FLIGHT PRICE PREDICTION

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

RAYIDI VENKAT

Regd. No: 20B91A12G3

Under Supervision of Mr. Gundala Nagaraju

Henotic Technology Pvt Ltd, Hyderabad

(Duration: 7th July, 2022 to 6th September, 2022)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(An Autonomous Institution)

Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada

CHINNA AMIRAM, BHIMAVARAM,

ANDHRA PRADESH

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(Autonomous)

Chinna Amiram, Bhimavaram

DEPARTMENT OF

INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the “**Summer Internship Report”** submitted by **Rayidi Venkat (20B91A12G3)** is work done by him/her and submitted during 2021 - 2022 academic year, in partial fulfillment of the requirements for the award of the Summer Internship Program for **Bachelor of Technology in Information Technology,** at **Henotic Technologies pvt Ltd** from *05.07.2022 to 04.09.2022*

Department Internship Coordinator **Dean -T & P Cell** **Head of the Department**

Table of Contents

Abstract

The Flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of flights. In the proposed system a predictive model will be created by applying machine learning algorithms to the collected historical data of flights. In this project we majorly targeted to uncover underlying trends of flight prices in India using historical data. This Project implements the validations or contradictions towards myths regarding the airline industry , a comparision study among various models in predicting the optimal time to buy ticket and the amount can be saved if done so. The prices of the flight tickets may vary on different features for example it depends on the departure time and the arrival time or the class of the seats like Economical and Business .From this Project you can also conclude which Airline at which time the ticket costs cheapest rate for a particular source to destination. The scope of the project can be extensively extended across the various routes to make significant savings on the purchase of flight prices across the Domestic Airline Market.

From the Dataset that we have taken from Kaggle we clearly observed that the prices of the tickets did not vary much with respect to the no of days the tickets prebooked.the arrival time and departure time is converted in to morning, evening categories..

According to the data set in india airlines only a few companies providing Business class tickets which hugely impacts the price of the ticket. With all these considerations we trained the data with seven different Supervised machine learning Regression Algorithms (Linear ,Decision Tree, Random Forest, Xgboost, Extratrees, gradientboosting, KNN algorithm) where three of them give approximately equal R2 Score (KNN,Extra trees,Xgboost) over 0.95.

1.0 Introduction

The tourism industry is changing fast and this is attracting a lot more travelers each year. The airline industry is considered as one of the most sophisticated industry in using complex pricing strategies. Now-a-days flight prices are quite unpredictable. The ticket prices change frequently. Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible. Using technology it is actually possible to reduce the uncertainty of flight prices. So here we will be predicting the flight prices using efficient machine learning techniques. Here we took previous records of the all the prices of tickets of the Indian Airlines, we trained the data with suitable machine Learning algorithms to predict the price of the tickets. The ultimate aim of the airlines is to get the maximum profit and where as for customer is to get the ticket at cheapest price.the Airline companies may chane their ticker prices at festival seasons where there is more traffic in booking the tickets they may usually increases the prices and where as in busy days the tend to low their prices inorder to get some attention of the travelling people.we cannot eventually predict the price of the tickets during this conditions but we can predict the price of the tickets considering all athe other features normal. Cost may likewise change with the seasons like winter, summer and celebration seasons. The extreme goal of the carrier is to build its income yet on the opposite side purchaser is searching at the least expensive cost. The least expensive accessible ticket changes over a period the cost of a ticket might be high or low.

Motivation is to help people who tends to pay more for the flight fare ticket for those who are naïve to this booking tickets process. This will also help us to get more exposure to the machine learning techniques that will help us to excel and improve in the existing skills.

1.2 What are the different types of Machine Learning?

There are four types of machine learning algorithms: supervised, semi-supervised, unsupervised and reinforcement.

Supervised learning:

In supervised learning, the machine is taught by example. The operator provides the machine learning algorithm with a known dataset that includes desired inputs and outputs, and the algorithm must find a method to determine how to arrive at those inputs and outputs. While the operator knows the correct answers to the problem, the algorithm identifies patterns in data, learns from observations and makes predictions. The algorithm makes predictions and is corrected by the operator – and this process continues until the algorithm achieves a high level of accuracy/performance.

UNDER THE UMBRELLA OF SUPERVISED LEARNING FALL: CLASSIFICATION, REGRESSION AND FORECASTING.

Classification:

In classification tasks, the machine learning program must draw a conclusion from observed values and determine to
what category new observations belong. For example, when filtering emails as 'spam' or 'not spam', the program must look at existing observational data and filter the emails accordingly.

Regression:

In regression tasks, the machine learning program must estimate – and understand – the relationships among variables. Regression analysis focuses on one dependent variable and a series of other changing variables – making it particularly useful for prediction and forecasting.

Forecasting:

Forecasting is the process of making predictions about the future based on the past and present data, and is commonly used to analyse trends.

Semi-supervised learning:

Semi-supervised learning is similar to supervised learning, but instead uses both labelled and unlabelled data. Labelled data is essentially information that has meaningful tags so that the algorithm can understand the data, whilst unlabelled data lacks that information. By using this
combination, machine learning algorithms can learn to label unlabelled data.

Unsupervised learning:

Here, the machine learning algorithm studies data to identify patterns. There is no answer key or human operator to provide instruction. Instead, the machine determines the correlations and relationships by analysing available data. In an unsupervised learning process, the machine learning algorithm is left to interpret large data sets and address that data accordingly. The algorithm tries to organise that data in some way to describe its structure. This might mean grouping the data into clusters or arranging it in a way that looks more organised.

As it assesses more data, its ability to make decisions on that data gradually improves and becomes more refined.Under the umbrella of unsupervised learning, fall:

Clustering:

Clustering involves grouping sets of similar data (based on defined criteria). It’s useful for segmenting data into several groups and performing analysis on each data set to find patterns.

Reinforcement learning:

Reinforcement learning focuses on regimented learning processes, where a machine learning algorithm is provided with a set of actions, parameters and end values. By defining the rules, the machine learning algorithm then tries to explore different options and possibilities, monitoring and evaluating each result to determine which one is optimal. Reinforcement learning teaches the machine trial and error. It learns from past experiences and begins to adapt its approach in response to the situation to achieve the best possible result.

1.3 Benefits of Using Machine Learning in Airline Pricing

- **Fraud detection** : by analyzing specific customers' flight and purchase patterns and coupling them with historical data, algorithms are able to identify passengers with suspicious credit card transactions and eliminate fraudulent cases, saving airline and travel companies millions of dollars every year.
- **Flight route Optimization** : machine learning-enabled systems that can find optimal flight routes, save money through lower operational costs, and result in higher customer retention. For this use case, various route characteristics, such as flight efficiency, air navigation charges, fuel consumption, and expected congestion level, can be analyzed.
- **Flight Delay Prediction :** as flight delays are dependent on a huge number of factors, including weather conditions and what's happening in other airports, predictive analytics and technology can be applied to analyze massive real-time data to predict flight delays, update departure time, and re-book customers' flights on time.
- **Flight Price Optimization :** Machine learning algorithms look for ways to maximize sales revenue in the longer term to ensure all flights are optimally booked. These include historical data such as past bookings, flight distance, willingness to pay, etc.

1.4 Airline Industry

The airline industry encompasses a wide range of businesses, called airlines, which offer air transport services for paying customers or business partners. These air transport services are provided for both human travellers and cargo, and are most commonly offered via jets, although some airlines also use helicopters.

Airlines may offer scheduled and/or chartered services and the airline industry forms a key part of the wider travel industry, providing customers with the ability to purchase seats on flights and travel to different parts of the world. The airline industry offers a variety of career paths, including pilots, flight attendants and ground crew.

1.3.1 AI / ML Role in Airline Industry

Machine Learning is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data. Machine learning (ML) is getting more and more attention and is becoming increasingly popular in many other industries. Within the Ainline Industry the Machine Learning plays a role in Predicting to set the Prices of the planes to the customers to book the flight.

2.0 Flight Price Prediction:

The purpose of this analysis is to analyze the data and build a Regressor using machine learning techniques such as decision tree Regression and random forest regression, Linear regression to predict the price of the particular flight based on the Airline company, source p and destination of airports , duration of travelling and class of the ticket and so on.

Main Drivers for Flight Price Prediction:

The following are the main drivers that influence Price of Flights :

- Airline
- Source
- Destination
- Duration of Travelling
- Departure Time
- Arrival Time
- Class of Ticket
- Total Number of Stops

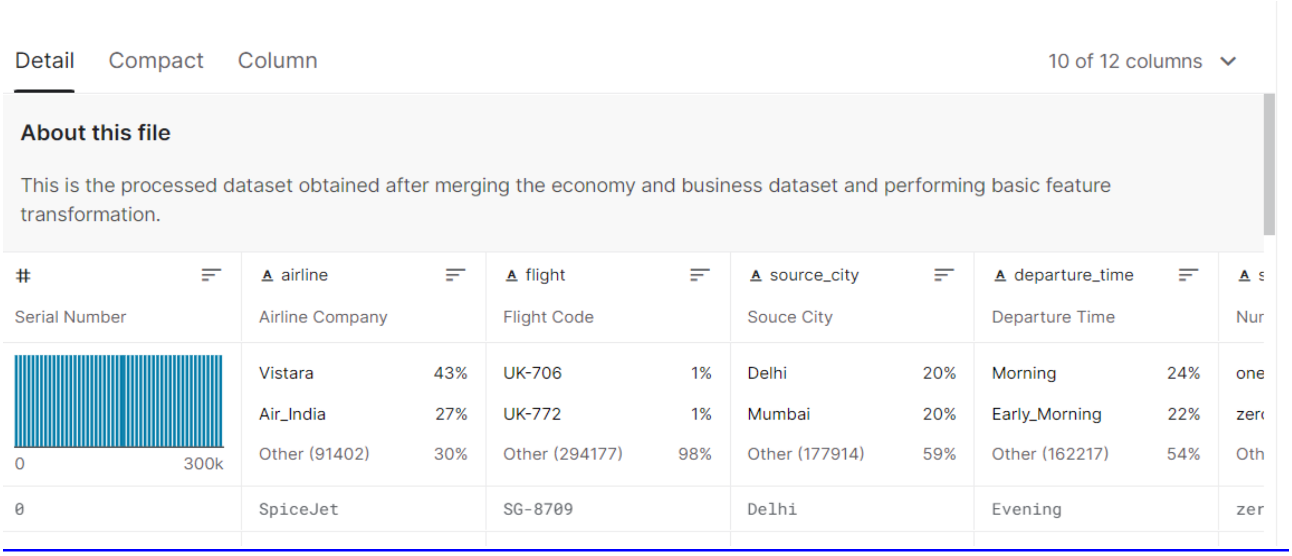
2.2 Internship Project - Data Link

The internship project data has taken from Kaggle and the link is

https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

Rows=300153

Columns=12



Detail Compact Column 10 of 12 columns

About this file

This is the processed dataset obtained after merging the economy and business dataset and performing basic feature transformation.

#	airline		flight		source_city		departure_time		s
Serial Number	Airline Company		Flight Code		Souce City		Departure Time		Nur
0 300k	Vistara	43%	UK-706	1%	Delhi	20%	Morning	24%	one
	Air_India	27%	UK-772	1%	Mumbai	20%	Early_Morning	22%	zer
	Other (91402)	30%	Other (294177)	98%	Other (177914)	59%	Other (162217)	54%	Oth
0	SpiceJet		SG-8709		Delhi		Evening		zer

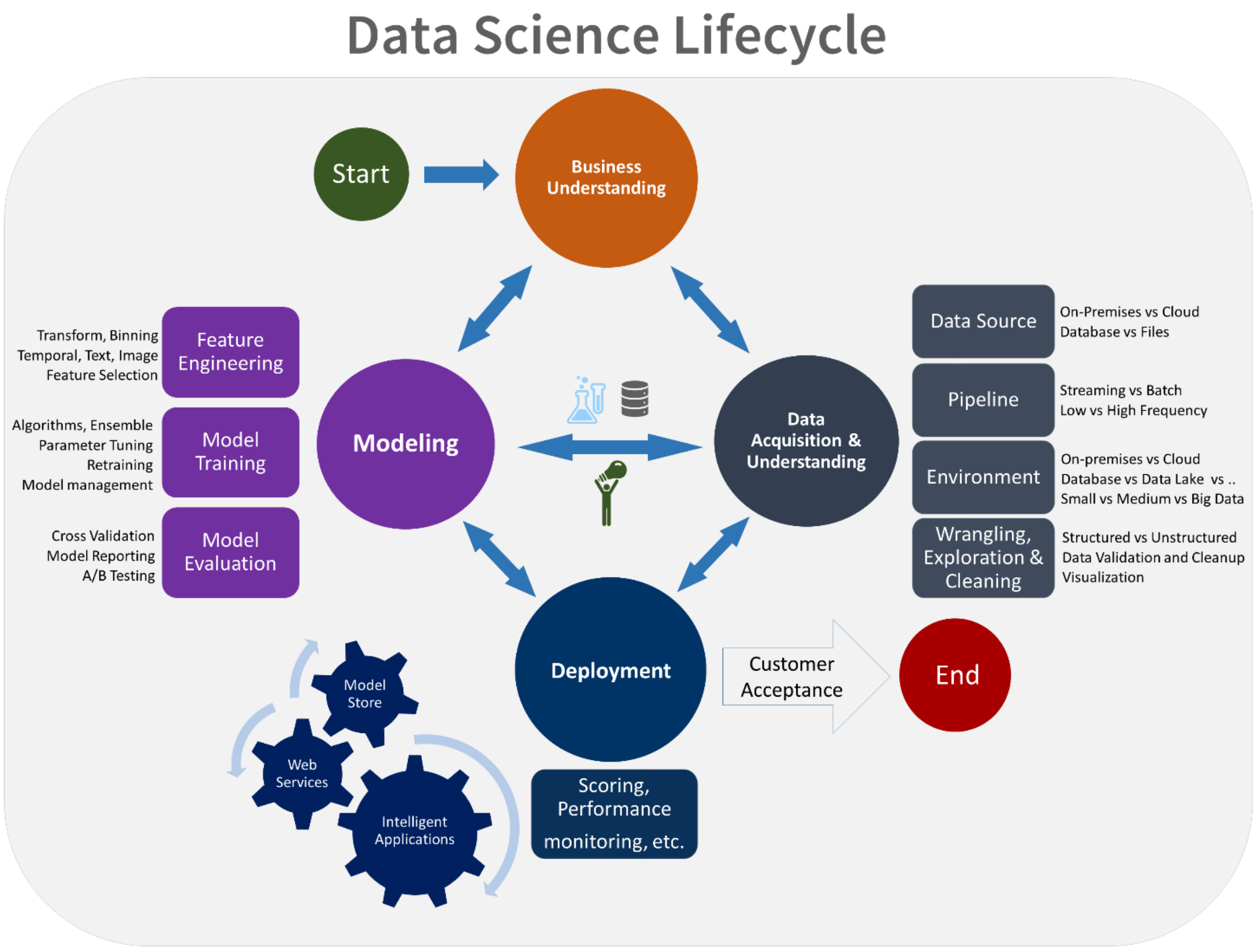
3.0 AI/ML Modelling and Results

3.1. Problem Statement:

Many People who want to travel in flights usually check for minimum or cheapest prices of flight . But they don't know how the flights prices vary .People who frequently travel on flights usually have idea on the Pricing of different flights. The task is to predict the pricing of different flights of india using Regression models of Supervised Machine learning

3.2 Data Science Project Life Cycle

Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of statistics and mathematics to extract useful insights and knowledge from data.



3.2.1 Data Exploratory Analysis

Exploratory data analysis has been done on the data to look for relationship and correlation between different variables and to understand how they impact or target variable.

3.2.2 Data Pre-processing

We removed variables which does not affect our target variable(Claimed_Target)as they mayadd noise and also increase our computation time,we checked the data for anomalous data points and outliers.We did principal component analysis on the data set to filter out unnecessary variables and to select only the important variables which have greater correlation with our target variable.

3.2.2.1 Check the Duplicate and low variation data

An important part of Data analysis is analyzing Duplicate Values and removing them. duplicated () method helps in analyzing duplicate values only. It returns a Boolean series which is True only for Unique elements.

3.2.2.2 Identify and address the missing variables

1. Use isnull() function to identify the missing values in the data frame
2. Use sum() functions to get sum of all missing values per column.
3. Now, remove the tuples with missing data.

3.2.2.3 Handling of Outliers

Following approaches can be used to deal with outliers:

1. Remove the observations
2. Imputation

1. **Remove the Observations**: We may explicitly delete outlier observation entries from our data so that they do not influence the training of our models.

2. **Imputation**: To impute the outliers, we can use a variety of imputation values, ensuring that no data is lost. As impute values, we can choose between the mean, median, mode, and boundary values.

3.2.2.4 Categorical data and Encoding Techniques

- **What is Categorical Data?**

Categorical data is a collection of information that is divided into groups. i.e., if an organisation or agency is trying to get a biodata of its employees, the resulting data is referred to as categorical. This data is called categorical because it may be grouped according to the variables present in the biodata such as sex, state of residence, etc.

There are two types of categorical data, namely; the nominal and ordinal data.

1. Nominal Data

This is a type of data used to name variables without providing any numerical value. Coined from the Latin nomenclature “Nomen” (meaning name), this data type is a subcategory of categorical data.

Nominal data is sometimes called “labelled” or “named” data. Examples of nominal data include name, hair colour, sex etc.

Mostly collected using surveys or questionnaires, this data type is descriptive, as it sometimes allows respondents the freedom to type in responses. Although this characteristic helps in arriving at better conclusions, it sometimes poses problems for researchers as they have to deal with so much irrelevant data.

2. Ordinal Data

This is a data type with a set order or scale to it. However, this order does not have a standard scale on which the difference in variables in each scale is measured.

3.2.2.5 Feature Scaling

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

Techniques to perform Feature Scaling:

1. **Min-Max Normalization**: This technique re-scales a feature or observation value with distribution value between 0 and 1.
2. **Standardization**: It is a very effective technique which re-scales a feature value so that It has distribution with 0 mean value and variance equals to 1.

3.2.3 Selection of Dependent and Independent variables

The dependent or target variable here is Claimed Target which tells us a particular policy holder has filed a claim or not the target variable is selected based on our business problem and what we are trying to predict.

The independent variables are selected after doing exploratory data analysis and we used Boruta to select which variables are most affecting our target variable.

3.2.4 Data Sampling Methods

The data we have is highly unbalanced data so we used some sampling methods which are used to balance the target variable so we our model will be developed with good accuracy and precision. We used three Sampling methods

3.2.4.1 Stratified sampling

Stratified sampling randomly selects data points from majority class so they will be equal to the data points in the minority class. So, after the sampling both the class will have same no of observations.

It can be performed using strata function from the library sampling.

3.2.4.2 Simple random sampling

Simple random sampling is a sampling technique where a set percentage of the data is selected randomly. It is generally done to reduce bias in the dataset which can occur if data is selected manually without randomizing the dataset.

We used this method to split the dataset into train dataset which contains 70% of the total data and test dataset with the remaining 30% of the data.

3.2.5 Models Used for Development

We built our predictive models by using the following seven algorithms

We built our predictive models by using the following seven algorithms

Total we discussed 7 models in here. They are:

1. Linear Regression

5. XGboost Regressor

2. Decision Tree Regressor
3. Random Forest Regressor
4. K Neighbours Regressor
6. Extra Trees Regressor
7. Gradient boosting Regressor

3.2.5.1 Linear Regression(Model 1)

Linear Regression is a **machine learning algorithm based on supervised learning**. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.

3.2.5.2 Decision Tree Regressor(Model 2)

A regression tree is a type of **decision tree**. analysis to predict values of the target field. The predictions are based on combinations of values in the input fields. A regression tree calculates a predicted mean value for each node in the tree.

3.2.5.3 RandomForest Regressor (Model 3)

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts thefinal output.

3.2.5.4 K Neighbours Regressor (Model 4)

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same *neighbourhood*. The size of the neighbourhood needs to be set by the analyst or can be chosen using cross-validation (we will see this later) to select the size that minimises the mean-squared error.

3.2.5.5 XGboost Regressor (Model 5)

XGBoost is a powerful approach for building supervised regression models.

These are some key members of XGBoost models, each plays an important role.

RMSE: It is the square root of mean squared error (MSE).

MAE: It is an absolute sum of actual and predicted differences, but it lacks mathematically, that's why it is rarely used, as compared to other metrics.

3.2.5.6 Extra Trees Regressor(Model 6)

Extra Trees (Extremely Randomized Trees) the ensemble learning algorithms. It constructs the set of decision trees. During tree construction the decision rule is randomly selected. This algorithm is very similar to Random Forest except random selection of split values.

3.2.5.7 Gradient Boosting Regressor(Model 7)

Gradient boosting Regression calculates the difference between the current prediction and the known correct target value.

This difference is called residual. After that Gradient boosting Regression trains a weak model that maps features to that residual. This residual predicted by a weak model is added to the existing model input and thus this process nudges the model towards the correct target. Repeating this step again and again improves the overall model prediction.

3.3 AI / ML Models Analysis and Final Results

We used our train dataset to build the above models and used our test data to check the accuracy and performance of our models.

3.3.1 K Neighbours Regressor (Top Model 1)

```
from sklearn.neighbors import KNeighborsRegressor
modelKNN = KNeighborsRegressor(n_neighbors=5)
# Fit the model with train data
modelKNN.fit(x_train, y_train)
   # predict the model with test data
y_pred = modelKNN.predict(x_test)
   # Print the model name
print('Model Name: ', modelKNN)
   # Evaluation metrics for Regression analysis
from sklearn import metrics
print('Mean Absolute Error (MAE):', round(metrics.mean_absolute_error(y_test, y_pred),3))
print('Mean Squared Error (MSE):', round(metrics.mean_squared_error(y_test, y_pred),3))
print('Root Mean Squared Error (RMSE):',
round(np.sqrt(metrics.mean_squared_error(y_test, y_pred)),3))
print('R2_score:', round(metrics.r2_score(y_test, y_pred),6))
print('Root Mean Squared Log Error (RMSLE):',
round(np.log(np.sqrt(metrics.mean_squared_error(y_test, y_pred))),3))
   # Define the function to calculate the MAPE - Mean Absolute Percentage Error
def MAPE (y_test, y_pred):
   y_test, y_pred = np.array(y_test), np.array(y_pred)
   return np.mean(np.abs((y_test - y_pred) / y_test)) * 100
   # Evaluation of MAPE
result = MAPE(y_test, y_pred)
print('Mean Absolute Percentage Error (MAPE):', round(result, 2), '%')


   # Calculate Adjusted R squared values


r_squared = round(metrics.r2_score(y_test, y_pred),6)
adjusted_r_squared = round(1 - (1-r_squared)*(len(y)-1)/(len(y)-x.shape[1]-1),6)
```

```
print('Adj R Square: ', adjusted_r_squared)
print('-----------------------------------------------------------------------------------------------------------')
   #------------------------------------------------------------------------------------------
new_row = {'Model Name' : modelKNN,
            'Mean_Absolute_Error_MAE' : metrics.mean_absolute_error(y_test, y_pred),
            'Adj_R_Square' : adjusted_r_squared,
            'Root_Mean_Squared_Error_RMSE' : np.sqrt(metrics.mean_squared_error(y_test,
y_pred)),
            'Mean_Absolute_Percentage_Error_MAPE' : result,
            'Mean_Squared_Error_MSE' : metrics.mean_squared_error(y_test, y_pred),
            'Root_Mean_Squared_Log_Error_RMSLE':
np.log(np.sqrt(metrics.mean_squared_error(y_test, y_pred))),
            'R2_score' : metrics.r2_score(y_test, y_pred)}
```

3.3.2 Extra Trees Python Code(Top Model 2)

```
from sklearn.ensemble import ExtraTreesRegressor
modelETR = ExtraTreesRegressor()
   # Fit the model with train data
modelETR.fit(x_train, y_train)
   # Predict the model with test data
y_pred = modelETR.predict(x_test)
   # Print the model name
print('Model Name: ', modelETR)
   # Evaluation metrics for Regression analysis
from sklearn import metrics
print('Mean Absolute Error (MAE):', round(metrics.mean_absolute_error(y_test, y_pred),3))
print('Mean Squared Error (MSE):', round(metrics.mean_squared_error(y_test, y_pred),3))
print('Root Mean Squared Error (RMSE):', round(np.sqrt(metrics.mean_squared_error(y_test, y_pred)),3))
print('R2_score:', round(metrics.r2_score(y_test, y_pred),6))
print('Root Mean Squared Log Error (RMSLE):', round(np.log(np.sqrt(metrics.mean_squared_error(y_test,
y_pred))),3))
```

```
# Define the function to calculate the MAPE - Mean Absolute Percentage Error
def MAPE (y_test, y_pred):
    y_test, y_pred = np.array(y_test), np.array(y_pred)
    return np.mean(np.abs((y_test - y_pred) / y_test)) * 100
    # Evaluation of MAPE
result = MAPE(y_test, y_pred)
print('Mean Absolute Percentage Error (MAPE):', round(result, 2), '%')
    # Calculate Adjusted R squared values
r_squared = round(metrics.r2_score(y_test, y_pred),6)
adjusted_r_squared = round(1 - (1-r_squared)*(len(y)-1)/(len(y)-x.shape[1]-1),6)
print('Adj R Square: ', adjusted_r_squared)
print('--------------------------------------------------------------------------------------------------------------')
    #---------------------------------------------------------------------------------------------
new_row = {'Model Name' : modelETR,
            'Mean_Absolute_Error_MAE' : metrics.mean_absolute_error(y_test, y_pred),
            'Adj_R_Square' : adjusted_r_squared,
            'Root_Mean_Squared_Error_RMSE' : np.sqrt(metrics.mean_squared_error(y_test, y_pred)),
            'Mean_Absolute_Percentage_Error_MAPE' : result,
            'Mean_Squared_Error_MSE' : metrics.mean_squared_error(y_test, y_pred),
            'Root_Mean_Squared_Log_Error_RMSLE': np.log(np.sqrt(metrics.mean_squared_error(y_test, y_pred))),
            'R2_score' : metrics.r2_score(y_test, y_pred)}
```

3.3.3 Remainings models code

```
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import ExtraTreesRegressor
# Create objects of Regression / Regressor models with default hyper-parameters
modelmlg = LinearRegression()
modeldcr = DecisionTreeRegressor()
modelrfr = RandomForestRegressor()
modelSVR = SVR()
modelKNN = KNeighborsRegressor(n_neighbors=5)
```

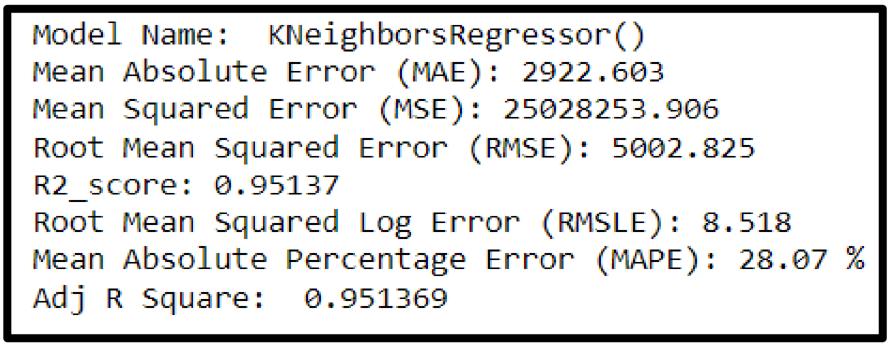
```
# Evalution matrix for all the algorithms
MM = [modelmlg, modeldcr, modelrfr, modelKNN, modelGBR]
for models in MM:
   # Fit the model with train data
   models.fit(x_train, y_train)
   # Predict the model with test data
   y_pred = models.predict(x_test)
   # Print the model name
   print('Model Name: ', models)
   # Evaluation metrics for Regression analysis
   from sklearn import metrics
   print('Mean Absolute Error (MAE):', round(metrics.mean_absolute_error(y_test, y_pred),3))
   print('Mean Squared Error (MSE):', round(metrics.mean_squared_error(y_test, y_pred),3))
   print('Root Mean Squared Error (RMSE):', round(np.sqrt(metrics.mean_squared_error(y_test, y_pred)),3))
   print('R2_score:', round(metrics.r2_score(y_test, y_pred),6))
   print('Root Mean Squared Log Error (RMSLE):', round(np.log(np.sqrt(metrics.mean_squared_error(y_test, y_pred))),3))
   # Define the function to calculate the MAPE - Mean Absolute Percentage Error
# Define the function to calculate the MAPE - Mean Absolute Percentage Error
   def MAPE (y_test, y_pred):
      y_test, y_pred = np.array(y_test), np.array(y_pred)
      return np.mean(np.abs((y_test - y_pred) / y_test)) * 100
# Evaluation of MAPE
   result = MAPE(y_test, y_pred)
   print('Mean Absolute Percentage Error (MAPE):', round(result, 2), '%')
   # Calculate Adjusted R squared values
   r_squared = round(metrics.r2_score(y_test, y_pred),6)
   adjusted_r_squared = round(1 - (1-r_squared)*(len(y)-1)/(len(y)-x.shape[1]-1),6)
   print('Adj R Square: ', adjusted_r_squared)
   new_row = {'Model Name' : models,
            'Mean_Absolute_Error_MAE' : metrics.mean_absolute_error(y_test, y_pred),
            'Adj_R_Square' : adjusted_r_squared,
            'Root_Mean_Squared_Error_RMSE' : np.sqrt(metrics.mean_squared_error(y_test, y_pred)),
            'Mean_Absolute_Percentage_Error_MAPE' : result,
            'Mean_Squared_Error_MSE' : metrics.mean_squared_error(y_test, y_pred),
            'Root_Mean_Squared_Log_Error_RMSLE': np.log(np.sqrt(metrics.mean_squared_error(y_test, y_pred))),
            'R2_score' : metrics.r2_score(y_test, y_pred)}
```

4.0 Conclusions and Future work

The model results in the following order by considering the model R2_Score and adj R square Score

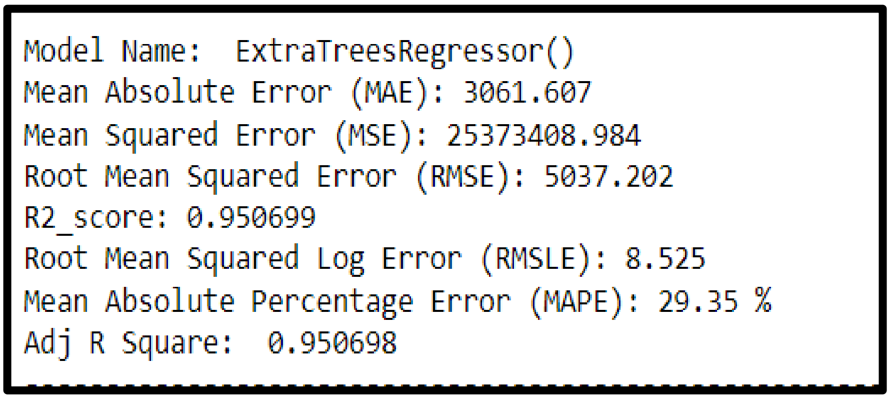
1) **K Neighbours Regressor**
2) **Extra Tree Regressor**
3) **XGB Regressor**

We recommend model - **K Neighbours Regressor model Which is the Best Fit Model with highest R2 score among all for the given data set**



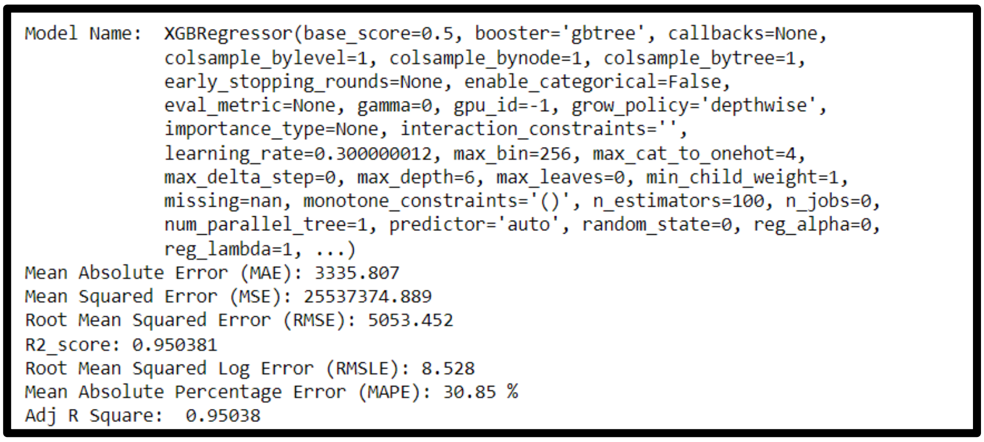
```
Model Name:  KNeighborsRegressor()
Mean Absolute Error (MAE): 2922.603
Mean Squared Error (MSE): 25028253.906
Root Mean Squared Error (RMSE): 5002.825
R2_score: 0.95137
Root Mean Squared Log Error (RMSLE): 8.518
Mean Absolute Percentage Error (MAPE): 28.07 %
Adj R Square:  0.951369
```

KNN Regressor



```
Model Name:  ExtraTreesRegressor()
Mean Absolute Error (MAE): 3061.607
Mean Squared Error (MSE): 25373408.984
Root Mean Squared Error (RMSE): 5037.202
R2_score: 0.950699
Root Mean Squared Log Error (RMSLE): 8.525
Mean Absolute Percentage Error (MAPE): 29.35 %
Adj R Square:  0.950698
```

Extra Tree Regressor



```
Model Name:  XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, ...)
Mean Absolute Error (MAE): 3335.807
Mean Squared Error (MSE): 25537374.889
Root Mean Squared Error (RMSE): 5053.452
R2_score: 0.950381
Root Mean Squared Log Error (RMSLE): 8.528
Mean Absolute Percentage Error (MAPE): 30.85 %
Adj R Square:  0.95038
```

XG boost Regressor

5.0 References

1. The Dataset is taken from Kaggle
2. Some of the above information is taken from many Websites like sklearn

The above information is taken from the websites:

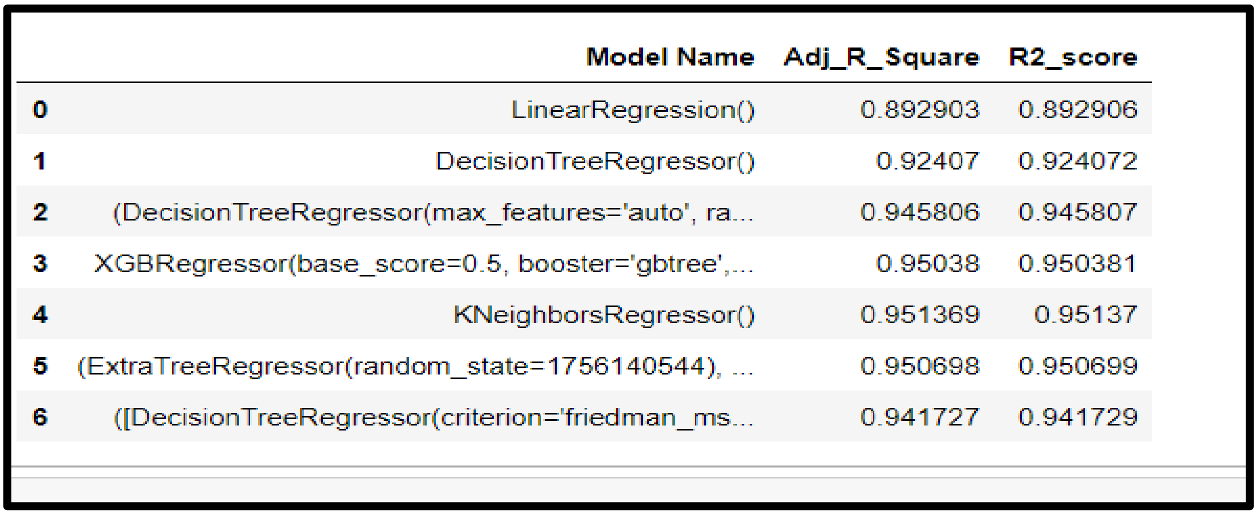
- Sklearn : scikit-learn: machine learning in Python — scikit-learn 1.1.2 documentation
- GeeksforGeeks : GeeksforGeeks | A computer science portal for geeks

About Dataset:

The objective of the study is to analyse the flight booking dataset obtained from “Ease My Trip” website and to conduct various statistical hypothesis tests in order to get meaningful information from it.

6.0 Appendices

6.1 Python Code Results

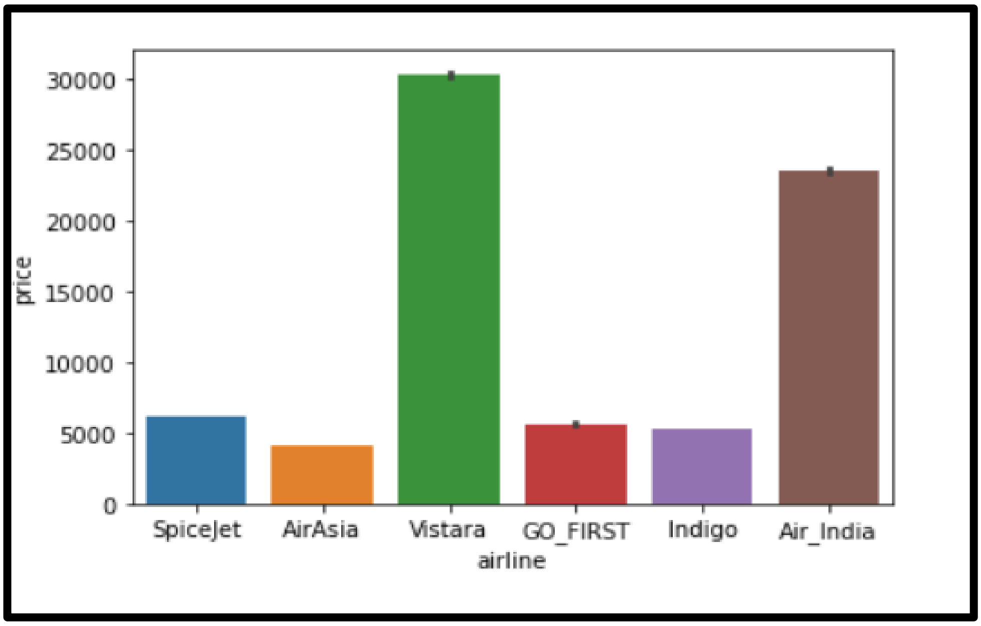


	Model Name	Adj_R_Square	R2_score
0	LinearRegression()	0.892903	0.892906
1	DecisionTreeRegressor()	0.92407	0.924072
2	(DecisionTreeRegressor(max_features='auto', ra...	0.945806	0.945807
3	XGBRegressor(base_score=0.5, booster='gbtree',...	0.95038	0.950381
4	KNeighborsRegressor()	0.951369	0.95137
5	(ExtraTreeRegressor(random_state=1756140544), ...	0.950698	0.950699
6	([DecisionTreeRegressor(criterion='friedman_ms...	0.941727	0.941729

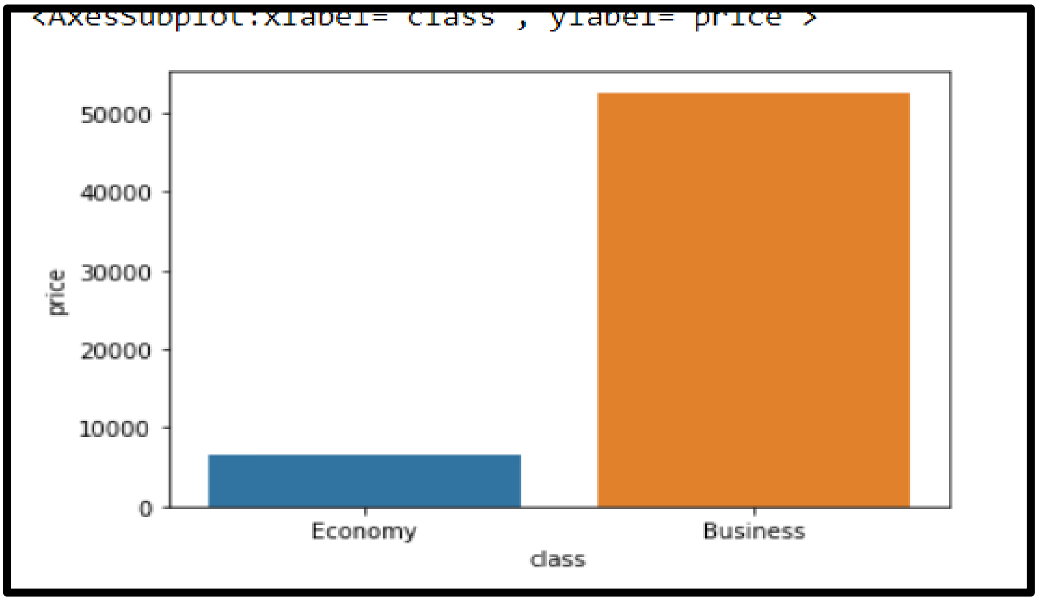
Adj_R_Square and R2_score of all Regressor Model used

6.2 List of Chart

6.2.1 Chart 01 : Highest Prices of Tickets by Airlines

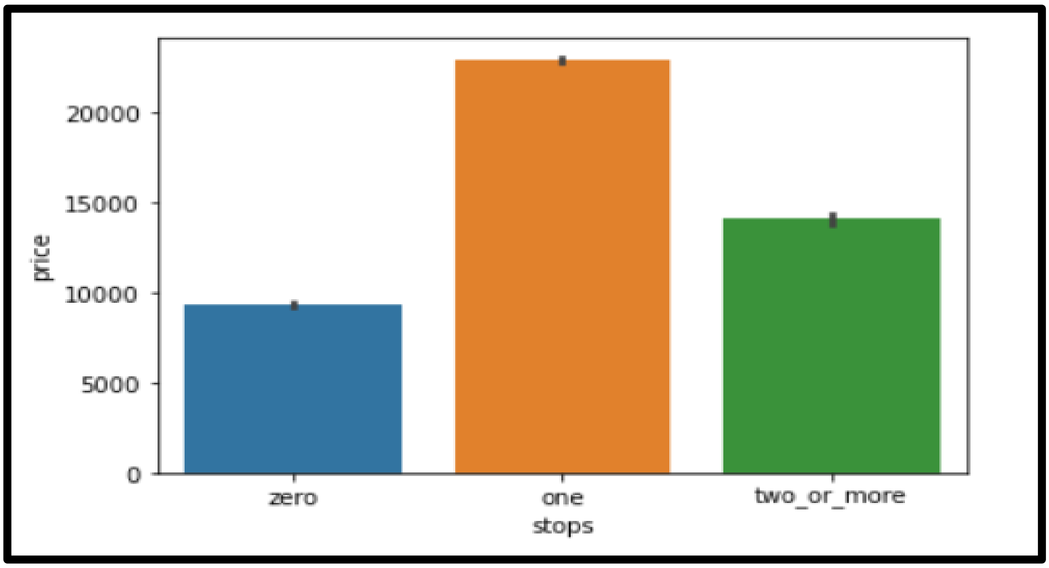


6.2.2 Chart 02 : Prices that vary between economy and Business Class



```
<AxesSubplot:xlabel= class , ylabel= price >
```

6.2.3 Chart 03 : Prices Bases on the No.of Stops:



price
zero
one
two_or_more
stops