Flight Ticket Price Prediction
PRML - CSL2050

Pranav Goswami (B20CS016)

April 30, 2022

1 Data Pre-processing and Cleaning

The given dataset was read using the *read_csv* function of the *pandas* library and the following things were done for it's pre-processing :-

- We checked for NaN values and since they were less in number so we dropped it.
- After that we Label encoded the following features *'Airline', 'Source', 'Destination', 'Additional_Info' and 'Total_Stops'* using the Label Encoder.
- We then dropped the *'Route'* feature since it was only telling us about the name of the stop, which doesn't provide us with any new information which *'Total_Stops'* feature can't give. We are concerned only about number of stops hence decided to drop this feature
- The *'Date_of_Journey'* feature was considered and all the data points were converted to their equivalent in total seconds using the *total_seconds()* function this function returns the total number of seconds covered for the specified duration of time instance
- After this, *'Dep_Time'* and *'Arrival_time* features were considered and the respective data points were converted to their equivalent in seconds using which we also updated the corresponding values of *'Duration* feature

2 Data visualizations and Analysis

Various plots for comparative analysis and getting the visual understanding of the data were made and are shown below

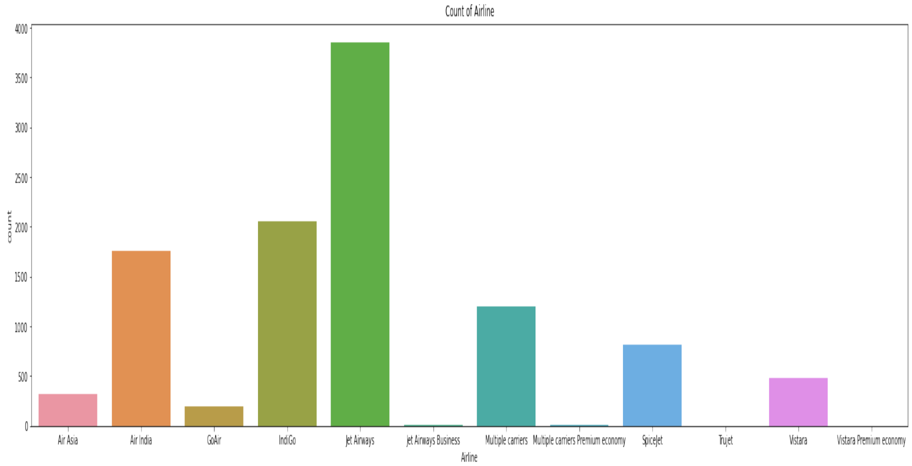


Figure 1: count of different airlines

The above plot shows that most of the flights belong to Jet Airways, Indigo and AirIndia.

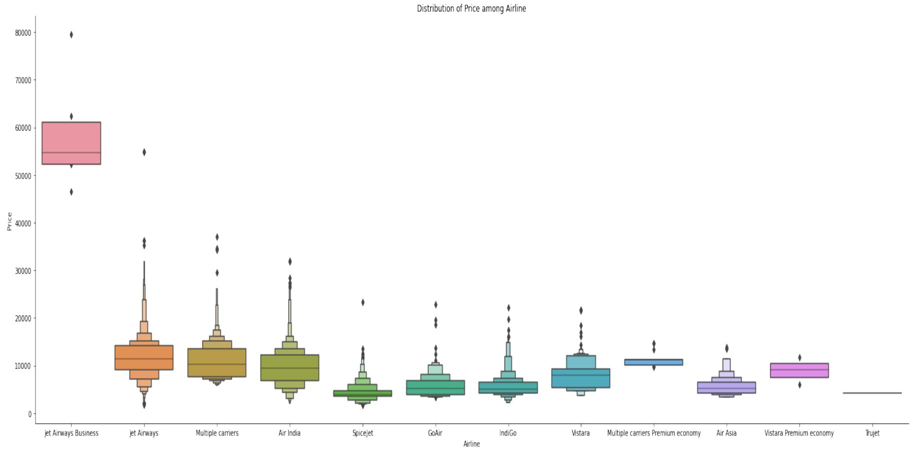


Figure 2: Airline v/s Price

The above plot shows that Jet Airways Business class is the most expensive flight and the median price of all the other flights are roughly equal.

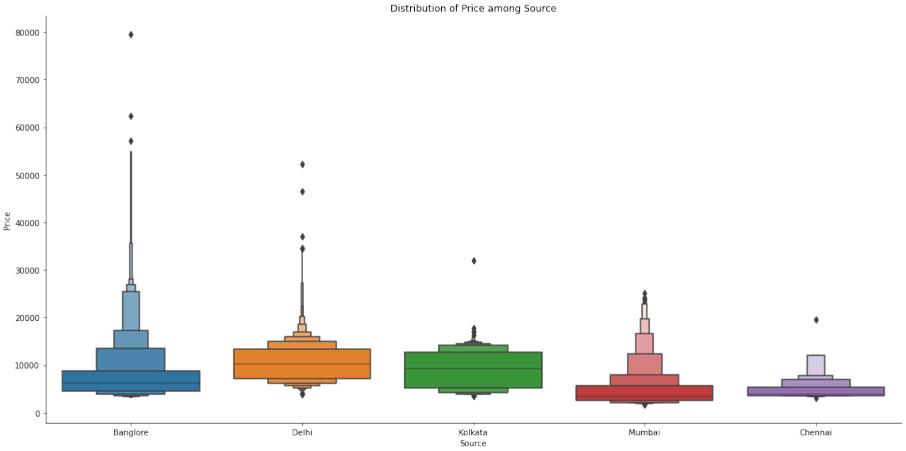


Figure 3: Source v/s Price

The above plot suggests that median price of flights from Delhi is the highest, with Bangalore having the most fluctuating range of price.

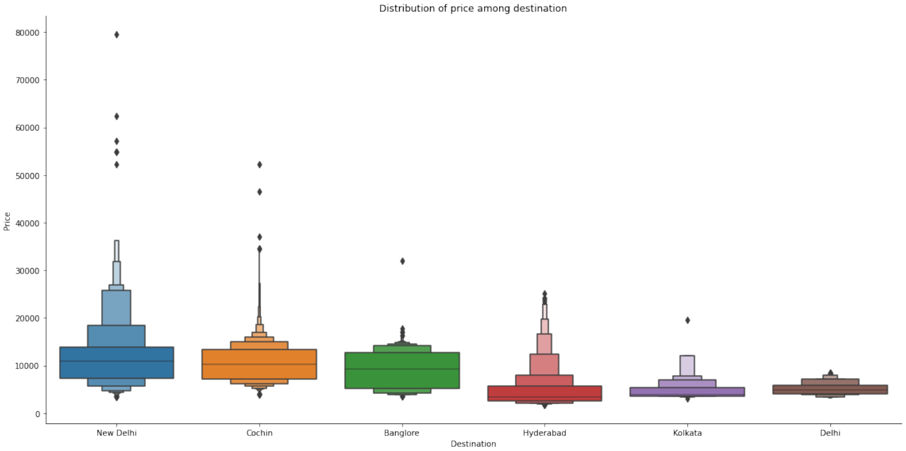
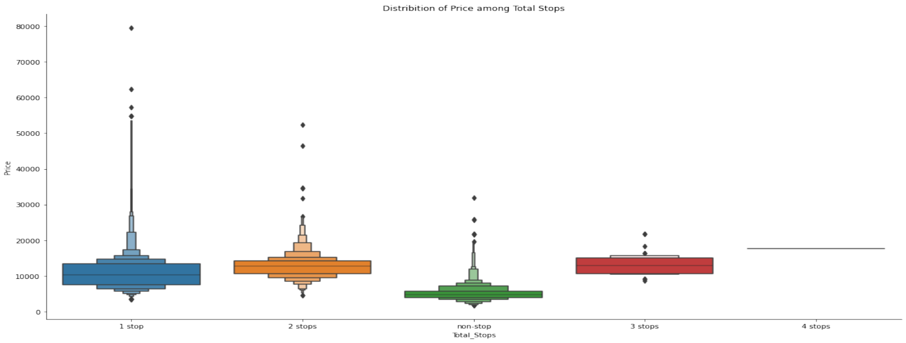
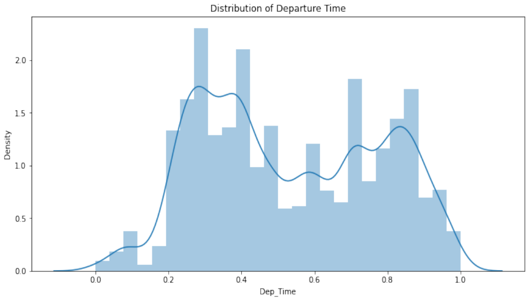


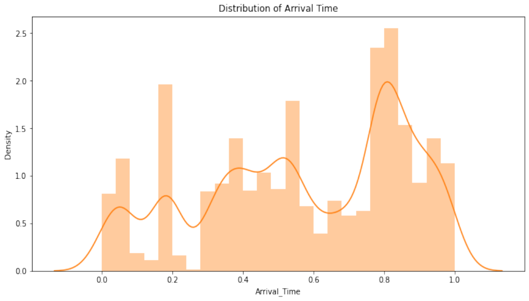
Figure 4: Destination v/s Price

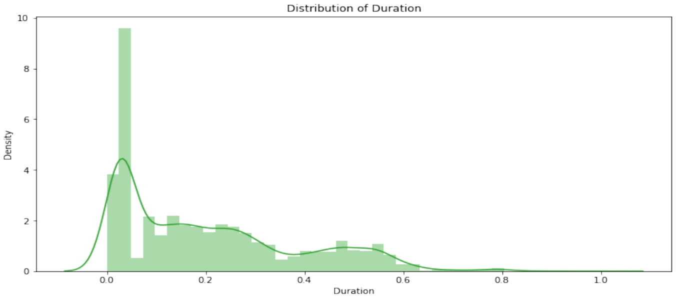
The above plot suggests that median price of flights going to New Delhi is the highest and also have the most fluctuating range of price.



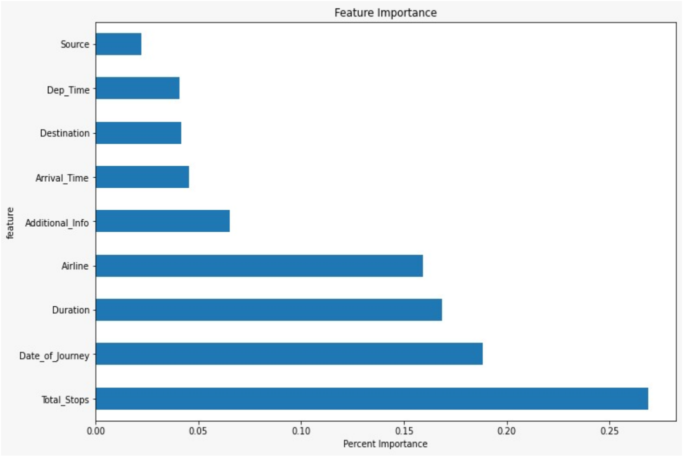
The above plot suggests that flights having 2 or 3 stops have the highest median price with outliers present in flights with 1 stop.







The above plots represent the Gaussian Distribution of Departure Time, Arrival Time and Duration of the flight.



The above plots shows the relative importance of different features and by looking at the plot we get to know that *'Total_Stops* has the most importance among all.

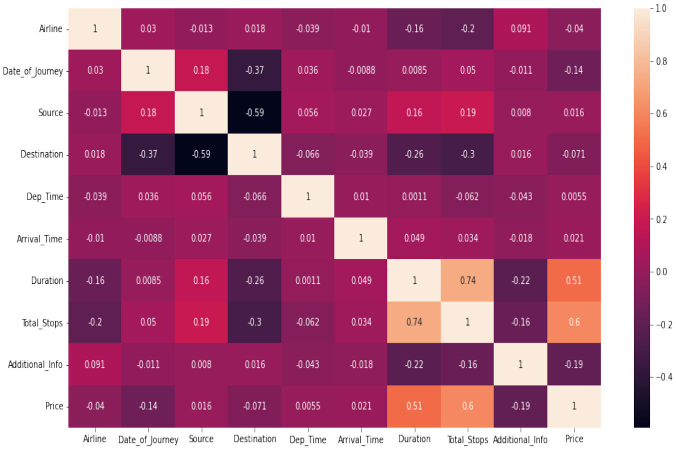


Figure 5: Correlation Heat map of features

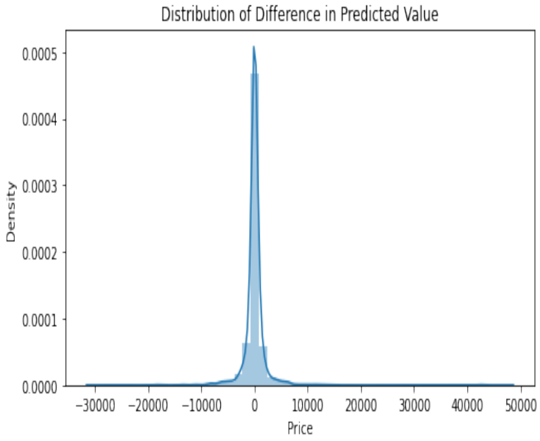
The heat map suggests that the features are neither too strongly nor too weakly correlated.

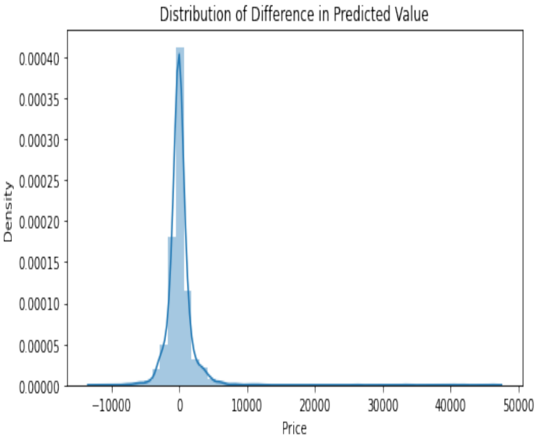
3 Applying Models

We applied different Models for predicting the price of the flight and their performance scores are given below in the table.

Models report				
Model	R2 score	MAE	MSE	RMSE
Random Forest regressor	0.8320	936.109	3714629.305	1927.33
Decision Tree regressor	0.7638	777.305	5223478.629	2285.49
Logistic regressor	0.4377	2486.06	12438636.10	3526.84
XGB regressor	0.85148	719.46	3285420.717	1812.57
Light GBM regressor	0.8756	739.687	2750901.9507	1658.86
SVM	0.5413	1719.1244	10146050	3185.286
DNN	0.83373	1145.4578	3568361	1889.010

The plots of distribution in difference between the actual and predicted price are plotted for some of the models and shown below.





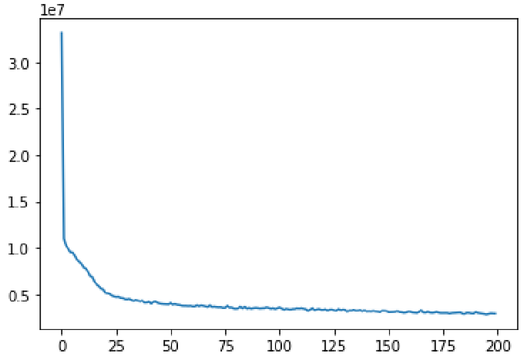


Figure 6: Loss v/s epoch

4 Model Optimization

4.1 Grid Search

We applied the concept of Grid search to find the best value of hyper-parameters like max_depth and n_estimators for various models and saw a significant increase in the R2 scores, the results are given below in the table

Grid search results				
Model	best max depth	min samples split	n_estimators	R2 score
Random Forest regressor	12	2	400	0.8320
Light GBM regressor	15	2	700	0.8809
XGB regressor	6	2	700	0.8870

5 Comparision of Models

We applied different models and then comparing it using various scoring metric as shown below:

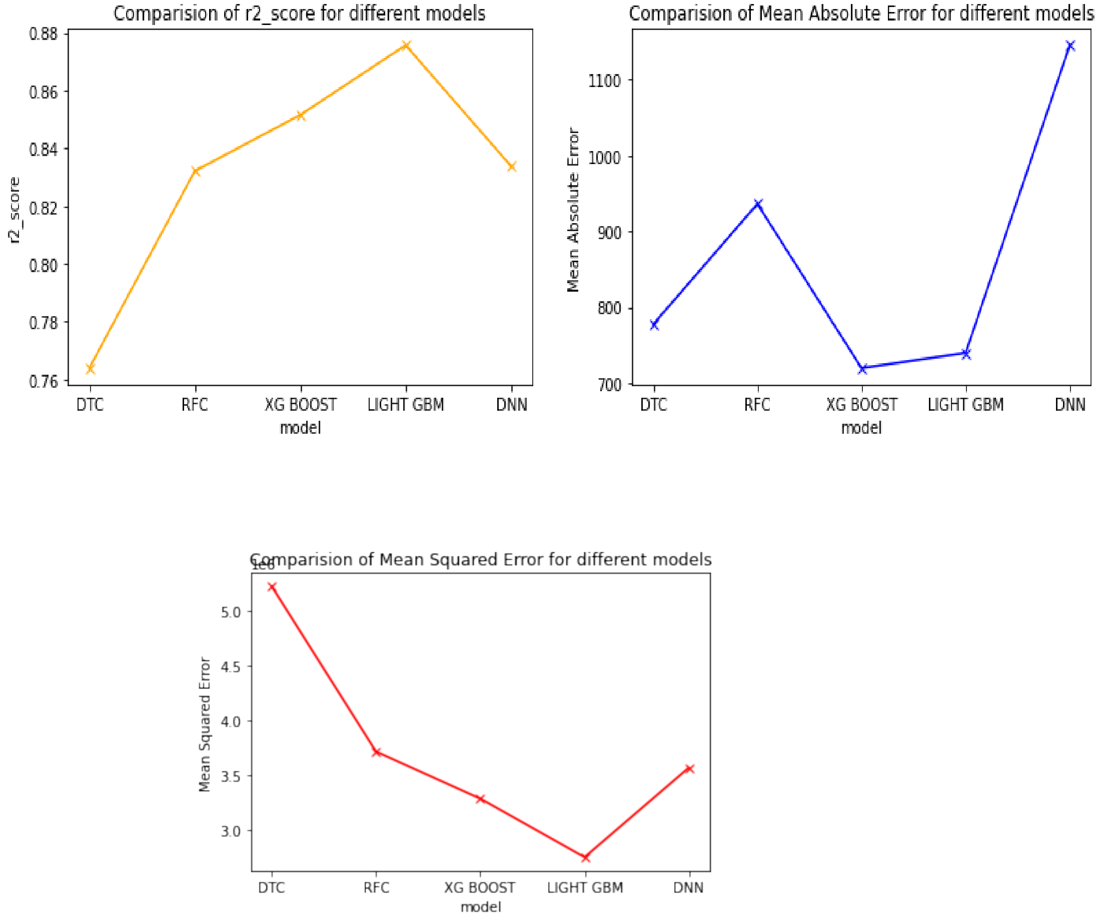


Figure 7: Metric v/s Models

6 Deployment of the Model

Machine learning research usually focuses on optimizing and testing some criteria, but more criteria are needed to deploy in public policy settings. The issue of technical and non-technical deployments has received relatively little attention. However, effective implementation is essential to the true benefits and impact of machine learning models

After Analyzing various models and techniques we decided to go with Multi Layer Perceptron (Deep Learning Technique) as the model which will work for predicting the flight prices based on user input.

6.1 The Keras Model

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models.
A keras model consists of mulitple components :-

- The architecture, or configuration, which specifies what layers the model contain, and how they're connected.
- A set of weights values (the "state of the model").
- An optimizer (defined by compiling the model).
- A set of losses and metrics (defined by compiling the model or calling add_loss() or add_metric()).

We basically need to save the architecture / configuration only, typically as a JSON file and the file which contains weights values only which is generally used when training the model.
Saving a Model :-

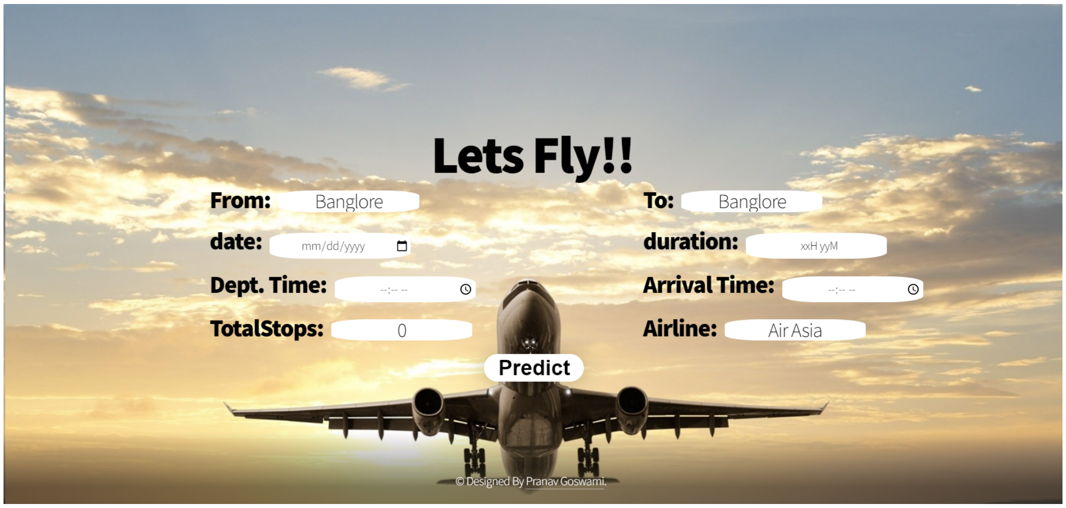
```
model = ...  Get model
model.save('path/to/location')
```

Loading a Model :-

```
from tensorflow import keras
model = keras.models.load_model('path/to/location')
```

6.2 Web Development

We designed the front end of our Website using HTML, CSS, SCSS and JavaScript and successfully deployed it using github, given below is one of the photo of our website's Interface



Lets Fly!!

From: Banglore

To: Banglore

date: mm/dd/yyyy

duration: xxH yyM

Dept. Time: --:-- --

Arrival Time: --:-- --

TotalStops: 0

Airline: Air Asia

© Designed By Pranav Goswami.

Figure 8: Hosted Website Interface

The user here enters the required details for flight price prediction, those details are then fetched in the back-end of our website and using them we created an input tensor of 9 Features which was then passed to the predict function of our Deep learning model, the prediction then is displayed on the Front-end of the website.

Link of the hosted website : -
Github Repository Link : -