Report

N.B: Comment From Proposal To Project Evaluation

Everything done above was our work done on the proposal which the Professor said is not necessary to add it to this one. Before moving on to the project evaluation part, we would like to highlight a couple of changes made from proposal to this point.

The original dataset we extracted from Kaggle was enormous (**300,154 rows**) and with variables such as the Airlines, the flights number, flight departures and destinations, the number of stops of the flights, the departure time and arrival time (moment of the day), the class of the flight (Economy or Business), duration of the flight, the days left before the flight and the price. We then move on to the next step, which is to prepare our data for analysis.

In our study, we will try to answer the following question: *What are the key determinants of flight pricing, and how can those factors be used to predict price variations?*

5. Data Preparation

a. Data selection

Among all the variables we have listed above, we have inspected the data (missing values were already removed) and have chosen only the variables that were found to be significant and helpful for this analysis. Here is a breakdown of what we found.

1) First, we consider the potential variables that could affect the price, meaning those that could be useful in building our model: *The Airlines, The Departure time, The Class, Duration, the Days left*.

2) We then check the relationship of each variable with the dependent variable while checking for outlier with different possible plots.

Comment 1: "Class": We notice that Business class has some unusually extremely high flight prices and seem to have little or no variation in prices with the number of days left. We were able to visualize those trends with a scatter plot and a histogram plot. For simplicity we decided to use only "Economy" Class.

Comment 2: The Departure time has two major changes in the averages with all the airlines together or within the airlines. "Late Night" seems to have a lower than other moments of the day (Standard). We decided to group them into "Late Night" and "Not Late Night" to minimize the number of variables and also easily handle the scenario. We convert this into a numerical variable.

b. Data cleaning

Afte applying those changes to our original dataset, we now have our clean up dataset named "***Simpler_Clean_Up***" with 194,464 rows with five (5) entity variables: Duration, Days_left, Stops, Departure Time, and Airlines.

Table 2: Simpler Clean Up Data

	A	B	C	D	E	F
1	price	duration	days_left	stops	departure_time	airline
2	5953	2.17	1	zero	Evening	SpiceJet
3	5953	2.33	1	zero	Early_Morning	SpiceJet
4	5956	2.17	1	zero	Early_Morning	AirAsia
5	5955	2.25	1	zero	Morning	Vistara
6	5955	2.33	1	zero	Morning	Vistara
7	5955	2.33	1	zero	Morning	Vistara
8	6060	2.08	1	zero	Morning	Vistara
9	6060	2.17	1	zero	Afternoon	Vistara
10	5954	2.17	1	zero	Early_Morning	GO_FIRST
11	5954	2.25	1	zero	Afternoon	GO_FIRST

c**. Prepare data.**

To make our current new data ready for analysis, we will need to use dummy variables for Stops, Departure time, and Airlines.

Stops: We will use "0" or no stop flights and "1" for flights with at least one stop.

Departure time: We will use "0" for "Late Night" and "1" for "Not Late Night."

Airlines: Since Vistara was the airline with the highest average prices, we have chosen that to be the reference, the benchmark. Now we have the other 5 airlines encoded as columns with the values either "0" or "1". We then save our new data file as "***Dummy_Variable_Created***" with a total of 9 predictor variables: Duration (*continuous*), Days_left (*numeric*), Stops (*binary*), AirAsia (*binary*), Air_India (*binary*), Go_FIRST (*binary*), Indigo (*binary*), SpiceJet (*binary*), and Not_Late_Night (*binary*). Now we can proceed with the analysis.

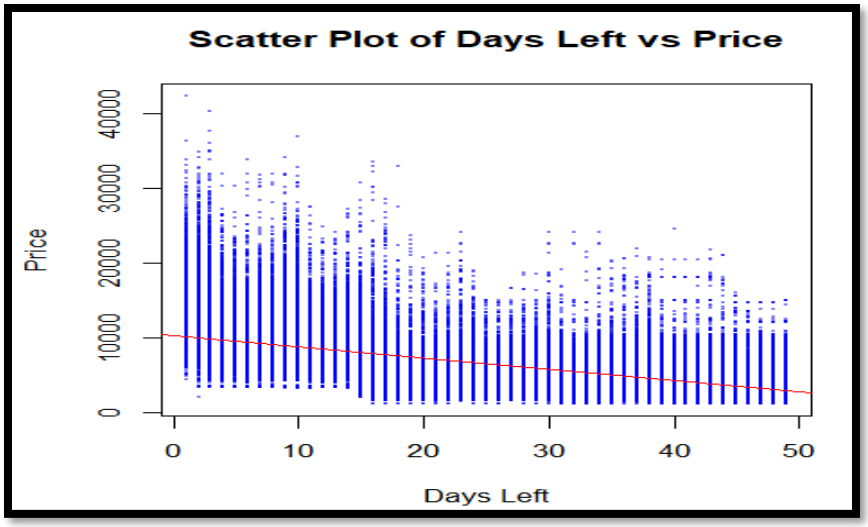
Table 3: Dummy Variables Created

	A	B	C	D	E	F	G	H	I	J	K
1	Price	Duration	Days_left	Stops	AirAsia	Air_India	GO_FIRST	Indigo	SpiceJet	Not_Late_Night	
2	5953	2.17	1	0	0	0	0	0	1	1	
3	5953	2.33	1	0	0	0	0	0	1	1	
4	5956	2.17	1	0	1	0	0	0	0	1	
5	5955	2.25	1	0	0	0	0	0	0	1	
6	5955	2.33	1	0	0	0	0	0	0	1	
7	5955	2.33	1	0	0	0	0	0	0	1	
8	6060	2.08	1	0	0	0	0	0	0	1	
9	6060	2.17	1	0	0	0	0	0	0	1	
10	5954	2.17	1	0	0	0	1	0	0	1	

Data visualization

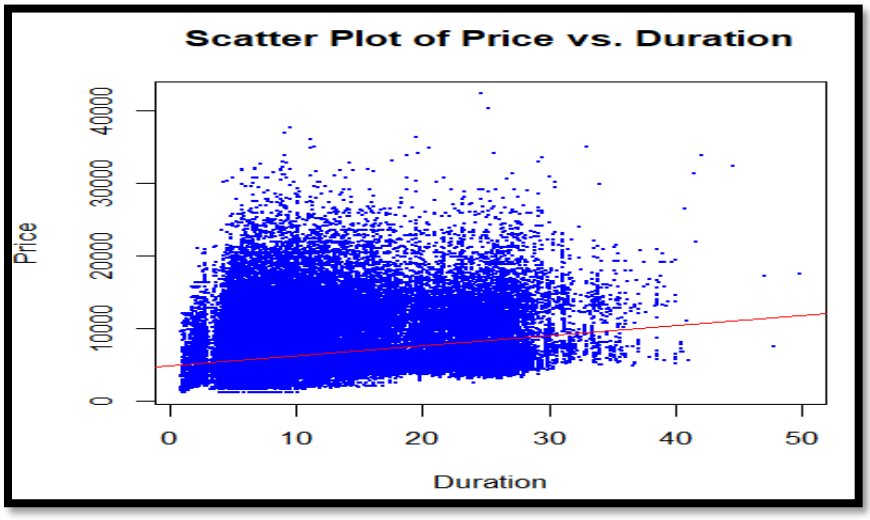
Before moving to data preparation, we will make some plots to visualize the variables and their relationships. The works and results of each graph can be found in our R scripts. We will just show the graphs and a brief summary here to stay focused on the main point of the analysis.

Plot 1: Scatter plot of "*Price*" vs. "*Days_left*". We make a scatter plot to visualize the change of the prices based on the days left, using R. (*See Appendix for the regression output, output 1*)



The plot shows a negative relationship between the *'price'* and the *'days_left'* as shown by the regression line. It has an $R^2 = 0.3175$, and a Standard Error= 2,986. Around 31.75% of the variation in the prices are due to the change in days left.

Plot 2: Scatter of *Price* vs. *Duration* of the flight. We will make a scatter plot to see how the price changes with respect to the duration, using R. (*See Appendix for the regression output, output 2)*



The plot shows a positive relationship as shown by the regression line in red. It has **R2 = 0.07582** and a standard deviation of **3,475**. The relationship isn't really strong because only **7.58%** of the variation in prices are due the changes in flight duration. It's still useful for our analysis.

Plot 3: Boxplot comparing prices between *'zero'* stops and *'one'* stop. We want to see the impact the stops have on the price. (*See Appendix, plot 3*). The plot shows that flights with one stop have

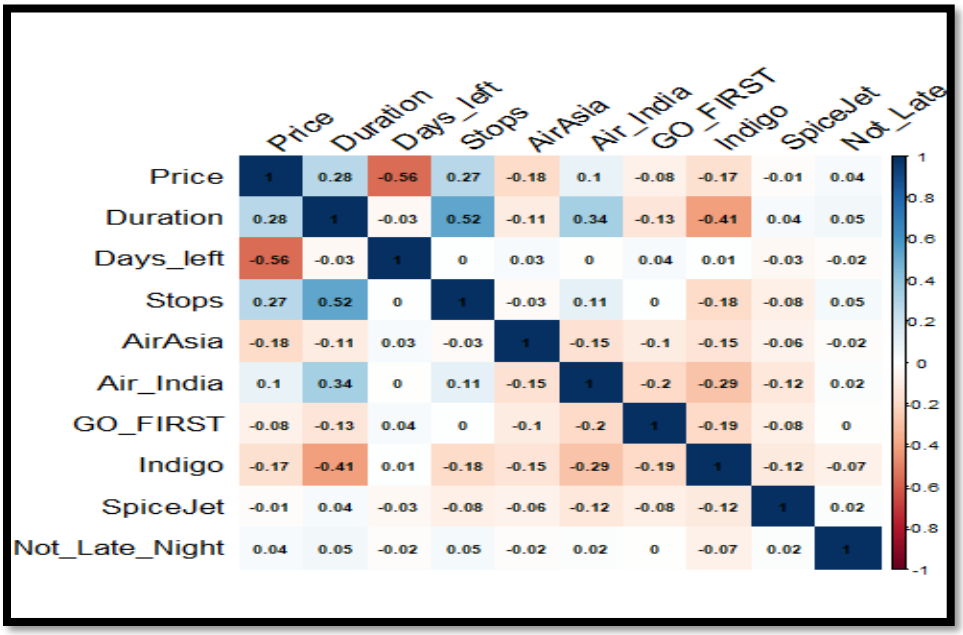
higher median and higher variation of price (variance). They tend to be more expensive than flights with no stops. So, we are expecting flights with more stops to be more expensive than flights with less stops.

Plot 4: *Price* variation based on *Departure time*. We make a bar plot to compare price variation based on the departures. (*See Appendix, plot 4*). As we can see from the plot, *'Late night'* tends to have lower prices and *'morning'* higher prices compared to other departure times.

Plot 5: Average price Across Airlines. We make a bar plot of the average price for each Airline. (*See Appendix, plot 5*). Vistara came to be the Airline with the highest prices and AirAsia with the lowest prices.

 # Grouped bar plot: We also plot a plot to compare the price based on departure time within the airlines. (*see Appendix Group bar plot*). As seen in the previous plot, we can see in this one that the *'early morning'* and 'late night' tend to have lower price for each Airline. However, '*Night*' flights interestingly stand out for each individual flight.

Plot 6: *The heatmap*. Checking for any potential multicollinearity, we use the heatmap to check for any correlation between the variables.



The heatmap indicates a moderately strong negative correlation between 'Price' and *'Days_left'* (**-0.56**), a positive correlation for 'Duration' (**0.28**) and 'Stops'(**0.27**) as expected. We can notice that *'Duration'* and *'Stops'* have the highest correlation coefficient (**0.52**). Which makes sense because the higher the stops, the longer we expect the flight to be. We think that correlation coefficient is not strong enough to be an issue for this analysis.

6. Modeling - Building Models

Based on the nature and category of our dataset, we have identified three potential models that we can use to process, analyze and build our prediction models.

Model 1: Multiple linear regression with a train-test split.

Before starting the analysis with the models, we first partition the data into "training", 70% and "validation" 30% done in R using the "**createDataPartition()**" function.

```
set.seed(1)
myIndex <- createDataPartition(m1Data$Price, p=0.7, list=FALSE)
trainSet <- m1Data[myIndex,]
validationSet <- m1Data[-myIndex,]
```

After checking the relationship of each predictor with the independent variable, we have started with the most relevant predictors and then consecutively add the rest to see how the addition of each predictor will affect the model. Starting with ***model_a*** which as all the *airlines* variables plus "*Days_left*", ***model_b*** has *airlines*, "*Days_left*" and "*Stops*", ***model_c*** has *airlines*, "*Days_left*", "*Stops*" and "*Not_Late_Night*" and ***model_d*** has *airlines*, "*Days_left*, "*Stops*", "*Not_Late_Night*", and "*Duration.*". (*The regression output results shown in the Appendix, Output 3*)

From the output of these models' trial showing the coefficients, the predictor "*Not_Late_Night*" was found not to be statistically significant. Also, "***Model_d"*** stands out with an Adjusted R2 of slightly higher (**0.4564**) and a lower Standard Error of estimate (2,666) lower than that of the other models. Using output 3 (*See Appendix*), we can write the mathematical regression equation.

Price *= 9,103.505 - 2892.594 * **AirAsia** - 537.368 * **Air_India** - 1386.746 * **GO_FIRST** - 1564.186 * **Indigo** - 1168.487 * **SpiceJet** - 154.186 ***Days_left** + 2119.802 * **Stops** + 31.203 * **Duration***

- The coefficient (***or slopes***) for each airline is the estimated average amount for the prices compared to the reference airline, Vistara. Example, you're estimated to pay ***INR 1,387*** less on your flight when you choose Air_india instead of Vistara if all other parameters are kept constant.
- You are estimated to save ***INR 155*** for each additional day left ahead before your flight.
- And ***INR 32*** for each additional hour to your flight.
- You will spend around ***INR 2,120*** for each additional stop you add to your flight.

The results of these models (different scenarios) are summarized in the table below. We also show the performance metrics of the best results (*model_d*) below.

Partition Method	Model_a	Model_b	Model_c	Model_d
Standard Error	2,794.0	2,671.0	2,671.0	2,666.0
Coef R2	0.4031	0.4544	0.4544	0.4564
Adjusted R2	0.4031	0.4543	0.4543	0.4564

Best Results	ME	RMSE	MAE	MPE	MAPE
Model_d	0.096	2663.149	1925.71	-12.993	35.337

Model 2: Multiple linear regression with a 4-fold Cross validation

To evaluate this model, we divide the dataset into 4-fold and then use then test experiments 1 through 4 according to each division and then take the average of the 4.

model_2 <- lm(Price ~ AirAsia + Air_India + GO_FIRST + Indigo + SpiceJet + Days_left + Stops + Duration, data = Tdata1)

The 4-fold cross validation method shows results that are roughly similar to those of Model 1, as expected. The results of the experiments of Model 2 are summarized in the table below.

4-fold Cross V.	Exp 1	Exp 2	Exp 3	Exp 4	Average
Standard Error	2654	2707	2646	2637	2661
Coef R2	0.463	0.4631	0.4511	0.4507	0.456975
Adjusted R2	0.463	0.4631	0.451	0.4507	0.45695

4-fold Cross V.	ME	RMSE	MAE	MPE	MAPE
Experiment 1	-218.7673	2704.712	1952.219	-17.93366	37.77673
Experiment 2	-259.9417	2549.635	1884.596	-15.77522	36.37857
Experiment 3	762.1891	2749.636	1954.288	2.389889	29.21204
Experiment 4	-294.6237	2761.527	2000.194	-22.37444	40.15361
average	-2.7859	2691.3775	1947.82425	-13.42335775	35.8802375

The average of the coefficients of each experiment can be used to write the mathematical equation of the "***price***" as a function of the predictors. The equation parameters have the same interpretation as seen with Model 1.

Price *= 9026.662 - 2,898.235 * **AirAsia** - 544.145 * **Air_India** - 1400.182 * **GO_FIRST** - 1575.675 * **Indigo** - 1199.042 * **SpiceJet** - 147.87 * **Days_left** + 2109.385 * **Stops** + 30.655 * **Duration***

Model 3: Regression Tree

This model needs the "rpart" and "rpart.plot" packages to be loaded in the environment before we can start using the functions rpart() and prp().

First, we make the default tree using the rpart() function and then we can create a visualization of the tree using the prp() function. (See the default tree in Appendix, *Plot 6)*

default_tree <- rpart(Price ~., data = m2tData, method = "anova")
prp(default_tree, type = 1, extra = 1, under = TRUE)

Next step is to create a full tree that contains pure subsets normally with a cp = 0. However, given the size of the dataset, it wasn't possible for the software to grow the tree to full depth. We then set cp = **0.00009**, so that it isn't difficult to compute a tree.
full_tree <- rpart(Price ~ ., data = m2tData, method = "anova", cp = 0.00009, minsplit = 2, minbucket = 1) prp(full_tree, type = 1, extra = 1, under = TRUE)
printcp(full_tree). The full Tree is shown in the Appendix, ***(Plot 7).***

Now we select a minimum error tree, and we try to find a cp value that will yield a simpler tree that has an error within one std of the minimum error tree.

101	0.000111273	134	0.32514	0.33920	0.0035378
102	0.000109397	135	0.32503	0.33924	0.0035378
103	0.000109364	139	0.32460	0.33891	0.0035271
104	0.000105892	140	0.32449	0.33878	0.0035265
105	0.000104447	141	0.32438	0.33898	0.0035402
106	0.000103897	143	0.32417	0.33907	0.0035414

We found Tree 104 to have the minimum error of **0.33878**, with a **cp = 0.000105892** and std = **0.0035265.**
After that, we will prune the tree to the best-pruned tree by selecting the cp associated with the error within 1 standard deviation, (0.33878 + 0.0035265=**0.3423065**). Our best pruned Tree comes to be Tree 53. We will use that cp = **0.000224152** to prune to the Tree. For consistency, we will the cp slightly higher, so our pruned Tree displays exactly 53 Tree.

50	0.000252292	55	0.33755	0.34250	0.0035430
51	0.000235382	56	0.33730	0.34255	0.0035442
52	0.000232923	57	0.33706	0.34239	0.0035437
53	0.000224152	59	0.33660	0.34226	0.0035435
54	0.000217099	61	0.33615	0.34227	0.0035372
55	0.000213273	63	0.33572	0.34216	0.0035365

Best pruned Tree
pruned_tree <- prune(full_tree, cp= 0.000224154)
prp(pruned_tree, type = 1, extra = 1, under = TRUE)
printcp(pruned_tree)
summary(pruned_tree)

The Diagram of the best pruned tree and R output are shown in the Appendix (*Plot 8*).

To evaluate the performance of our regression tree, we compute the performance measures for prediction using the '*accuracy()*' function.

We use accuracy() function to evaluate the model with the parameters such as ME, RMSE, MAE, MPE, MAPE

accuracy(predicted_value, validationSet$Price)

```
          ME      RMSE       MAE       MPE      MAPE
: 8.792973 2117.348 1378.839 -8.23056 23.10078
```

7. Model Evaluation

The performance metrics of the three models are summarized in the table below for comparison.

Models Comparaison	ME	RMSE	MAE	MPE	MAPE	Standard Error	Coef R2	Adjusted R2
Model 1 (train-test split)	0.096	2663.149	1925.710	-12.993	35.337	2666.000	0.456	0.4564
Model 2 (4-fold cross Validation	-2.786	2691.378	1947.824	-13.423	35.880	2661.000	0.457	0.4570
Model 3 - Regression Tree	8.792973	2117.348	1378.839	-8.231	23.101	N//A	N/A	N/A

From the table, we can observe that Model 3, which is a Regression Tree, has the lowest RMSE and MAE values among the three models, suggesting that it may have the best predictive performance in terms of these particular metrics. The Mean Error (ME) is also lowest for Model 1, indicating less bias in predictions.

Model 1: Model 1 has a lower **ME = 0.096** (Mean Error) value close to **0** which suggests that it has the tendency to make more accurate predictions compared to other models, especially model 2 which has approximately the same performance metrics (RMSE, MAE, MPE, MAPE, Standard Error, R2) as model 1. So, Model 1 would be preferable to model 2.

Model 2: With a negative **ME = -2.786** suggests that it usually slightly overpredicts the values. However, even though it has a slightly higher Adjusted **R2 = 0.457** than other models, it still performs worse than both models in other performance metrics such as *MPE, MAE, MAPE, RMSE*. Hence the errors in this model are larger. That's a sign that the model has a poor performance, more inaccurate and inadequate and also possibly affected by outliers.

Model 3: Model 3 has a positive **ME = 8.793**, suggesting that it will mostly underpredict the values. However, it emerges as the best of the three models in terms of *MPE, MAE, MAPE, RMSE* by a larger margin. As a result, even though this model tends to underpredict the values, will mostly perform better them other models.

8. Discussion

In the developed regression tree models for predicting flight prices, we observed distinct influences from various predictors. The number 'days left' before the flight is a crucial predictor;

prices tended to increase as the departure date approached, possibly due to higher demand and lower seat availability. Our linear model assumes a perfect linear relationship between the price and the number of days left. However, the variation in price is not actually linear. We can notice on the graph that around **14 days** the prices jump to a higher level which cannot be foreseen from the earlier trend in the data. One would expect the same amount of change for each additional or less day, but what actually happens is that there is a noticeable surge in ticket price at around **15 days**. Therefore, our recommendation would be to buy a ticket with the number of days before the flight greater than **16**, which the model will underpredict due to the prices being constant after that interval.

We also saw that the duration of the flight has a positive influence on the price. This likely reflects the increased operational costs associated with longer journeys. The correlation for the duration isn't strong (**0.07**) and we can notice on the graph that there's more variation of the prices independently of the durations. So, the duration in fact, doesn't really affect the prices.

The choice of airlines also showed a marked impact on pricing, suggesting that different carriers have unique pricing strategies and service offerings that affect ticket costs. The number of stops was another influential factor, with direct flights often priced differently (lower) than those with one or more stops, which could be attributed to the added complexity and time of multi-stop journeys.

Some enhancements or adjustments we think might help support our decision include the way the data was gathered. We think factors like moment of the year such as New Year's Day, Christmas Day, Thanksgiving could greatly affect the way the prices change. Therefore, any data collected during those periods will not be a good for building a model. Also, our analysis was based on more flights with 1 or a few days left. That can create bias due to the fact that the analysis does not have enough information for the higher number of days left like 30, 40 or 49. We think getting as much data as possible for each particular range of days for the flights will help us increase the prediction performance of the model.

9 -Appendix

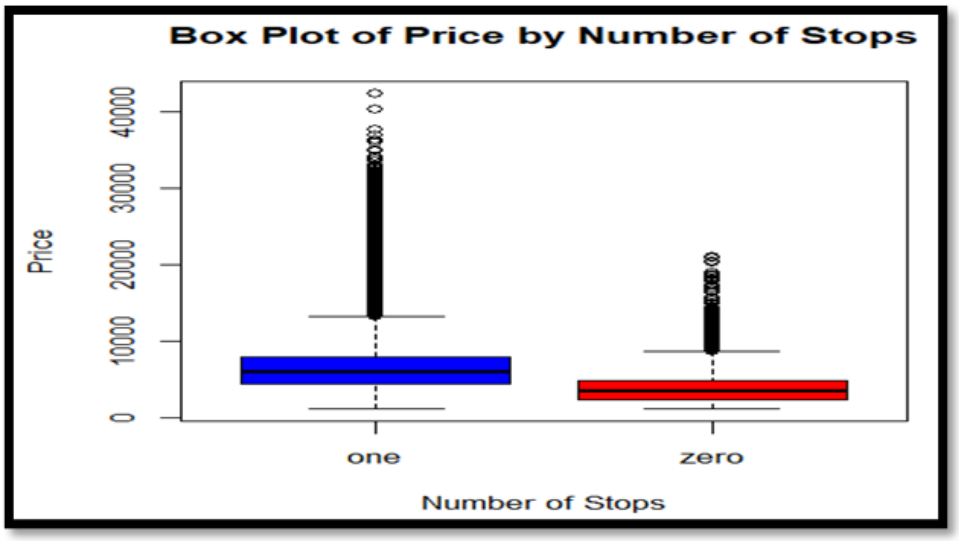
Output 1: Scatter plot of "*Price*" vs. "*Days_left*". Regression output results from R.

```
Coefficients:
             Estimate Std. Error t value            Pr(>|t|)
(Intercept) 10358.0220    14.7655   701.5 <0.0000000000000002 ***
days_left    -150.6552     0.5008  -300.8 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2986 on 194461 degrees of freedom
Multiple R-squared:  0.3175,    Adjusted R-squared:  0.3175
F-statistic: 9.048e+04 on 1 and 194461 DF,  p-value: < 0.00000000000000022
```

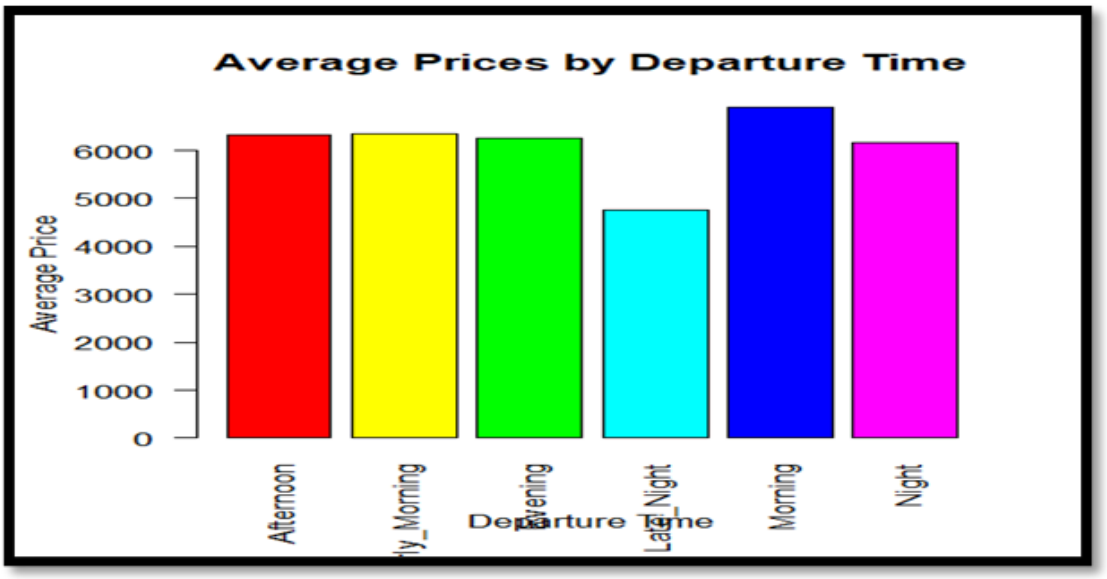
Output 2: Scatter of *Price* vs. *Duration* of the flight output

```
Coefficients:
            Estimate Std. Error t value            Pr(>|t|)
(Intercept) 4839.191     14.730   328.5 <0.0000000000000002 ***
duration     139.003      1.101   126.3 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3475 on 194461 degrees of freedom
Multiple R-squared:  0.07582,   Adjusted R-squared:  0.07582
F-statistic: 1.595e+04 on 1 and 194461 DF,  p-value: < 0.00000000000000022
```

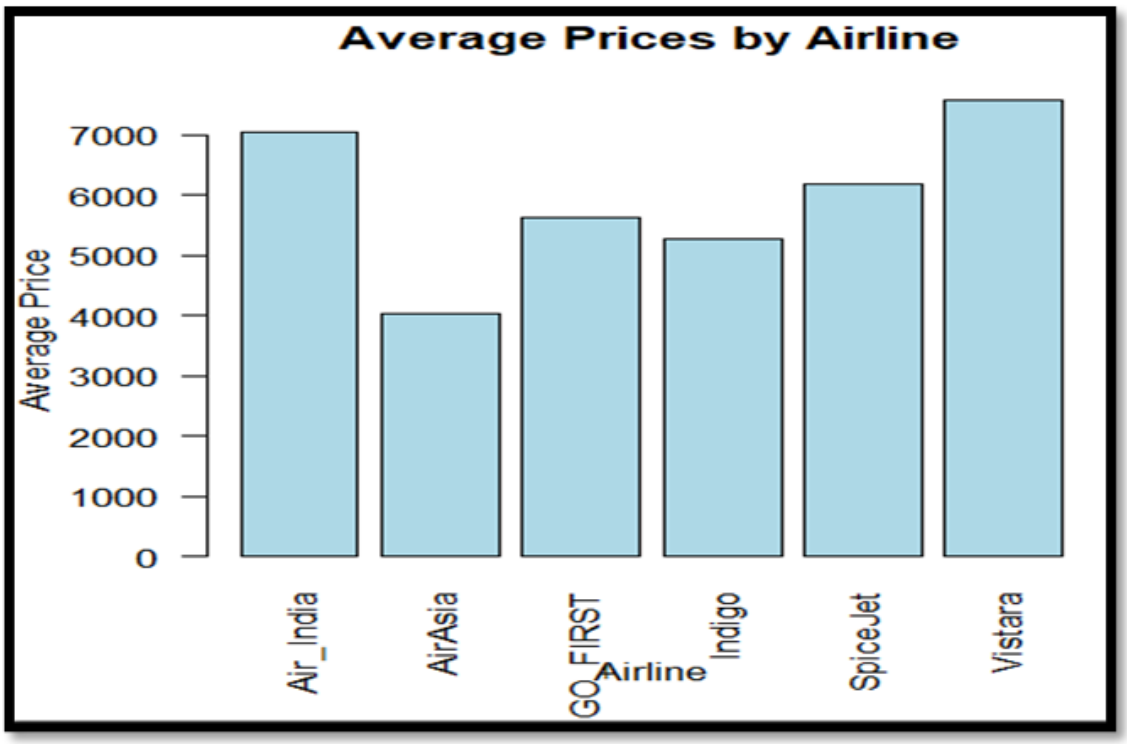
Plot 3: Boxplot comparing prices between *'zero'* stops and *'one'* stop.



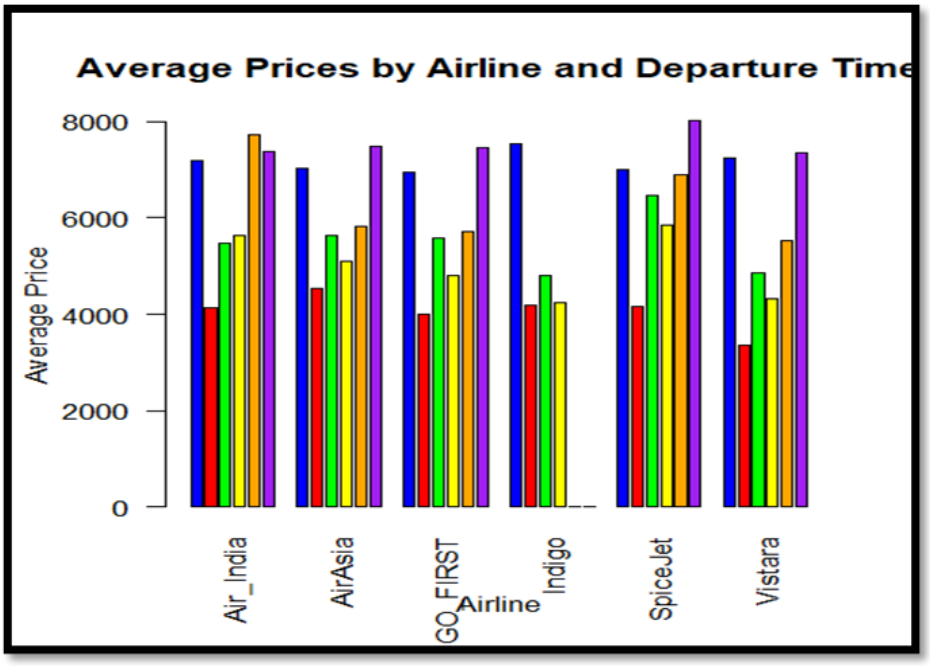
Plot 4: *Price* variation based on *Departure time*.



Plot 5: Average price Across Airlines.



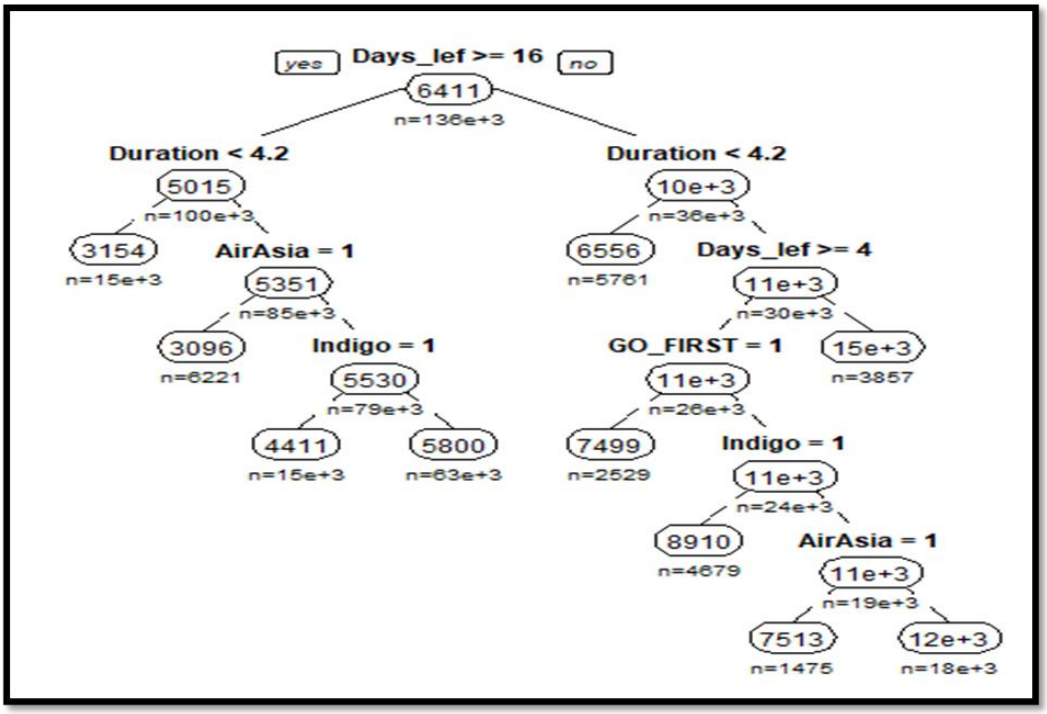
Grouped bar plot:



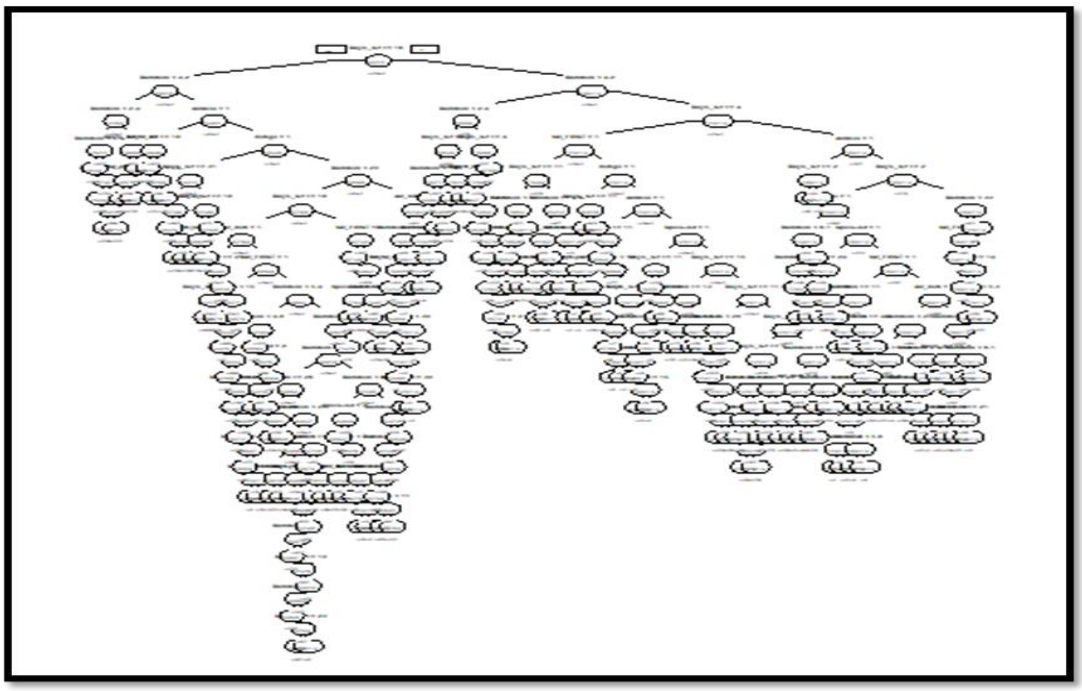
Output 3: Model 1 Coefficients

```
Coefficients:
                Estimate Std. Error  t value             Pr(>|t|)
(Intercept)   9103.5049    98.6032   92.325 <0.0000000000000002 ***
AirAsia      -2892.5940    30.4783  -94.907 <0.0000000000000002 ***
Air_India      -537.3683    20.0579  -26.791 <0.0000000000000002 ***
GO_FIRST      -1386.7464    25.4119  -54.571 <0.0000000000000002 ***
Indigo        -1564.1841    22.0844  -70.828 <0.0000000000000002 ***
SpiceJet      -1168.4866    36.3644  -32.133 <0.0000000000000002 ***
Days_left       -148.1667     0.5354 -276.748 <0.0000000000000002 ***
Stops           2119.8022    24.5820   86.234 <0.0000000000000002 ***
Not_Late_Night   -92.6839    94.9772   -0.976                0.329
Duration          31.2035     1.3748   22.696 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2666 on 136116 degrees of freedom
Multiple R-squared:  0.4564,    Adjusted R-squared:  0.4564
F-statistic: 1.27e+04 on 9 and 136116 DF,  p-value: < 0.00000000000000022
```

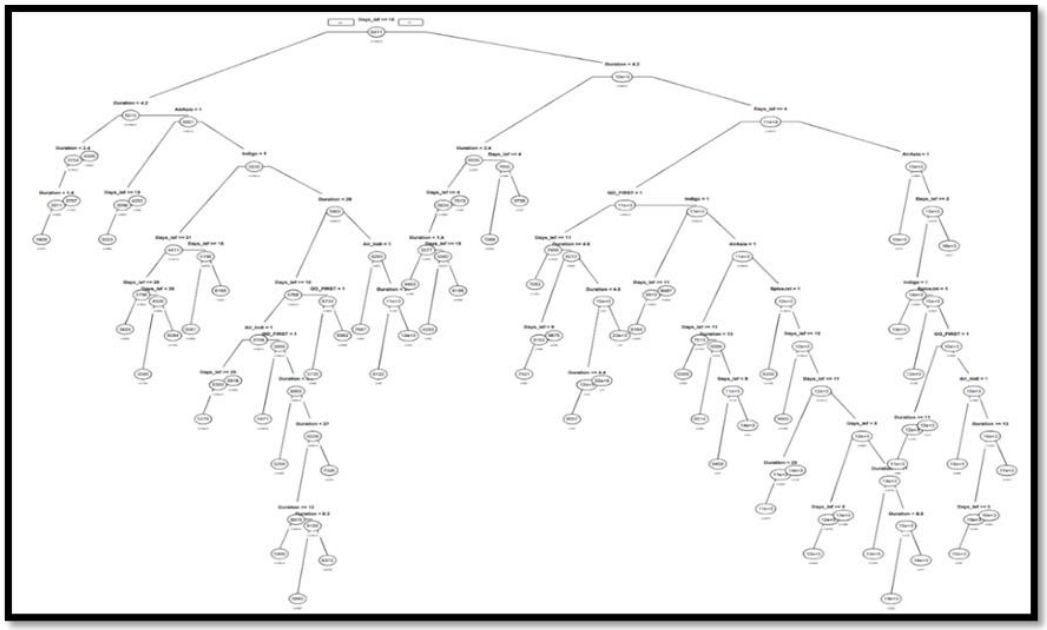
Plot 6: The default Decision Tree



***Plot 7**: Full Decision Tree*



Plot 8: The Diagram of the Best Pruned Tree



Best pruned Tree output

```
> printcp(full_tree)

Regression tree:
rpart(formula = Price ~ ., data = m2tData, method = "anova",
    cp = 0.00009, minsplit = 2, minbucket = 1)

Variables actually used in tree construction:
[1] Air_India AirAsia   Days_left Duration  GO_FIRST  Indigo    SpiceJet

Root node error: 1786611184978/136126 = 13124687

n= 136126

            CP nsplit rel error  xerror      xstd
1   0.414204888      0   1.00000 1.00001 0.0070238
2   0.055315709      1   0.58580 0.58582 0.0046111
3   0.035929063      2   0.53048 0.53051 0.0043351
4   0.031526603      3   0.49455 0.49464 0.0042886
5   0.018834421      4   0.46302 0.46314 0.0038506
6   0.014569604      5   0.44419 0.44432 0.0038392
7   0.013264536      6   0.42962 0.42975 0.0038199
8   0.012156933      7   0.41636 0.41644 0.0038079
9   0.011573774      8   0.40420 0.40506 0.0037766
10  0.007691591      9   0.39262 0.39274 0.0037926
11  0.004165650     10   0.38493 0.38506 0.0037695
12  0.003883399     11   0.38077 0.38118 0.0037640
```