CSL2050: Pattern Recognition and Machine Learning

Sanidhya S. Johri B20CS061

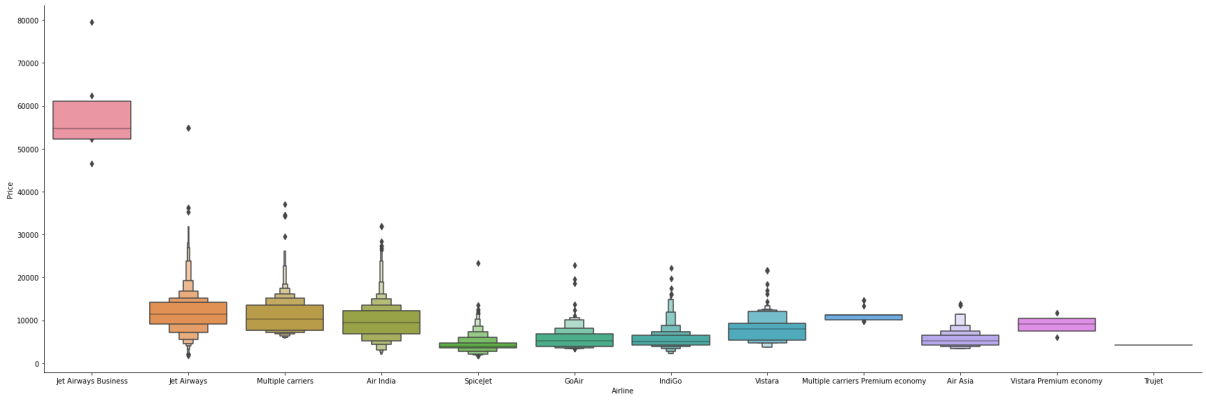
Bonus Project Report

Data Importing:

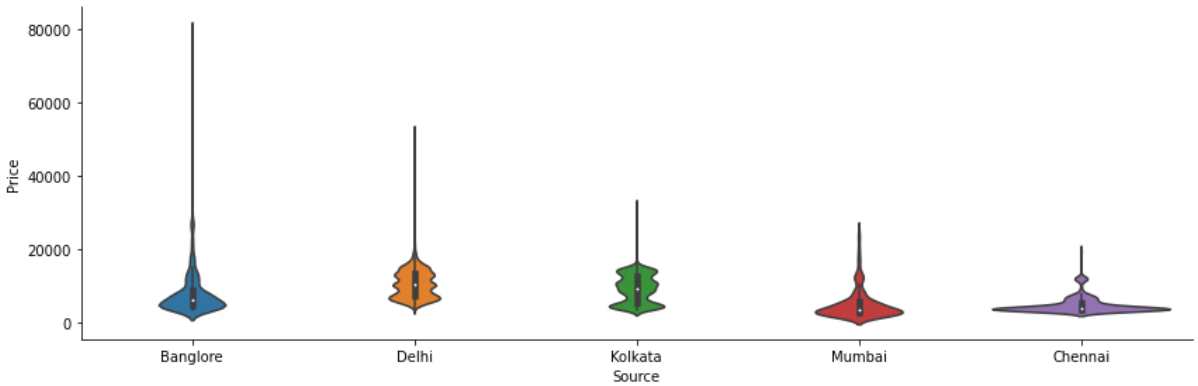
- The drive was mounted in the given file and the dataset was added to the **drive folder**.
- Then the dataset was read using the **read_csv ()** function present in the **panda's library.**

Pre-Processing And Visualisations:

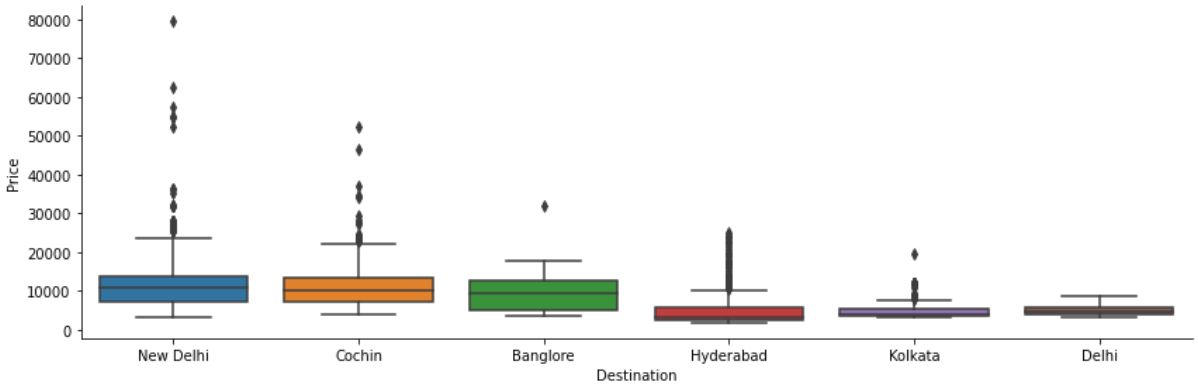
- After that the **pre-processing** of data was done in which the following things mentioned below were done:
    1. Checking for NaN values, since there were some, they were dropped.
    2. Then we plotted some distributions to see the outliers of various features and their ideal range in which most of the data lies. Those plots are shown below.



- The plot of Price vs Airlines show us that Jet Airways has most number of outliers.

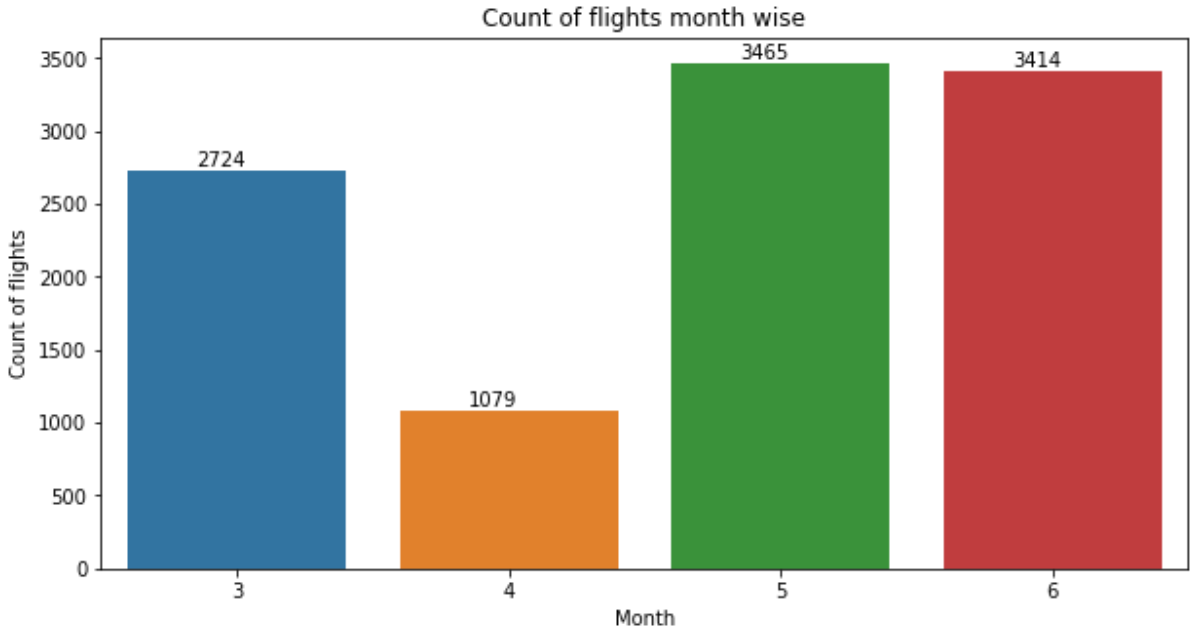


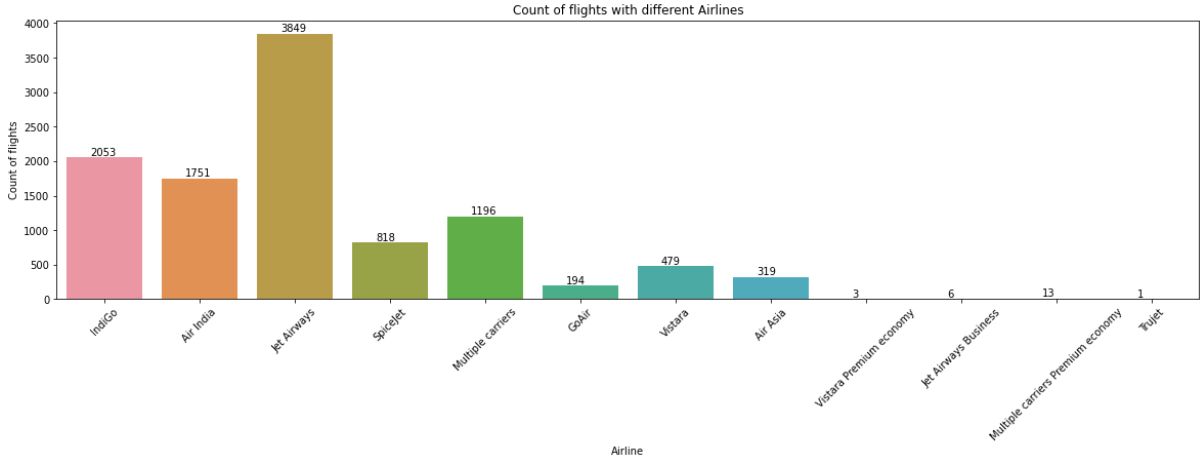
- The plot of Price vs. Source shows us that Bangalore has the most number of outliers and Chennai has the least number of outliers.

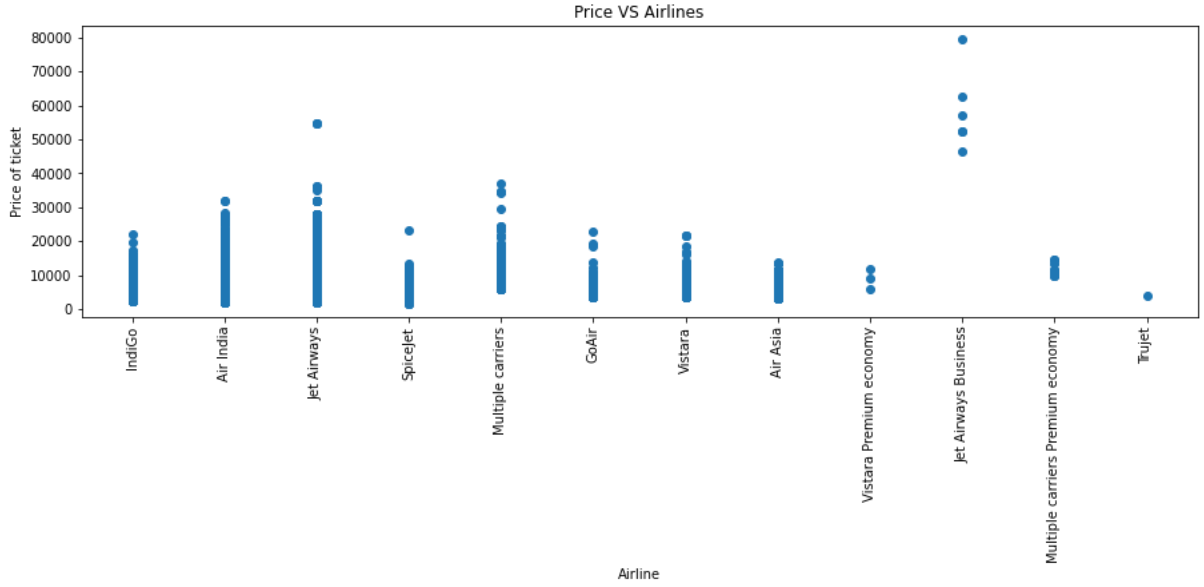


- The plot of Price vs. Destination shows us that New Delhi has the greatest number of outliers and Bangalore has the least.

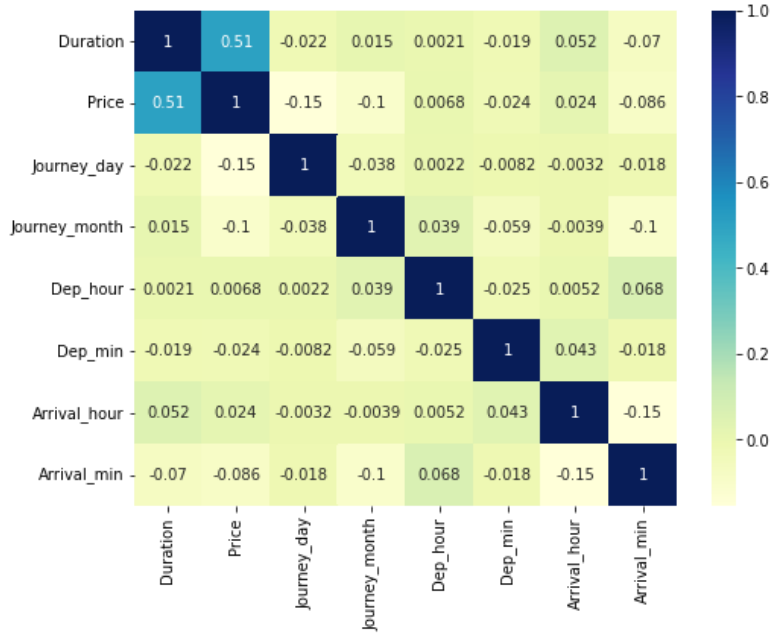
3. Then Feature Engineering was performed on features to make them into usable data for application of Models. The following was performed in the Dataset:
    - The Duration in Hours and Minutes was converted fully into minutes.
    - The Journey date was split into two columns as journey date and journey month.
    - The Departure Time was converted into Departure Hour and Departure min. Same was done with the arrival time.
    - Some columns which were not providing any important information were dropped. For Example: Route, since it was not providing any extra information other that the number of stops which was also being provided into another column in the Dataset.
4. Some plots to see the number of distributions of the new columns were also plotted. Those are shown below.







5. The correlation matrix was also plotted to check the correlation between the features. It is shown below.



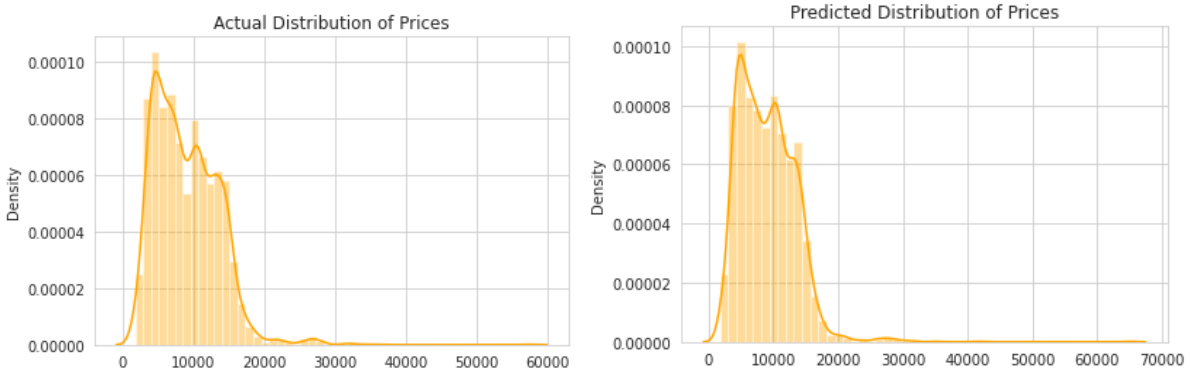
6. After that, label encoding of the categorical features was done.
7. Then, the train-test split was done in 70:30 ratio to evaluate and compare various models and their performance.

Implementing Models:

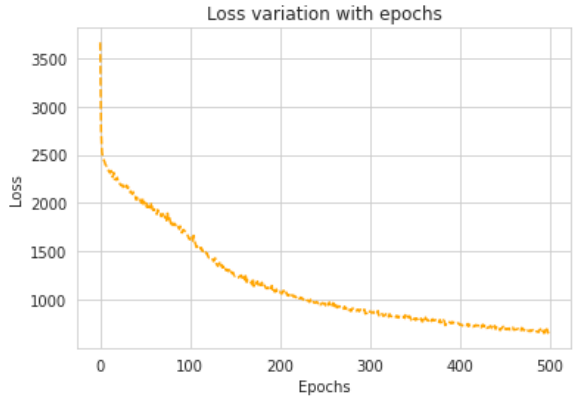
- We applied various model for predicting the price of flight and their performance scores are given below in the table.

Model Report				
Models	R2- Score	MSE	MAE	RMSE
Decision Tree Regressor	0.7999	4081553.998	767.154	2020.28
Random Forest Regressor	0.870	2638658.482	676.88	1624.39
Linear Regressor	0.448	11256968.97	2450.52	3355.14
XGB Regressor	0.830	3452633.006	1241.95	1858.126
LGBM Regressor	0.880	2444307.305	908.58	1563.42
SVR	0.158	17172883.31	3078.426	4144.01
Deep Neural Networks	0.836	3326487.41	1059.527	1823.86

- The plots to show the difference between actual and predicted price are shown below:



- The variation of loss with number of epochs is also shown below:



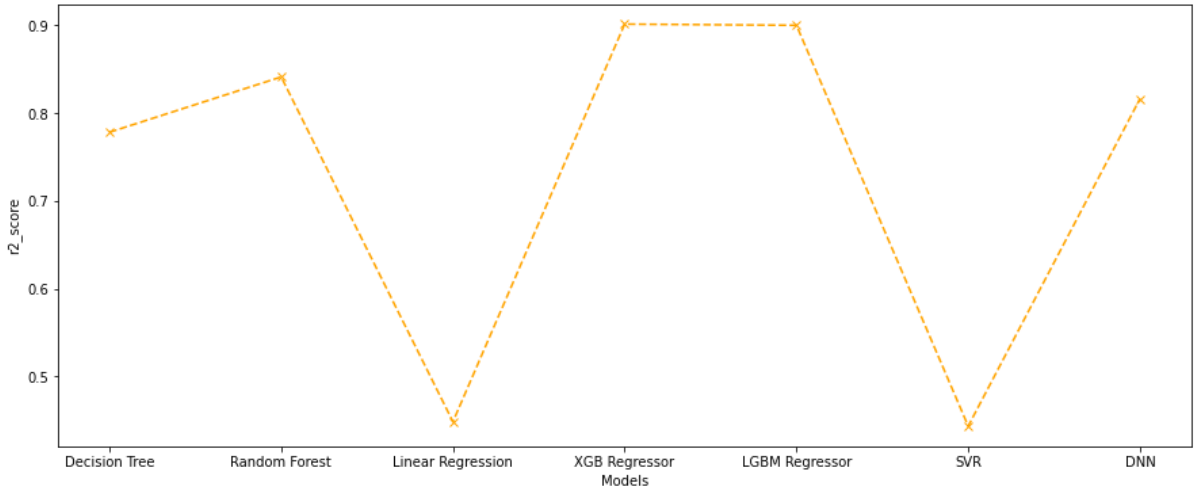
Model Optimisation using GridSearchCV:

- We applied Grid Search CV above models on some hyperparameters and the results after that are shown below:

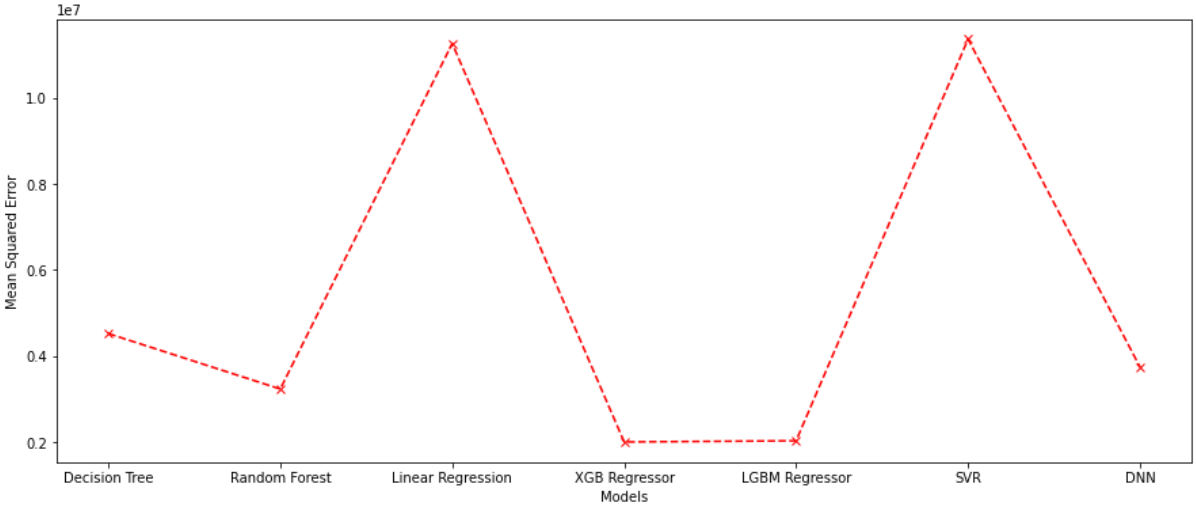
Model Report (After Hyper Parameter Optimisation)				
Models	R2- Score	MSE	MAE	RMSE
Decision Tree Regressor	0.778	4525253.85	1161.14	2127.26
Random Forest Regressor	0.841	3238548.449	1016.766	1799.59
XGB Regressor	0.901	2009996.78	735.7081	1417.74
LGBM Regressor	0.900	2037756.82	802.84	1427.50
SVR	0.443	11358923.04	2223.922	3370.30

Comparative Analysis of Models:

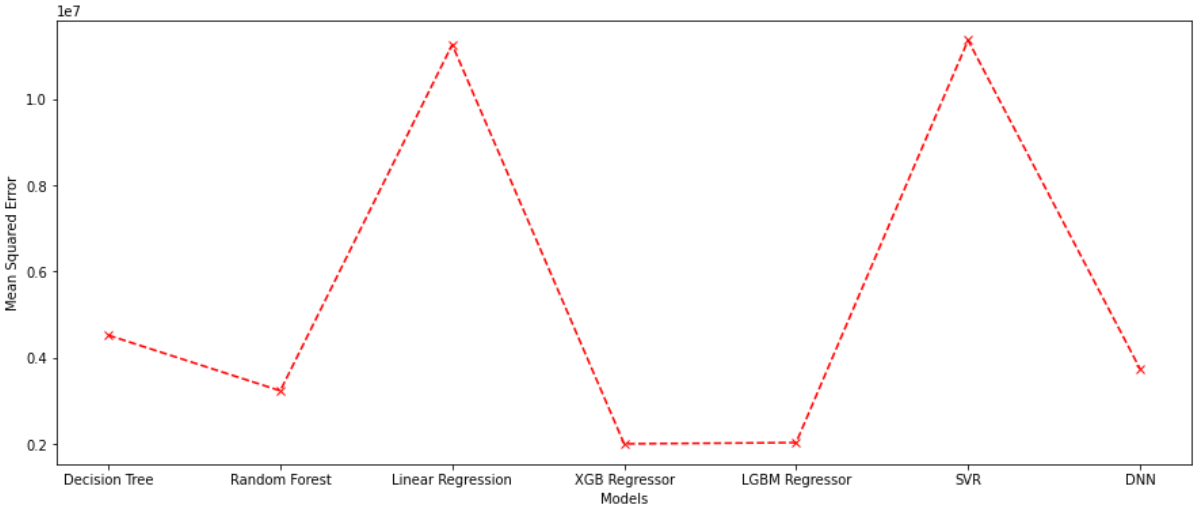
- We applied various models and their comparison reports using different evaluation metrics are shown below:



R2- Score Comparison of Models



Mean Squared Error Comparison of Models



Mean Absolute Error Comparison of Models

Deployment of Model:

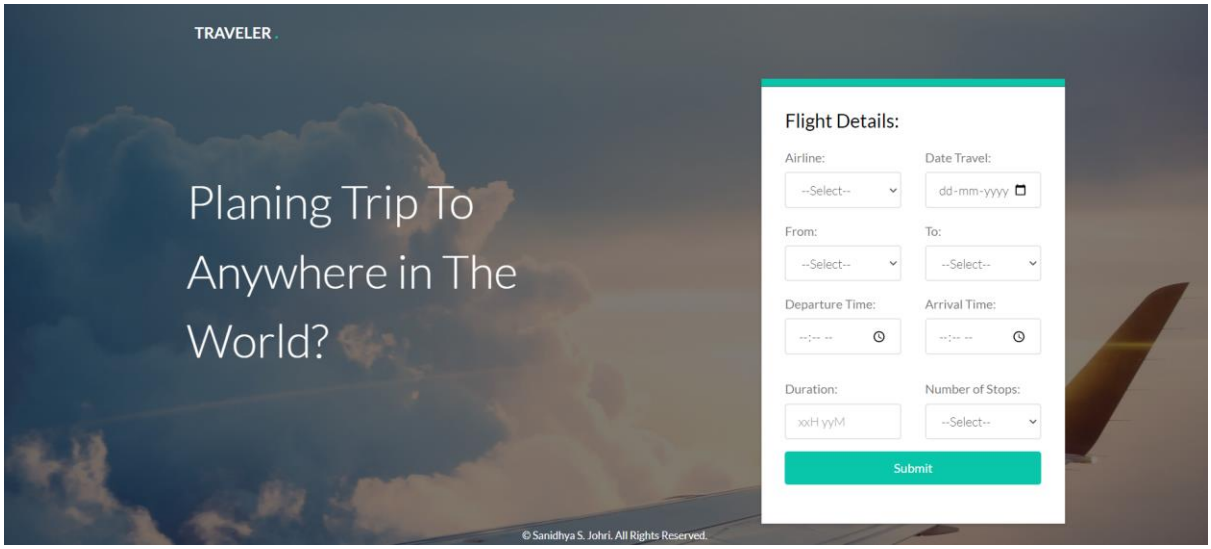
- Machine learning research normally focuses on optimising and testing a few criteria, however in public policy contexts, more criteria are required. The distinction between technical and non-technical deployments has gotten little attention. However, to reap the full benefits and impact of machine learning models, good implementation is required.
- After Analysing 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.

The Keras Model:

- A Keras Model is a powerful, easy to use, open-source Python Library used for developing and evaluating Deep Learning Models. It consists of the following features:
    - The configuration or architecture, which specifies what are the layers contained in it and how they are connected.
    - A set of weight values.
    - An optimiser function (defined by compiling the model)
    - A set of losses and metrics.
- We basically need to save the configuration and the architecture only, typically as the .json file and the file which contains the weight values which are generated while training the model.

Website Deployment:

- We successfully developed our website using HTML, CSS, SCSS and JavaScript and deployed it successfully on GitHub. Below it the Photo of our Website's 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 12 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 To the Web Deployment: - https://unstoppablevenom.github.io/Flight-Price-Prediction-/