FLIGHT PRICE PREDICTION APPLICATION

Mohit Pal Singh

Student, School of Computer Science and Engineering

Lovely Professional University

Phagwara, India

mohitpalsingh239@gmail.com

Abstract

Machine learning is a widely accepted technology as the solution for the future's high computational needs. We are already seeing its implementations around us from suggesting our next meal to anomaly detection in various crucial systems such as banking etc. One such application of Machine Learning can be Price Prediction. This paper proposes a system to predict the price of upcoming flight months before. It uses Machine Learning model which can be trained once a year to increase it's efficiency and then the whole application can be run offline and on a simple website. It has been tested from the data of 2017 and 2018 Flight Data from several Flight companies operating in India. The promising results from the rigorous testing obtained in terms of Model accuracy justify the proposed Application.

keywords-Machine Learning, Prediction, Flight Price, Accurate Model

1 INTRODUCTION

Machine Learning has emerged as the most trending technology in recent times and we have seen research organisations proposing solutions to age old problem which required huge calculations but now can be done in milliseconds, thanks to evolving machine learning models. Machine Learning is the concept where a system does something for which it is not **explicitly** trained. For example- A well trained machine learning model to identify all dogs and cats doesn't need all the photos of dogs and cats from the whole universe to work. The model can be trained to satisfactory **accuracy** with just few hundred photos of dogs and cats and it will be able identify from any picture that you show it. It is just the tip of the ice-berg. Machine Learning is divided into three categories- **Supervised learning, Unsupervised learning and Reinforcement learning**. Supervised learning is something in which we **specify** the features and resultant feature. Unsupervised learning can be determined as the training where there is no classification and the model is supposed to work on the basis of

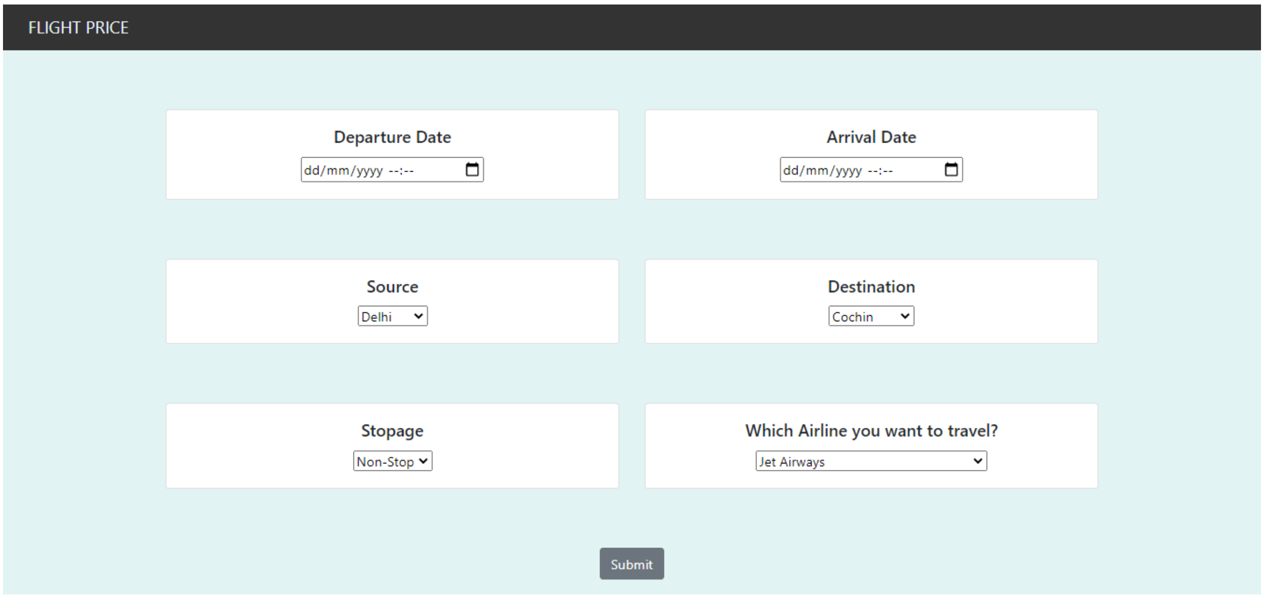
mimicry or visible patterns. In Reinforcement Learning, the model is **rewarded** with every right decision it takes and **punished** for every wrong one and that's how it will reach the global optimal. One such evolving application of Machine Learning is **Price Prediction**. In this, the Machine Learning model is required to predict the price of any commodity or service based on the described properties or features. The model is given past data with commodities and services with similar properties or features along with their prices. After training. The model will be able to predict the price of a new commodity or service which it has never seen before with **high accuracy**. This paper proposes a new system which can predict the cost of a **flight ticket cost** based on the type of ticket. Rest of the paper is organised as follows: Section II comprises of the related work followed by the details of the application in Section III and IV. Analysis is done in Section V. The paper is concluded in Section VI.

2 RELATED WORKS

There are several ticket-booking companies which provide similar working and features such as BOOKING.COM, TRIVAGO.COM, MAKEMYTRIP.COM, etc. All have slightly different workings when observed closely. MAKEMYTRIP and BOOKING utilises the **rush** of users for a particular flight and based on that it predicts the cost of the ticket like how many users have shown interest in this **particular destination** for that particular week plus how many view this particular flight is getting maybe for the same source and destination or for any **intersecting journey**. From data like this, these websites predicts the price of the ticket for the particular flight on that particular date. On the other hand, TRIVAGO is a vacation booking website and it works on **parameters** such as which is vacation destinations are **trending** as of now and will take into account the rush that this particular destination is gonna get the time you are trying to find flight. Apart from this, just like the previous two websites, it also works on viewership and **traffic monitoring** through various other sources like GOOGLE ANALYTICS and other social media networks. Now these systems require constant data hogging and data feeding as these are all **real-time systems**. They require powerful systems hosted on **powerful servers** with enough throughput to serve 100 thousand users at the same time whilst processing new data. Because of such heavy work, the **capital** required to build and run these systems is also huge and hence these companies prefer to keep their source code **private** and either charge monthly **fees** to use them or show ads in great number to **generate revenue**. Through various surveys done by different organisations, it is found that users are **not satisfied** with the user experience that they get while using these services for many reasons. The system proposed by this paper is different from currently present systems such as mentioned above,it is inspired from the output given by several surveys and is discussed in the next section.

3 PROPOSED SYSTEM

Unlike the solutions that are available right now in the market. This system **doesn't** process real-time data and doesn't require **any server** to work. It can be directly used by users on their devices completely **offline** and off the servers. This is possible by pre-trained model which can be included in the application package either **hosted** on a website or **bundled** and available to download through mobile apps marketplace. The model can be trained from the yearly **data collected** by the developer and then training the model again with this new data and then by pushing an **update** over the air thus ensuring accuracy for the **next year's predictions**. In its current form, the included model is trained of the data from a KAGGLE Challenge and trained with 70-30% system and has shown **95% accuracy**. After packing it through **pickle**, it is being used by a **Flask Application**. This Web-App has several input fields i.e. DEPARTURE DATE and TIME, ARRIVAL DATE and TIME, SOURCE, DESTINATION, NO. OF STOPS and preferred AIRLINE. Based on these inputs the web application returns a **predicted price** when clicked by the user.



Departure Date: dd/mm/yyyy --:--

Arrival Date: dd/mm/yyyy --:--

Source: Delhi

Destination: Cochin

Stopage: Non-Stop

Which Airline you want to travel?: Jet Airways

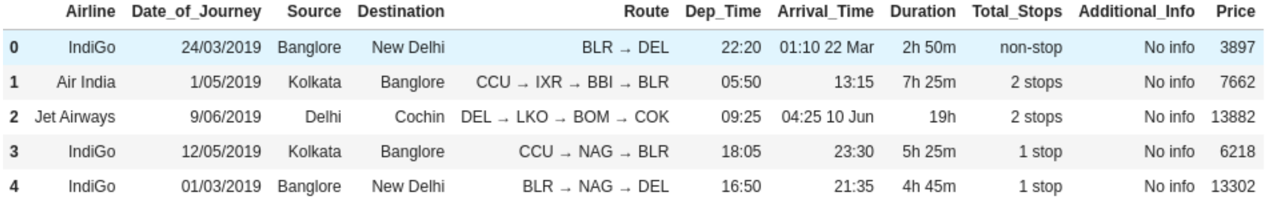
The DEPARTURE DATE field will accept any calendar date and time.

Just like DEPARTURE DATE, ARRIVAL DATE field will accept any date and time from calendar but it should be **ahead** from the set departure entries. SOURCE field will accept any city from BANGALORE, DELHI, COCHIN, KOLKATA, MUMBAI, CHENNAI and HYDERABAD. The DESTINATION field will also accept the above cities but it **should not** be the same from the source field.

In the STOPPAGE field, user can fill from ZERO stop, ONE stop and TWO stops. And finally in the AIRLINE field, user can enter among JET AIRWAYS, INDIGO, AIR INDIA, SPICEJET, VISTARA, AIR ASIA and GOAIR.

4 ALGORITHMS and TECHNIQUES

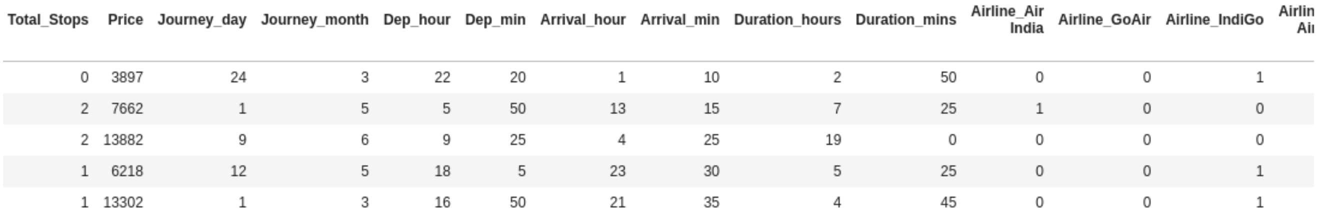
Preprocessing data -



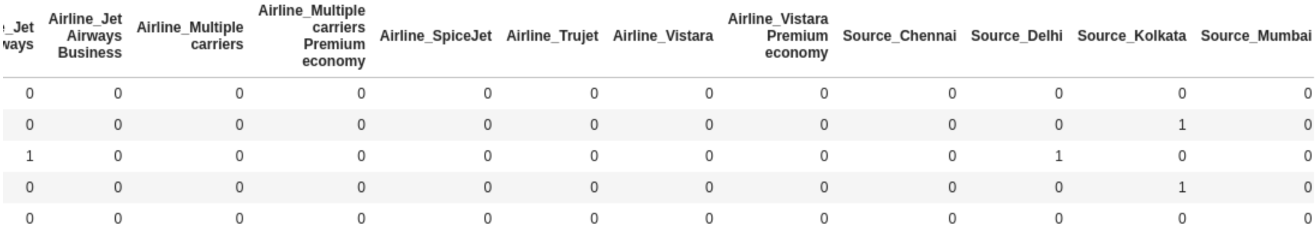
	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

This the raw data and we first need to preprocess it as the follows- First the **NaN** values are dropped then the **Date_of_journey** are converted into separate columns as **Journey_day** and **Journey_month**. Following this, **Dep_Time** and **Arrival_Time** are converted into separate columns as **Dep_hour**, **Dep_min**, **Arrival_hour** and **Arrival_min**. **Original** columns are dropped now.

Now we need to handle **categorical data** in our dataset. The data which is not in any order is called **Nominal Data** and **OneHotEncoder** is required in this case on the other hand, the data which is in order is called **Ordinal Data** and **LabelEncoder** is required in this case. Next is **Airline** column, It is **nominal** in nature, so the **OneHotEncoding** is done on it creating new columns for all possible entries of this column and filled with 0s and 1 to mark the correct airline for every entry. **Source** and **Destination** are both **nominal** in Nature and hence **OneHotEncoding** is done on both of them as well. The original columns are being dropped as we are moving forward with preprocessing them with different techniques. Our model can predict just with the **No. Of Stops** and doesn't necessarily need the **Route** column hence we can drop it. Now we need to change **No stops**, **1 stop**, **2 stops** ,**3 stops** to **0,1,2,3** respectively with the help of replace function. Now our Data set is good to go and before running it we need to preprocess the test set in the same way. The data looks like this now-

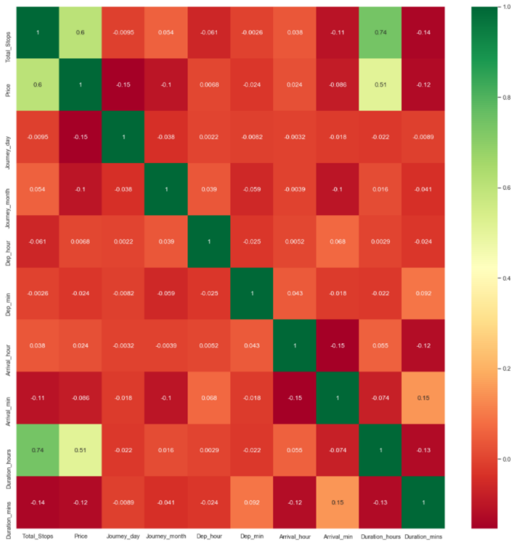


Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Duration_hours	Duration_mins	Airline_Air India	Airline_GoAir	Airline_IndiGo	Airlin Ai
0	3897	24	3	22	20	1	10	2	50	0	0	1	
2	7662	1	5	5	50	13	15	7	25	1	0	0	
2	13882	9	6	9	25	4	25	19	0	0	0	0	
1	6218	12	5	18	5	23	30	5	25	0	0	1	
1	13302	1	3	16	50	21	35	4	45	0	0	1	

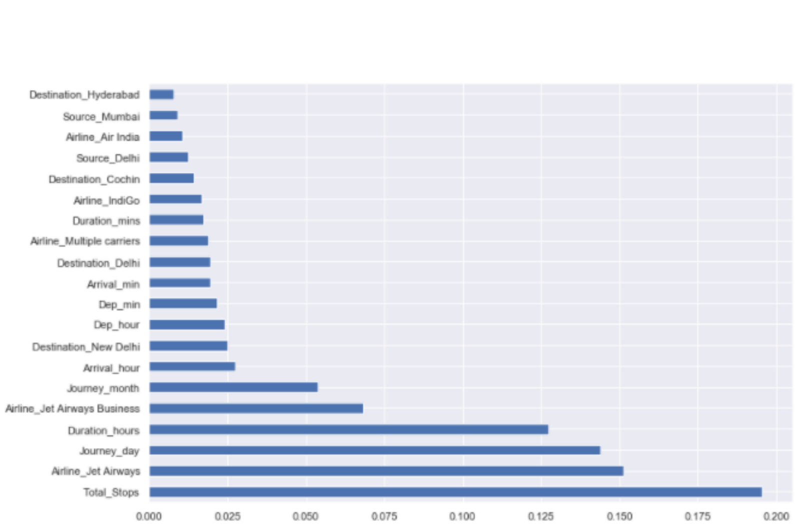


_Jet ways	Airline_Jet Airways Business	Airline_Multiple carriers	Airline_Multiple carriers Premium economy	Airline_SpiceJet	Airline_Trujet	Airline_Vistara	Airline_Vistara Premium economy	Source_Chennai	Source_Delhi	Source_Kolkata	Source_Mumbai
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0

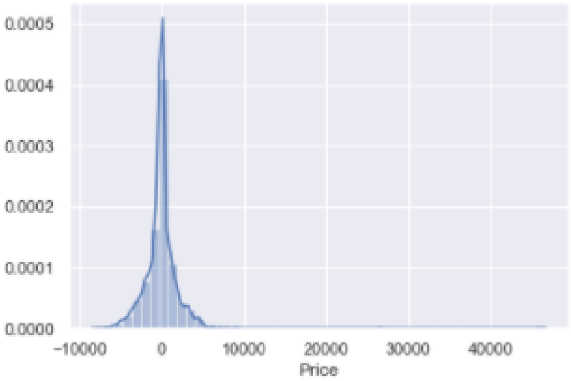
Now to check highly co-related features, heat map is one of the best tool and we can use **ExtraTreeRegressor** from the **sklearn** library to extract these feature and plot them out for easy visualization.

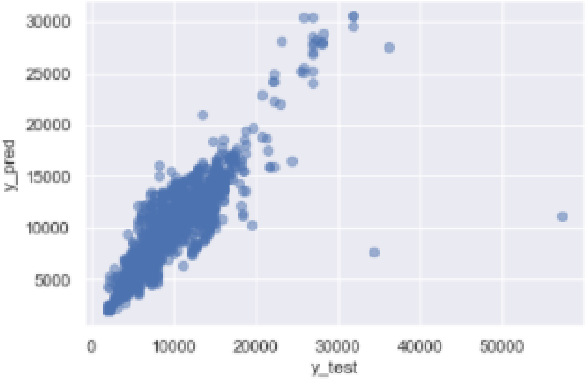


	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Duration_hours	Duration_mins
Total_Stops	1	0.6	-0.0095	0.054	-0.061	-0.0026	0.038	-0.11	0.74	-0.14
Price	0.6	1	-0.15	-0.1	0.0068	-0.024	0.024	-0.086	0.51	-0.12
Journey_day	-0.0095	-0.15	1	-0.038	0.0022	-0.0082	-0.0032	-0.018	-0.022	-0.0089
Journey_month	0.054	-0.1	-0.038	1	0.039	-0.059	-0.0039	-0.1	0.016	-0.041
Dep_hour	-0.061	0.0068	0.0022	0.039	1	-0.025	0.0052	0.068	0.0029	-0.024
Dep_min	-0.0026	-0.024	-0.0082	-0.059	-0.025	1	0.043	-0.018	-0.022	0.092
Arrival_hour	0.038	0.024	-0.0032	-0.0039	0.0052	0.043	1	-0.15	0.055	-0.12
Arrival_min	-0.11	-0.086	-0.018	-0.1	0.068	-0.018	-0.15	1	-0.074	0.15
Duration_hours	0.74	0.51	-0.022	0.016	0.0029	-0.022	0.055	-0.074	1	-0.13
Duration_mins	-0.14	-0.12	-0.0089	-0.041	-0.024	0.092	-0.12	0.15	-0.13	1

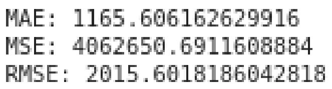


Now our data is ready to be fitted in the **RandomForestRegressor** model from **sklearn** library. The prediction can be visually displayed with the help of **bar graph** and **scatter plot** between **y_test** and **y_pred**.

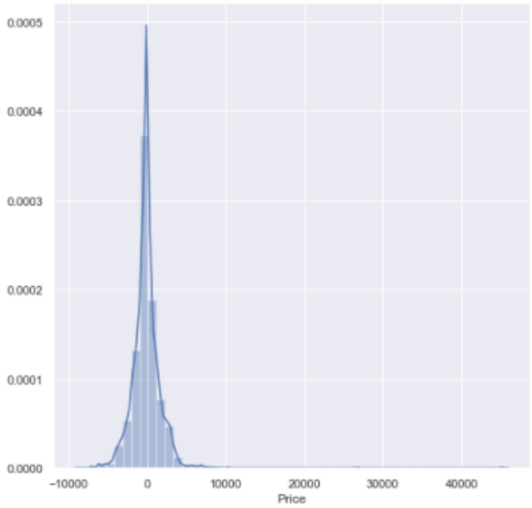


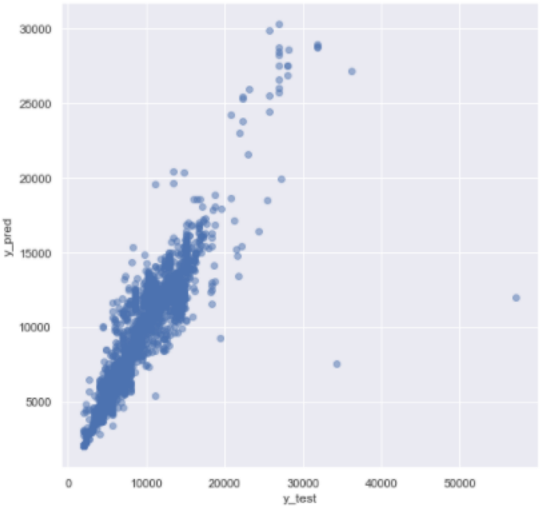


For Hyperparameter tuning, there are two popular options - **RandomizedSearchCV** and **GridSearchCV**, currently the former one is used as it is considerably **faster** from the latter. After fitting the model, we can see the **improved results** with the help of bar graph and scatter plot as well as the **MAE(Mean Absolute Error), MSE(Mean Squared Error) and RMSE(Root Mean Squared Error)**.



```
MAE: 1165.606162629916
MSE: 4062650.6911608884
RMSE: 2015.6018186042818
```





To save the current state for reuse, **pickle** is used to dump the model into .pkl file. User can load this file to a variable and then use predict directly to obtain results for **x_test** and **r2_score**.

5 ANALYSIS

From the **rigorous testing** of the application, I have found that this model works with promised **accuracy** on any date of the year including highly busy days like annual holidays as well as vacation weeks. It also has **fault tolerance** if the input given is not appropriate. The web application will give **proper alert** in all cases of what the application is doing. The annual update system also works perfectly fine with re-bundling of the application and then pushing it as an update **over the air**. It is future proof because while **serving** for many users, the companion program is constantly **scraping** newer data from all supported airline's **APIs** and improving model on a daily basis as well as **back-testing** the current data with older model to measure the change in **trends** in the process of booking an airline ticket in India.

6 CONCLUSION

Air-travel is still a **luxury** in India but we have seen more and more people are now accepting it as a preferred way of travel. The airline companies are also working to make the travel **cheaper** to make it more accessible to middle class citizens. This project was just one way to aid the movement. I will be **open-sourcing** this project on my GitHub so that anyone with the same thought and motivation can make it better to help people and making this part of their life just a tad bit **easier**. There is a lot of scope in this project and I would like to invite you, the anonymous viewer to use the application and give your valuable **feedback** since it is of the utmost importance.

References

1) **COURSERA** - www.coursera.com/

-> Learnt about several frameworks and tools used in the project.

2) **KAGGLE** - www.kaggle.com/

-> Got the initial dataset for training the model.

3) **TOWARDS DATA SCIENCE** - https://towardsdatascience.com/

-> Learnt about different Machine Learning algorithms

4) **DEEP LEARNING.AI** - https://www.deeplearning.ai/

-> Used for researching about what algorithm and visualization methods are best for this project.

Statutory Declaration

I hereby declare that the paper presented is my own work and that I have not called upon the help of a third party. In addition, I affirm that neither I nor anybody else has submitted this paper or parts of it to obtain credits elsewhere before. I have clearly marked and acknowledged all quotations or references that have been taken from the works of others. All secondary literature and other sources are marked and listed in the bibliography. The same applies to all charts, diagrams and illustrations as well as to all Internet resources. Moreover, I consent to my paper being electronically stored and sent anonymously in order to be checked for plagiarism. I am aware that the paper cannot be evaluated and may be graded “failed” (“nicht ausreichend”) if the declaration is not made.

Mohit

Signature