**Flight price prediction using Machine Learning and IBM Cloud**

**Smartinternz Externship Program project Report**

**Course: Applied Data Science**

**Campus: VIT Vellore**

**Group no: 609**

**Group Members**

**Jai Khare 20BDS0297**

**Syed Qasim Kaif 20BDS0343**

**Vedant Kumar Khetan 20BCE0503**

**Shubhankar Gulati 20BCE2942**

**Abstract:**

Everyone loves to travel by flight, flight travel has become the most frequent way of travelling across the globe. Going along with the study, the charge of travelling by plane changes frequently, which also includes the day and night Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off, this will provide the perfect time to purchase the plane ticket. The motivation behind this project is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project we have implemented the Random Forest, Gradient boosting, and multiple linear Regression algorithms in order to get the most accurate prediction of flight fares.

**Introduction**

The ticketing system is to buy the ticket many days before the flight takes off to avoid the effects of the most extreme fees. Air routes usually do not agree with this procedure. Airlines can reduce costs at times when they need to build a market and when tickets are less affordable. They can maximize costs. So the price may depend on various factors. In order to predict the costs, this business uses AI to show the ways of the tickets after some time. All groups are free to adjust the price of their tickets at any moment. An explorer can save cash by booking the lowest cost flight. People who have traveled frequently by plane are aware of the fluctuations in prices.

Airlines use comprehensive Revenue Management principles to implement distinctive rating systems. As a result, the evaluation system changes the fee depending on the time, season and holidays to change the header or footer on the following pages. The ultimate goal of airways is to make a profit while the customer seeks the minimum rate. Customers usually try to buy a ticket well in advance of the departure date to avoid the increase in ticket prices as the date approaches. But in reality it is not. A customer may end up paying more than they should for the same location.

This project aims to predict flight prices for different flights using the machine learning model. The user receives the expected values, and using these as a guide, they can choose whether to purchase tickets.

In the current scenario, airlines are trying to manipulate ticket prices to maximize their profits. Many people frequently take flights, so they are aware of the optimum times to get affordable tickets. But there are also many people who have no experience while booking tickets and they end up falling into the trap of discounts from the companies where they actually end up spending more than they should have. By providing clients with the information they need to order tickets at the proper moment, the suggested system can help them save millions of rupees

The proposed problem statement is “Flight price prediction System".

**Proposed Model**

* **Random forest**

Random Forest is a supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Working: Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

* **Gradient Boosting Algorithm**

Gradient boosting is a method standing out for its prediction speed and accuracy, particularly with large and complex datasets. From Kaggle competitions to machine learning solutions for business, this algorithm has produced the best results. We already know that errors play a major role in any machine learning algorithm. There are mainly two types of error, bias error, and variance error. The gradient boost algorithm helps us minimize bias error of the model.

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. But how do we do that? How do we reduce the error? This is done by building a new model on the errors or residuals of the previous model.

When the target column is continuous, we use Gradient Boosting Regressor whereas when it is a classification problem, we use Gradient Boosting Classifier. The only difference between the two is the “Loss function”. The objective here is to minimize this loss function by adding weak learners using gradient descent. Since it is based on loss function hence for regression problems, we’ll have different loss functions like Mean squared error (MSE) and for classification, we will have different for e.g., log-likelihood.

* **Multiple Linear Regression**

**Multiple linear regression** is a generalization of simple linear regression, in the sense that this approach makes it possible to evaluate the linear relationships between a response variable (quantitative) and several explanatory variables

In the real world, multiple linear regression is used more frequently than simple linear regression. This is mostly the case because:

* Multiple linear regression allows to evaluate the relationship between two variables, while **controlling for the effect** (i.e., removing the effect) **of other variables**.
* With data collection becoming easier, more variables can be included and taken into account when analyzing data.

**Results and Discussions**

**Data Set**

The data set was obtained through the Kaggle website. The data set contains the data of flights for the year 2016 across various destinations in India the dataset comprises of 11 features namely.

Name of airlines, date of Journey, Source, Destination,Route,Dep\_Time,Arrival\_time,Duration,Total\_stop,Additional\_info,price

**Data Preprocessing**

In this phase we preprocessed the data and explored the data checked for null values, replaced the missing values with approximate values and we proceeded further by checking the outliers for the ‘price’ column and did label encoding and scaling the data before splitting them into training and testing.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Model building**

The second phase of our project we went about building the model once our data was cleaned we build our model using multi linear regression, Gradient boosting and Random forest algorithms.

**Models:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Model Evaluation**

Once our model was built we imported the required evaluation metrics from sklearn library in order to test each model and to get a clear idea as to how each algorithm is performing and choosing the best algorithm to deploy the model.

The following performance metrics were used for evaluation.

**MAE (Mean Absolute Error)**

The mean of the absolute difference between the expected and actual numbers is effectively added to determine the mean absolute error.

**MAE = 1/n [∑ (y-ý)]**

The expected output values are y' and the actual output values are y. There are n total data points. Your model will perform better the lower the MAE number is.

**MSE (mean square error)**

The root mean square error exponentiates the difference of the true a predicted output values before summing them instead using an absolute value.

**MSE = 1/n [∑ (y-ý) 2]**

y=actual output values ý=predicted output values n = Total number of data points

**RMSE (root mean square error)**

RMSE is measured by taking the square root of the mean squared difference between forecast and actual value.

**RMSE = √1/n [∑ (y-ý) 2]**

The expected output values are y' and the actual output values are y.

There are n total data points. The higher the performance of a model, the more RMSE is bigger than MAE and smaller than RMSE value comparing different models.

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

**Model deployment**

Out of the three models selected gradient boosting gave us the best prediction results as a result we selected gradient boosting model and deployed the model on IBM Watson cloud interface.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Application building**

This is the final phase of our project where we built a web application in order to display the prediction results to the user. We integrated this web application with the model that we have built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

The application building task is done using Flask.

A screenshot of a computer program

Description automatically generated

A computer screen shot of a plane

Description automatically generated

A computer screen shot of a plane

Description automatically generated

A screenshot of a computer

Description automatically generated

**Conclusion and future scope**

More routes can be added, and the same analysis can be extended to major airports and travel routes in India. More data points and historical data should be considered for analysis. This will train the model better and provide better accuracy and more savings.

Additional rules may be added to rule-based learning based on our understanding of the industry, including offer periods provided by airlines. Development of a more user-friendly interface for different routes giving users more flexibility

Currently, there are many fields where prediction services are used such as stock price prediction tools used by stockbrokers and services like estimating that provide estimated value of house prices. That's why there exists the demand for services like this in the aviation industry which can assist customers in booking tickets. There are many of them that examine the work that has been done on it using various techniques and further research is needed to improve prediction accuracy using different algorithms. Results can be more accurately obtained by using data that is more accurate and has better features.