Flight Fare Prediction

Vishvjeet Yadav

11905631

KM018

Github Link : https://github.com/vishvjeet-yadav/Flight-Fare-Prediction

Problem Statement

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travelers saying that flight ticket prices are so unpredictable. As data scientists, we are gonna prove that given the right data anything can be predicted. Here you will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities. Size of training set: 10683 records.

Literature Review

"RWA: A Regression-based Scheme for Flight Price" Prediction by Zhenbang Wang, April 2020.

Flying has become the primary transportation method for long-distance travel. Most of the travelers are intend to purchase the tickets with lowest cost. In practice, many travelers tend to purchase flight tickets as early as possible to avoid possible price hikes. However, this type of purchase behavior does not always lead to the most economical flight tickets. In our research, we proposed a regression-based scheme, RWA, to improve the accuracy of flight price prediction. Specifically, we first collected a variety of different flight price data sets from publicly-available travel websites. After that, we devised a data splitting method to divide the training data set into two partitions because the price change patterns in these partitions are entirely different. Finally, RWA is applied to each of the partitions to arrive at the accuratelypredicted flight price. To verify the effectiveness of RWA, extensive experiments were carried out in our research.

In our research, we proposed a slope-based dataset spitting method and a regression-based weighted average algorithm, RWA, for flight price prediction. Flight price prediction can be regarded as a time series problem. In the old study, all the studies always treat the data as a whole part. Comparing with their method, we split the data into two parts according to the trend of the price curve and make a hypothesis that the different datasets can have different characters. To get the best performance of each dataset, we should have different feature sets on different datasets. Then we use a two-phase experiment to get the best feature set for each dataset. In phase one, we use a brute force algorithm to search the feature combination in two redundant feature sets, which has the best performance on our datasets. Moreover, in phase two, we test the importance of different features for each dataset and remove one more feature, which has the worst influence on the prediction result to get the best performance. The result shows that the best feature sets for each dataset are different from each other.

In 2003, Etzioni et al. proposed a model that aims to tell users this is the perfect timing to buy the ticket or not [18]. The model combines moving average, rule learning and Q-learning together. Their data contains two routes: Los Angeles to Boston and Seattle to Washington, D.C. And there are five features in their model, which are flight number, hours until the departure date, airline, price and route. They generate several rules, for example, when the hours before takeoff is greater or equals to 252 and the current price is greater or equals to 2223 and route is from LA to Boston, we should wait. After that, there should be several "buy" and "wait" suggestions. The final result is achieved by using an ensemble method doing the voting. By utilizing the sequence of buy or wait signal, the cost of each stimulating passenger was calculated. The total amount of moneysaving by using their strategy can reach 61.8%. However, their method is not able to tell the user what is the specific price of one day and how when will the price drop down to the lowest point. Similar to Etzioni et al. work, Bingchuan and Yudong use a Bayes classification method to tell the probability of price change in hours or days [21]. They selected the route between Shanghai and Tokyo and only focused on the flights which departure time is around 9-10 a.m. During a whole year data collection (from July 2015 to June 2016), they had over two million 8 records, which have query days before the departure date is within 4 to 119 days. By using the probability, they build a decision system to smartly provide user buy-or-wait decisions. Faker and Bejugum [20] provided an approach to design a decision-support system. It used the information of one specific airline, including general features and the percentage of discounts which are provided by users to calculate the probability of different situations. Then use the probability value to make the decision: it is the perfect timing to buy or not. Their model utilizes the interactive nature of the online environment providing pieces of advice to users and let users make their final decision. In 2011, Groves and Gini proposed a regression model that is using the history diagram to predict the perfect timing for purchasing airline tickets [19]. They collected their data from Feb. 22 2011 until Jun. 23, 2011, and over 140 thousand records in total. There are two steps in their model. At first, they used a regression model to make predictions on the daily price. Secondly, after having a reliable threshold, they developed a

reliable rule, which is if the price is lower than the value which is prediction price minus the threshold, travelers should buy the ticket. Otherwise, travelers should wait. Their results showed when the purchase date is over two months away from the departure date, their model can effectively lower the average cost. Their model also enables travelers to input their preference such like how many stops that travelers can accept. Wohlfarth et al. in the same year proposed a preprocess method naming MPP (Marked Point Process) [23]. It is focusing on predict the price will fall or drop at one specific point. They reduced the size of feature set and using a clustered method and a tree model to make predictions. Their data was collected from 9 flight tickets providers focusing on six roundtrips. To cover the most common stay length, they chose 3, 7, or 14 days as the staying time. 9 However, all these work above were trying to give suggestions to users on the trend of airline route prices. They did not provide any information about the whole period's price of the airline, which will leave a small space for users to make their choice. In 2015, Yuwen et al. proposed an ensemble method, which is basing on learn++ [19]. They were focusing on five one-trip routes in China and chose KDD (K Nearest Neighbors) and PA (Passive-Aggressive) to finish their comparison task. Their strategy is that different routes should use different parameters to get the best result. Although for route CAN-SEL, they could get 2.85% error rate, the error rate of route BJS-HKG can even reach 17.87%.

Description of Data Set

The given Data Set consists of following columns & properties :-

- Size of test set: 2671 records
- FEATURES: Airline: The name of the airline.
- Date_of_Journey: The date of the journey
- Source: The source from which the service begins.
- Destination: The destination where the service ends.
- Route: The route taken by the flight to reach the destination.
- Dep_Time: The time when the journey starts from the source.

- Arrival_Time: Time of arrival at the destination.
- Duration: Total duration of the flight.
- Total_Stops: Total stops between the source and destination.
- Additional_Info: Additional information about the flight
- Price: The price of the ticket

Predict The Flight Fare Based On User Ticket Details.

Sowing Data Set demo:

```
Data columns (total 11 columns):
 #    Column           Non-Null Count  Dtype
---   ------           --------------  -----
 0    Airline          10683 non-null  object
 1    Date_of_Journey  10683 non-null  object
 2    Source           10683 non-null  object
 3    Destination      10683 non-null  object
 4    Route            10682 non-null  object
 5    Dep_Time         10683 non-null  object
 6    Arrival_Time     10683 non-null  object
 7    Duration         10683 non-null  object
 8    Total_Stops      10682 non-null  object
 9    Additional_Info  10683 non-null  object
 10   Price            10683 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

In [3]:

```
# Load The Dataset
df =pd.read_excel("../input/flight-fare-prediction-mh/Data_Train.xlsx")
df.head()
```

Out[3]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duratio
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h

EDA (Exploratory Data Analysis)

Exploratory Data Analysis, or EDA, is an important step in any Data Analysis or Data Science project. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset.

EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better. In this article, we will understand EDA with the help of an example dataset. We will use **Python** language (**Pandas** library) for this purpose.

Missing Values : -

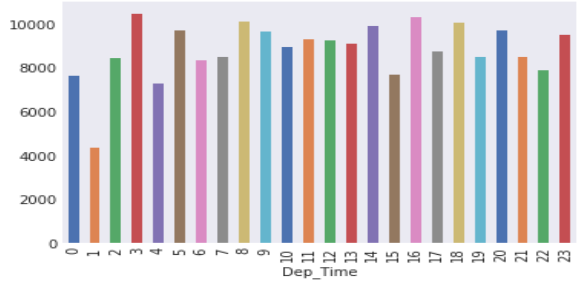
```
In [10]:
## Missing Values
df.isna().sum()
```

```
Out[10]:
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              1
Dep_Time           0
Arrival_Time       0
Duration           0
Total_Stops        1
Additional_Info    0
Price              0
dtype: int64
```

Plotting Graph

```
In [20]:
## The Average Price of Flights Based on Their Departure Time
df['Dep_Time'] = pd.to_datetime(df['Dep_Time'], format='%H:%M')
df.groupby(df['Dep_Time'].dt.hour)['Price'].mean().plot.bar(color=deep)
```

```
Out[20]:
<AxesSubplot:xlabel='Dep_Time'>
```



What is Feature Engineering?

The feature engineering pipeline is the preprocessing steps that transform raw data into features that can be used in machine learning algorithms, such as predictive models. Predictive models consist of an outcome variable and predictor variables, and it is during the feature engineering process that the most useful predictor variables are created and selected for the predictive model. Automated feature engineering has been available in some machine learning software since 2016. Feature engineering in ML consists of four main steps: Feature Creation, Transformations, Feature Extraction, and Feature Selection. Feature engineering consists of creation, transformation, extraction, and selection of features, also known as variables, that are most conducive to creating an accurate ML algorithm. These processes entail:

Feature Creation: Creating features involves identifying the variables that will be most useful in the predictive model. This is a subjective process that requires human intervention and creativity. Existing features are mixed via addition, subtraction, multiplication, and ratio to create new derived features that have greater predictive power.

Transformations: Transformation involves manipulating the predictor variables to improve model performance; e.g. ensuring the model is flexible in the variety of data it can ingest; ensuring variables are on the same scale, making the model easier to understand; improving accuracy; and avoiding computational errors by ensuring all features are within an acceptable range for the model.

Feature Extraction: Feature extraction is the automatic creation of new variables by extracting them from raw data. The purpose of this step is to automatically reduce the volume of data into a more manageable set for modeling. Some feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis.

Feature Selection: Feature selection algorithms essentially analyze, judge, and rank various features to determine which features are irrelevant and should be removed, which features are redundant and should be removed, and which features are most useful for the model and should be prioritized.

```
In [21]: df.Airline.value_counts()
```

```
Out[21]:
Jet Airways                          3849
IndiGo                               2053
Air India                            1752
Multiple carriers                    1196
SpiceJet                              818
Vistara                               479
Air Asia                              319
GoAir                                 194
Multiple carriers Premium economy      13
Jet Airways Business                    6
Vistara Premium economy                 3
Trujet                                  1
Name: Airline, dtype: int64
```

Approach

Load and cleans the dataset, perform thorough EDA on it to apply effective feature engineering.

Once the data is filtered. Split the data and train it. There comes Model Training.

Initially, basic models are implemented with default parameters and compared with each other. Dataset is then fitted with KNeighbours Regressor, Decision Tree Regressor. The outcome is not upto the mark that means the model is not nicely fitting. To improve it further lets proceed towards ensemble approach. In this, RandomForest Regressor and GradientBoosting Regressor is used and its fitted over the data.

Results implies that RandomForest gives better outcome for the given data.

To improve it further, tune this model for more optimized approach.

HyperParameter tuning improves the model and gives the best possible outcome.

Workflow of Project

K-Neighbors Regression Analysis in Python

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. **Algorithm** A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors.

82]:
```
## Fitting Each Model With Base Parameters One By One
fit_and_evaluate(KNeighborsRegressor())
```

```
###### MACHINE LEARNING MODEL : KNeighborsRegressor()
Training score: 0.7353783201025581
Predictions:
 [16315.   5158.2  8536.  ...  6471.8  7535.8 11467.6]


r2 score is: 0.5743709506218349
MAE:1879.4638277959757
MSE:9177437.535891436
RMSE:3029.428582404846
```

Decision Tree - Regression

Decision tree builds regression or classification models in the form of a tree structure.
It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
The final result is a tree with **decision nodes** and **leaf nodes**.
A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested.
Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called **root node**.
Decision trees can handle both categorical and numerical data.

In [83]:
```
fit_and_evaluate(DecisionTreeRegressor())
```

```
###### MACHINE LEARNING MODEL : DecisionTreeRegressor()
Training score: 0.9692484150527355
Predictions:
 [16840.  4959.  9187. ...  6152. 13339. 14335.]


r2 score is: 0.7263312105882094
MAE:1338.4948136016224
MSE:5900861.8514622
RMSE:2429.168963135788
```

Ensemble Learning:

As we know, Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote. Bagging and Boosting are two types of Ensemble Learning. These two decrease the variance of a single estimate as they combine several estimates from different models. So the result may be a model with higher stability. Let's understand these two terms in a glimpse.

Bagging: It is a homogeneous weak learners' model that learns from each other independently in parallel and combines them for determining the model average.

Applying bagging method on our dataset.

Boosting: It is also a homogeneous weak learners' model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

Random Forest Regression

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

```
In [84]: fit_and_evaluate(RandomForestRegressor())

         ###### MACHINE LEARNING MODEL : RandomForestRegressor()
         Training score: 0.952911072002993
         Predictions:
          [16836.85         5650.16         8901.08        ...   6711.54
          13202.22166667 13061.00716667]


         r2 score is: 0.7950213456694
         MAE:1177.4161062921378
         MSE:4419761.289927288
         RMSE:2102.322831994955
```

Gradient Boosting Regressor

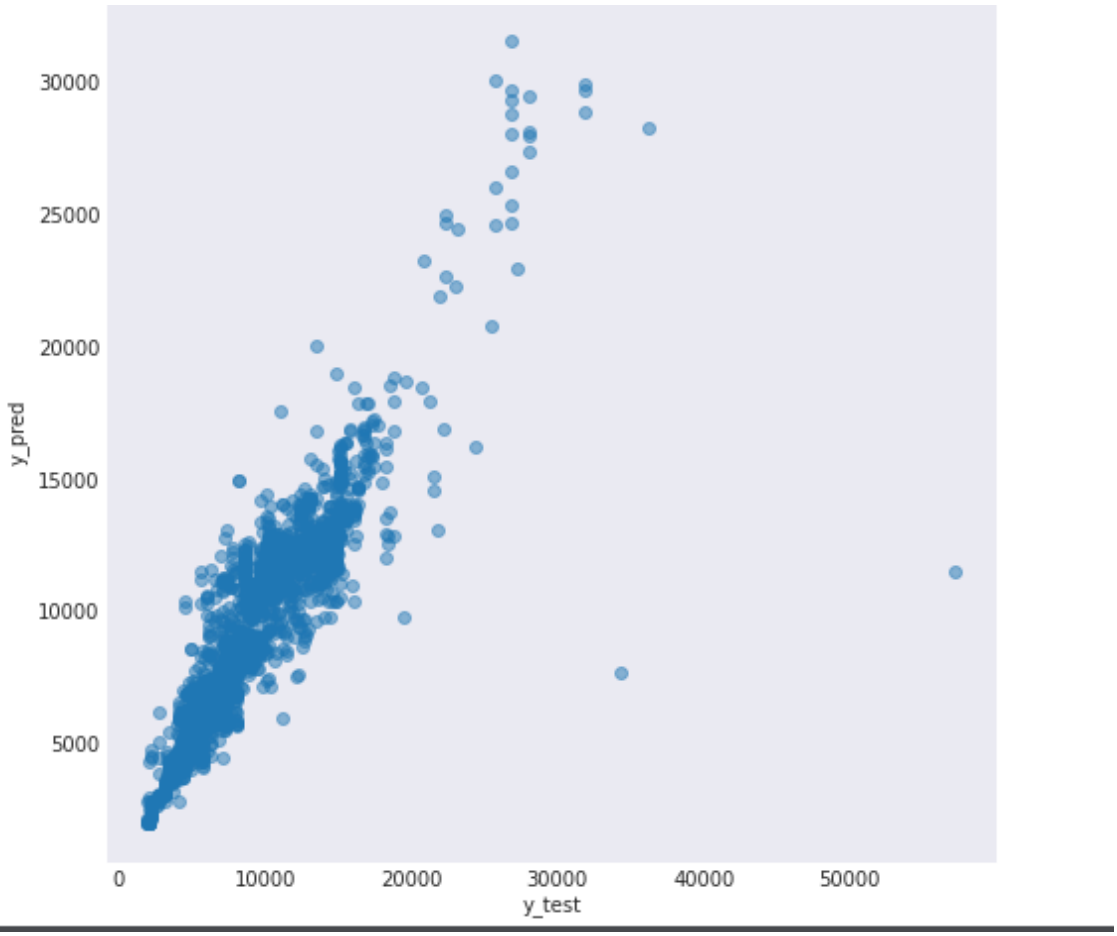
Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.

There is a technique called the **Gradient Boosted Trees** whose base learner is CART (Classification and Regression Trees).

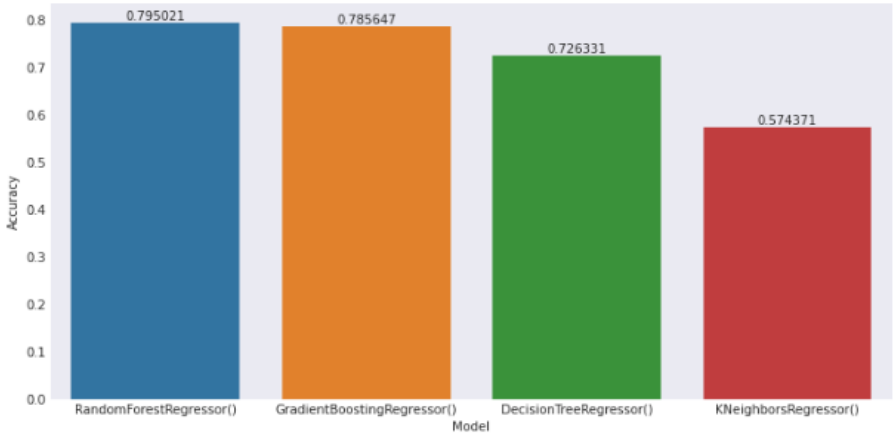
```
In [85]: fit_and_evaluate(GradientBoostingRegressor())

###### MACHINE LEARNING MODEL : GradientBoostingRegressor()
Training score: 0.7825102431771656
Predictions:
 [15115.3030793   7631.92313112  9311.56615764 ...  7226.34578254
 10993.44358889 12496.0873803 ]


r2 score is: 0.7856472504714556
MAE:1527.7244389206612
MSE:4621886.058573419
RMSE:2149.8572181829704
```



```
e),x='Model',y='Accuracy')
ax.bar_label(ax.containers[0]);
```



Conclusion

In our research, we proposed a slope-based dataset spitting method and a regression-based weighted average algorithm, RWA, for flight price prediction. Flight price prediction can be regarded as a time series problem. In the old study, all the studies always treat the data as a whole part. Comparing with their method, we split the data into two parts according to the trend of the price curve and make a hypothesis that the different datasets can have different characters. To get the best performance of each dataset, we should have different feature sets on different datasets. Then we use a two-phase experiment to get the best feature set for each dataset. In phase one, we use a brute force algorithm to search the feature combination in two redundant feature sets, which has the best performance on our datasets. Moreover, in phase two, we test the importance of different features for each dataset and remove one more feature, which has the worst influence on the prediction result to get the best performance. The result shows that the best feature sets for each dataset are different from each other.

RandomForest Regressor bagging ensemble approach gave a nice accurate result, further with hyperparameter tuning to it, it is the best possible approach for the given data.

Bibliography

[1] Mahapatra, D. M., & Patra, S. K. (2019). A New Destination of Online Travel Business: A Case Study. SEDME (Small Enterprises Development, Management & Extension Journal), 46(2), 130-137.

[2] Malighetti, P., Paleari, S., & Redondi, R. (2009). Pricing strategies of low-cost airlines: The Ryanair case study. Journal of Air Transport Management, 15(4), 195-203.

[3] Malighetti, P., Paleari, S., & Redondi, R. (2010). Has Ryanair's pricing strategy changed over time? An empirical analysis of its 2006–2007 flights. Tourism management, 31(1), 36-44.

[4] Smith, B. C., Leimkuhler, J. F., & Darrow, R. M. (1992). Yield management at American airlines. interfaces, 22(1), 8-31.

[5] Daudel, S., Vialle, G., & Humphreys, B. K. (1994). Yield Management: Applications to Air Transport and Other Service Industries; this New English Version is Published with Additional Material and Updated Statistics. Presses de l'Institut du Transport aérien.

[6] McGill, J. I., & Van Ryzin, G. J. (1999). Revenue management: Research overview and prospects. Transportation science, 33(2), 233-256.

[7] Pritscher, L., & Feyen, H. (2001). Data mining and strategic marketing in the airline industry. Data Mining for Marketing Applications, 39.

[8] Avineri, E., & Ben-Elia, E. (2015). Prospect theory and its applications to the modelling of travel choice. Bounded Rational Choice behavior: Applications in Transport, 233.

[9] Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

[10] Russell, S., & Norvig, P. (2002). Artificial intelligence: a modern approach.

[11] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. Adaptive computation and machine learning. MIT Press, 31, 32.

[12] Alpaydin, E. (2020). Introduction to machine learning. MIT press

- www.stat.yale.edu
- www.simplilearn.com
- www.geeksforgeeks.org
- www.analyticsvidhya.com