Flight price prediction

Submitted by:

SURENDRAN . G

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) So, you have to work on a project where you collect data of flight fares with other features and work to make a model to predict fares of flights.

- **Conceptual Background of the Domain Problem**
  This ML model will help to find the flight at the best possible cheapest price. This helps the user to understand how early that they should book a ticket to get the ticket at low price and which airways and which time of travel to be chosen to find the cheapest ticket.
- **Review of Literature**
  This research focuses on finding the best reasonable flight price to travel with in India between available cities in india. This is derived based on the data scrapped from Yarta.com which is a web service that helps the end user to book the ticket from their own place.

  This ML model will help the user to buy the ticket at best possible low cost which will help the user to plan their travel in order to save their money.

- **Motivation for the Problem Undertaken**

Now a days mode of Air travel had been increased between cities. This model will help the end user to plan for their travel in order to find the ticket at the best possible low cost and book the ticket in right time to save their cost.

Analytical Problem Framing

- **Mathematical/ Analytical Modeling of the Problem**

  Since all the independent variables are categorical and date, this model does not need much statistical approach. Model that we had used is XGBoost regressor with the accuracy of 91%.

- **Data Sources and their formats**

  Complete data to develop this model is scrapped from Yatra.com. features as such, destination, origin, Date of departure, date of arrival, no of stops, departure time arrival time and price.

- **Data Preprocessing Done**

  Since all independent variables are categorical , we encoded them with both label encoding and on hot encoding techniques.

  Departure date is converted to new feature as the number of days before the departure date with the help of date in which data is scrapped.

- **Data Inputs- Logic- Output Relationships**

  Model takes the input features as, destination, origin, number of stops, number of days before the departure date, travel duration, Airways name. All these features has impact on predicting the fight fare.

- **State the set of assumptions (if any) related to the problem under consideration**

This model will help the user to predict the flight fare one month before the departure date due to data restriction.

This model will help the user to predict the flight price between cities in india.

- **Hardware and Software Requirements and Tools Used**
  Software's used are python 3.7.6, selenium
  Libraries used are – Pandas , Numpy , Matplot , seaborn , scikit-learn

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

  As said above intention of this model is to derive the best reasonable flight fare. To achieve this we had collected data from Yatra.com. Then the scrapped data went through data pre-processing steps as mentioned above.

  Since the expected output is a numerical feature, algorithms used to develop this model are regression models.

  We had tried regression models like, Linear regression, ADA boost regression, Random forest regression, KNN regression and XGBoost regression.

- **Testing of Identified Approaches (Algorithms)**

Algorithms used to test the data set and find the best performing models are as follows.,

i. Linear regression
ii. ADA boost regression
iii. Random forest regression

iv. KNN regression
v. XGBoost regression.

- **Run and Evaluate selected models**

a. Linear regression

```
best_acc=0
best_random=0

for i in range(1,100):
    X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=.20,random_state=i)
    LR=LinearRegression()
    LR.fit(X_train,Y_train)
    LR_pred=LR.predict(X_test)
    score=r2_score(Y_test, LR_pred)
    if score>best_acc:
        best_acc=score
        best_random=i

print("Best Accuracy score is : ",best_acc,' and Random_state is : ',i)
```

```
Best Accuracy score is :  0.4265920286424051  and Random_state is :  99
```

b. ADA boost regression

```
ADA=AdaBoostRegressor()
ADA.fit(x_train,y_train)
ADA_pred=ADA.predict(x_test)

r2score=r2_score(y_test,ADA_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, ADA_pred))
print('Mean squared error: ', mean_squared_error(y_test,ADA_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,ADA_pred)))
```

```
R2score is :  0.5768477019275178

Errors:
mean Absolute error:  1291.9258064618364
Mean squared error:  2678395.563227656
Root mean squared error:  1636.5804481380242
```

c. Random forest regression

```
RFR=RandomForestRegressor()
RFR.fit(x_train,y_train)
RFR_pred=RFR.predict(x_test)

r2score=r2_score(y_test,RFR_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, RFR_pred))
print('Mean squared error: ', mean_squared_error(y_test,RFR_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,RFR_pred)))
```

```
R2score is :  0.9357572750185753

Errors:
mean Absolute error:  252.9338717575092
Mean squared error:  406632.3882528671
Root mean squared error:  637.6773386696967
```

d. KNN regression

```
KNN=KNeighborsRegressor(n_neighbors=2)
KNN.fit(x_train,y_train)
KNN_pred=KNN.predict(x_test)
r2score=r2_score(y_test,KNN_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, KNN_pred))
print('Mean squared error: ', mean_squared_error(y_test,KNN_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,KNN_pred)))
```

```
R2score is :  0.8285288307053396

Errors:
mean Absolute error:  469.1656
Mean squared error:  1085348.2804
Root mean squared error:  1041.8004993279665
```

e. XGBoost regression.

```
XGB=XGBRegressor()
XGB.fit(x_train,y_train)
XGB_pred=XGB.predict(x_test)
r2score=r2_score(y_test,XGB_pred)

print('R2score is : ',r2score)
print('')
print('Errors:')

print('mean Absolute error: ', mean_absolute_error(y_test, XGB_pred))
print('Mean squared error: ', mean_squared_error(y_test,XGB_pred))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,XGB_pred)))
```

```
R2score is :  0.9074109011986221

Errors:
mean Absolute error:  314.2815302734375
Mean squared error:  586054.3179429435
Root mean squared error:  765.5418459776994
```

- **Key Metrics for success in solving problem under consideration**

  The R-squared statistic provides a measure of fit. It takes the form of a proportion—the proportion of variance explained—and so it always takes on a value between 0 and 1. In simple words, it represents how much of our data is being explained by our model.


```
R2score is :  0.9074109011986221

Errors:
mean Absolute error:  314.2815302734375
Mean squared error:  586054.3179429435
Root mean squared error:  765.5418459776994
```

To ensure that the data is not over fitting and perform better with different sub set of sample we had user cross validation technique as shown below.

```
from sklearn.model_selection import cross_val_score
print("Linear regression cross validation accuracy is :", cross_val_score(LR,x,y,cv=10,scoring='r2').mean())
```

```
Linear regression cross validation accuracy is : 0.33538012722688154
```

```

```

```
print("Random forest regressor cross validation accuracy is :", cross_val_score(RFR,x,y,cv=10).mean())
```

```
Random forest regressor cross validation accuracy is : 0.9192166127131103
```

```
print("ADA boost regressor cross validation accuracy is :", cross_val_score(ADA,x,y,cv=10).mean())
```

```
ADA boost regressor cross validation accuracy is : 0.47988623412621045
```

```
print("KNN cross validation accuracy is :", cross_val_score(KNN,x,y,cv=10).mean())
```

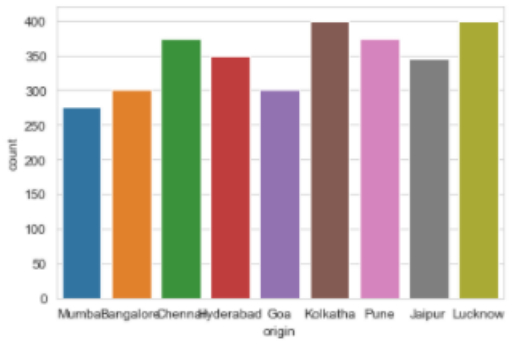
```
KNN cross validation accuracy is : 0.7391326520852959
```

```
print("XGB cross validation accuracy is :", cross_val_score(XGB,x,y,cv=10).mean())
```

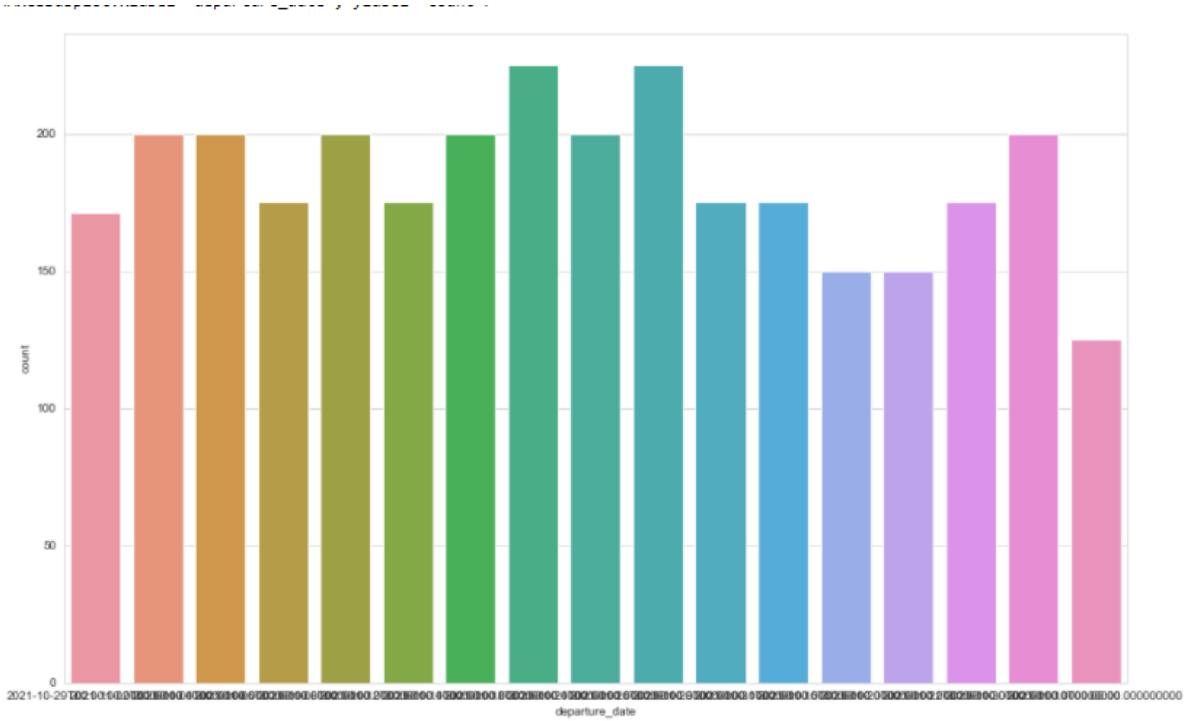
```
XGB cross validation accuracy is : 0.9201810258031358
```

- **Visualizations**

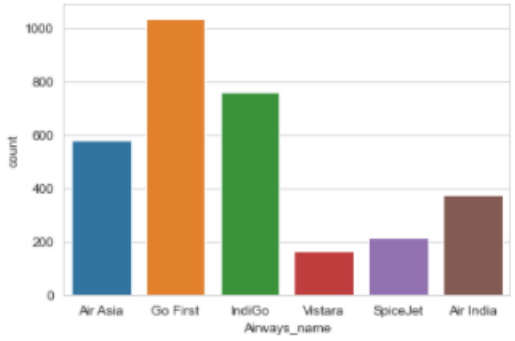
all the fight details scrapped are originated from Mumbai, Bangalore, Chennai, Hyderabad, Goa, Kolkata, Pune, Jaipur and Lucknow



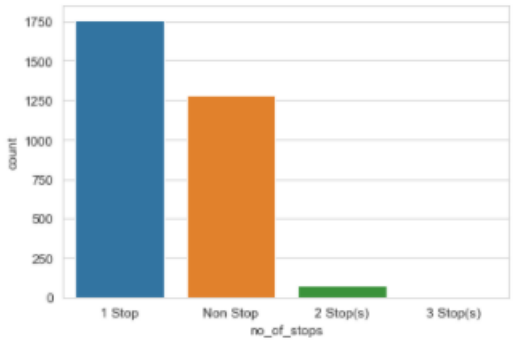
This data set contains fight details between 26th of Oct 2021 to 31st of Nov 2021.



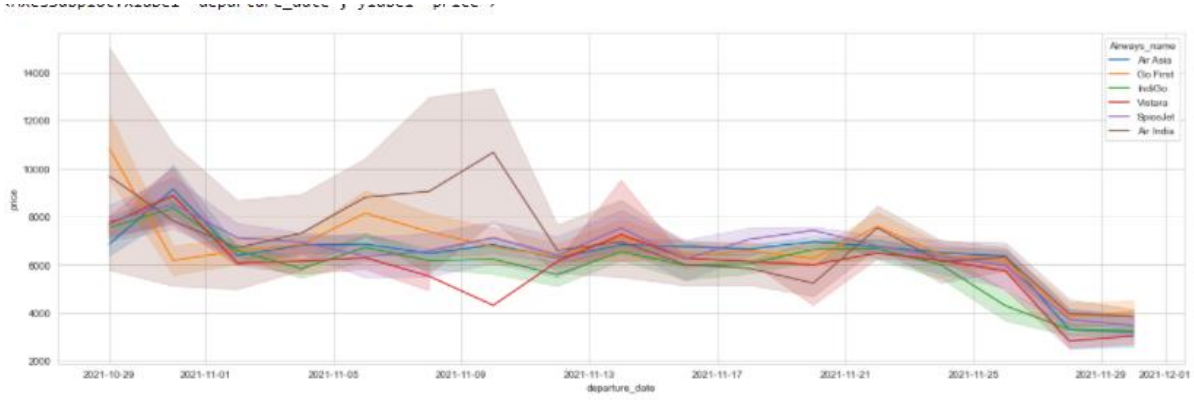
This data set contains data of 6 airways as shown below and IndiGo and Go First is high in the given data set



1 stop flight are high in this data set, i.e., 1781, non stop flights are nearly 1316, 2 stops flights are 73, and 1 3 stops flight

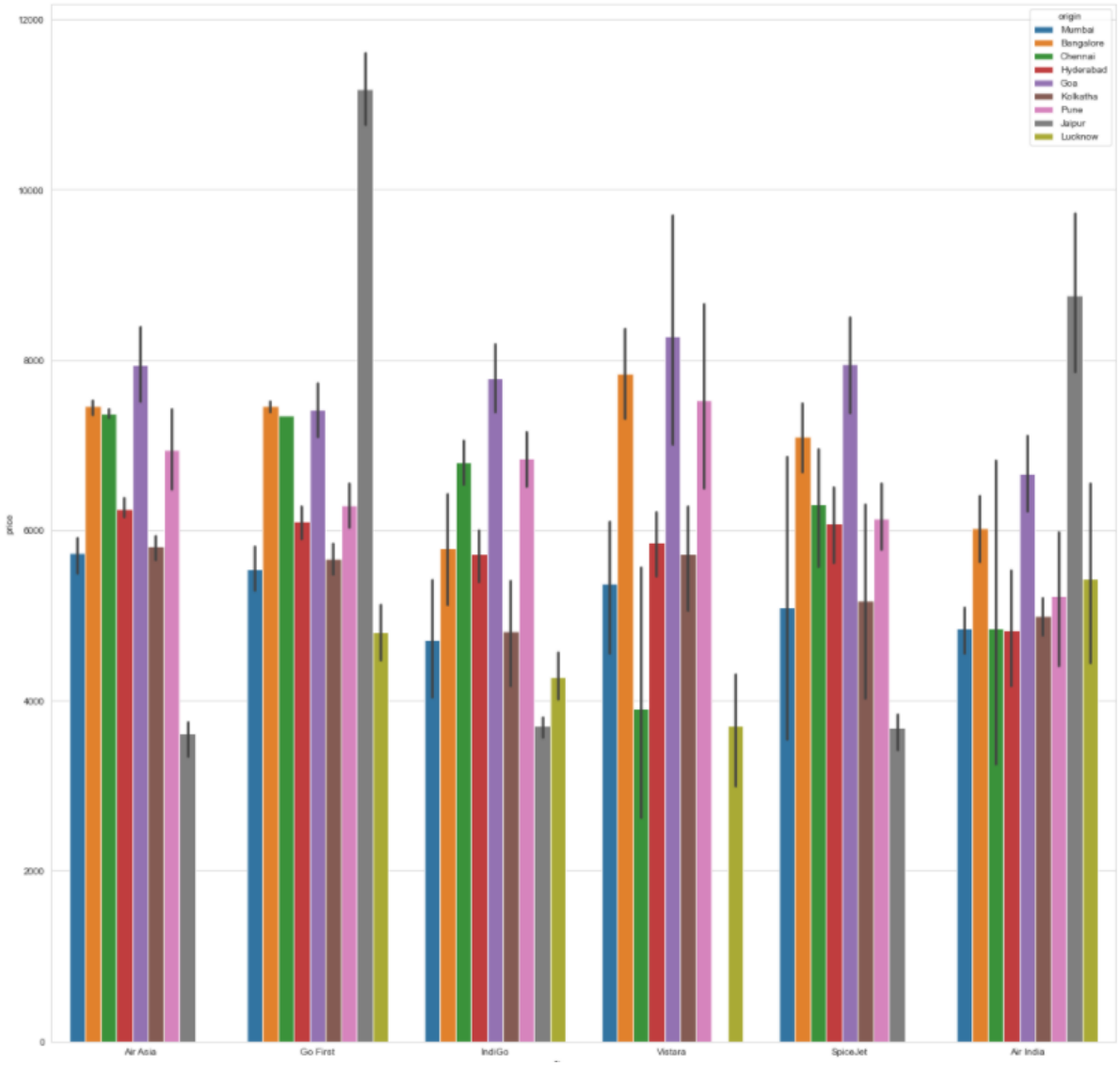


light ticket prices are not frequently changing, but the flight ticket prices are at high before between 3 to 7 days from departure date in this data set and flight ticket price is at its cheapest almost a month before the departure date as shown above.

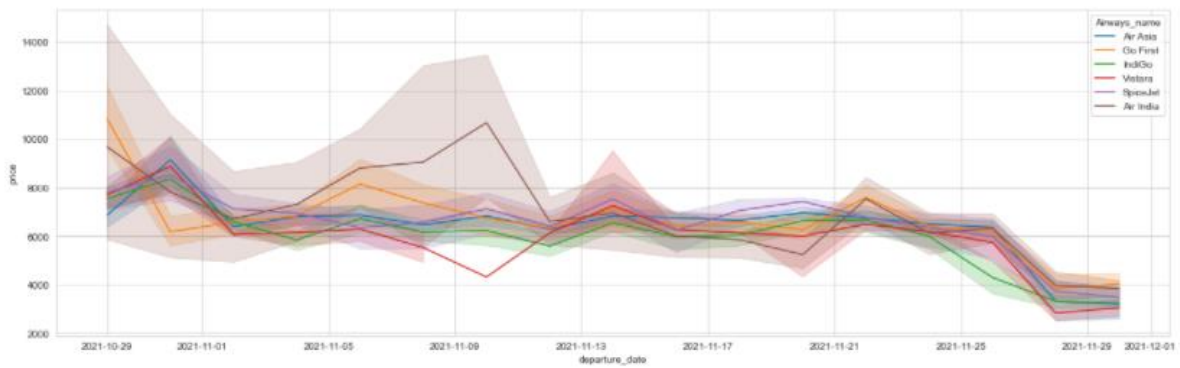


in this data set,

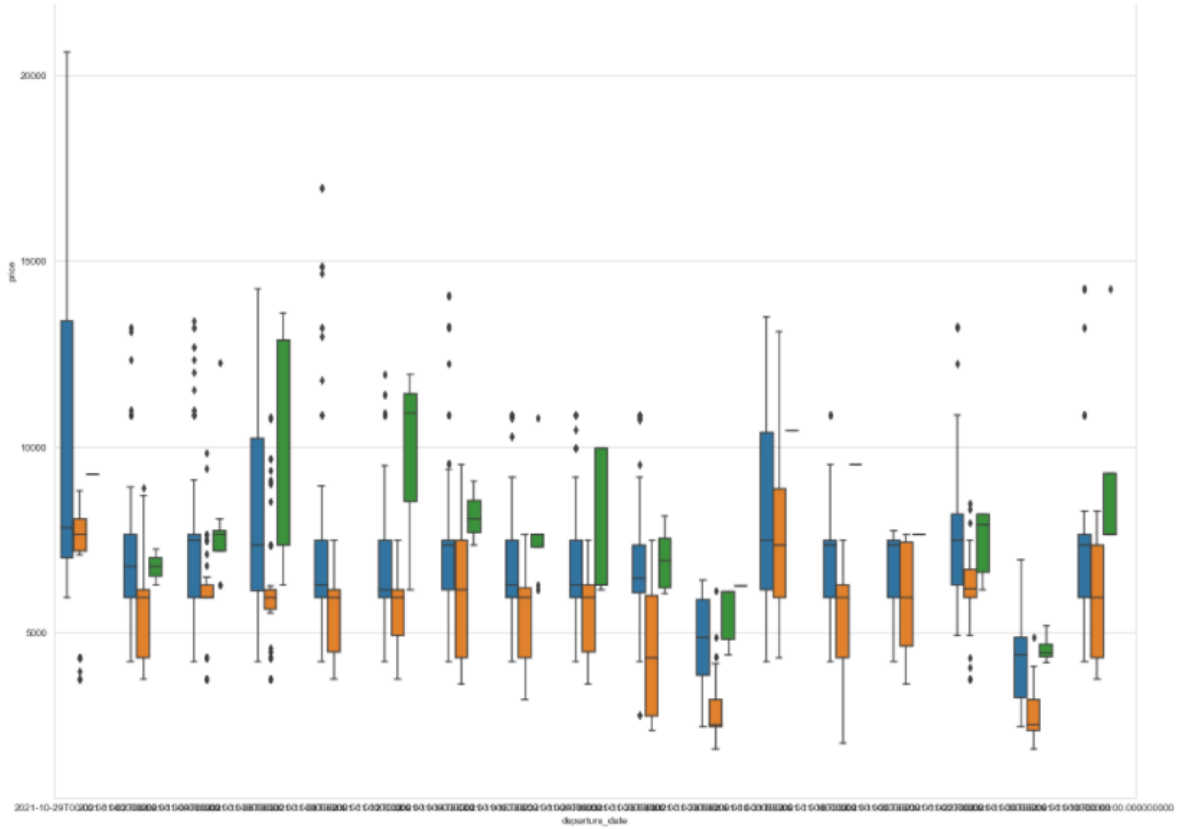
1. Indigo airways charge lessesbetween Delhi to Mumbai
2. Indigo airways charge lessesbetween Delhi to Bangalore
3. Vistara airways charge lessesbetween Delhi to Chennai
4. Airindia airways charge lessesbetween Delhi to Hyderabad
5. Airindia airways charge lessesbetween Delhi to Goa
6. Indigo airways charge lessesbetween Delhi to Kolkatha
7. Airindia airways charge lessesbetween Delhi to Pune
8. Air Asia airways charge lessesbetween Delhi to Jaipur
9. Vistara airways charge lessesbetween Delhi to Lucknow



Vistara is cheaper in this given data set, followed by Indigo, Air asia, Spice jet, Air india and go first



Non stop flights are cheaper than 1 or more stop flights in this data set



Evening flights price are cheaper in this data set an after noon flights price are higher

- **Interpretation of the Results**
    A. Tickets that are booked a month before the departure date cost low comparatively.
    B. 1 stop fights are cheaper
    C. Evening flights are cheaper
    D. Vitara airways is the cheapest airways

CONCLUSION

- **Key Findings and Conclusions of the Study**
  A. User should book their ticket a month before the departure date
  B. User should plan for evening flights to travel at low cost
  C. User should choose, Vistara, Indigo flight to travel at low cost

- **Learning Outcomes of the Study in respect of Data Science**

  I had a chance to do an end to end project starts from data scrapping based on requirement and develop a model which can be deployed in different platforms.

  I had a chance to learn python techniques and advance excel techniques to clean a data I scrapped

- **Limitations of this work and Scope for Future Work**

We collected prices for flights having departure date from 28-aug-2021 to 31-Nov. This implies that we collected data with max 31 days to departure. Our model is restricted to predict a maximum of 30 days to departure and thus we collected data for maximum of 31 days. This made sure that we had enough data for 31 days of departure.

We need to collect data for atleast 9 months and develop this model to build s stable model which can be used by most of the end user as many of us plan for travel maximum 6 month before.