Flight_Price Capstone

Nitin Yadav – PGP-BABI April Batch

Table of Contents

List of Figures

1. Introduction

We have seen that any individual who has booked a flight ticket previously knows how dynamically flight price change. Aircraft uses advanced strategies like Revenue Management to execute a distinctive valuing strategy. This valuing method naturally modifies the price as per the time of day like morning, afternoon or night. Cost may likewise change with the seasons like winter, summer and celebration seasons. The intent of the airline is to build its revenue and income and on the opposite side purchaser is searching for the price with minimum cost. Purchasers generally endeavor to purchase the flight ticket in advance from the actual takeoff day. Since they trust that airfare will be most likely high when the date of buying a ticket is closer to the takeoff date, yet it is not generally true always. Airline industries uses advance machine learning techniques for fair price and as a result purchaser may finish up with the paying more than they ought to for a similar seat even if they have purchased in advance.

1.1. Problem Statement

Airline industries are in continuous tussle to get more and more customers and in turn are working on very thin margins. The price of flight tickets are very unpredictable considering the dynamic nature of business and governing the law of demand and supply. At times we have noted that for a particular city or destination when we search for flight price , the price keeps getting dynamically updated depending on the search criteria, seat availability, date and time of travel etc. Hence it becomes very important for the Airline industry to have a right price prediction mechanism which is backed up by data and helps the industry to take a data driven decision.

1.2. Need of Study Project

This is a problem of machine learning where we have been given 2 data sets i.e Train and Test set. Train data consist of 10683 records and Test data consist of 2671 records

1.3. Understanding business/social opportunity

This is a Machine learning problem based on supervised learning. Here we train the algorithm using the Train dataset. In supervised machine learnng we know the Target variable and we try to identify the kep predictors on which the response variable (Y) is dependent. Based on the trained machine model, we then try to predict the target using a Test set. This is very crucial in Airline industry as price of a flight is very important parameter for a customer to take the travel decision and a right price point will be beneficial for both customer and the Airline company. Hence the better the Machine learning model, better would be the accuracy and hence minimum error.

2. Machine Learning Process and Methodology

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. **The primary aim is to allow the computers learn automatically** without human intervention or assistance and adjust actions accordingly.

Machine learning can be classified into three broad categories:

1. Supervised learning – Also called predictive learning. A machine predicts the class of unknown objects based on prior class-related information of similar objects. In case of numeric variables, it tries to predict the continuous value.
2. Unsupervised learning – Also called descriptive learning. A machine finds patterns in unknown objects by grouping similar objects together.
3. Reinforcement learning – A machine learns to act on its own to achieve the given goals.

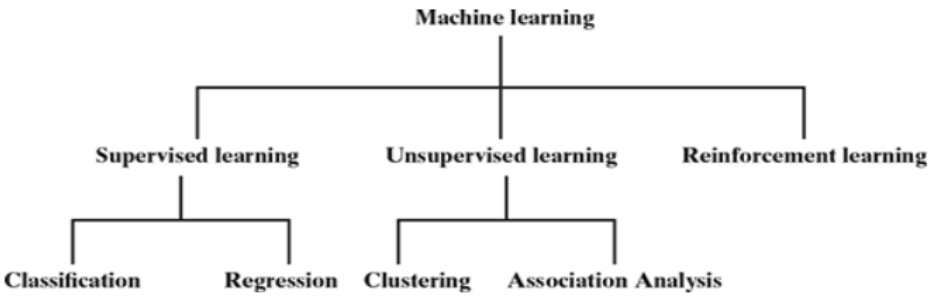


Figure 1: Types of Machine Learning

Machine learning algorithms are often categorized as supervised, unsupervised or Reinforcement.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

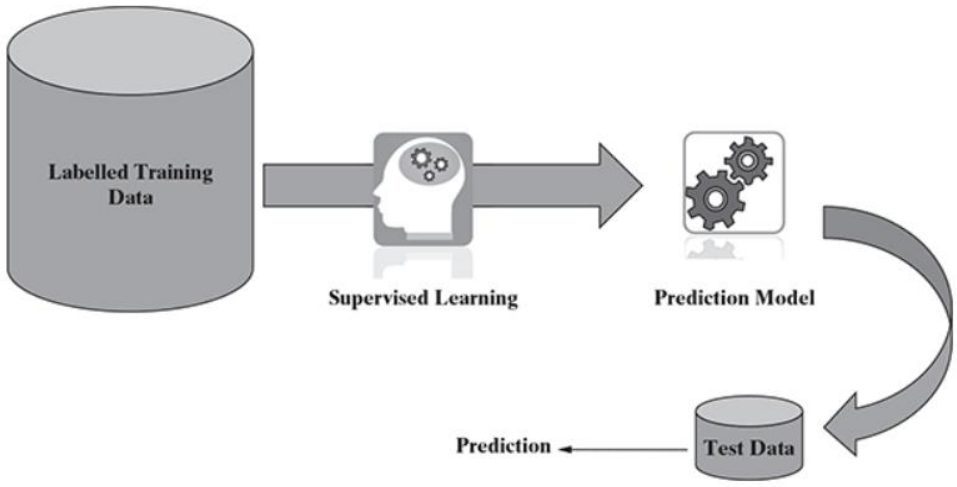


Figure 2: Supervised Machine Learning

In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

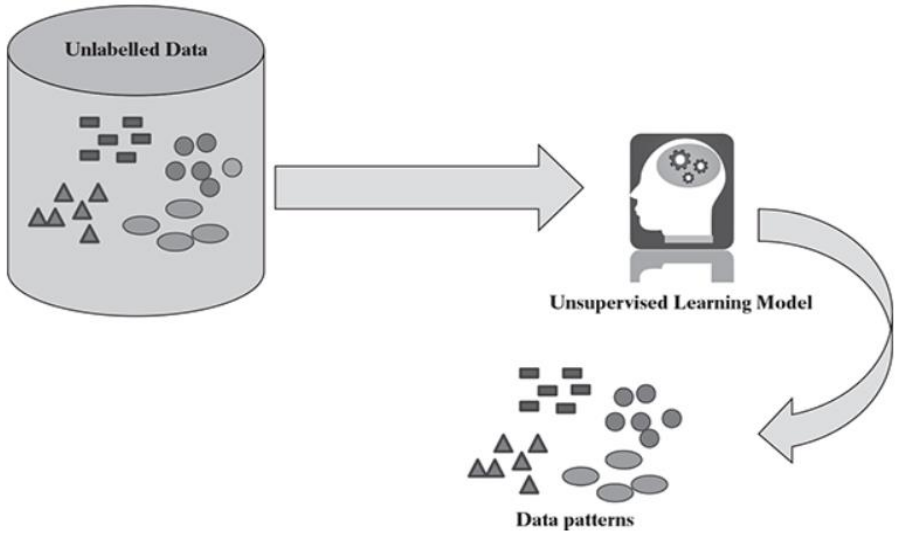


Figure 3: Un-Supervised Machine Learning

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

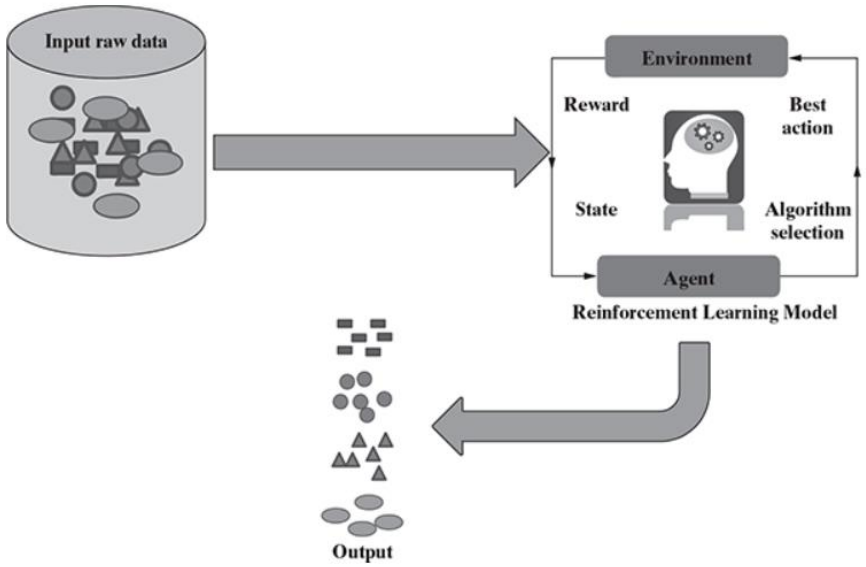


Figure 4: Reinforcement Machine Learning

3. Data Report

3.1. Data collection methodology

If we see the data collected, we notice that data provided comprises of 4 months data starting from March till June 2019 and the data is provided for Weekday, weekends and for 24-hour time period across all days. There can be various methodologies to collect data i.e through APIs as the direct hstorical data for airline flights is not available, however different travel websites provided data in various fields which has to be cleaned first to get data in desired format.

3.2. Visual inspection of Data

The Train dataset comprises of 10683 rows and 10 columns. Test set consist of 2671 row items and 10 variables. We observe the following in dataset

1. Price is dependent variable; all other variable is independent or predictors
2. Except Price which is numeric, all other variables are in “Char” format which needs to be converted to categorical or right class

3. Date of Journey column needs to be separated into “Date”, “Month” and “Year” columns and convert to Date format
4. Route Info has starting city as “Source” and end city as “Destination”. We need to do feature engineering to create 2 columns i.e. for Source and Destination using the separate function and see if this matches with the existing source and destination information provided.
5. Departure time and arrival time have to be converted to time format and the duration has to be put in either “Total hours” or “Total mins”. We have taken “Total Mins”
6. Total stops have to be converted to factor category
7. Jet Airways and Indigo have the maximum number of flights followed by Air India
8. Delhi, Kolkata and Bangalore have the maximum flights starting from them as Source City
9. Cochin, Bangalore and Delhi have the maximum flights reaching there as Destination City There are 3491 non-stop flights and 5625 flights with 1 stop.
10. There is a huge variation in the price, minimum is 1750 and maximum goes up to 79500. There is possibility of outliers in the Price column.
11. Dates within 1st to 10th of month have highest number of flights and maximum flight are in month of May-June (Possibility of Summer Holidays)

3.3. Understanding of Attributes

1. Convert Date of Journey in Date, month and Year columns
2. Convert the required variables to Factor or Date formats
3. Separate the Duration column in Hour and minutes to calculate Total Minutes
4. Convert the Departure time in two brackets i.e day time (9am-9pm) and night time (9pm-9am)
5. Get the weekday information and create a separate column for the day of week from date of journey field

4. Exploratory Data Analysis

4.1. Univariate Analysis

1. Price and Total Duration are numeric categories and all other columns are either categorical or date class
2. Boxplot and histogram of Price shows the presence of outliers
3. Skewness is a mesure of symmetry , positive skewness for price (1.85) means the mean is more than median of the entries and hence it is right skewed
4. Kurtosis define the tail shape of data distribution , in this we have excess kurtosis (13.5) which is towards positive hence it indicates Fat tailed distribution or leptokurtic

5. Day of Travel shows that maximum number of flights are on Monday , Wednesday and Thursday 6 Departure Time and arrival time shows that maximum number of flights arrive and depart around 7 pm in evening
6. Minimum of duration (in mins) is 75 mins and maximum is 2860.
7. Total count of flights is highest during Daytime, on wednesday as day of week and flight with 1 stop

4.2. Bivariate Analysis

1. Average flight price on Sunday and Friday are highest and on Monday are lowest
2. Price of Daytime flight is more than night time
3. Jet Airways, Air India and Indigo have highest number of flights in May june month which is maximum or peak season from flights perspective due to summer season
4. Delhi, Kolkata and Bangalore are the popular choice as Source for boarding the flights
5. Cochin, Bangalore are the popular choice as Destination
6. Average Flight price per week is high in the months of May and June compared to MArch April
7. Jet Airways command the highest price among the Airline categories as evident from box plot
8. Average flight price is high during the first 15 days of month compared to the month end days unless there is some specific festive occasion
9. Delhi and Kolkata commands the highest median price among the other source cities
10. Delhi and Kolkata has highest number of flights as source city and also the count of 1 stops is high for these cities
11. Flight price and Flight duration in mins have a positive correlation of 0.56, means as duration increases flight price increases.

4.3. Outlier treatment and Missing value Treatment

1. We remove unwanted variable like Route information and Additional info from our dataset as they are not contributing to the model and we have already extracted source and destination information from Route information.

2. For outlier treatment, we notice that outlier present in price , we take maximum value of price as 22500, and drop the data points above that point. By doing this we have eliminated around 322 entires of flight price having value higher than 22500

3. For NA values , we notice that there is 1 NA present in Total Stops column, hence we take complete cases and drop the single entry. After doing this transformation the final row count is 10361 and 16 rows

5. Insights from EDA

We have already covered insights from EDA. For data imbalance, it make more sense when the classification is binary (0 or 1) but in our case the response variable (Price) is numeric so data imbalance would not play much role here. also the imbalance due to oultier entries is around 3% and very minimal.

Also techniques like clustering and PCA would have played role where we didnt have target column and we are trying to predict the target, but in our case we have been given the price information and we need to use the same to predict the test data once the model gets sufficient learning and tuning from train data.

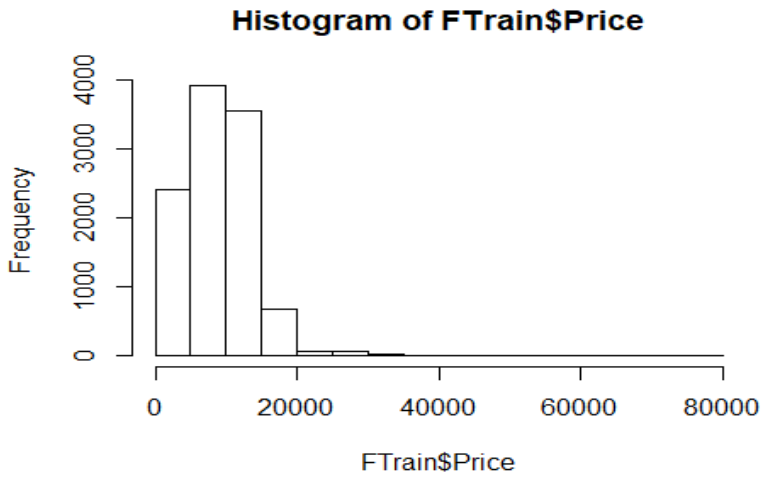
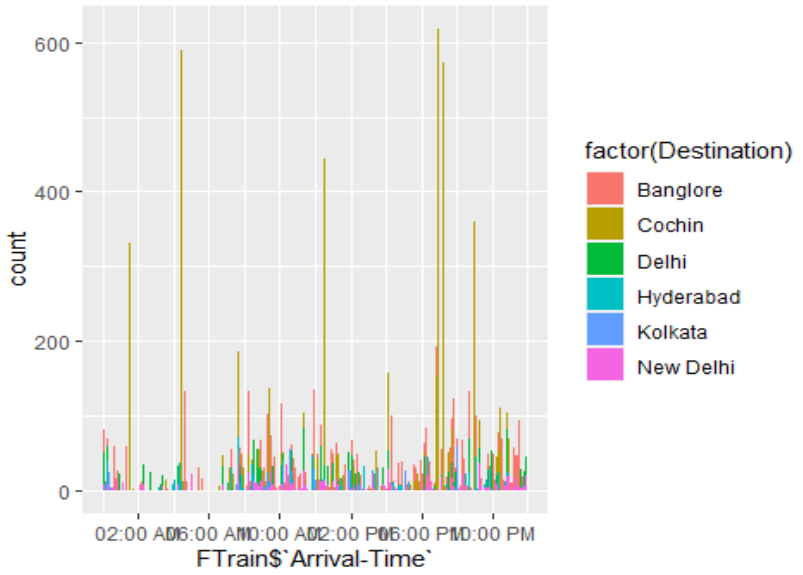
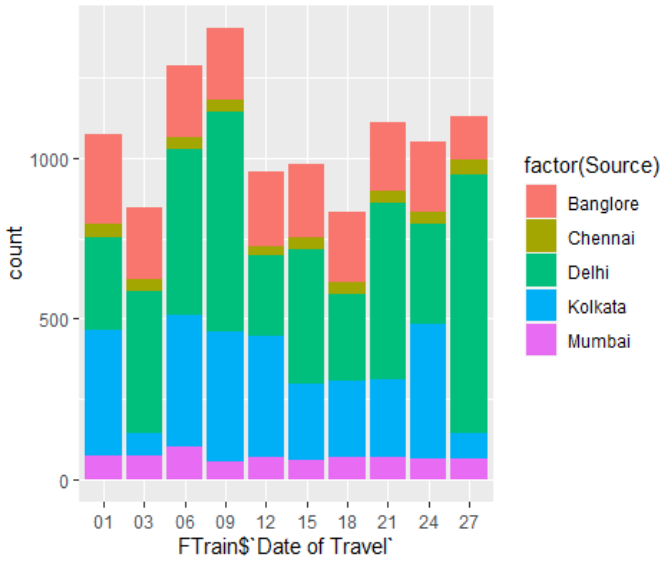
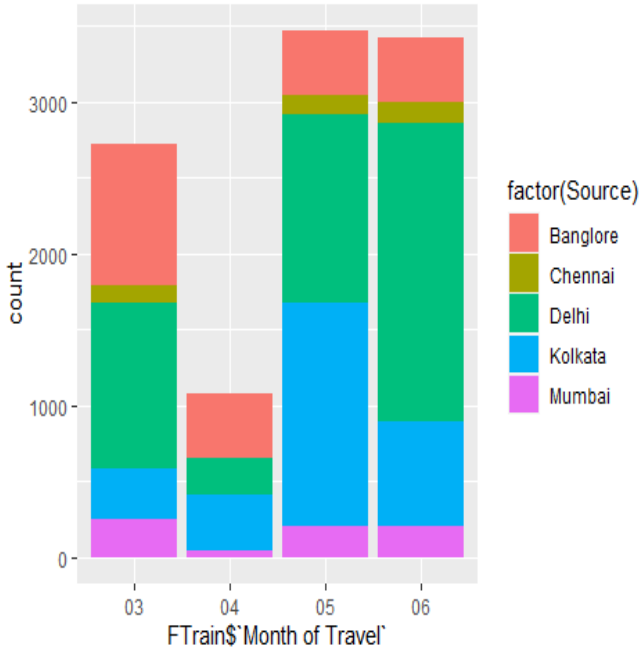
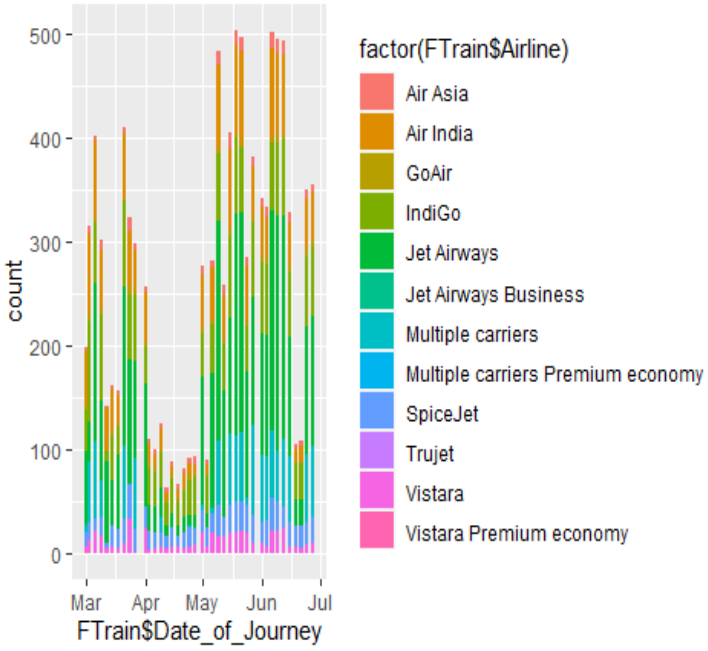


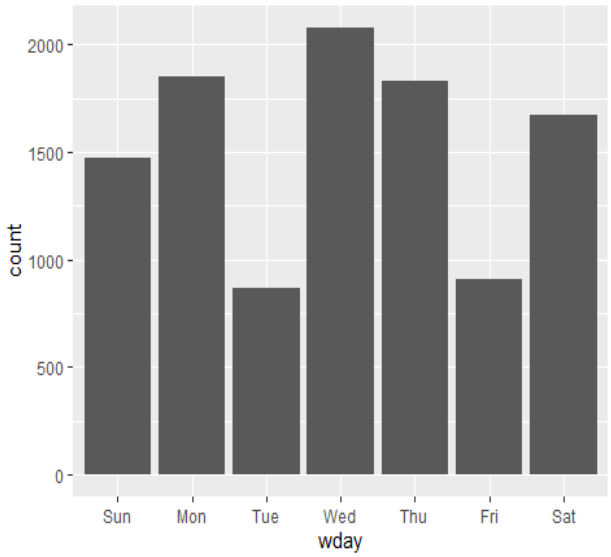
Figure 5: Price Histogram

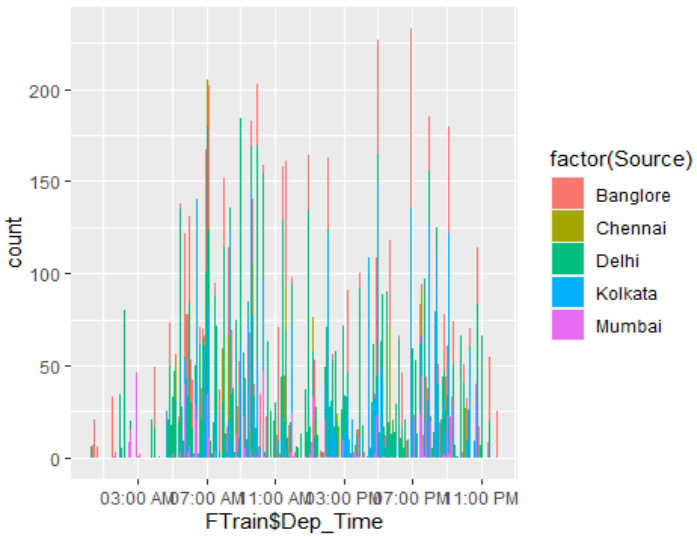

count
600
400
200
0
02:00 AM
06:00 AM
10:00 AM
02:00 PM
06:00 PM
10:00 PM
FTrain$`Arrival-Time`
factor(Destination)
Banglore
Cochin
Delhi
Hyderabad
Kolkata
New Delhi

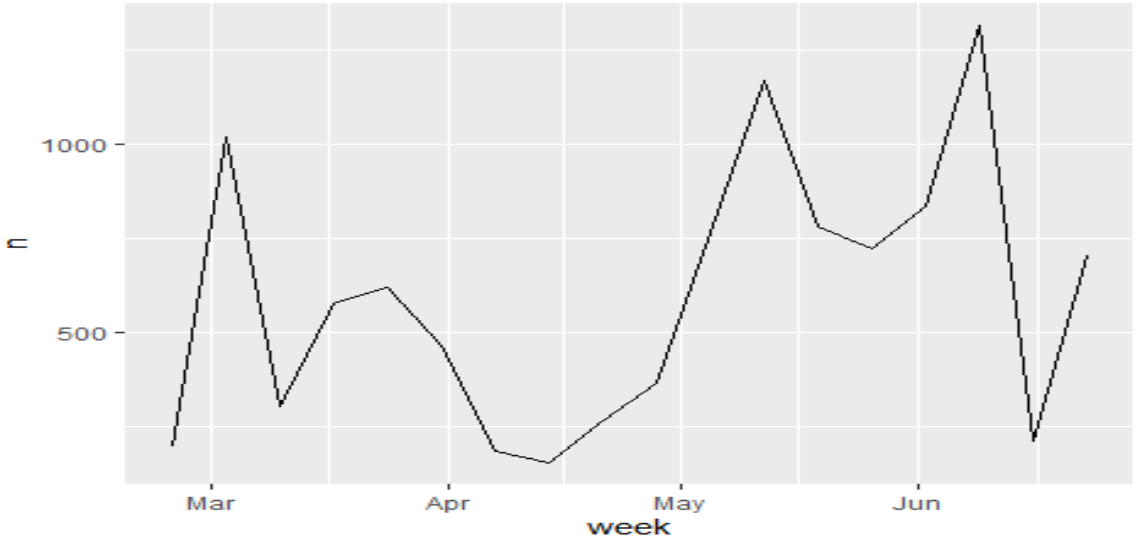

count
1000
500
0
01
03
06
09
12
15
18
21
24
27
FTrain$`Date of Travel`
factor(Source)
Banglore
Chennai
Delhi
Kolkata
Mumbai

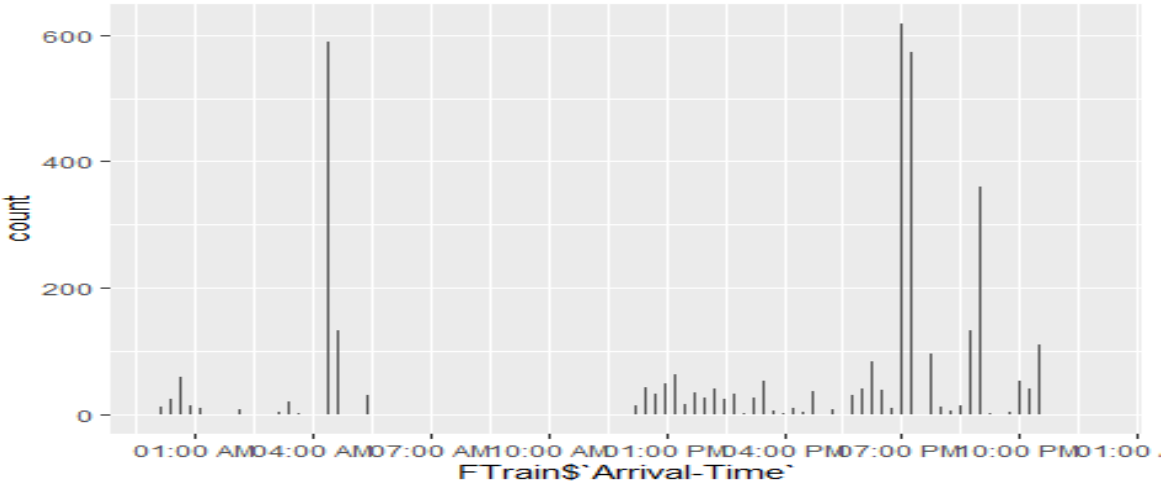

count
3000
2000
1000
0
03
04
05
06
FTrain$`Month of Travel`
factor(Source)
Banglore
Chennai
Delhi
Kolkata
Mumbai

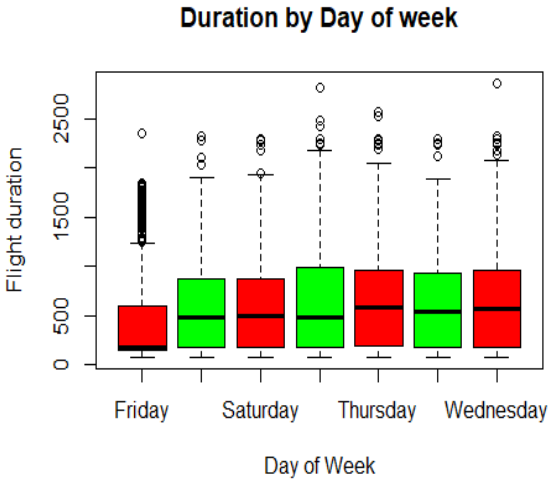

count
500
400
300
200
100
0
Mar
Apr
May
Jun
Jul
FTrain$Date_of_Journey
factor(FTrain$Airline)
Air Asia
Air India
GoAir
IndiGo
Jet Airways
Jet Airways Business
Multiple carriers
Multiple carriers Premium economy
SpiceJet
Trujet
Vistara
Vistara Premium economy

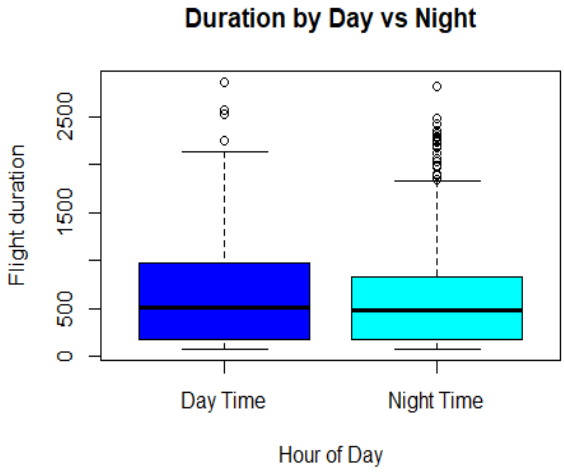

2000
1500
1000
500
0
count
Sun
Mon
Tue
Wed
Thu
Fri
Sat
wday

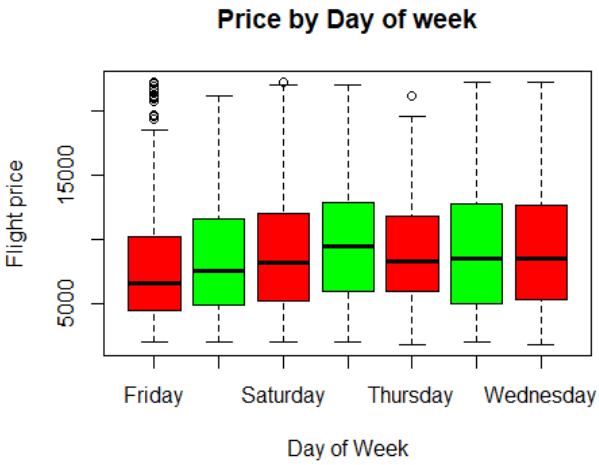

200
150
100
50
0
count
03:00 AM
07:00 AM
11:00 AM
03:00 PM
07:00 PM
11:00 PM
FTrain$Dep_Time
factor(Source)
Banglore
Chennai
Delhi
Kolkata
Mumbai

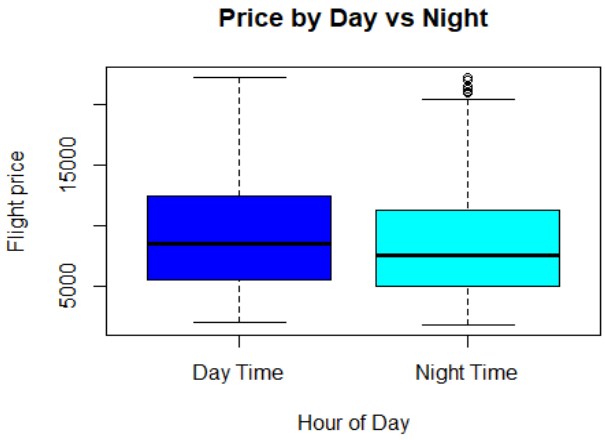

1000
500
n
Mar
Apr
May
Jun
week

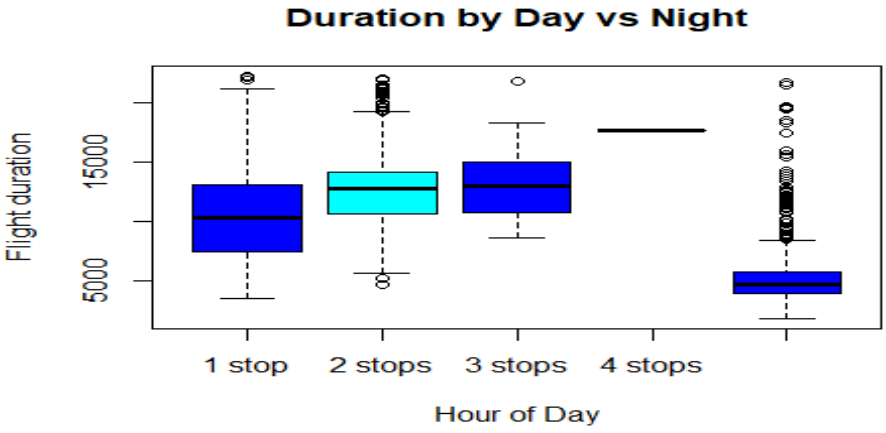

600
400
200
0
count
01:00 AM
04:00 AM
07:00 AM
10:00 AM
01:00 PM
04:00 PM
07:00 PM
10:00 PM
FTrain$`Arrival-Time`


Duration by Day of week
Flight duration
0
500
1500
2500
Friday
Saturday
Thursday
Wednesday
Day of Week


Duration by Day vs Night
Flight duration
0
500
1500
2500
Day Time
Night Time
Hour of Day


Price by Day of week
Flight price
5000
15000
Friday
Saturday
Thursday
Wednesday
Day of Week


Price by Day vs Night
Flight price
5000
15000
Day Time
Night Time
Hour of Day


Duration by Day vs Night
Flight duration
5000
15000
1 stop
2 stops
3 stops
4 stops
Hour of Day

5.1. Independent variables that are singificant

Based on the data transformation and feature engineering we have done above, we can say that except the columns “Route” and “Additional Info”, all other columns are significant in the model building. The same will get validated once we start building the model using Multiple linear regression, Decision Tree, Random Forest, Gradient Boost etc.

5.2. Relationship between time of journey and Flight prices

Response to this section we have covered earlier. Flight prices are costlier during the day time and specific during the evening time. Also flight price on weekend are costlier compared to weekdays. Flight price in the morning hours 8-9 am and in evening 4-6 pm are higher compared to other time.

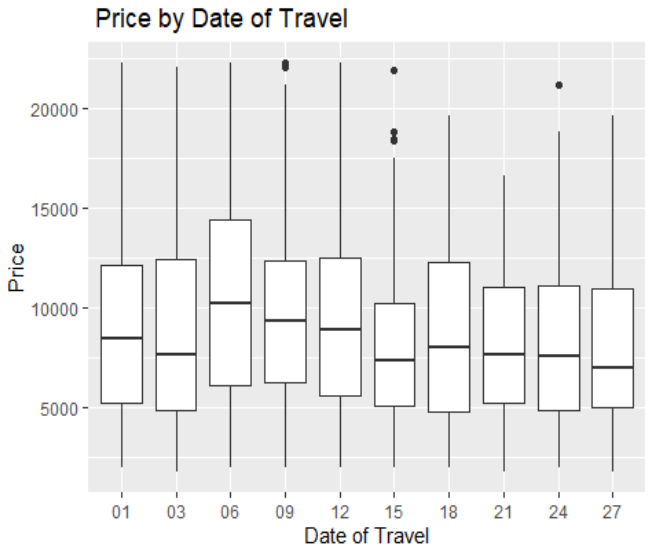
5.3. Hypothesis Testing

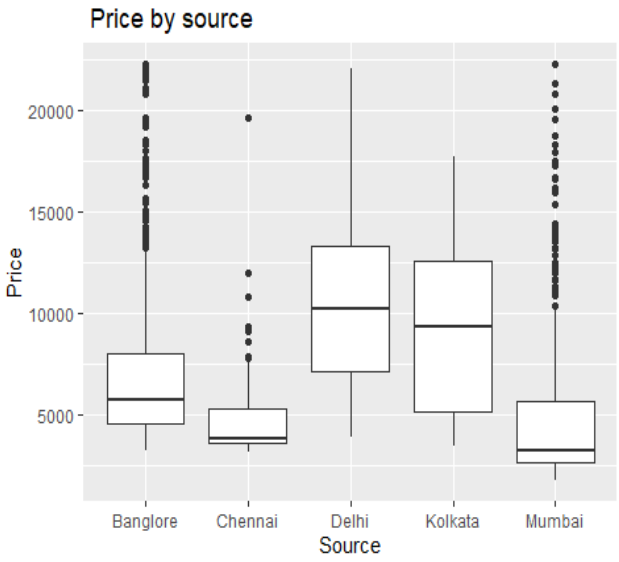
1. **Weekday price cheaper than weekend**

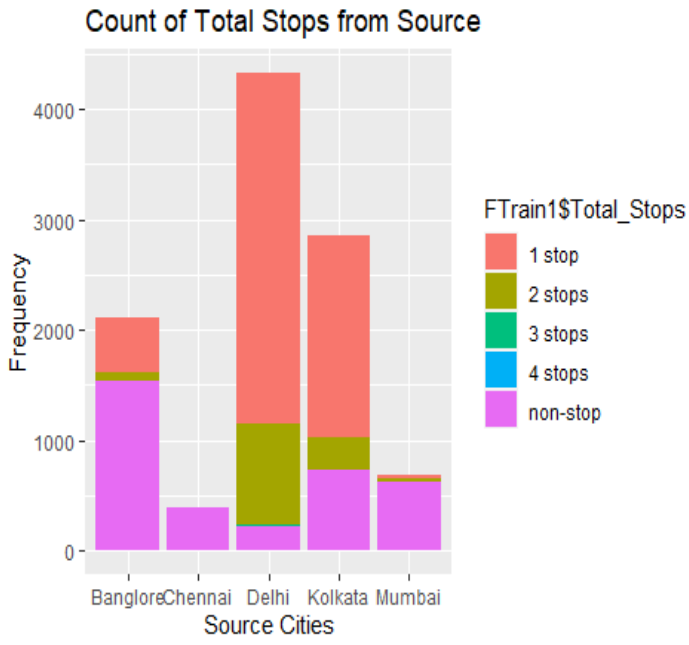
We did anova testing on the linear model built using Price and “DayofWeek” and found the P value very small and hence null hypothesis is rejected and we can say that flight price on weekends are costlier compared to weekdays

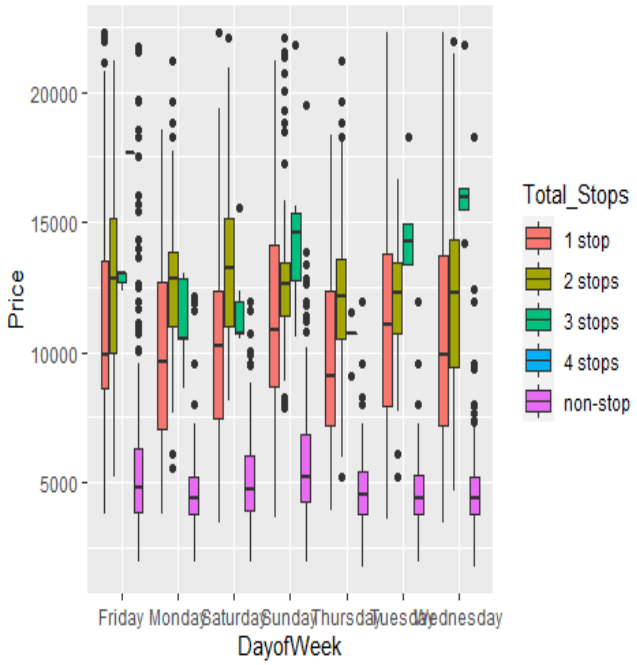
2. **Peak hour price is more than non-peak**

We did a 2-tail t test for the same and found that P value is very small and less than 0.05, hence null hypothesis is rejected and hence Flight price during peak hours 9am-9pm are higher than non peak hours ie 9pm-9am.


Price by Date of Travel
Price
20000
15000
10000
5000
01
03
06
09
12
15
18
21
24
27
Date of Travel


Price by source
Price
20000
15000
10000
5000
Banglore
Chennai
Delhi
Kolkata
Mumbai
Source


Count of Total Stops from Source
Frequency
4000
3000
2000
1000
0
Banglore
Chennai
Delhi
Kolkata
Mumbai
Source Cities
FTrain1$Total_Stops
1 stop
2 stops
3 stops
4 stops
non-stop


Price
20000
15000
10000
5000
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
DayofWeek
Total_Stops
1 stop
2 stops
3 stops
4 stops
non-stop

6. Regression based models and Interpretation

6.1. Multiple Linear Regression

It utilizes error minimization to fit the best possible line in statistical methodology. However, in machine learning methodology, squared loss will be minimized with respect to β coefficients. Linear regression also has a high bias and a low variance error.

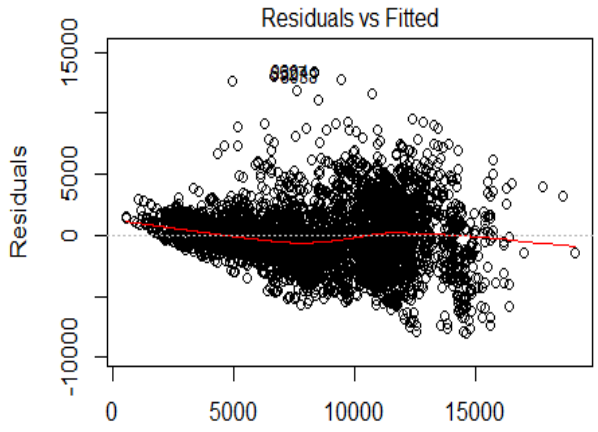
Model Interpretation

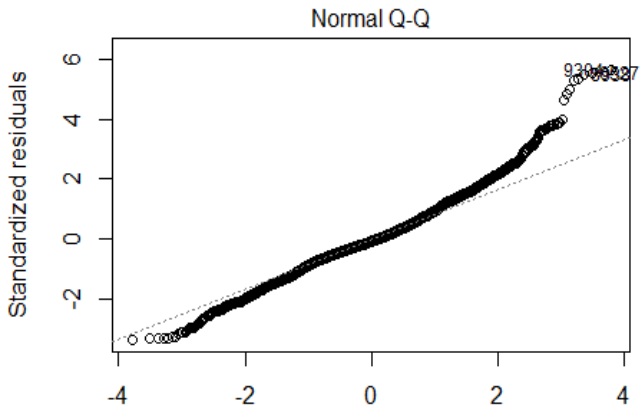
For the model building, we take encoded data set set where all the categorical variables have been encoded with dummy variables as 0 or 1. The intercept consider the effect of variables that we have not considered. We can intrepret the model such as For each one unit change of independent variable the price increase or decrease by the amount represented as slope keeping other variable constant

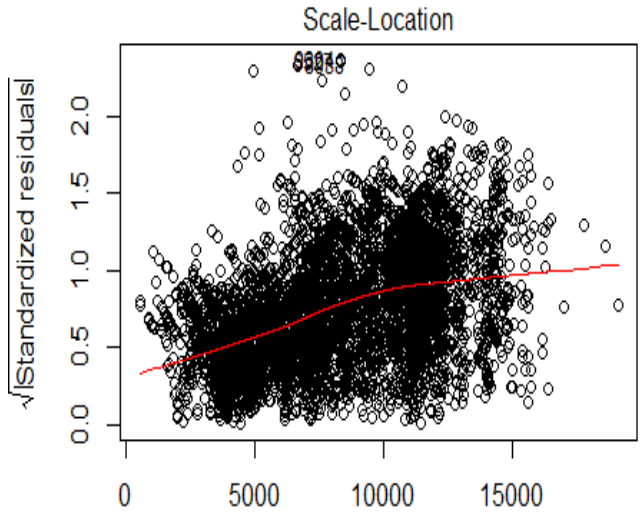
R square interpretation - 63% of the variation in the flight price is explained by the independent variables used in the model. We reject null hypothesis as t-stat has extremely low p-values which is $p<0.05$ and there is significant evidence that regression model exists. Adjusted R square shows the effect of adding more variables in the model. For eg current model is explained 63% by the independent variables which means remaining 40% of model is unexplained or by residuals.so we inflate the error component by multiplier which is division of Total degree of freedom by residual degree of freedom

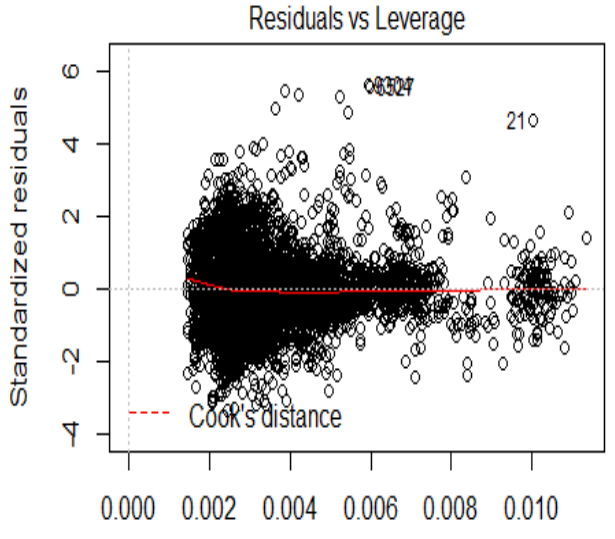
Based on the model and P-stat we keep only the significant variables in the final model building and again compute the R2 and root mean square error. We also check for correlation between the models using VIF

Model validation on Test set →RMSE - 2629.421, R2 - .633


Residuals vs Fitted
Residuals
15000
5000
0
-10000
0
5000
10000
15000
Fitted values
lm(Price ~ Airline.Air.Asia + Airline.Air.India + Airline.GoAir + Airline


Normal Q-Q
Standardized residuals
6
4
2
0
-2
-4
-2
0
2
4
Theoretical Quantiles
lm(Price ~ Airline.Air.Asia + Airline.Air.India + Airline.GoAir + Airline


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
0
5000
10000
15000
Fitted values
lm(Price ~ Airline.Air.Asia + Airline.Air.India + Airline.GoAir + Airline


Residuals vs Leverage
Standardized residuals
6
4
2
0
-2
-4
Cook's distance
0.000
0.002
0.004
0.006
0.008
0.010
Leverage
lm(Price ~ Airline.Air.Asia + Airline.Air.India + Airline.GoAir + Airline

6.2. Ridge Regression

Regularization

This uses regularization to control overfitting issues by applying a penalty on coefficients. In ridge regression, a penalty is applied on the sum of squares of coefficients, whereas in lasso, a penalty is applied on the absolute values of the coefficients. The penalty can be tuned in order to change the dynamics of the model fit. Ridge regression tries to minimize the magnitude of coefficients, whereas lasso tries to eliminate them.

Normal regression works by selecting the coefficients that minimize the loss function, however if the coefficient are large there may be chance of overfitting the training data and will not generalize well on unseen test data. To overcome this, we will do regularization that will impact the large coefficients

Ridge Regression - Loss function is minimized by adding a penalty parameter equivalent to square of magnitude of coefficients. The model will be tuned thru the hyperparameter lambda. This model will generalize well on test data as it will be less sensitive to extreme variance.

Model validation → RMSE – 2579, R2 - .634

6.3. Lasso Regression

Lasso Regression - in this loss function is modified to minimize the complexity of model by limiting sum of absolute values of model coefficients

Model Validation→ RMSE – 2626, R2 - .633

6.4. Elastic Net Regression

Elstic Net - It uses the properties of both ridge and lasso . It works by penalizing the model using both optimum alpha and lambda values. We use caret package to find optimal values and using the hyper tuning parametrs

Model validation → RMSE – 2627, R2 - .633

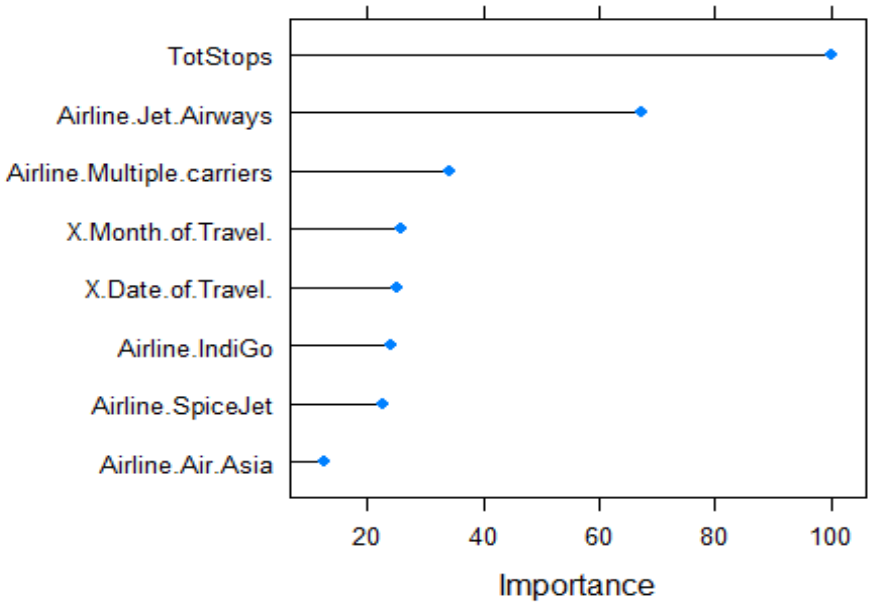


Figure 6: Elastic Net regressor Feature Importance

6.5. KNN Regressor

K nearest neighbors is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification.

KNN – For this used caret package, to find the optimal number of nearest neighbors to predict the value. We used cross validation method where 9 samples were used for train and one is used for test

Minimum RMSE is achieved when K is 5, we use both default method of RMSE and Rsquared method, and find out that Rsquare gives a slight better RMSE reduction

Model validation → RMSE – 2116, R2 - .73

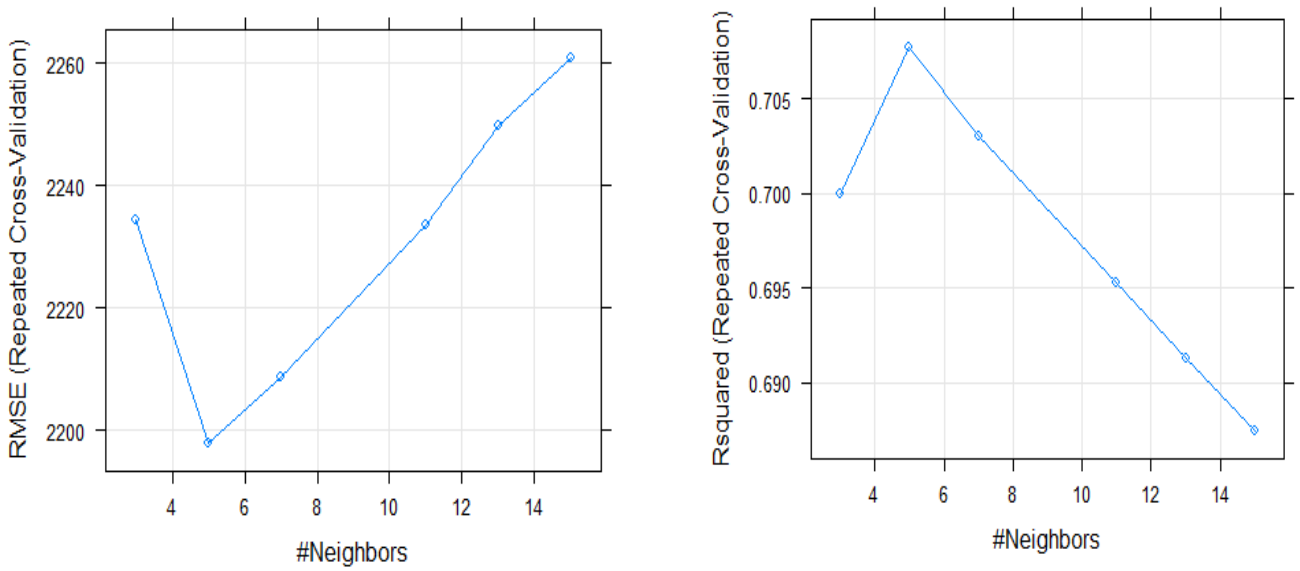


Figure 7: Optimum K value as per RMSE and R-squared

7. Decision Trees

Decision Tree - In this method, we split the tree starting from root node based on the condition that each split will have minimum impurity or less Gini index. Impure nodes will have more Gini index and have maximum variance .Pure node will have zero Gini index and impure as maximum Gini of 0.5 The split should be such that when we come from root node to child node the Gini gain is maximum for that particular split compared to other split for different predictors and Gini impurity reduces for that node compared to root node.

Recursive binary splitting is applied to split the classes at each level to classify observations to their purest class. The classification error rate is simply the fraction of the training observations in that region that do not belong to the most common class. Decision trees have low a bias and a high variance error.

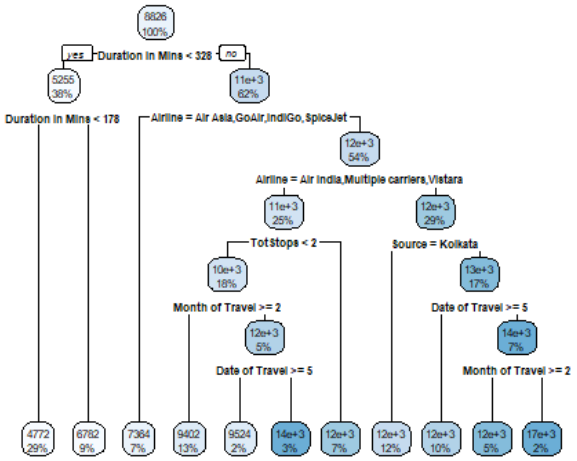
Decision Trees are powerful but have tendency to over fitting the data. We prune the tree to avoid overfitting based on complexity Parameter which says that error decrease should be more than Alpha, if it is less than we stop adding the branches. Alpha is threshold and we generally take around 0.015

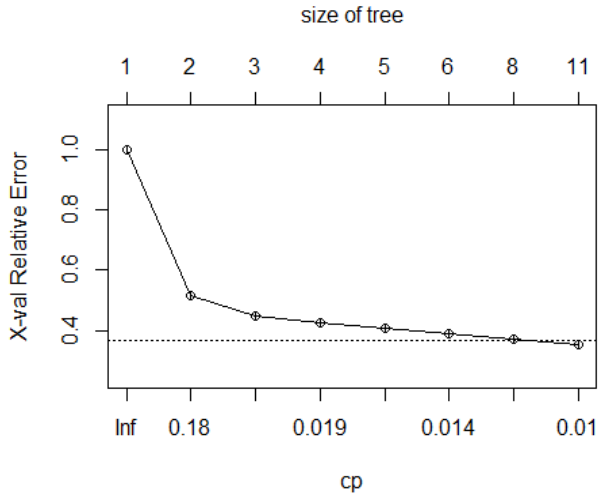
Common problem with Decision Trees are -

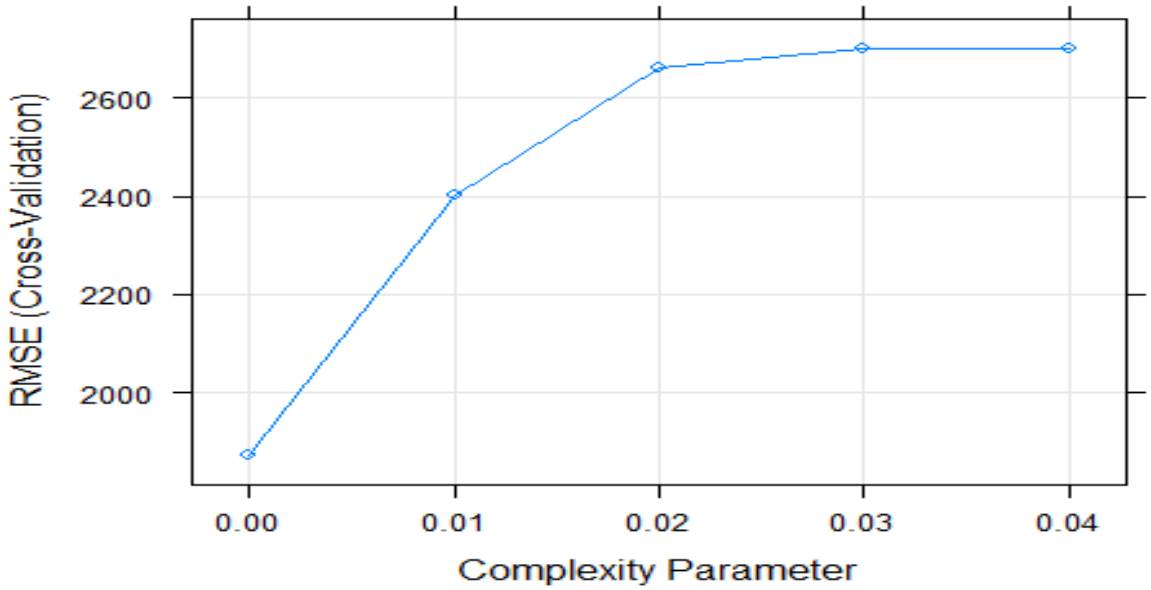
Too many branches - chances of overfitting (capture of noise) Too less branch - under fitting (Misclassification error)

Optimum number of branches are arrived based on pruning.CP is error reduction per node. We choose the node where cross validation error is minimum i.e xerror.

Model validation → RMSE - 1885.7, R2 -.787







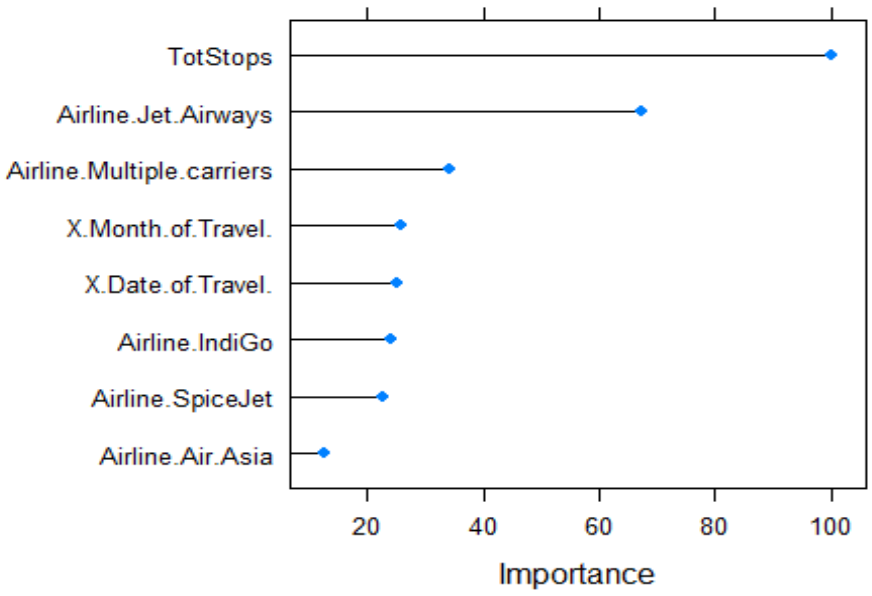


Figure 8: Feature Importance Decision Tree

8. Random Forest and Ensemble Technique

Ensemble Technique

This is a technique applied on decision trees in order to minimize the variance error and at the same time not increase the error component due to bias. In bagging, various samples are selected with a subsample of observations and all variables (columns), subsequently fit individual decision trees independently on each sample and later ensemble the results by taking the maximum vote (in regression cases, the mean of outcomes calculated)

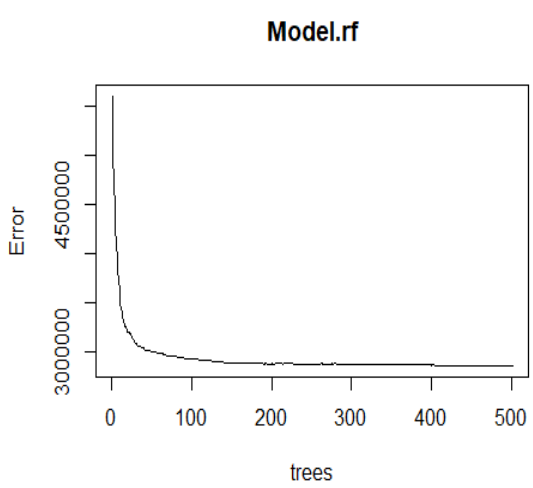
Random Forest

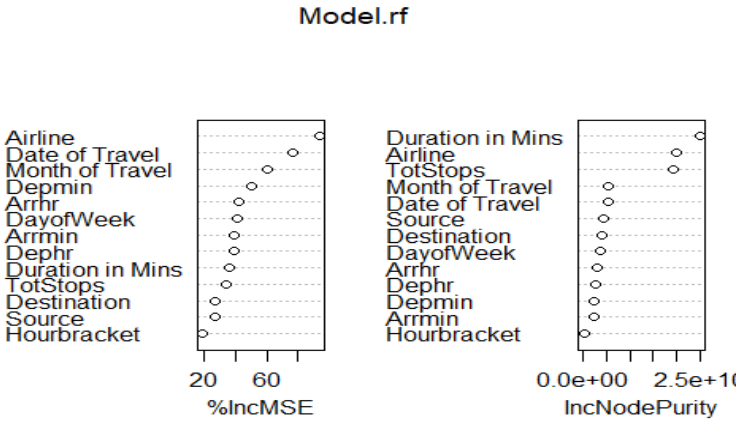
This is similar to bagging except for one difference. In bagging, all the variables/columns are selected for each sample, whereas in random forest a few sub columns are selected. The reason behind the selection of a few variables rather than all was that during each independent tree sampled, significant variables always came first in the top layer of splitting which makes all the trees look more or less similar and defies the sole purpose of ensemble: that it works better on

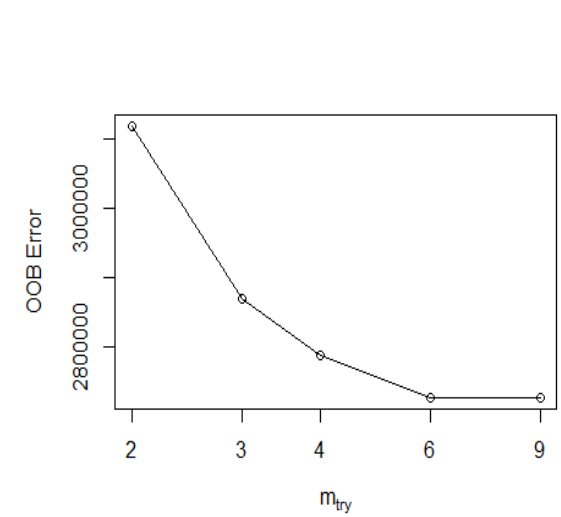
diversified and independent individual models rather than correlated individual models. Random forest has both low bias and variance errors.

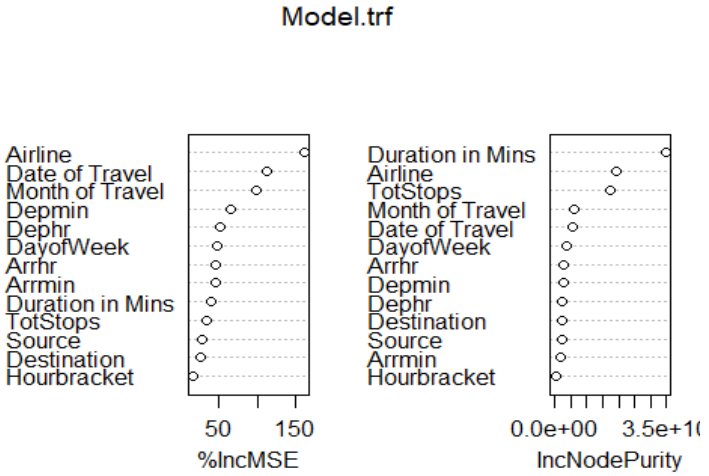
We use bootstrap aggregating (which is generating new training subset by sampling the data subset over and over again with replacement). Through this we ensure that each tree built within the forest is diverse and at the same time share communality that they have been built from the same subset of data. We have to be careful with value of m “mtry in mode”, if m is large variables become too correlated, if m is small, then predictive power of model decreases. Optimal choice of m plays vital role in random forest model.

Model validation → RMSE - 1654.5 R2 - .836









8.1. Bias Variance Tradeoff

Every model has both bias and variance error components in addition to white noise. Bias and variance are inversely related to each other; while trying to reduce one component, the other component of the model will increase. The true art lies in creating a good fit by balancing both. The ideal model will have both low bias and low variance. Errors from the bias component come from erroneous assumptions in the underlying learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs; this phenomenon causes an underfitting problem. On the other hand, errors from the variance component come from sensitivity to change in the fit of the model, even a small change in training data; high variance can cause an overfitting problem:

An example of a high bias model is logistic or linear regression, in which the fit of the model is merely a straight line and may have a high error component due to the fact that a linear model could not approximate underlying data well. An example of a high variance model is a decision tree, in which the model may create too much wiggly curve as a fit, in which even a small change in training data will cause a drastic change in the fit of the curve.

At the moment, state-of-the-art models are utilizing high variance models such as decision trees and performing ensemble on top of them to reduce the errors caused by high variance and at the same time not compromising on increases in errors due to the bias component. The best example of this category is random forest, in which many decision trees will be grown independently and ensemble in order to come up with the best fit

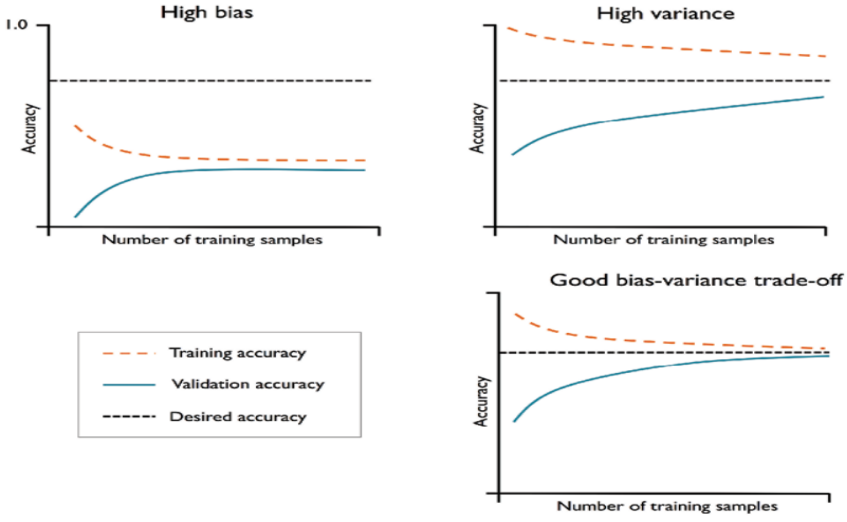


Figure 9: Bias Variance Tradeoff

8.2. Boosting method

This is a sequential algorithm that applies on weak classifiers such as a decision stump (a one-level decision tree or a tree with one root node and two terminal nodes) to create a strong classifier by ensembling the results. The algorithm starts with equal weights assigned to all the observations, followed by subsequent iterations where more focus was given to misclassified observations by increasing the weight of misclassified observations and decreasing the weight of properly classified observations. In the end, all the individual classifiers were combined to create a strong classifier. Boosting might have an overfitting problem, but by carefully tuning the parameters, we can obtain the best of the self-machine learning model

We will do boosting to sequentially train the weak learners. Difference in bagging and boosting is that bagging is parallel and boosting is sequential. Gradient boosting method - It builds on each model, trying to fit the next model based on the residuals of previous model. We will use several tuning parameters to arrive at optimal model performance

Model validation → RMSE - 2388.8 R2 - .66

8.2.1. Extreme gradient Boosting

Extreme gradient boosting – This is a specialized implementation of gradient boosting decision trees designed for performance. Types are gradient boosting ,stochastic and regularized boosting. Some of advantages of using XGboost are

1. *Parallel computing - it is enabled with parallel processing*
2. *Regularization - it is used to avoid overfitting in linear and tree-based models*
3. *Enabled cross validation - it is enabled with internal CV and not needed any additional package*
4. *Missing values - model can handle missing values*
5. *Tree pruning - it grows the tree upto a max depth and then prune backward until improvement in loss function.*
6. *Bias and Variance - Unlike bagging model like Random forest which takes care of overfitting (high variance) but still retains some bias, boosting method can handle both bias and variance*

We will use several hyper parameters to tune the model. These values can be further optimized to get a most optimum XGBoost model. In my current case i have take learning rate as 0.045, depth as 6 and number of rounds as 650

Model validation → RMSE - 1677 R2 - .832

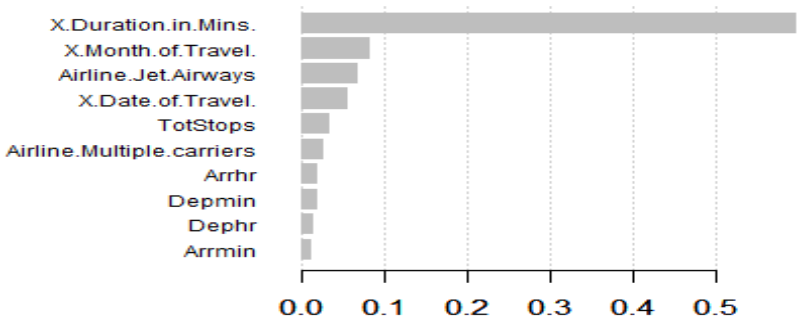


Figure 10: Feature importance XGBoost model

9. Model Comparison

1. Feature Engineering is most crucial for these types of supervised machine learning problem
2. Although both Random forest and Xgboost gave similar result but we choose Xgboost for final model building considering it takes care of both bias and variance
3. The model accuracy can be further built by tuning the hyperparameters

Figure 11: Model comparison

10. Conclusion and Future Recommendation

1. Supervised machine learning technique can be a good predictor of airfare prices for historical data
2. The most important factors in airfare price prediction are the data collection (Type and nature of date) and the feature engineering (techniques to apply for model building)
3. Our finding suggests that the key important factors in pricing models are “Duration in mins”, “Total stops”, “Month of Travel”, Type of airline” etc. These are not limited to but can be extended to other variables which were missing like airline delay, number of seats etc.
4. Large datasets collected over a longer duration and inclusion of other important predictors can increase the prediction accuracy of model.