Flight Price Prediction

Group 3: ML project

Pratham Singhal 2021082
Akshat Kumar 2021230
Hitesh Kumar 2021257
Harsh 2020061

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Content

- Motivation
- Literature Review
- Dataset Description
- Methodology
- Result/Analysis
- Timeline
- Contribution

Motivation

- Why do travelers compare flights?
- Why different airlines have different prices for the same route?
- Suppose you are a business personal, and travel delhi to mumbai quite often for business purpose, what would you prefer?
- you are a student from Chennai, and studies in Kolkata, what is your preference to select a flight?
- what is the average time spent by the online user to select the flight of his /her choice?

Motivation

Flight rates are affected by a number of variables in today's dynamic travel business, including airlines, destinations, departure schedules, and more.

Making educated judgments might be difficult for travelers due to this complexity. The goal of this research is to use machine learning techniques to create a prediction model that can accurately anticipate flight costs. Travelers may thus maximize their itinerary and perhaps cut costs.

Literature Review | Research Paper 1

https://www.ijraset.com/research-paper/flight-price-prediction

Paper Id : IJRASET 43666 | ISSN : 2321-9653 | Publish Date : 2022-05-31 | Name : IJRASET.

A Btech Project by the students of Guru Nanak Institute of Technology, Kolkata, focuses on the prediction of the flight price of different Airlines.

They initially did some feature engineering by changing the units of times from hours into minutes, arrival and departure days into rescaling factor, and characterization of certain features.

Then they tried to model the problem using Decision Trees, Random Forests, and XGBoosts. They calculated the accuracy of the model, and found that Random Forest Overfit the model, and the best result could be found using Decision Trees only showing an accuracy of 78% on the testing data.

Inference: We can improve the accuracy of models like these, by using more **feature engineering**, and **pruning**. Also we will try to implement the model using **regression**, which might increase the chances of accuracy of the hypothesis.

Literature Review | Research Paper 2

https://www.researchgate.net/publication/335936877_A_Framework_for_Airfare_Price_Prediction_A_Machine_Learning_Approach

A Framework for Airfare Price Prediction: A Machine Learning Approach

Feature Name	Description
Distance	Market distance between the origin and destination airports
Seat Class	Indicator for economy or premium seat type
Passenger Volume	Total number of passengers traveled between the origin and destination airports
Load Factor	The ratio of the total number of passenger to the total number of seat in a market
Competition Factor	The market HHI
LCC Presence	Indicator of LCC operating in the market
Crude Oil Price	Quarterly average of crude oil price
CPI	Quarterly average of Consumer Price Index
Quarter	Indicates the three month period of the year

Method	RMSE	R^2_{adj}
LR	112.039	0.599
SVM	109.914	0.615
MLP	94.569	0.715
XGBoost	90.419	0.739
RF	**70.575**	**0.804**

Dataset Description

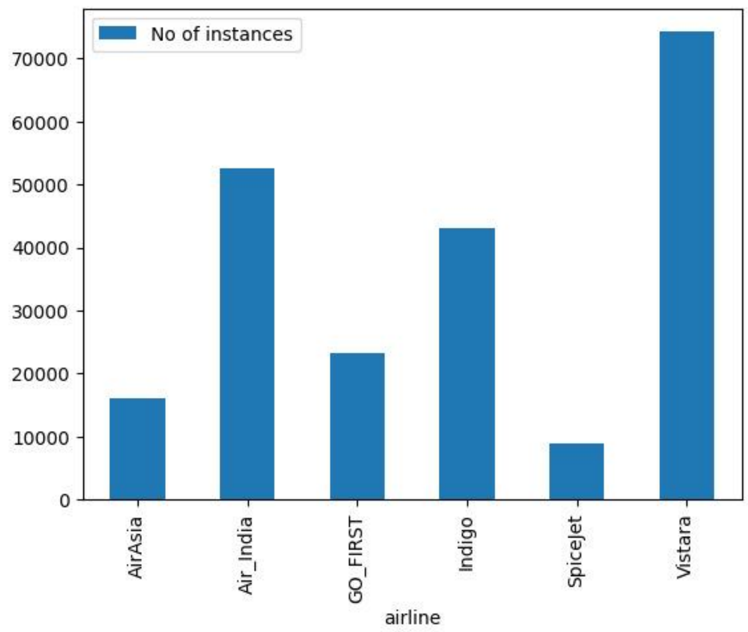
Feature	**Description**
Unnamed	It refers to the column which represent the serial number
Airline	It represent the company name of the airline whose flight is scheduled
Flight	It refer the Airplane number which is ready to fly
Source_City	City name, from where the flight will depart from
Departure_Time	Time of take-off
Stops	Number of stops the flight will wait, until reaching the destination
Arrival_Time	Time of landing
Destination_City	City name where the flight will land
Class	Seat type in the flight
Duration	Total time of the flight in air
Days_left	Number of days left for the flight to fly
Price	The price of the flight.

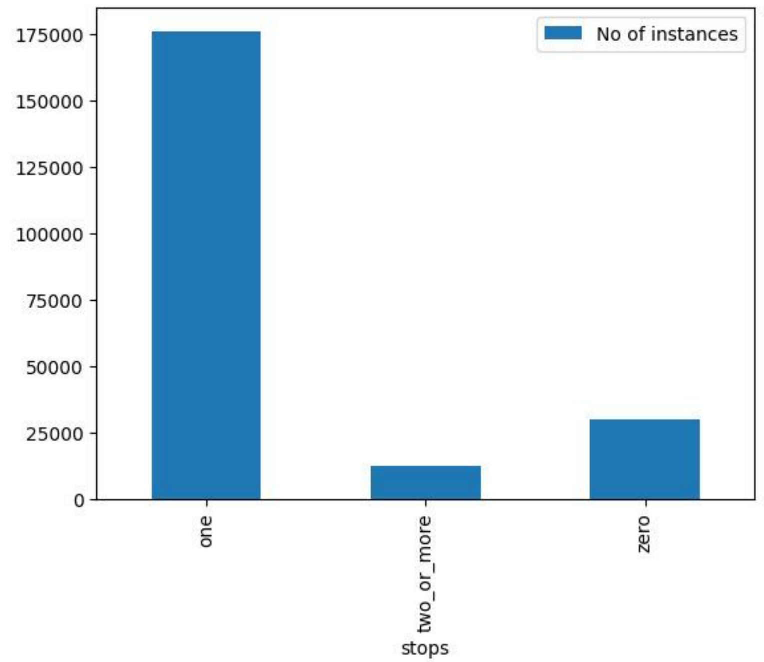
Data preprocessing:

- Removing the NA values from the dataset.
- Differentiating the Categorical and Numerical features.
- Label Encoding of the categorical data. We have assigned different numbers to different classes in Categorical Variables which has made it easy to train our models on the preferred features.
- Normalization of the numerical data. In order to discard the anomalies in the variation of the numerical data, we normalized these features so that our model does not count the bias created from them.
- Bootstrapping. Observing our dataset, we found that certain features involve high bias for certain values in the feature. hence we added certain more data points in those classes of features where the number was less using bootstrap that is repeating the number of the same instances. This is done because we do not want out model to lose out any class while training.

Dataset visualization:

1. Number of instances of each airline
2. Number of flights with different number of stops

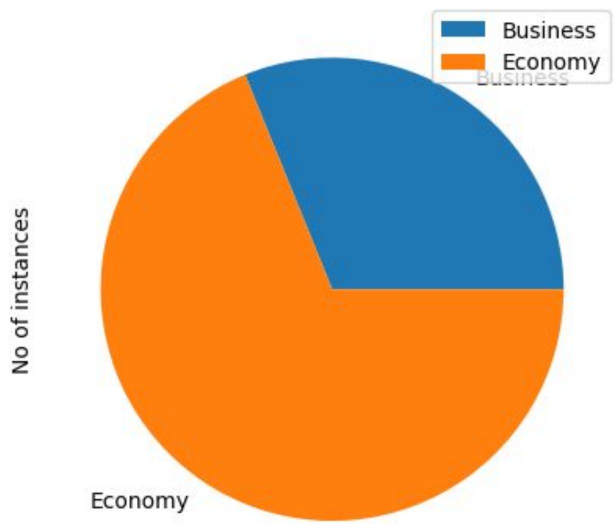


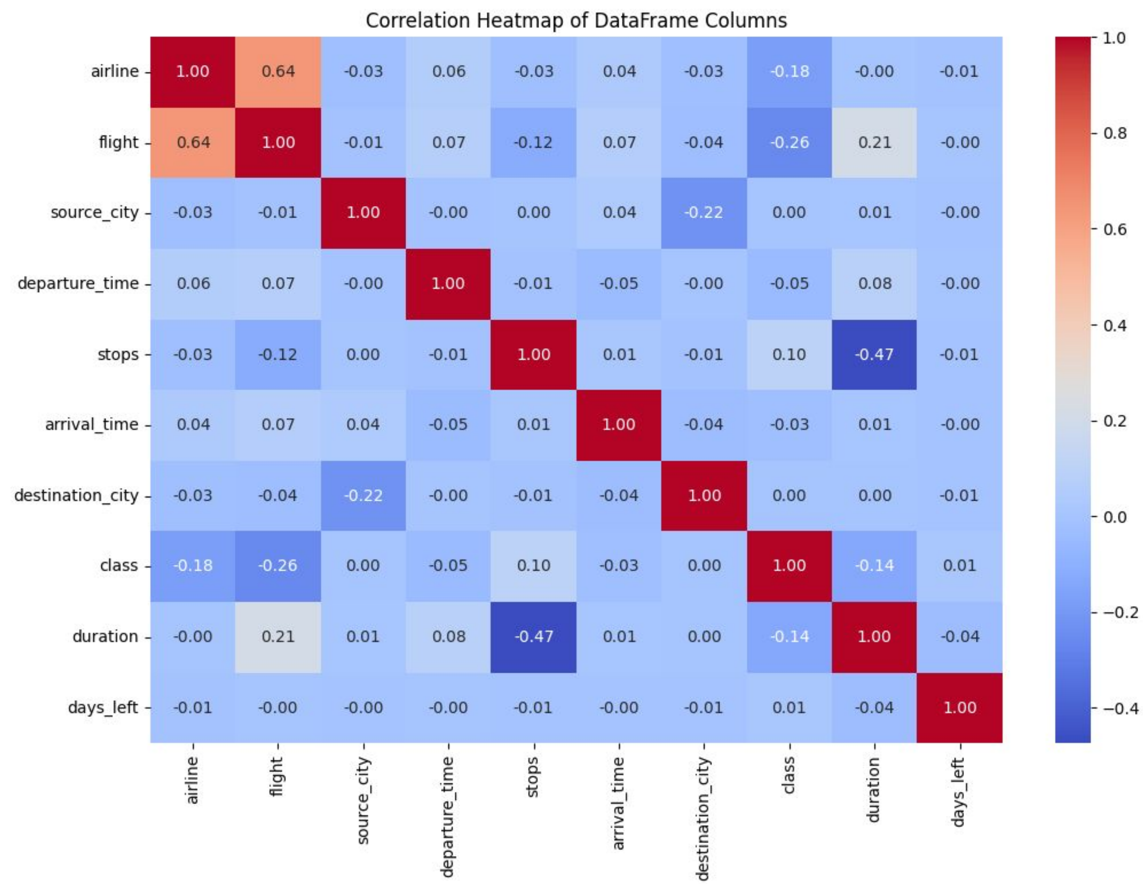


Data Visualization

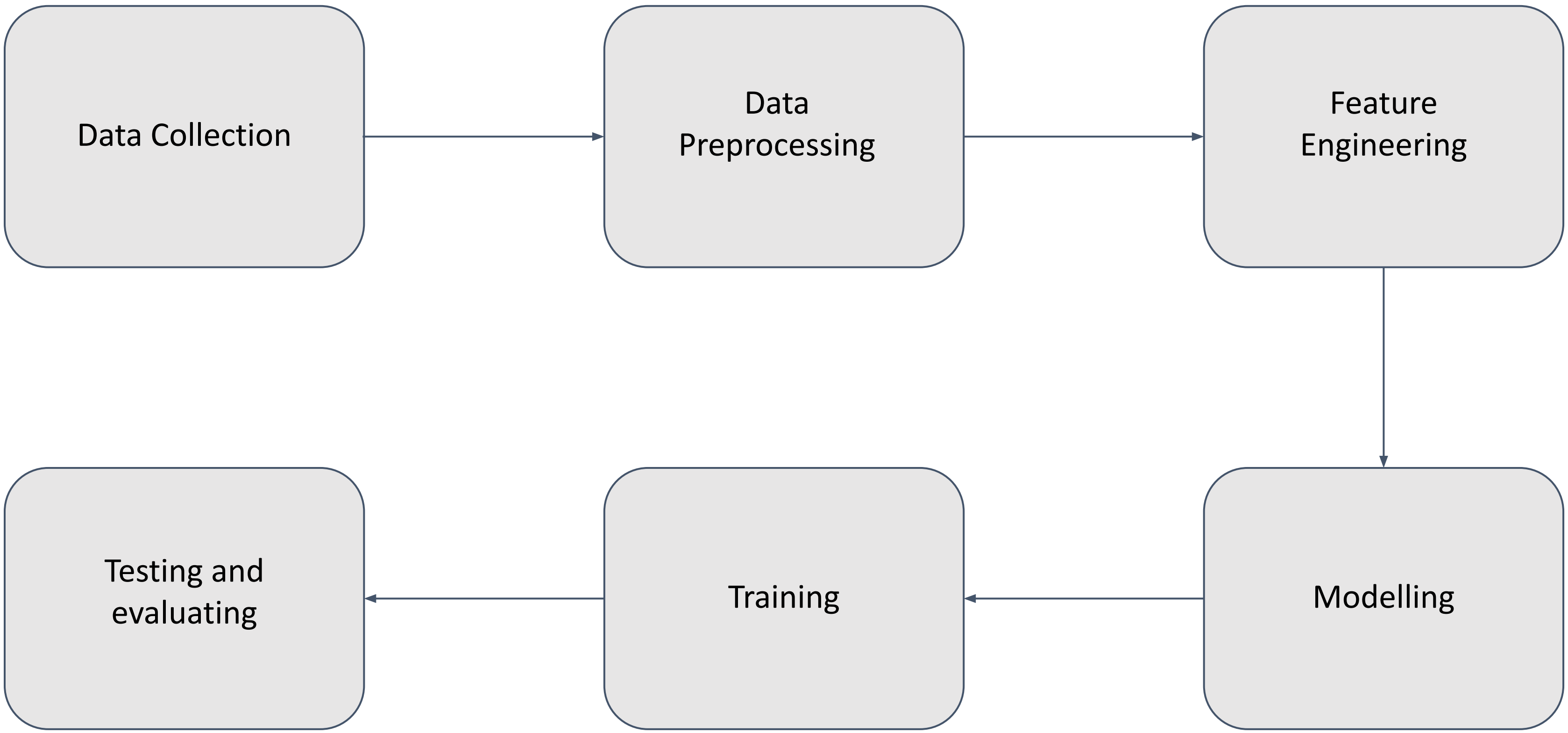
3. Instances of economy vs business class:

4. Heatmap





Methodology



Results:

Sno.	Model	R2	MAE
1	LINEAR REGRESSION	0.904	4622.18
2	LASSO	0.900	4833.36
3	RIDGE	0.904	4617.12
4	SGD REGRESSOR	0.904	4640.27
5	NAIVE BAYES	0.934	3021.48
6	SVM	0.310	15927.34
7	DECISION TREES	0.982	892.61
8	RANDOM FOREST	0.989	865.05
9	POLYNOMIAL(DEG=2,3,4,5)	0.94	3666.12

Results (Linear Regression):

We ran the linear regression model with the given dataset and the required preprocessing steps to get the following results:

```
Training data:
Mean squared error:  12040282.749627912
R2:  0.8908763028346185
RMSE:  3469.9110578843247
MAE:  2243.0603399825404

Testing data:
Mean squared error:  9257727051025222.0
R2:  -85674945.86000177
RMSE:  96217082.94801512
MAE:  997400.4031276882
```

Results (Lasso Regularization):

After linear regression, we used l1 regularization to get the following results:

```
Using L1 linearization:
Training data:
Mean squared error:  14799692.741018897
R2:  0.8908763028346185
RMSE:  3847.0368780424888
MAE:  2409.1198133526887

Testing data:
Mean squared error:  14383990.244153365
R2:  0.8668844314581352
RMSE:  3792.6231350021276
MAE:  2377.5686157244627
```

Results (Ridge Regularization):

After linear regression, we used L2 regularization to get the following results:

```
Using L2 linearization:
Training data:
Mean squared error:  12984139.594986424
R2:  0.8908763028346185
RMSE:  3603.3511617640634
MAE:  2281.780119479221

Testing data:
Mean squared error:  12436619.556287775
R2:  0.8849062287394812
RMSE:  3526.559166707369
MAE:  2237.5176282487564
```

Results (SGD Regressor):

We also used SGD Regressor to get the following results:

```
Analytics based on SGDRegressor:
Training data:
Mean squared error:  12984139.594986424
R2:  0.8908763028346185
RMSE:  3603.3511617640634
MAE:  2281.780119479221
Testing data:
Mean squared error:  12984139.594986424
R2:  0.8908763028346185
RMSE:  3603.3511617640634
MAE:  2281.780119479221
```

Results (Naive Bayes)

Using naive bayes we got the following results:

```
Training data:
Mean squared error:  31899550.115320545
R2:  0.9380727239933645
RMSE:  5647.968671595173
MAE:  2850.4332006230165

Testing data:
Mean squared error:  33969198.11069281
R2:  0.9341021023370764
RMSE:  5828.310056156313
MAE:  3021.4864819843083
```

Results (SVM):

Using SVM we got the following results:

```
Training data:
Mean squared error:  675948287.6920439
R2:  -0.31357812882865166
RMSE:  25999.00551352001
MAE:  15958.32732892266

Testing data:
Mean squared error:  680179753.0063303
R2:  -0.3103842639545684
RMSE:  26080.255999631794
MAE:  15927.035800705196
```

Results (Decision tree)

Using decision tree we got the following results:

```
Training data:
Mean squared error:  52732.817387966665
R2:  0.9998976286585489
RMSE:  229.63627193448048
MAE:  11.837750532368268

Testing data:
Mean squared error:  8798775.039998315
R2:  0.982930984262405
RMSE:  2966.2729206865497
MAE:  892.6192828149901
```

Results (Random Forest)

Using random forest we got the following results:

```
Training data:
Mean squared error:  769307.1466485786
R2:  0.9985065276522034
RMSE:  877.1015600536682
MAE:  324.9695370308122

Testing data:
Mean squared error:  5395104.464499838
R2:  0.9895338700453314
RMSE:  2322.736417353428
MAE:  865.0561041108258
```

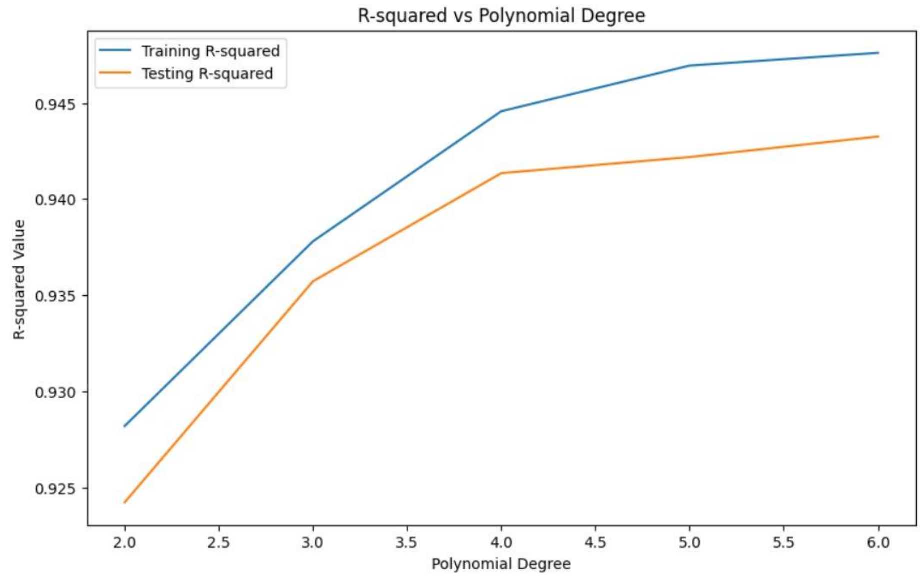
Results (Polynomial Regression)

Using polynomial regression we got the following results:

```
Results for Polynomial Degree 3:
Training data:
Mean squared error:  32004002.335908167
R2:  0.9378062519442567
RMSE:  5657.2079982892765
MAE:  3744.330147252241
Testing data:
Mean squared error:  33361636.64792932
R2:  0.9357279256108499
RMSE:  5775.953310747008
MAE:  3785.006589806098

Results for Polynomial Degree 4:
Training data:
Mean squared error:  28524338.526775714
R2:  0.944568322887523
RMSE:  5340.818151442316
MAE:  3541.834768998882
Testing data:
Mean squared error:  30443347.68356327
R2:  0.9413500863994877
RMSE:  5517.549064898587
MAE:  3628.5739423097066

Results for Polynomial Degree 5:
Training data:
Mean squared error:  27302995.70408381
R2:  0.946941772526944
RMSE:  5225.226856710033
MAE:  3516.4624171681535
Testing data:
Mean squared error:  30009373.674694195
R2:  0.9421861487928093
RMSE:  5478.081203733128
MAE:  3666.0846119049666
```



Timeline

1. Data Preprocessing ⟶ Week 1
2. Data Visualization & Feature Engineering⟶ Week 2
3. Model Training and optimization ⟶ Week 4
4. Front End Ideation and Application ⟶Week 5
5. Model Evaluation and Fine Tuning ⟶ Week 6

Contribution:

Work to be done by:

1. Akshat and Pratham:

    Data Preprocessing and Exploration

    Feature Engineering and Selection

2. Hitesh and Harsh:

    Model Selection and Training

    Model Evaluation and Fine-Tuning

3. All members:

    Create comprehensive documentation covering the entire project.