FLIP ROBO

Flight Price Prediction

Submitted by:

Aneesha B Soman

ACKNOWLEDGMENT

The success and final outcome of this project required a lot of guidance and assistance from Keshav Bansal Sir and I am Extremely fortunate to have got this all along the completion of my project work.I am also grateful to Fliprobo Company for assigning this project to me.

Various references were used like:

Anlyticsvidhya, Medium,Data trained Reference materials and Github which helped me in completion of the project

Research papers referred to:

- Predicting Flight Prices in India by Achyut Joshi
- O. Etzioni, R. Tuchinda, C. A. Knoblock, and A. Yates. To buy or not to buy: mining airfare data tominimize ticket purchase price
- Manolis Papadakis. Predicting Airfare Prices.
- Groves and Gini, 2011. A Regression Model For Predicting Optimal Purchase Timing For Airline Tickets.
- Modeling of United States Airline Fares – Using the Official Airline Guide (OAG) and Airline Origin and Destination Survey (DB1B), Krishna Rama-Murthy, 2006.

INTRODUCTION

Business Problem Framing

Optimal timing for airline ticket purchasing from the consumer's perspective is challenging principally because buyers have insufficient information for reasoning about future price movements. In this project we majorly targeted to uncover underlying trends of flight prices in India using historical data and also to suggest the best time to buy a flight ticket.

<u>The project addresses the following central questions:</u>

- ❑ Frequency of change in Air flight prices
- ❑ Movement of fare is in small increments or in large jumps
- ❑ Movement goes up or down over time
- ❑ the best time to buy so that the consumer can save the most by taking the least risk?
- ❑ Relation of booking date to departure date
- ❑ Is Indigo cheaper than Jet Airways
- ❑ Are morning flights expensive

Conceptual Background of the Domain Problem

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. Airlines use using sophisticated quasi-academic tactics known as "**revenue management**" or "**yield management**". The cheapest available ticket for a given date gets more or 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, if we could inform the Early purchase travellers with the optimal time to buy their flight tickets based on the historic data and also show them various trends in the airline industry we could help them save money on their travels. This would be a practical implementation of a data analysis, statistics and machine learning techniques to solve a daily problem faced by travellers.

The objectives of the project can broadly be laid down by the following questions :

Flight Trends	**Best Time To Buy**	**Verifying Myths**
Do airfares change frequently? Do they move in small increments or in large jumps? Do they tend to go up or down over time?	What is the best time to buy so that the consumer can save the most by taking the least risk? So should a passenger wait to buy his ticket, or should he buy as early as possible?	Does price increase as we get near to departure date? Is Indigo cheaper than Jet Airways? Are morning flights expensive?

Review of Literature

Type of flight ticket purchasers

early purchasers

generally can wait some time to find the best deal on a flight, but often will simply buy a relatively affordable ticket, since predicting when the lowest price point is can be too difficult.

Last-minute purchasers

often pay full price for a ticket and do not have the flexibility of waiting for cheaper deals. As a result, prices will tend to spike radically within a few days of a flight, since airlines know some consumers have no other option

Factors affecting Price fluctuations:

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)

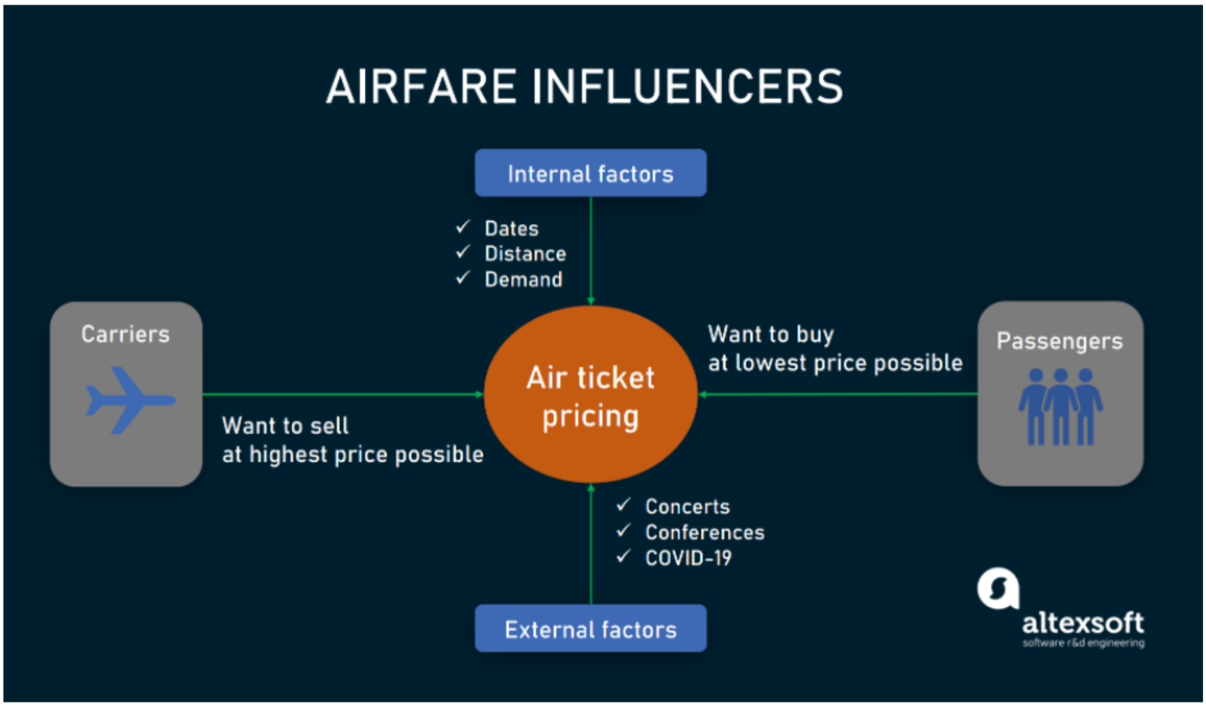
Destination Popularity

Airlines always are trying to maximize their profit based on the forecast demand for a destination

---

Flight Closures

Economic realities, airline mergers and global events can sometimes cause aircraft to be removed from service. When this happens, overall capacity for a route is reduced, leaving fewer seats to be filled. Airlines will thus suspect that flights will be fuller and will increase ticket prices.



<u>Previous work in the AI community on the problem of predicting product prices over time has been :</u>

1.**Trading Agent Competition (TAC)** :In 2002, TAC focused on the travel domain. TAC relies on a simulator of airline, hotel, and ticket prices and the competitors build agents to bid on these. The problem is different from ours since the competition works as an auction (similar to Price line.com). Whereas we gathered actual flight price data from the web, TAC simulates flight prices using a stochastic process that follows a random walk with an increasingly upward bias. Also, the TAC auction of airline tickets assumes that the supply of airline tickets is unlimited. Several TAC competitors have explored a range of methods for price prediction including historical averaging, neural nets, and boosting. It is difficult to know how these methods would perform if reconfigured for our price mining task.

2. There has been some previous work on building prediction models for airfare prices using Machine Learning techniques The various research groups have focused on mostly different sets of features and trained their models on different kinds of flights.

A major distinction among these projects is the specific trend they are trying to predict. Specifically, we can categorize projects into 2 approaches:

- **studying the factors that influence the average price of a flight**
- **those factors that influence the price of a specific flight in the days leading up to departure**

Our project will focus on the second part, that is the factors influencing the price of specific flight in the days leading upto departure.

3. Flight Price prediction over time have been studied in statistics under the heading of “time series analysis” and in computational finance under the heading of “optimal stopping problems”.

Computational finance is concerned with **predicting prices and making buying decisions in markets for stock, options, and commodities**. Prices in such markets are **not determined by a hidden algorithm**, as in the product pricing case, but **rather by supply and demand** as determined by the actions of a large number of buyers and sellers. Thus, for example, stock prices tend to move in small incremental steps rather than in the large, tiered jumps observed in the airline data

Time series analysis is a large body of statistical techniques that apply to a sequence of values of a variable that varies over time due to some underlying process or structure .The observations of product prices over time are naturally viewed as time series data. Standard data mining techniques are “trained” on a set of data to produce a predictive model based on that data, which is then tested on a separate set of test data. In contrast, time series techniques would attempt to predict the value of a variable based on its own history. For example, our moving average model attempts to predict the future changes in the price of a ticket on a flight from that flight’s own price history.

4**. Bing Travel’s “Fare Predictor**” and **AirHint Travels Air Predictor** are some of the current problems

5. As per **William Groves and Maria Gini Paper,**Generating a feature set hierarchy requires some domain knowledge, but does not require expert level understanding. The inclusion of lagged features in the model captures temporal relationships among feature and improves the predictions. By examining the best lag schemes, domain knowledge can be extracted: the significance of individual features can be discovered by observing their presence in the scheme.

6**. Manolis Papadakis** developed a model where Given a flight that the user is interested in booking, it runs the predictor on said flight, and, based on the result, advises the user to either buy right away ("buy"), or wait for a predicted future drop in price ("wait"), perhaps with an accompanying measure of confidence.

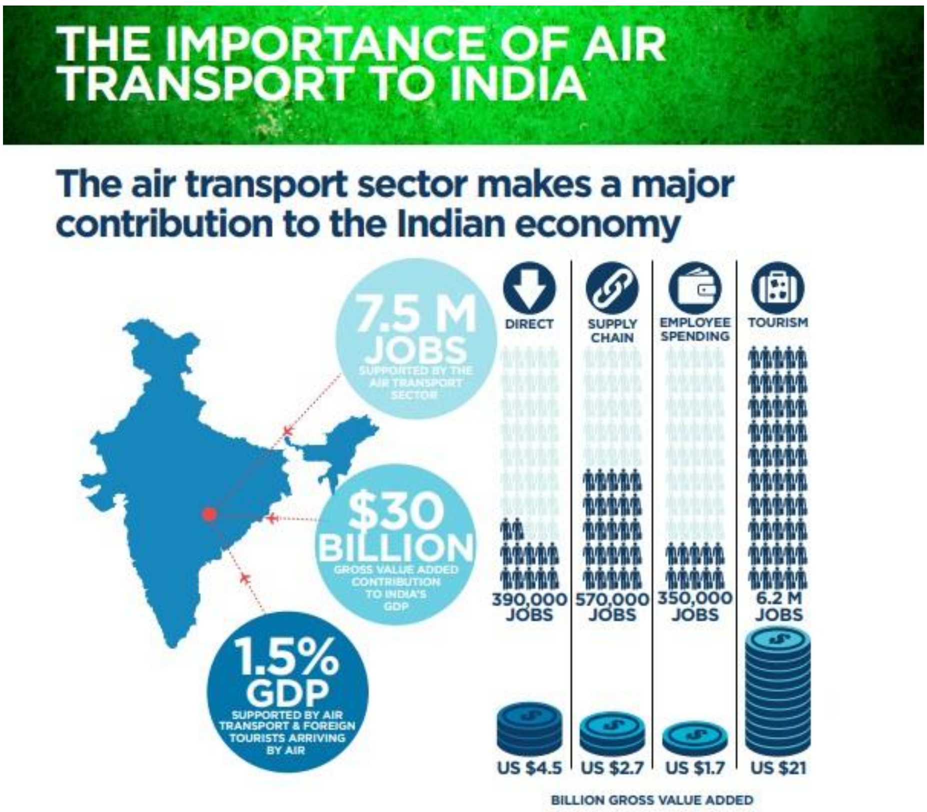
7. Oren Etzioni and Craig A. Knoblock ,Hamlet data mining method achieved 61.8% of the possible savings by appropriately timing ticket purchases. They have used statistics (time series methods), computational finance (reinforcement learning) and classical machine learning (Ripper rule learning). Each algorithm was tailored to the problem at hand (e.g., we devised an appropriate reward function for reinforcement learning), and the algorithms were combined using a variant of stacking to improve their predictive accuracy of the Flight fare

Motivation for the Problem Undertaken

1.As product prices become increasingly available on the World Wide Web, consumers attempt to understand how corporations vary these prices over time. However, corporations change prices based on proprietary algorithms and hidden variables (e.g., the number of unsold seats on a flight). But by building a model which can predict prices which does not need the information about internal flight arena and only needs Data mining would facilitate Price prediction of Fares

2.This is particularly useful for Middle class consumers who need to travel long distance and would like to know the time to take the ticket so as to obtain the Least Fare Price.

3.If the passengers have better means of predicting the price,it will directly result in the growth of the Aviation sector of India.
This will further impact the Economy of India

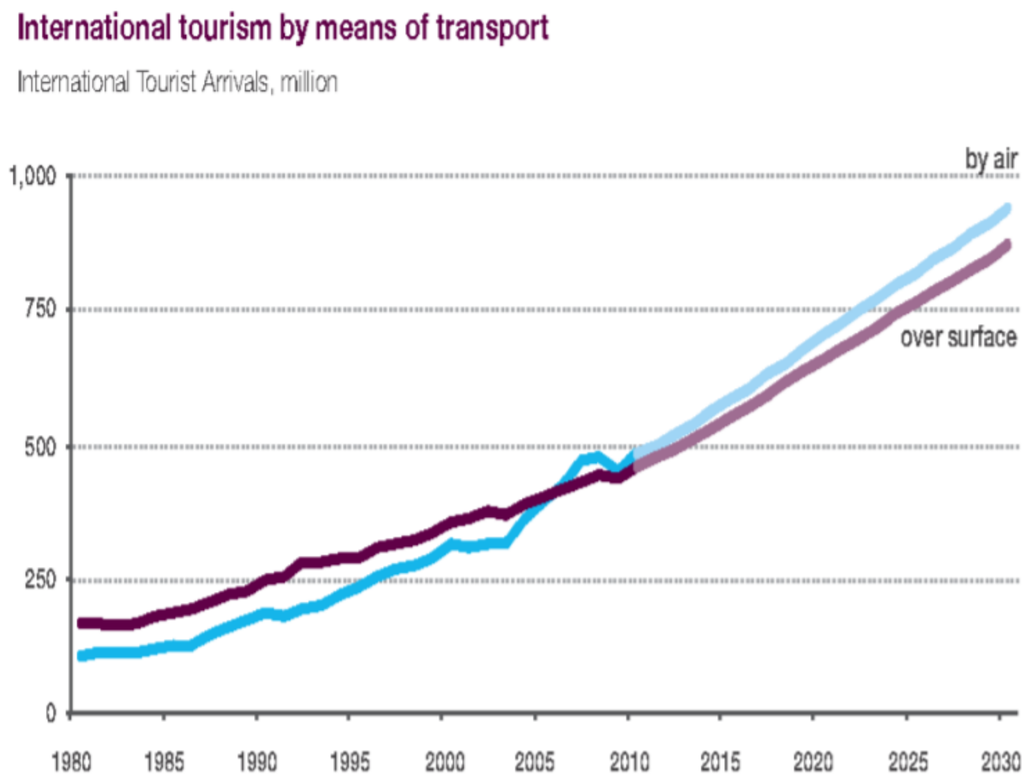


4.A model created to understand the working of the Fare of Flights will also assist Tourism Industry in India, hence building the GDP of the country.

<u>a.Internal toursim</u>

> The “Incredible India Campaign” helped in establishing the country as a highend destination, leading to a 16% increase in tourist traffic in 2002, its first year of Campaign

<u>b.External tourism</u>:



Indian government has been initiating various schemes to build aviation tourism and a model which can enhance the sector is in the right direction

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

a.**Statistical models used:**

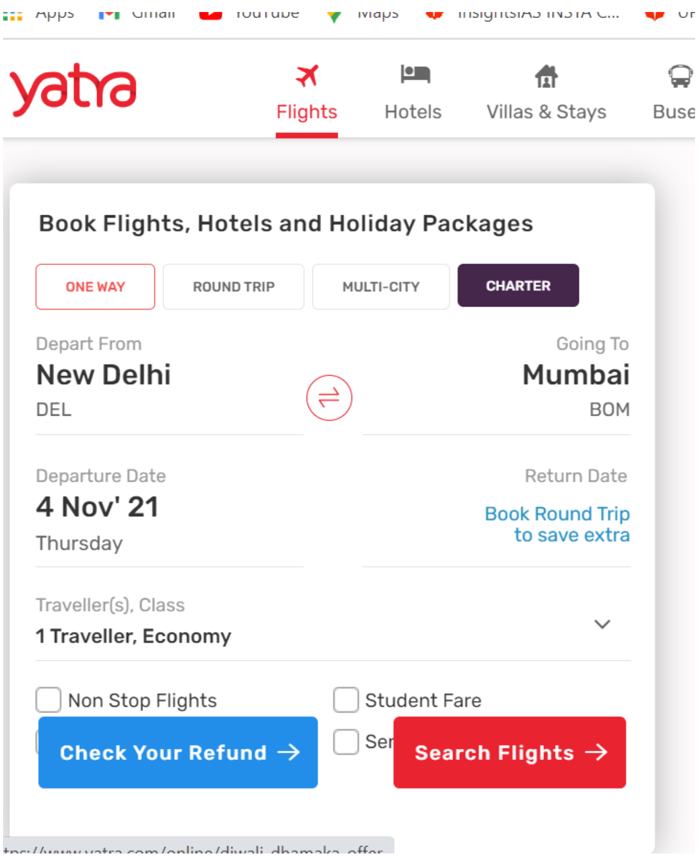
- pipeline for applying regression
- Ordinal encoding

b. **b.Analytical models:**

Descriptive Analytics	**Diagnostic Analytics**	**Predictive Analytics**	**Prescriptive Analytics**
Data visualization done through matplotlib and seaborn between Features and label.Also heatmap, feature,Correlation between feature and label graph and feature importance graph is taken.	The reason for change is understood to be filling up of seats, keeping seats for the end days(last minute customers) and Competitive pricing	Prediction is done through various Regression techniques	Prescriptive analysis is done through the Model created.

Data Sources and their formats

To undertake the experiments we have Mined Data from yata.com between New Delhi and Mumbai using Selenium.




```
Price=[]
Dateday=[]
Flightname=[]
Sourceplace=[]
SourcePlace=[]
Arrivaltimedestination=[]
DestinationPlace=[]
ArrivalTime=[]
Timetaken=[]
Stops=[]
Stop_list=[]
Sourcetime=[]



price=driver.find_elements_by_xpath("//div[@class='i-b tipsy fare-summary-tooltip fs-18']")
for i in price:
    try:
        Price.append(i.text)
    except NoSuchElementException:
        Price.append("--")


dateday=driver.find_elements_by_xpath("//div[@class='day-li text-center cursor-pointer pr active font-primary-color']//p")
for i in dateday:
    try:
        Dateday.append(i.text)
    except NoSuchElementException:
        Dateday.append("--")


flightname=driver.find_elements_by_xpath("//span[@class='i-b text ellipsis']")
for i in flightname:
    try:
        Flightname.append(i.text)
    except NoSuchElementException:
        Flightname.append("--")
```

```python
#sourceplace
sourceplace=driver.find_elements_by_xpath("//div[@class='i-b col-4 no-wrap text-right dtime col-3']//p")
for i in sourceplace:
    try:
        Sourceplace.append(i.text)
    except NoSuchElementException:
        Sourceplace.append("--")

#finding second elements
for i in range(1,len(Sourceplace),2):
    SourcePlace.append(Sourceplace[i])


arrivaltimedestination=driver.find_elements_by_xpath("//div[@class='i-b pdd-0 text-left atime col-5']//p")
for i in arrivaltimedestination:
    try:
        Arrivaltimedestination.append(i.text)
    except NoSuchElementException:
        Arrivaltimedestination.append("--")

#finding second elements
for i in range(1,len(Arrivaltimedestination),2):
    DestinationPlace.append(Arrivaltimedestination[i])

for i in range(0,len(Arrivaltimedestination),2):
    ArrivalTime.append(Arrivaltimedestination[i])

#timetaken
timetaken=driver.find_elements_by_xpath("//p[@class='fs-12 bold du mb-2']")
for i in timetaken:
    try:
        Timetaken.append(i.text)
    except NoSuchElementException:
        Timetaken.append("--")
```

```python
#stops
stops=driver.find_elements_by_xpath("//div[@class='stop-cont pl-13']")
for i in stops:
    try:
        Stops.append(i.text)
    except NoSuchElementException:
        Stops.append("--")

#finding number of stops
for i in range(len(Stops)):
            a,b = Stops[i].split("\n")
            Stop_list.append(b)

#source time
sourcetime=driver.find_elements_by_xpath("//div[@class='i-b pr']")
for i in sourcetime:
    try:
        Sourcetime.append(i.text)
    except NoSuchElementException:
        Sourcetime.append("--")
print("done for 1st november")
```

```python
#finding date and day
Dateday=[]
Datedayprice=[]
Date=[]
day=[]

datedayprice=driver.find_elements_by_xpath("//div[@class='day-li text-center cursor-pointer pr active font-primary-color']//
for i in datedayprice:
    try:
        Datedayprice.append(i.text)
    except NoSuchElementException:
        Datedayprice.append("--")

#finding second elements
for i in range(0,len(Datedayprice),2):
    Dateday.append(Datedayprice[i])


#split Dateday into date and day
datedaylist=Dateday[0].split(',')
day=datedaylist[0]
date=datedaylist[1]

Date1=[]
day1=[]

a=len(Price)
for k in range(a):
    Date1.append(date)
    day1.append(day)


df1=pd.DataFrame({'Price':Price,
                'Flightname':Flightname,
                 'Sourceplace':SourcePlace,
                'DestinationPlace':DestinationPlace,
                 'ArrivalTime':ArrivalTime,
                'Stops':Stop_list,
                'Sourcetime':Sourcetime,
                  'Date':Date1,
                   'day':day1
                })
df1
```

The features are:

Flightname	2385	non-null	object	Name of the flight which is being studied
Sourceplace	2385	non-null	object	Source region-New Delhi
DestinationPlace	2385	non-null	object	Destination-Mumbai
day	2385	non-null	object	The day of the flight
Date number	2385	non-null	int64	The date of the flight in the particular month
Date Month	2385	non-null	object	Month in which the particular flight is scheduled for departure
numberofstops	2385	non-null	object	The number of stops taken by the flight
DepartureHour	2385	non-null	int64	Hour of flights departure in the day
DepartureMin	2385	non-null	int64	Minute of flights departure in the day
extraday	2385	non-null	object	Travelling in arrival has extra day
Arrival_hours	2385	non-null	int64	Hour of flights arrival in the day
Arrival_minutes	2385	non-null	int64	Minute of flights arrival in the day

The **label is Prices** in object format

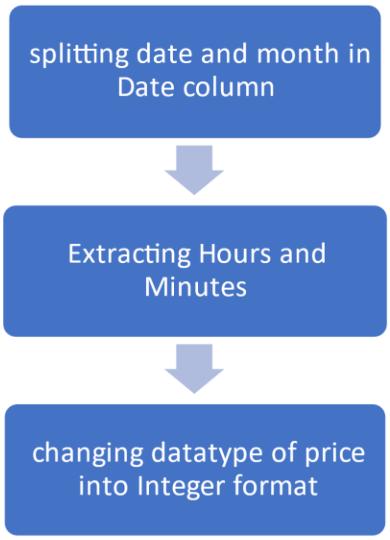
Format for data:excel

Train data

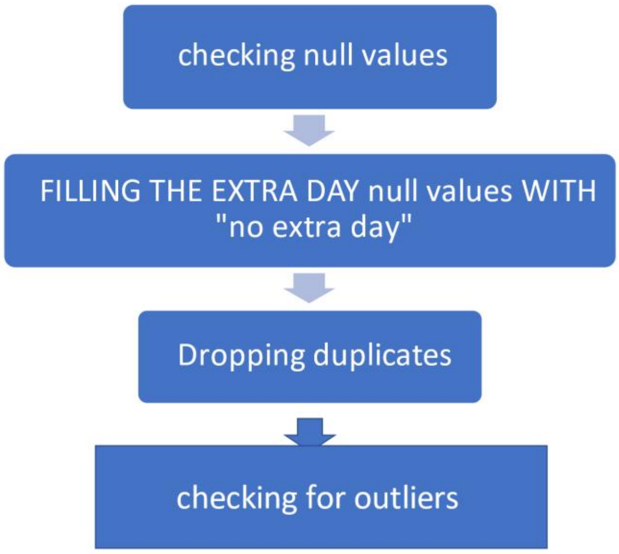
	A	B	C	D	E	F	G	H	I
1	Price	Flightname	ourceplac	stinationPl	ArrivalTime	Stops	Sourcetime	Date	day
2	5,953	Air Asia	New Delhi	Mumbai	06:45+ 1 d	1 Stop	21:25	1 Nov	Mon
3	5,953	Air Asia	New Delhi	Mumbai	07:15+ 1 d	1 Stop	21:25	1 Nov	Mon
4	5,953	Air Asia	New Delhi	Mumbai	06:45+ 1 d	1 Stop	18:35	1 Nov	Mon
5	5,953	Air Asia	New Delhi	Mumbai	07:15+ 1 d	1 Stop	18:35	1 Nov	Mon
6	5,953	Air Asia	New Delhi	Mumbai	12:25+ 1 d	1 Stop	22:10	1 Nov	Mon
7	5,954	Go First	New Delhi	Mumbai	09:10	Non Stop	07:00	1 Nov	Mon
8	5,954	Go First	New Delhi	Mumbai	10:10	Non Stop	08:00	1 Nov	Mon
9	5,954	Go First	New Delhi	Mumbai	00:40+ 1 d	Non Stop	22:30	1 Nov	Mon
10	5,954	Go First	New Delhi	Mumbai	04:15	Non Stop	02:00	1 Nov	Mon
11	5,954	Go First	New Delhi	Mumbai	16:35	Non Stop	14:20	1 Nov	Mon
12	5,954	Go First	New Delhi	Mumbai	17:15	Non Stop	15:00	1 Nov	Mon
13	5,954	Go First	New Delhi	Mumbai	21:55	Non Stop	19:40	1 Nov	Mon
14	5,954	Go First	New Delhi	Mumbai	23:15	Non Stop	21:00	1 Nov	Mon
15	5,954	Go First	New Delhi	Mumbai	12:50	Non Stop	10:30	1 Nov	Mon

Data Preprocessing Done

Feature engineering

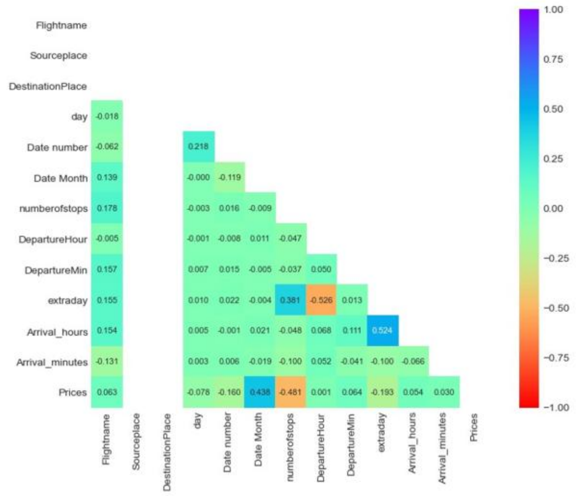


Exploratory data Analysis

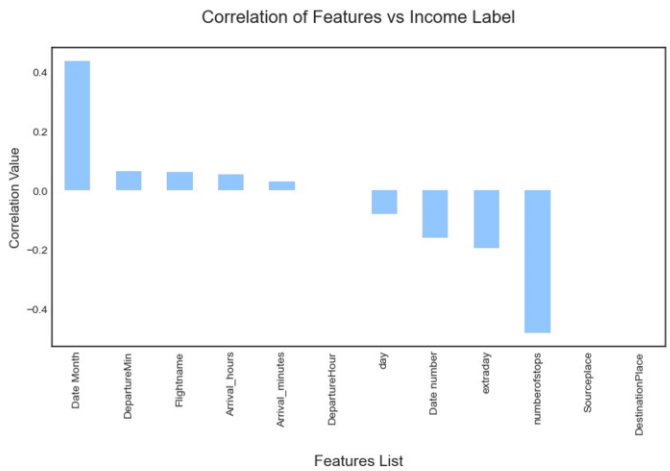


Data Inputs- Logic- Output Relationships

Heat map



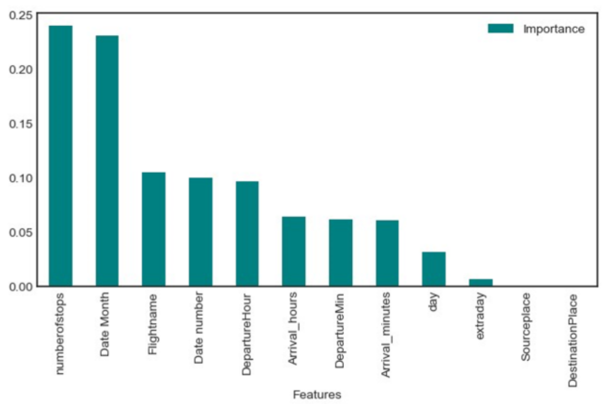
there are no multi collinearity concerns in our dataset



we can see that columns**Date Month,DepartureminFlightname,arrival_hours and arrival_minutes** are positively correlated with our target.

Date Number,extra day and numberofstops are negatively correlated where numberofstopsis highly negatively correlated indicating that as the number of total stops in an itinerary increases the price of that particular flight increases and vice a versa

Feature Importance



Features	Importance
numberofstops	0.240
Date Month	0.231
Flightname	0.105
Date number	0.100
DepartureHour	0.097
Arrival_hours	0.064
DepartureMin	0.062
Arrival_minutes	0.061
day	0.032
extraday	0.007
Sourceplace	0.000
DestinationPlace	0.000

weightage in predicting our labe.

The largest relationship with the label can be seen with the numbersbystop. Then Datemonth also influences the label highly.

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

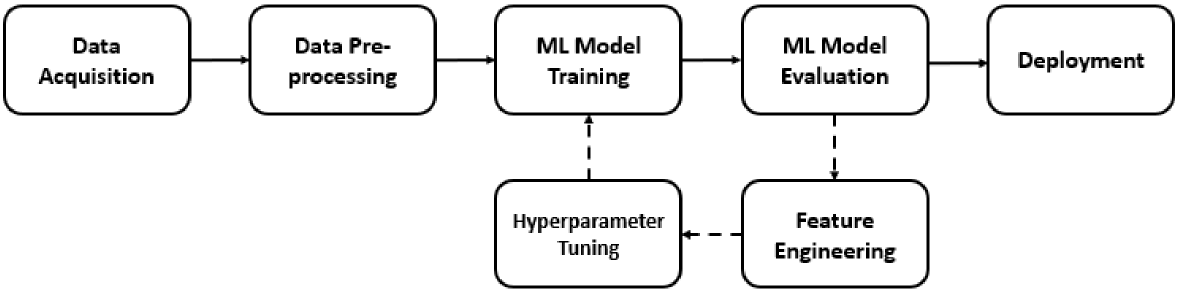
- ❖ Competitive pricing is not considered
- ❖ External factors such as festivals is not considered

Hardware and Software Requirements and Tools Used

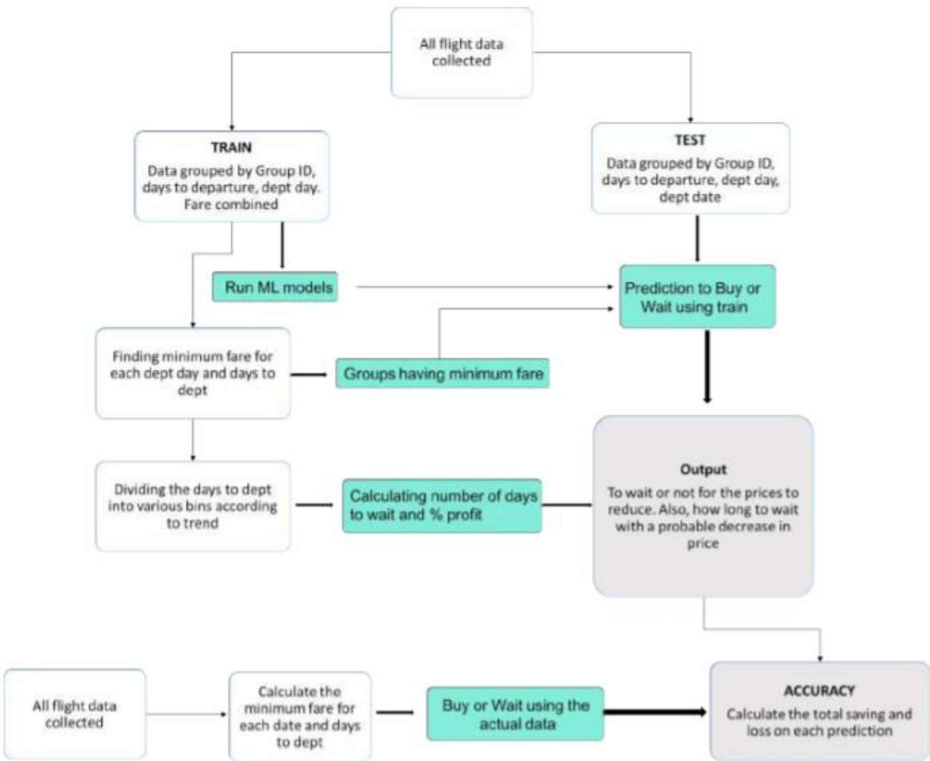
- Hardware technology being used.
  - RAM : 8 GB
  - CPU :Intel® Core™ i7-10510U CPU @ 1.80GHz
- Software technology being used.
  - Programming language : Python
  - Distribution : Anaconda Navigator
  - Browser based language shell : Jupyter Notebook
- Libraries/Packages specifically being used.
  - Pandas , NumPy, matplotlib, seaborn, scikit-learn, pandas-profiling, missingno

Model/s Development and Evaluation

Identification of possible problem-solving approaches



The problem at Hand is a regression problem.



Identified Approaches (Algorithms) for Regression

Run and Evaluate selected models using key metrics

```
# Regression Model Function

#splits the training and testing features and labels
#then trains the model
#predicts the label,
#calculates the RMSE score
#generates the R2 score
#calculates the Cross Validation score
#finds the difference between the R2 score and Cross Validation score.

def reg(model, X, Y):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=754)

    # Training the model
    model.fit(X_train, Y_train)

    # Predicting Y_test
    pred = model.predict(X_test)

    # RMSE - a lower RMSE score is better than a higher one
    rmse = mean_squared_error(Y_test, pred, squared=False)
    print("RMSE Score is:", rmse)

    # R2 score
    r2 = r2_score(Y_test, pred, multioutput='variance_weighted')*100
    print("R2 Score is:", r2)

    # Cross Validation Score
    cv_score = (cross_val_score(model, X, Y, cv=5).mean())*100
    print("Cross Validation Score:", cv_score)

    # Result of r2 score minus cv score
    result = r2 - cv_score
    print("R2 Score - Cross Validation Score is", result)
```

A common function is created to evaluate all the algorithms

Linear Regression Model	`# Linear Regression Model`
	`model=LinearRegression()`
	`reg(model, X, Y)`
		RMSE Score is: 2438.789058288971
		R2 Score is: 43.9997075050449
		Cross Validation Score: 3.758363160299494
		R2 Score - Cross Validation Score is 40.24134434474541

Ridge Regression	`# Ridge Regression`
	`model=Ridge(alpha=1e-2, normalize=True)`
	`reg(model, X, Y)`
		RMSE Score is: 2438.8813326146405
		R2 Score is: 43.99546975683402
		Cross Validation Score: 4.055024524355588
		R2 Score - Cross Validation Score is 39.94044523247843

Lasso Regression	`# Lasso Regression`
	`model=Lasso(alpha=1e-2, normalize=True, max_iter=1e5)`
	`reg(model, X, Y)`
		RMSE Score is: 2438.7034014638693
		R2 Score is: 44.00364119746918
		Cross Validation Score:

		3.767193873255572 R2 Score - Cross Validation Score is 40.23644732421361

Support Vector Regression	`# Support Vector Regression`
	`model=SVR(C=1.0, epsilon=0.2, kernel='poly', gamma='auto')`
	`reg(model, X, Y)`
		RMSE Score is: 3298.764391434288
		R2 Score is: -2.457655877110043
		Cross Validation Score: -40.758774675142185
		R2 Score - Cross Validation Score is 38.30111879803214

Decision Tree Regressor	`# Decision Tree Regressor`
	`model=DecisionTreeRegressor(criterion="poisson", random_state=111)`
	`reg(model, X, Y)`
		RMSE Score is: 2979.8182661862934
		R2 Score is: 16.397093594345048
		Cross Validation Score: -290.4215139672732
		R2 Score - Cross Validation Score is 306.81860756161825

Random Forest Regressor	`# Random Forest Regressor`
	`model=RandomForestRegressor(max_depth=2, max_features="sqrt")`
	`reg(model, X, Y)`
		RMSE Score is: 2661.3949815871515
		R2 Score is: 33.310035402996064
		Cross Validation Score: -4.917191242714118
		R2 Score - Cross Validation Score is 38.22722664571018

K Neighbors Regressor	`# K Neighbors Regressor`
	`KNeighborsRegressor(n_neighbors=2, algorithm='kd_tree')`
	`reg(model, X, Y)`
		RMSE Score is: 2672.971706735006
		R2 Score is: 32.72858803600812
		Cross Validation Score: -4.760782472737028
		R2 Score - Cross Validation Score is 37.48937050874515

Gradient Boosting Regressor	`# Gradient Boosting Regressor`
	`model=GradientBoostingRegressor(loss='quantile', n_estimators=200, max_depth:`
	`reg(model, X, Y)`
		RMSE Score is: 2264.631157663613
		R2 Score is: 51.712271524712335
		Cross Validation Score: -60.67370974525765
		R2 Score - Cross Validation Score is 112.38598126996999

Ada Boost Regressor	`# Ada Boost Regressor`
	`model=AdaBoostRegressor(n_estimators=300, learning_rate=1.05, random_state=42)`
	`reg(model, X, Y)`
		RMSE Score is: 2275.577827747495
		R2 Score is: 51.244321264861334
		Cross Validation Score: 14.595684912708418
		R2 Score - Cross Validation Score is 36.64863635215292

Extra Trees Regressor	`# Extra Trees Regressor`
	`model=ExtraTreesRegressor(n_estimators=200, max_features='sqrt', n_jobs=6)`
	`reg(model, X, Y)`
		RMSE Score is: 1524.3175510392498
		R2 Score is: 78.1227504931184
		Cross Validation Score:

		30.02822513189347 R2 Score - Cross Validation Score is 48.09452536122493

XGB Regressor	`# XGB Regressor`
	`model=XGBRegressor()`
	`reg(model, X, Y)`
		RMSE Score is:
		1345.1846589979662
		R2 Score is: 82.96250932122629
		Cross Validation Score:
		22.32671846156723
		R2 Score - Cross Validation Score is 60.63579085965907

LGBM Regressor	`# LGBM Regressor`
	`model=LGBMRegressor()`
	`reg(model, X, Y)`
		RMSE Score is:
		1325.1746345021627
		R2 Score is: 83.46561490681431
		Cross Validation Score:
		44.63585914866849
		R2 Score - Cross Validation Score is 38.82975575814582

Hyper parameter tuning

I have chosen the XGB regressor as my best model since it is able to provide me the highest R2 score plus the model is doing better in Cross validation score too. However the LGBM model is not chosen even though it has high score and low difference between r2 score and cross val score is because LGBM algorithm is better for datasets above 10,000 rows

In the below cell all the parameters for LGBM regressor that can be used for hyper tuning our final model are listed

```
# Choosing XGB Regressor

fmod_param = {'booster' : ['gbtree','dart','gblinear'],
              'importance_type' : ['gain','split'],
              'n_estimators' : [100,200,500],
              'eta' : [0.001, 0.01, 0.1]
             }
GSCV = GridSearchCV(XGBRegressor(), fmod_param, cv=5)
GSCV.fit(X_train,Y_train)
```

```
GridSearchCV(cv=5,
             estimator=XGBRegressor(base_score=None, booster=None,
                                    colsample_bylevel=None,
                                    colsample_bynode=None,
                                    colsample_bytree=None,
                                    enable_categorical=False, gamma=None,
                                    gpu_id=None, importance_type=None,
                                    interaction_constraints=None,
                                    learning_rate=None, max_delta_step=None,
                                    max_depth=None, min_child_weight=None,
                                    missing=nan, monotone_constraints=None,
                                    n_estimators=100, n_jobs=None,
                                    num_parallel_tree=None, predictor=None,
                                    random_state=None, reg_alpha=None,
                                    reg_lambda=None, scale_pos_weight=None,
                                    subsample=None, tree_method=None,
                                    validate_parameters=None, verbosity=None),
             param_grid={'booster': ['gbtree', 'dart', 'gblinear'],
                         'eta': [0.001, 0.01, 0.1],
                         'importance_type': ['gain', 'split'],
                         'n_estimators': [100, 200, 500]})
```

```
GSCV.best_params_
```

```
{'booster': 'gbtree',
 'eta': 0.1,
 'importance_type': 'gain',
 'n_estimators': 200}
```

```
Final_Model = XGBRegressor(booster='gbtree', eta=0.1, importance_type='gain', n_estimators=200)
regressor = Final_Model.fit(X_train, Y_train)
fmod_pred = Final_Model.predict(X_test)
fmod_r2 = r2_score(Y_test, fmod_pred)*100
print("R2 score for the Best Model is:", fmod_r2)
```

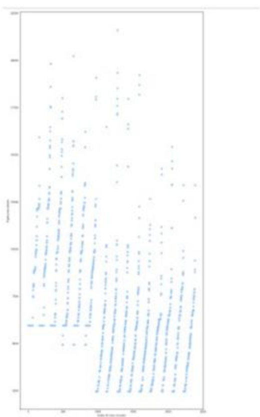
```
R2 score for the Best Model is: 82.4465367200635
```

final model is built using the hyper tuned parameters

The r2 score of the final model is 82.44

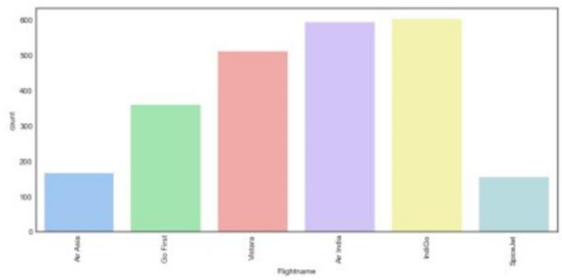
Visualizations

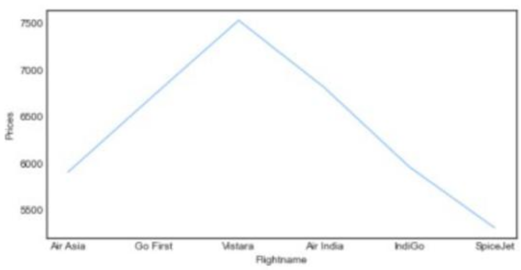
Price of the Fare



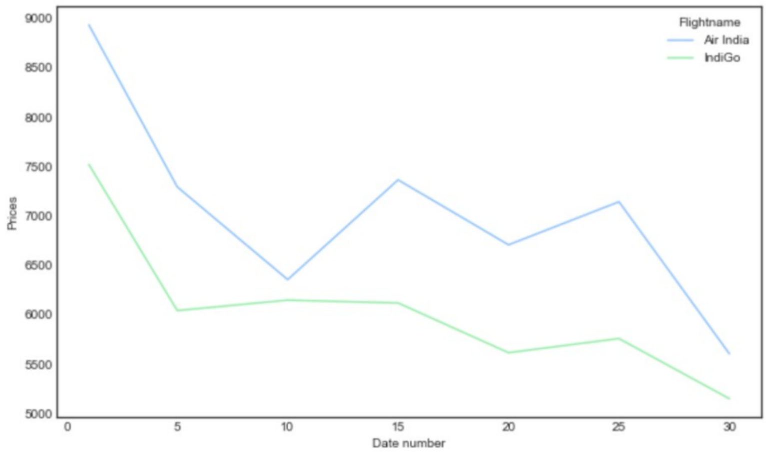
we are able to see that most of the flight price values are accumulated between 2500 and 12500 and very rare data points are distributed abov that number.

Flights



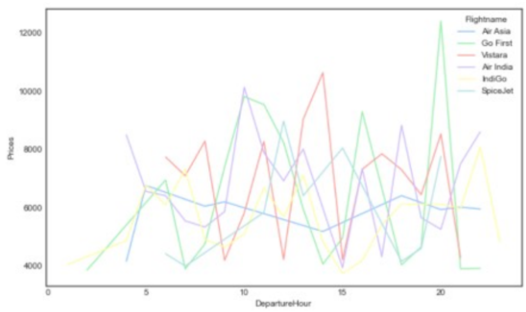


- .Maximum number of flights are of Air India and Indigo
- Vistara shows having the highest price

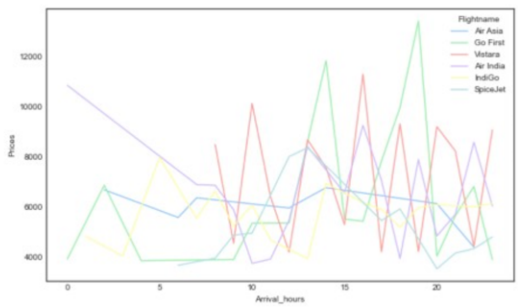


Indigo flights are cheaper than AirIndia

Departure hour of flights vs Price of Fare

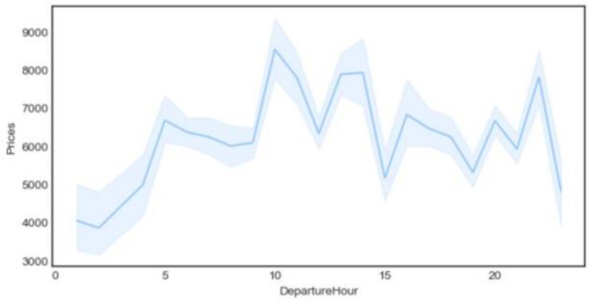


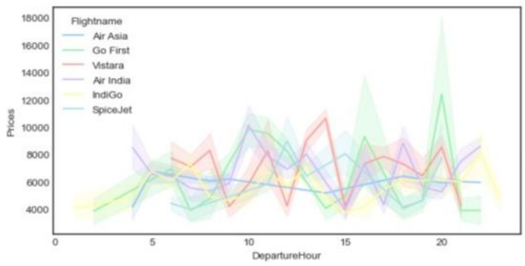
Arrival hour of flights vs Price of Fare



- Arrival and Departure at 8pm Has highest Fare Price
- 5am to 10am arrival has lowest Price

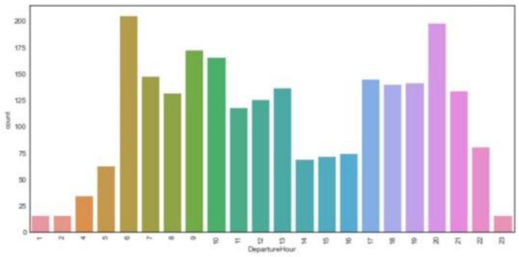
Morning vs Afternoon vs Evening





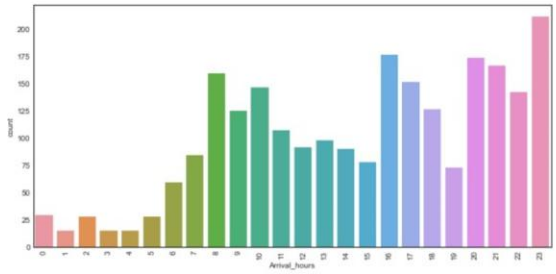
Prices are lower in the morning than
in the afternoon and evenings

Departure hour of flights



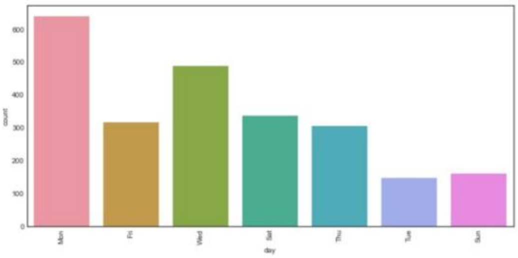
Maximum flights take off between 6am -10am and 5pm to 9pm

Arrival hour of flights

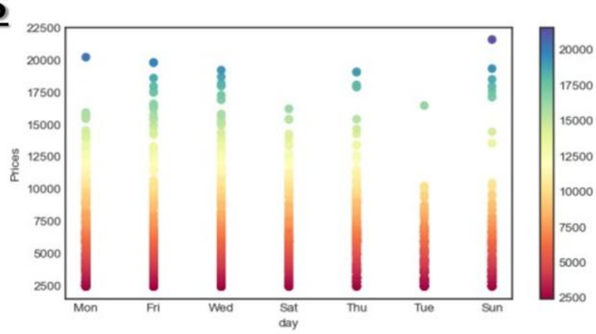


Most of the flights arrive between 8am to 10am and 4pm to 11pm

DAY OF WEEK FLIGHT DEPARTS

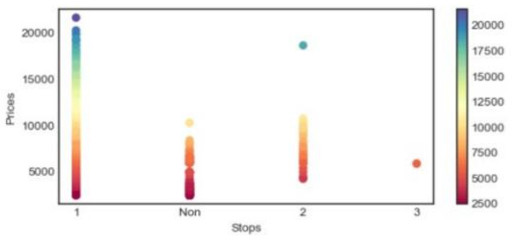


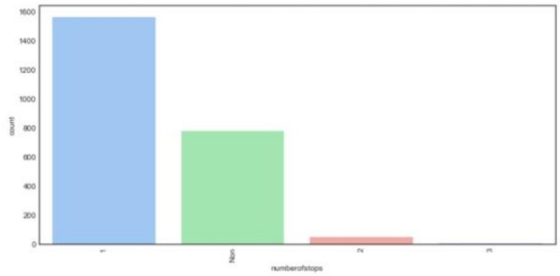
Maximum number of flights are taken on Monday

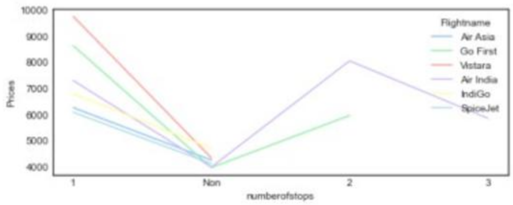


Highest prices is for Fridays,wednesdays and sundays

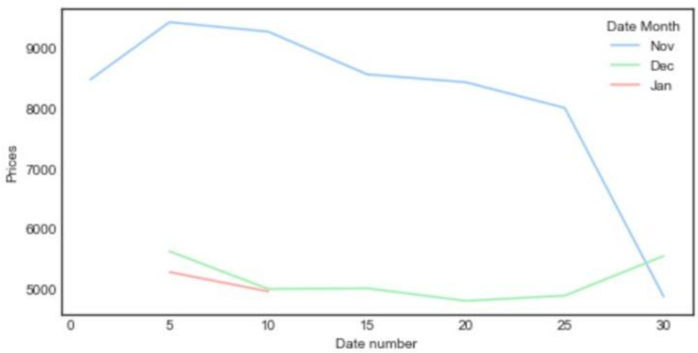
Number of stops







- ❖ 1 stop is more than no stop flights
- ❖ The maximum prices are seen for 1 stop and lesser for non stop
- ❖ minimum flights have 3 stops



- ❖Data collected are of November, December and January months.
- ❖Prices are **seen to be higher till November 25th** , the data was collected on october 30th. It shows a 30ay gap will cause the price to fall
- ❖There **shows to be a steep fall in prices after 30 days** .
- ❖Prices are seen to be decrementing in small intervals till 30days and in steep interval after 30days of booking date
- ❖Prices are also seen to be increasing as we book near to the departure date

Interpretation of the Results

- While watching the prices fluctuation with the available data mined, we can observe that from the date of booking after 30days the prices start to drop.It shows **a steep fall after 30 days.**Prices are seen to be decrementing in small intervals till 30days and in steep interval after 30days of booking date

- The difference between no stops and 1 stop is also seen to be largely different.

- The most minimum the price falls between Mumbai and Delhi **is Rs2500**

- The **Indigo flights are cheaper than AirIndia,** and the costliest Flights are Vistara. The Air India and Indigo have the highest number of flights

- Arrival and Departure at 8pm Has highest Fare Price and 5am to 10am arrival has lowest Price

- Prices are **lower in the morning** than in the afternoon and evenings

- Maximum flights take off between 6am-10am and 5pm to 9pm
  Most of the flights arrive between 8am to 10am and 4pm to 11pm

- Maximum number of flights are taken on Monday
  Highest prices is for Fridays,wednesdays and sundays

- 1 stop is more than no stop flights
  The maximum prices are seen for 1 stop and **lesser for non stop**
  minimum flights have 3 stops

CONCLUSION

Key Findings and Conclusions of the Study

The main aim is to predict the price change in Flights and how various factors influences it. Through analysis the highest factors that impact it are Stops the flight takes and the difference between the booking date and the departure date.

The XGB Booster regressor is tuned with the best parameter and the model is built with 82% r2 score which can predict the prices of the flight in the future

Learning Outcomes of the Study in respect of Data Science

Various Algorithms were used to train the data such as Linear Regression Model, Ridge Regression, Lasso Regression, Support Vector Regression, Decision Tree Regressor, Random Forest Regressor, K Neighbors Regressor, Gradient Boosting Regressor, Ada Boost Regressor, Extra Trees Regressor, XGB Regressor and LGBM Regressor

The LGBM has a higher score than XGB Regressor, but the XGB is preferred in the current dataset as LGBM is preferred for datasets above 10,000 rows and for the current dataset the problem of overfitting can arise in LGBM

Best parameters are given into the XGB model and the model is built and saved as a file which could be used by anyone to predict Flight fare with 82% accuracy.

Limitations of this work and Scope for Future Work

- The number of data that is mined is limited
- Other external factors influence such as festival, economic recession is not considered
- Competitive pricing is not considered

There is further scope to improve the project with a larger dataset and by predicting if the test data will have low price or high.

****