FLIGHT PRICE PREDICTION

Submitted by:

PRAKHAR PRAKASH

Email: [mr.prakhar@gmail.com](mailto:mr.prakhar@gmail.com)

Github: prakhar3146

**ACKNOWLEDGMENT**

**In this project,I have used the study material,vlogs and sample projects provided(as a part of my data science with nlp course) by the DataTrained institute(based in Noida) for references.Apart from that, certain articles and definations were obtained from** www.wikipedia.org, www.towardsdatascience.com, www.cnbc.com, www.machinelearningmastery.com **,www.scikit-learn.org, www.scipy.org.**

INDEXING

|  |  |
| --- | --- |
| INDEX | CONTENT |
| 1. | Name of the project |
| 2. | Acknowledgment |
| 4. | Introduction |
| 7. | Analytical problem framing |
| 14. | Tools used |
| 15. | Model Development and evaluation |
| 21. | Conclusion |

**INTRODUCTION**

* Business Problem Framing

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on –

1. Time of purchase patterns (making sure last-minute purchases are expensive)

2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases).

In this project, I have web-scraped data for flight fares and relevant information,between various airports and made a model which can predict the flight fares from the relevant information.

* DOMAIN RELATED PROBLEMS

**While working on this project I came across a few domain related problems in which domain expertise would have made things a lot easier and would have led to an improvement in the model’s performance.**

**Some of these problems are discussed below:**

**1)Special Occasions**

**The occasions on/near which the flight prices are higher/lower than usual.**

**The discounts/offers provided(if any) on travelling during these occasions.**

**Other information related to such occasions which may impact the flight fares.**

**This is important because it will increase the significance of the ‘Date of journey’ and the ‘Days\_to\_journey’ feature in this project which will change the dynamics of the prediction.**

**2)Fuel Prices**

**The impact of fuel prices on the flight fares.A domain expert might be able to help determining the bars(range) of fuel price variations above/below which the flight/fares may increase/decrease as it ticket prices do not vary for small fluctuations which happen mostly on a daily basis.Therefore I am assuming that there must be some criteria/bars to determine the same.**

**3) Other relevant information**

**Other relevant information which may vary from case to case, like a pandemic such as the covid-19 forced the flights to operate on 50% capacity which impacted the ticket prices or a strike by the employees of a particular airline which may impact have a direct/indirect impact on the flight fares.**

**Other informations like weather,model of the aeroplane,seating capacity etc. might improve the efficiency of the model in predicting the flight fares.**

* Review of Literature

According to an article published on the cnbc:

PUBLISHED FRI, AUG 3 201811:59 AM EDTUPDATED FRI, AUG 3 201812:07 PM EDT

by

[Tom Chitty](https://www.cnbc.com/tom-chitty/)[@MRTOMCHITTY](https://twitter.com/mrtomchitty)

**Today, airlines price tickets “as much as the customer and market will bear,” according to consultant and former airline planning executive Robert W. Mann.**

**Airlines also profile their customers to help them adjust prices.**

**This often means placing passengers into one of two groups: leisure or business. And the way each group is priced is very different.**

**Leisure passengers usually book months in advance, so airlines tend to start the prices for these seats relatively high. It then adjusts the prices according to market response.**

**For typical business routes, airlines will start with low prices to fill a minimum capacity, then increase prices steeply as corporate passengers tend to book last minute.**

**What ticket prices aren’t actually focused on is a ticket’s combined cost, including taxes and fuel. According to Mann, it’s technology that determines the price you see online instead.**

**“Technology has allowed some airlines to create a ‘basic economy fare’ with limited amenities to compete with minimal service, low cost carriers,” he said.**

**That’s because the lower fare allows full-service carriers to appear on the first page of search engines such as Google Flights.**

**But it’s not just airlines that are using artificial intelligence (AI) technology to their advantage.**

**Consumers now have access to sites that monitor fares, using their own algorithms and past data to predict the lowest price a seat will reach, then alert their customers.**

**The threat airfare search websites pose to airlines’ dynamic pricing system even saw**[**United Airlines suing the website Skiplagged**](https://money.cnn.com/2015/12/31/investing/aktarer-zaman-how-i-beat-united-airlines/index.html)**, which helps passengers find loopholes for cheaper tickets.**

* MOTIVATION

**I had received this project as an assignment during my internship at the Flip-robotechnologies,which is a banglore based software company.**

**Company address: Floor, Suite No.1759, #39, 2nd, NGEF Ln, Indiranagar, Bengaluru, Karnataka 560038.**

**Apart from that, I have a knack for taking up different problems as I believe,contrary to the popular opinion ,”experience is not acquired by age but by the number of problems one has dealt with”.**

**Also, this was fun.**

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling

**A number of tools/libraries were used for the mathematical/analytical modelling and visualization of the dataset.**

**The libraries used were:**

**NUMPY**

**NumPy is a Python library used for working with arrays.**

**It also has functions for working in domain of linear algebra, fourier transform, and matrices.**

**NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.**

**NumPy stands for Numerical Python.**

**PANDAS**

**pandas is a**[**software library**](https://en.wikipedia.org/wiki/Software_library)**written for the**[**Python programming language**](https://en.wikipedia.org/wiki/Python_(programming_language))**for data manipulation and**[**analysis**](https://en.wikipedia.org/wiki/Data_analysis)**. In particular, it offers**[**data structures**](https://en.wikipedia.org/wiki/Data_structure)**and operations for manipulating numerical tables and**[**time series**](https://en.wikipedia.org/wiki/Time_series)**. It is**[**free software**](https://en.wikipedia.org/wiki/Free_software)**released under the**[**three-clause BSD license**](https://en.wikipedia.org/wiki/3-clause_BSD_license)**.**

**Use in this project:**

**Most of the work in this project was done with the help of the Pandas library as it has a number of library features that made life easier while doing the analytical/mathematical modelling.**

## **Library features[**[**edit**](https://en.wikipedia.org/w/index.php?title=Pandas_(software)&action=edit&section=1)**]**

* **DataFrame**[**object**](https://en.wikipedia.org/wiki/Object-oriented_programming)**for data manipulation with integrated indexing.**
* **Tools for reading and writing data between in-memory**[**data structures**](https://en.wikipedia.org/wiki/Data_structure)**and different**[**file formats**](https://en.wikipedia.org/wiki/File_format)**.**
* **Data alignment and integrated handling of missing data.**
* **Reshaping and pivoting of data sets.**
* **Label-based slicing, fancy indexing, and subsetting of large data sets.**
* **Data structure column insertion and deletion.**
* **Group by engine allowing split-apply-combine operations on data sets.**
* **Data set merging and joining.**
* **Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.**
* **Time series-functionality: Date range generation**[**[6]**](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-6)**and frequency conversions, moving window**[**statistics**](https://en.wikipedia.org/wiki/Statistics)**, moving window**[**linear regressions**](https://en.wikipedia.org/wiki/Linear_regression)**, date shifting and lagging.**
* **Provides data filtration.**

**Time**

**The time library was used to put the programs to sleep while giving the url time to open.**

* Data Sources and their formats

**The data was extracted from the website-www.kayak.co.in by web-scraping using selenium.**

**They were then organised according to their respective categories in a pandas DataFrame and stored as a csv aswell as an excel file to be used for performing the complete life cycle of data science.**

**The web-scraping worksheet(ipynb) and the dataset is present in a excel format in “Flight\_price\_prediction” repository on my github account.**

* Data Preprocessing Done

**The cleaning of the data was a time taking challenge as the data was mixed up with irrelevant information in some features or had to be divided into separate columns where two relavant features were mixed up.**

**The features containing dates(“date\_recorded”,”Date\_of\_journey”) was converted into datetime format and the days and months had to be extracted into separate features.**

**The departure and arrival times were present in 00:00 format as an object datatype.**

**The “arrival\_time” feature had extra data present which contained the number of days taken to reach destination.**

**The ”stops” feature also had the number of stops made during flight along with their names mixed up in it.The information was present in the format : 1 stop,HYD, direct and so on .**

**Hence,the information was split into two new columns namely “stops” and “route”.**

**Also,the target feature “Price” had extra data like:₹ 9,280.**

**Here the comma and the rupee symbols had to be removed from the feature before converting it into integer format.**

* Data Inputs- Logic- Output Relationships

**The relationships of the independent features with the output feature(Price) were visualized with the help of a catplot and a barplot using pandas.groupby() method.**

**There were interesting patterns/trends obsereved.**

**The relationships have been stated in the table below:**

|  |  |
| --- | --- |
| **Feature Names** | **Observations** |
| **Airline\_name vs Price** | **1)'Etihad Airways' has the highest mean Price followed by 'Aeromexico' in this sample.**  **2)'AirAsia India' has the lowest mean Price.**  **3)The mostly domestic airlines like 'Vistara','Indigo',''GoFirst','Frontier','AirAsia India' have the lowest mean Prices in this sample.**  **4)The Arab('Etihad,Qatar Airways) and the western Airlines(Aeromexico,AirCanada,British Airways) have the highest mean SalePrices in this sample.**  **5)The highest recorded Price(Rs 6,60,190) was from MuliAirlines journey jointly by Emirates and AirCanada flying from Cape Town Intl. to Pearson Intl.**  **6)The lowest recorded Price(Rs 660) was by Indigo flights travelling from Ahmedabad to Chhatrapati Shivaji Intl(Mumbai) airport.** |
| **Website Vs Price** | **1)The websites: Agoda.com,CheapTicket.in,BudgetTicket,Spirit Airlines,akbartravels.com,happyfares,Travomint have the lowest mean prices amongst all the websites.**  **2)The AirCanada,Aeromexico and the Qatar Airways have the highest mean Price.**  **3)The highest price in the sample was from 'Bravofly' and the lowest from 'happyfares'.** |
| **Day\_of\_journey Vs Price** | **Flights leaving on Friday had a higher mean Price than the ones leaving on Wednesday or Sunday.Although they can be probably due to the fact that the ones leaving on friday were closer to the booking date.** |
| **Month\_of\_journey VS Price** | **The flights departing in january have a higher mean of prices than those departing in Feb.** |
| **Dep\_hour VS Price** | **1)The early morning flights(departing at:3,4,5am.) have the highest mean of prices with the ones departing at 3am being the costliest.**  **2)The morning flights(departing at: 6,7 and 8am) have the lowest mean of prices with the ones departing at 6am. Being the cheapest.**  **3)The highest ticket fare in this sample however,was of a flight departing at 6pm.** |
| **Days\_since\_departure VS Price** | **1)The flights taking 2 days to reach destination have the highest mean of ticket prices.**  **2)The flights arriving to their destinations on the same day they depart,have the lowest mean of ticket prices.**  **3)The costliest ticket price in this sample was of a flight taking 1 day to reach its destination airport.** |
| **Days\_to\_journey VS Price** | **1)Contrary to the popular belief, the flights booked 30 days prior to the journey had the highest average of ticket prices.**  **2)A downward trend is present with the increasing number of days if we leave the '30' out.**  **3)The highest recorded ticket price in this sample was also booked 30 days prior to the flight.** |
| **Origin\_city VS Price** | **1)The flights originating from 'Cape Town Intl' had the costliest airfares(averaged.) and indivisual.**  **2)The flights originating from 'Ahmedabad' had the cheapest(average.) aswell as indivisual ticket prices.** |
| **Destination city VS Price** | **1)The flights flying to 'Cape Town Intl' has the highest(average) Price.**  **2)The flights flying to 'ahmedabad' has the lowest(average) ticket price.**  **3)The costiliest ticket in this sample is of a flight flying to 'Pearson Intl' airoprt(Toronto).**  **4)The cheapest ticket in this sample is of 3 flights flying to Chatrapati shivaji Intl.(Mumbai).** |
| **Stops VS Price** | **1)The flights having to 2stops in their routes have the highest ticket prices(both average and extreme) closely followed by the ones having 3stops.**  **2)The direct flights have the cheapest airfares.** |
| **Arrival\_hour VS Price** | **1)The flights arriving at 4 and 5am. have the highest ticket prices(mean).**  **2)The flights arriving at 9am and 11pm. have the lowest ticket prices(mean).**  **3)The costliest flight in this sample has an arrival time of 7pm(IST).**  **4)The cheapest flights in this sample have an arrival time of 8am,8pm and 11pm respectively.** |
| **Duration hour VS Price** | **1)The flights having a travel time of 25 and 24 hours(respectively) have the highest mean of ticket prices.**  **2)The flights having a travel time of 1,2,14 hours have the lowest mean of ticket prices.**  **3)An upward trend of price is present from duration hours 1 to 10.**  **4)The costliest ticket in this sample is of a flight having a travel time of 32 hours.**  **5)The cheapest ticket in this sample is of 3 flights having a travel time of 1 hour(respectively).** |

* ASSUMPTIONS

**Due to the lack of domain expertise,some assumptions had to be made while proceeding with the project.**

**A few of them were:**

**1)The factors(information) not included in the prediction have negligible impact on the flight fares or they were constant over the period this sample was taken.(eg:Fuel prices,special occasions,government taxes etc.)**

**2)The impact of unique values of factors(features) present in this sample will be similar if new/unseen unique values are present in the test dataset.**

**For example: If a new source/destination airport is present which was not used in training our model or the number of stops/duration hours is increased/decreased in test datset.etc.**

TOOLS USED

Jupyter Notebook

**Jupyter Notebook (formerly IPython Notebooks) is a**[**web-based interactive**](https://en.wikipedia.org/wiki/Web_application)**computational environment for creating**[**notebook**](https://en.wikipedia.org/wiki/Notebook_interface)**documents.**

**A Jupyter Notebook document is a browser-based**[**REPL**](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop)**containing an ordered list of input/output cells which can contain code, text (using**[**Markdown**](https://en.wikipedia.org/wiki/Markdown)**), mathematics,**[**plots**](https://en.wikipedia.org/wiki/Plot_(graphics))**and**[**rich media**](https://en.wikipedia.org/wiki/Interactive_media)**. Underneath the interface, a notebook is a**[**JSON**](https://en.wikipedia.org/wiki/JSON)**document, following a versioned schema, usually ending with the ".ipynb" extension.**

**Almost all of the work was done on jupyter notebooks and uploaded as an ‘ipynb’ file on github.**

**ANACONDA**

**Anaconda is a**[**distribution**](https://en.wikipedia.org/wiki/Software_distribution)**of the**[**Python**](https://en.wikipedia.org/wiki/Python_(programming_language))**and**[**R**](https://en.wikipedia.org/wiki/R_(programming_language))[**programming languages**](https://en.wikipedia.org/wiki/Programming_language)**for**[**scientific computing**](https://en.wikipedia.org/wiki/Scientific_computing)**(**[**data science**](https://en.wikipedia.org/wiki/Data_science)**,**[**machine learning**](https://en.wikipedia.org/wiki/Machine_learning)**applications, large-scale**[**data processing**](https://en.wikipedia.org/wiki/Data_processing)**,**[**predictive analytics**](https://en.wikipedia.org/wiki/Predictive_analytics)**, etc.), that aims to simplify**[**package management**](https://en.wikipedia.org/wiki/Package_management)**and**[**deployment**](https://en.wikipedia.org/wiki/Deployment_environment)**. The distribution includes data-science packages suitable for**[**Windows**](https://en.wikipedia.org/wiki/Microsoft_Windows)**,**[**Linux**](https://en.wikipedia.org/wiki/Linux)**, and**[**macOS**](https://en.wikipedia.org/wiki/MacOS)**. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and**[**Travis Oliphant**](https://en.wikipedia.org/wiki/Travis_Oliphant)**in 2012.**[**[8]**](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-8)**As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.**

**The libraries used were:**

**1)Datetime**

**2)pickle**

**3)sklearn**

**4)pandas**

**5)seaborn**

**6)matplotlib**

**7)Scipy**

**8)string**

**9)numpy**

**Model/s Development and Evaluation**

**STATISTICAL/ANALYTICAL APPROACH**

**The dataset was checked for outliers using a a boxplot and a presence of outliers was detected in the 'Price' and 'Days since Departure' feature.**

**The rows containing outliers were removed from the dataset with the help of the zscore class from the scipy.stats library and the amount of data lost was 3.34% approximately.**

**The other problem faced was that of the high amount of skewness present in the target(Price) and ‘duration\_hours’ features.**

**This was resolved by using log transformation on both the features.**

**The impact of transformation on predicted values was reversed later by using np.exp() function to get the real time values.**

* ALGORITHMS

**The algorithms used for training and testing are:**

**1)Linear Regression**

**2)Random Forest Regressor**

**3)Adaboost Regressor**

**4)Gradient Boosting Regressor**

**5)KNeighbors Regressor**

**6)Support Vector Regressor**

* EVALUATION AND METRICS

Linear Regression

Training score: 0.9992049445348246

mean squared error: 5.407798139006106e+17

R.M.S.E: 735377327.5676988

mean absolute error: 98928697.15849565

R2\_score -4.639354447310444e+17

Cross\_val\_score

cross val score: [-3.66698743e+14 -3.88786215e+14 -5.42578063e+ 14 -1.47322966e+14

-1.60864422e+16 9.95924417e-01 9.97058931e-01 9.94445894e-01

-1.28396841e+12 -1.54552233e+14]

Random Forest Regressor

score : 0.9999964783614999

mean\_squared error: 3.520795742311681e-05

R.M.S.E: 0.005933629363477029

mean\_absolute\_error: 0.0008902485923359216

r2 score: 0.9999697950645248

Cross\_val\_score

Cross val score: [0.99999768 ,0.99999712, 0.9999984 , 0.98937706, 0.99998047, 0.9999947,0.99999621 ,0.9999915 ,0.99999677, 0.99995057]

Adaboost Regressor

score : 0.9950437093289188

mean\_squared error: 0.005642257161900708

R.M.S.E: 0.07511495964121066

mean\_absolute\_error: 0.05864323005720215

r2 score: 0.9951595029651488

Cross\_val\_score

cross val score: [0.99760007 ,0.99042323 ,0.99559433 .0.96533453 , 0.98887585 ,0.9898699,0.99081815, 0.99538926 ,0.99381902 , 0 .98671976]

Gradient Boosting Regressor

score : 0.9999532816357484

mean\_squared error: 7.40866546588398

R.M.S.E: 2.721886380046746

mean\_absolute\_error: 2.5015284700081013

r2 score: -5.35590016385138

Support Vector Regressor

score : 0.9949742653162437

mean\_squared error: 0.011694959711032522

R.M.S.E: 0.10814323701014558

mean\_absolute\_error: 0.07732743377701307

r2 score: 0.9899668844968265

KNeighbors Regressor

score : 0.9588649714428898

mean\_squared error: 0.12970449574319756

R.M.S.E: 0.3601451037334779

mean\_absolute\_error: 0.17973789290194653

r2 mmsore: 0.88872640699696

cross val score: [0.88851494 ,0.87536496 ,0.88355057 ,0.87569565 0.86440725, 0.81914134,0.82599164, 0.87886788 ,0.86481893 ,0.8960 008 ]

* METRICS

**The key metrics used in this project were :**

1. **Model.score() : Gives the training accuracy of the model.**
2. **Mean\_squared\_error() : Gives the average of the squared differences between the actual and the estimated values.**
3. **Mean\_absolute\_error() : In statistics, mean absolute error (MAE) is a measure of**[**errors**](https://en.wikipedia.org/wiki/Error_(statistics))**between paired observations expressing the same phenomenon.**
4. **Cross\_val\_score : The k-fold cross-validation procedure is a standard method for estimating the performance of a machine learning algorithm or configuration on a dataset.**

**R2\_score() : In**[**statistics**](https://en.wikipedia.org/wiki/Statistics)**, the coefficient of determination, denoted R2 or r2 and pronounced "R squared", is the proportion of the variation in the dependent variable that is predictable from the independent variable(s).**

**It is a**[**statistic**](https://en.wikipedia.org/wiki/Statistic)**used in the context of**[**statistical models**](https://en.wikipedia.org/wiki/Statistical_model)**whose main purpose is either the**[**prediction**](https://en.wikipedia.org/wiki/Prediction#Statistics)**of future outcomes or the testing of**[**hypotheses**](https://en.wikipedia.org/wiki/Hypotheses)**, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.(Wikipedia definition).**

**The r2\_score was the most important metric in choosing our best model followed by the training and cross\_val scores.**

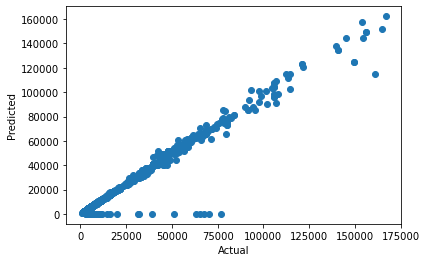
Visualizations

**The results(predicted vs actual) were visualized with the help of a scatterplot for all the algorithms used after reverse transforming the target feature(Price) by using np.exp() function from numpy library.**

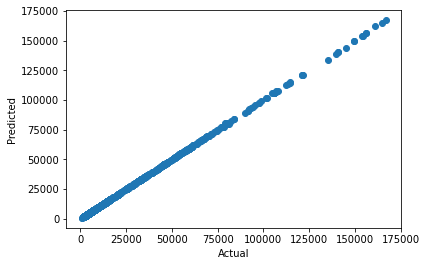
**The reverse transformation was done as the predicted feature was transformed by using np.log() method to remove skewness,which in turn made the algorithms predict the log of the actual values.**

**The snapshots of visualisations (actual VS predicted) for each algorithms are :**

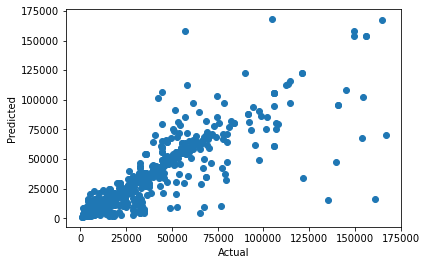
Linear Regression



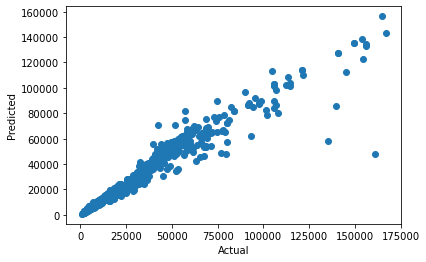
Random Forest Regressor



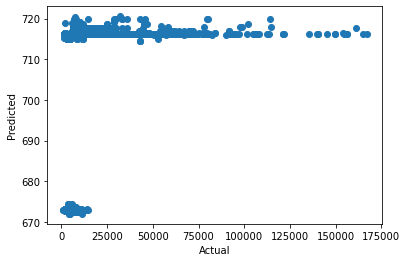
KNeighbors Regressor



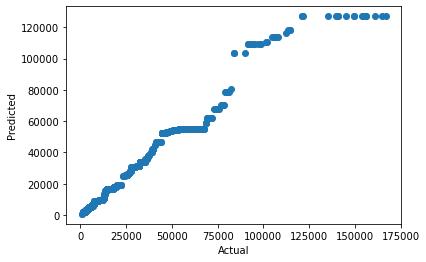
Support Vector Regressor



Gradient Boosting Regressor



Adaboost Regressor



Observation:

The Random Forest Regressor depicted the lowest fluctuations from the best fit line.

* INTERPRETATIONS

**The Results interpreted were:**

**1) The Random Forest Regressor had the best Training score.**

**2) The Random Forest Regressor had the best r2\_score.**

**3) Random Forest Regressor had the lowest difference between r2 and cross\_val\_score.**

**4) Random Forest Regressor had the best visualization plot.**

**5) The Random Forest Regressor also had a very low root of mean\_squared\_error and mean absolute error values,186.96 and 43.93 respectively.**

Therefore, Random Forest Regressor was choosen as the best model.

**CONCLUSION**

**The best model was hyperparameter tuned using RandomizedSearchCV and was saved using the pickle.dump() method from the pickle library.**

**Due to the log transformation of the target feature,the values predicted by the model are the log of the real time values.Hence,the values have to be reverse transformed by using the exponents before comparing them with the actual values or using them as the same.**

* Learning Outcomes of the Study in respect of Data Science

**While working on this project I found out that the ensemble techniques work better on datasets such as this one,where the data is non-linear and where a lot of continous features behave like categorical(nominal) ones.**

**Also,I learned that the pd.get\_dummies() has certain limitations like if the test dataset has an unseen value or is missing a value/feature that was present in the training dataset,it can lead to problems in predicting the test values.The pd.get\_dummies also is not an algorithm like OneHotEncoder from the sklearn library and hence it does not learn anything from training(past experiences) as it does not have a fit method.i.e it creates fresh dummies for each unique value every single time it is used.**

**Hence, using pandas.get\_dummies() method should be avoided to encode the categorical features.**

**The OneHotEncoder from the sklearn library can be quite useful in performing the same.**

* Limitations of this work and Scope for Future Work

**There were however,limitations to this work,improving on which may enhance the performance and the scope of the model.**

**Some of them are discussed below:**

1) The data scraped was for 35 airports which were used both as the origin aswell as the destination.Hence,the model was trained on these specific origins/destinations and may/may not perform so well if those parameters are changed.

2) The journey dates were 3,13,29 and 30 days post the date of record.The prices may vary differently for different months.

3) External factors such as Fuel prices,special occasions,government taxes, public holidays,climate,covid restrictions etc. were not accounted for.These factors are likely to impact the ticket prices.

4) Lack of domain expertise.The availability of which might provide an insight into various unseen trends/criteria of determining the price.

5) The dataset was confined to January-feburary 2022.This may restrict the model from identifying the change of prices with years.

**The remedies to these limitations can be:**

1)Availablity of domain expertise.

2)Scraping of a larger dataset containing flight data from as many airports as possible.

3)Availablity of a larger dataset containing flight data over a reasonably long period(as per the domain expert).

4)Accounting for the external factors such as Fuel prices,special occasions,government taxes, public holidays,climate,covid restrictions etc.

\*\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*\*