,	Adjusted R-squared:  0.7614 
F-statistic: 1.107e+04 on 4 and 13872 DF,  p-value: < 2.2e-16
```

4.1.3 Decision Tree Regressor

The decision trees is used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve.

We can see that if the maximum depth of the tree (controlled by the max_depth parameter) is set too high, the decision trees learn too fine details of the training data and learn from the noise, i.e. they overfit.

Decision Tree Regression:

Multiple linear regression is the most common form of linear regression analysis. Multiple regression is an extension of simple linear regression. It is used as a predictive analysis, when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable).

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented by a discrete, known set of numbers or values.

```
DecisionTreeRegressor(criterion='mse', max_depth=3, max_features=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0,
presort=False, random_state=None, splitter='best')
```

4.1.4 Random Forest Regression

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Approach :

Pick at random K data points from the training set.

Build the decision tree associated with those K data points.

Choose the number Ntree of trees you want to build and repeat step 1 & 2.

For a new data point, make each one of your Ntree trees predict the value of Y for the data point, and assign the new data point the average across all of the predicted Y values.

4.1.5 Lasso Regression model

In statistics and machine learning, lasso (least absolute shrinkage and selection operator) (also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.

Lasso is able to achieve both of these goals by forcing the sum of the absolute value of the regression coefficients to be less than a fixed value, which forces certain coefficients to be set to zero, effectively choosing a simpler model that does not include those coefficients. This idea is similar to ridge regression, in which the sum of the squares of the coefficients is forced to be less than a fixed value, though in the case of ridge regression, this only shrinks the size of the coefficients, it does not set any of them to zero.

```
Lasso(alpha=0.005, random_state=0)

predict_lasso=lasso_model.predict(X_test)
```

4.1.6 Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

4.1.7 Hyper Parameters Tunings for optimizing the results

Model hyperparameters are set by the data scientist ahead of training and control implementation aspects of the model. The weights learned during training of a linear regression model are parameters while the number of trees in a random forest is a model hyperparameter because this is set by the data scientist.

Hyperparameters can be thought of as model settings. These settings need to be tuned for each problem because the best model hyperparameters for one particular dataset will not be the best across all datasets. The process of hyperparameter tuning (also called hyperparameter optimization) means finding the combination of hyperparameter values for a machine learning model that performs the best - as measured on a validation dataset - for a problem.

Here we have used two hyper parameters tuning techniques

- Random Search CV
- Grid Search CV

1. **Random Search CV:** This algorithm set up a grid of hyperparameter values and select random combinations to train the model and score. The number of search iterations is set based on time/resources.
2. **Grid Search CV:** This algorithm set up a grid of hyperparameter values and for each combination, train a model and score on the validation data. In this approach, every single combination of hyperparameters values is tried which can be very inefficient.

Chapter 5 Conclusion

5.1.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability

3. Computational Efficiency

Evaluating a model is a core part of building an effective machine learning model There are several evaluation metrics, like confusion matrix, cross-validation, AUC-ROC curve, etc. Different evaluation metrics are used for different kinds of problems.

5.1.2 MAPE

Here in regression model the evaluation is done by MAPE. RMSE is not used because dropoff_Datetime is not given in the data.

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation, also used as a loss function for regression problems in machine learning. It usually expresses accuracy as a percentage, by Multiplying 100 to percentage error.

For R

```
mape=function(av,pv){

mean(abs((av-pv)/av))*100 #av=actual value and pv= predicted value

}
```

For python

```
#mape              #av= actual value and pv= predicted value

def mape(av, pv):

mape = np.mean(np.abs((av - pv) / av))*100 return mape
```

From above we can define the mape function in both R and Python. The internal mape operation is

$$\frac{\sum_{t=1}^{n} \left| \frac{(Y_t - \hat{Y}_t)}{Y_t}(100) \right|}{n}$$

5.1.3 **Regr.eval function available in R**

Calculate Some Standard Regression Evaluation Statistics

This function is able to calculate a series of regression evaluation statistics given two vectors: one with the true target variable values, and the other with the predicted target variable values.

regr.eval(trues, preds, stats = if (is.null(train.y)) c("mae","mse","rmse","mape") else

c("mae","mse","rmse","mape","nmse","nmae"), train.y = NULL)

The regression evaluation statistics calculated by this function belong to two different groups of measures: absolute and relative. The former include "mae", "mse", and "rmse" and are calculated as follows:

"mae": mean absolute error, which is calculated as sum(|t_i - p_i|)/N, where t's are the true values and p's are the predictions, while N is supposed to be the size of both vectors.

"mse": mean squared error, which is calculated as sum((t_i - p_i)^2)/N "rmse": root mean squared error that is calculated as sqrt(mse)

The remaining measures ("mape", "nmse" and "nmae") are relative measures, the two later comparing the performance of the model with a baseline. They are unit-less measures with values always greater than 0. In the case of "nmse" and "nmae" the values are expected to be in the interval [0,1] though occasionally scores can overcome 1, which means that your model is performing worse than the baseline model. The baseline used in our implementation is a constant model that always predicts the average target variable value, estimated using the values of this variable on the training data (data used to obtain the model that generated the predictions), which should be given in the parameter train.y. The relative error measure "mape" does not require a baseline. It simply calculates the average percentage difference between the true values and the predictions.These measures are calculated as follows:

"mape": sum(|(t_i - p_i) / t_i|)/N

"nmse": sum((t_i - p_i)^2) / sum((t_i - AVG(Y))^2), where AVG(Y) is the average of the values provided in vector train.y

"nmae": sum(|t_i - p_i|) / sum(|t_i - AVG(Y)|)

MAPE TEST

Model	Mape value(%)	Accuracy(%)
Linear Regression	7.88	92.12
Decision Tree	7.93	92.15
Random Forest	7.34	92.61
Lasso	7.48	92.52

s

Model	Parameters	Mape value	Accruacy(%)
Random Search CV	Random Forest	7.20	92.80
	Gradient Boosting	8.27	91.73
Grid search CV	Random Forest	7.19	92.81
	Gradient Boosting	7.81	92.19

Above table shows the results after tuning the parameters of our two best suited models i.e. Random Forest and Gradient Boosting. For tuning the parameters, we have used Random Search CV and Grid Search CV under which we have given the range of n_estimators, depth and CV folds.

Regr.eval function from R

Model/Regr.eval	mae	mse	rmse	mape
Linear Regression	0.17510848	0.08089190	0.28441501	0.07778767
Decision Tree	0.19292435	0.07946429	0.28189411	0.08617598
Random Forest	0.22312437	0.09489477	0.30804995	0.10069728
xgboost	0.16544827	0.06464456	0.25425294	0.07449069

5.1.3 Model Selection(Observation)

- On the basis of mape results a good model should have least mape value. So, from above tables we can see:
- Random Forest perform comparatively well while comparing their MAPE Values with other models.
- After this, I chose Random Forest CV and Grid Search CV to apply cross validation technique and see changes brought about by that.

- So finally, we can say that Random forest model is the best method to make prediction for this project with parameter tuning technique Grid Search CV.
- We select the model with less errors and high accuracy.