DSAI

India Flight Prices dataset

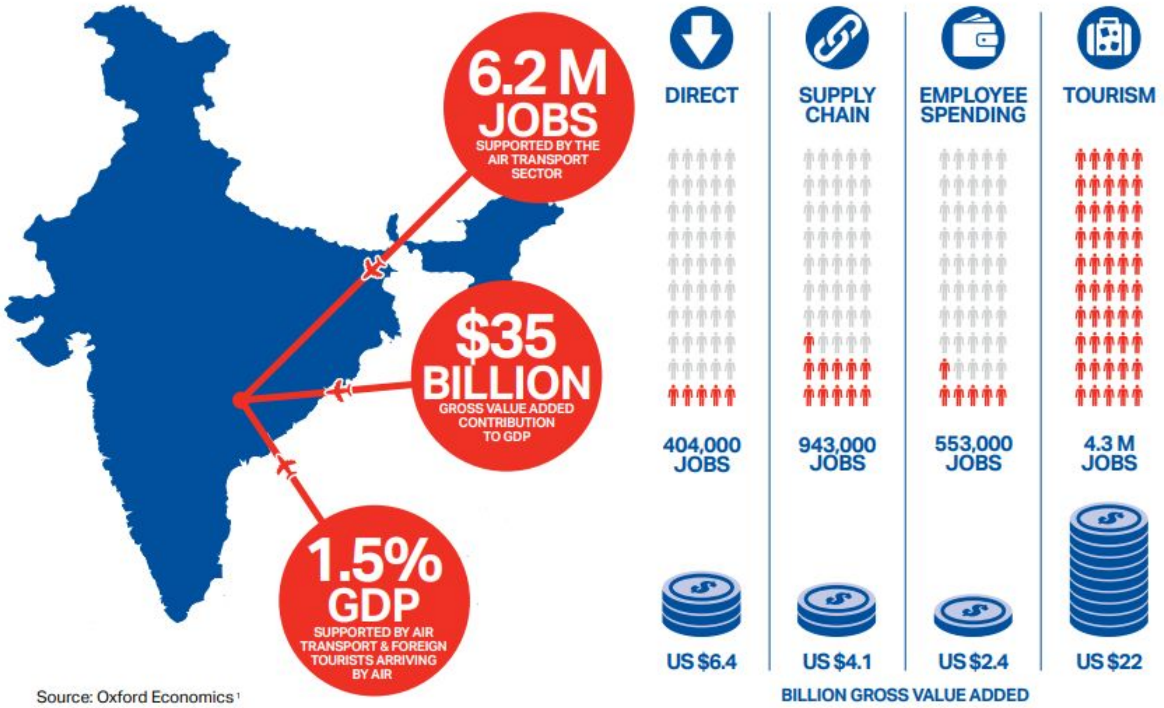
Han ye, Hakim, Siang Jen (**Team C**)

Air Travel Industry in India

Contributed 35 Billion USD to GDP

Expected to grow by 262% in the next 20 years

Additional 370.3 million passenger journeys by 2037



Real Life Problem

‘When should passengers purchase their airline ticket to get the cheapest price when flying in India?'

The Process


Art and Craft of DATA SCIENCE
Sample COLLECTION
Practical MOTIVATION
Data PREPARATION
Problem FORMULATION
Exploratory ANALYSIS
Statistical DESCRIPTION
Analytic VISUALIZATION
Pattern RECOGNITION
Algorithmic OPTIMIZATION
Machine LEARNING
Information PRESENTATION
Statistical INFERENCE
Ethical CONSIDERATION
Intelligent DECISION

Data Set

- An internet platform for booking flight tickets
- Popular platform for intra-country travel within India

Variables

1) airline
2) flight
3) source city
4) departure time
5) stops
6) arrival time
7) destination city
8) class
9) duration
10) days left
11) Price

Sample Collection

- Data size:
  300153 samples
- No **null** values

```
In [16]: flightData.isnull().sum()

Out[16]: Unnamed: 0          0
         airline             0
         flight              0
         source_city         0
         departure_time      0
         stops               0
         arrival_time        0
         destination_city    0
         class               0
         days_left           0
         price               0
         roundDuration       0
         dtype: int64
```

Retained

- Airline
- Source city
- Departure time
- Stops
- Arrival time
- Destination city
- Duration
- Days left
- Price

Removed

- Class
- Flight

Data Science Problem

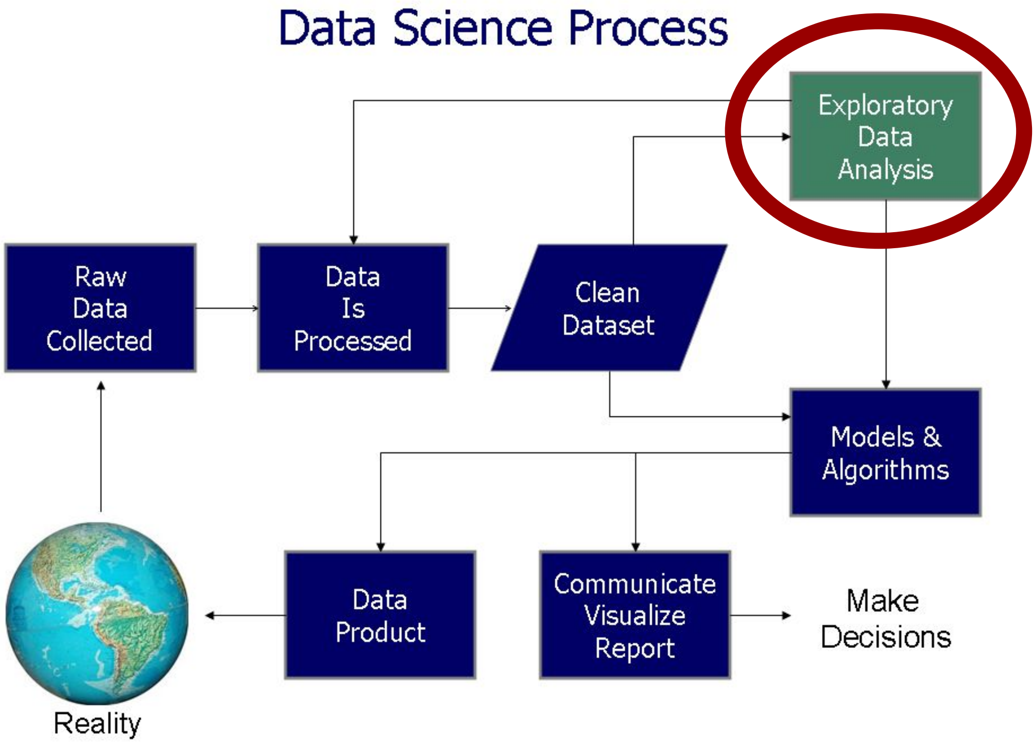
Data Optimization Problem

When can someone get the cheapest Airfare traveling from Delhi with relevant variables involved?

Why Delhi?

- Capital of India
- Major transport hub
- 17.91 million visitors in 2017

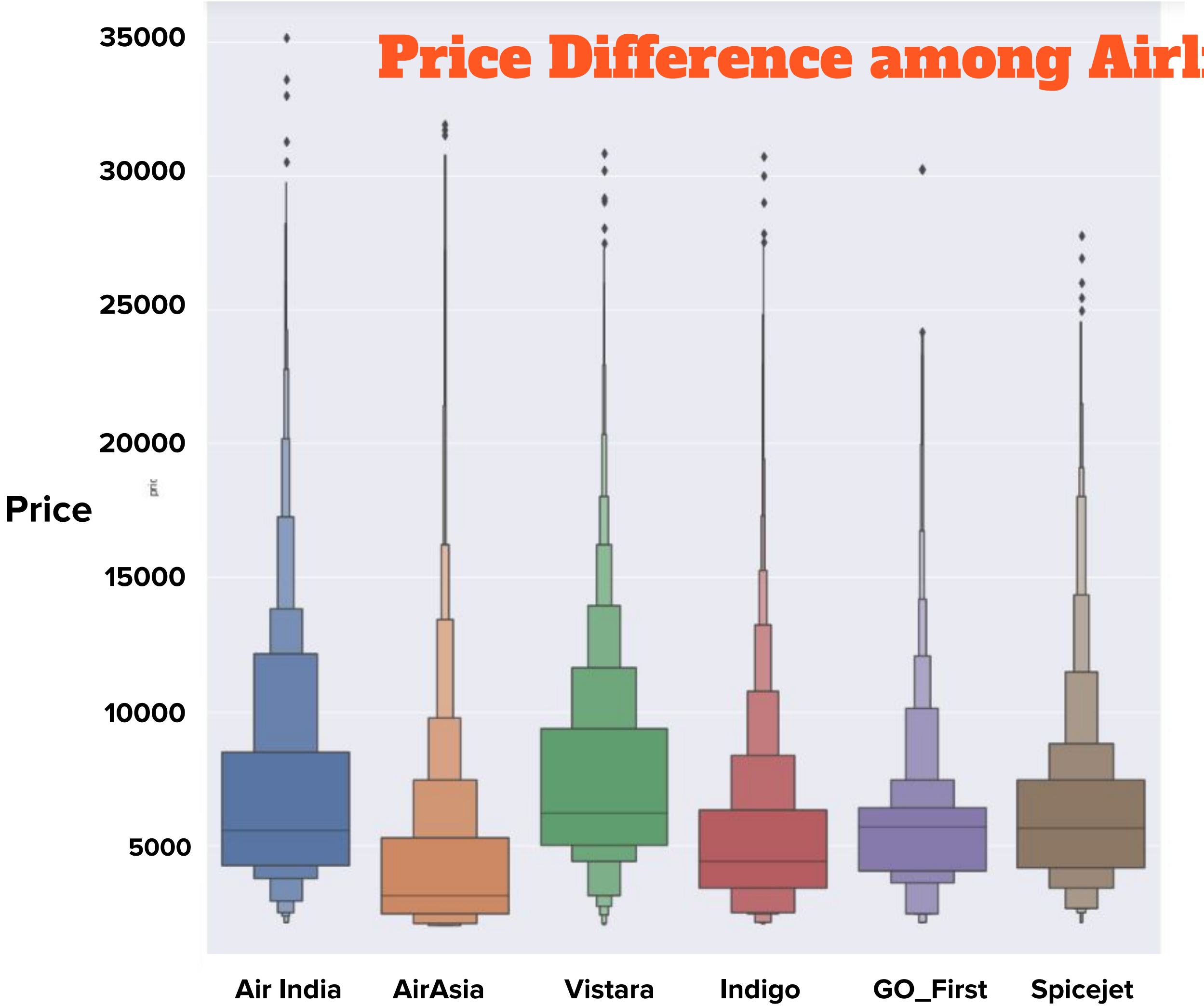
Exploratory Data Analysis



Exploratory Data Analysis

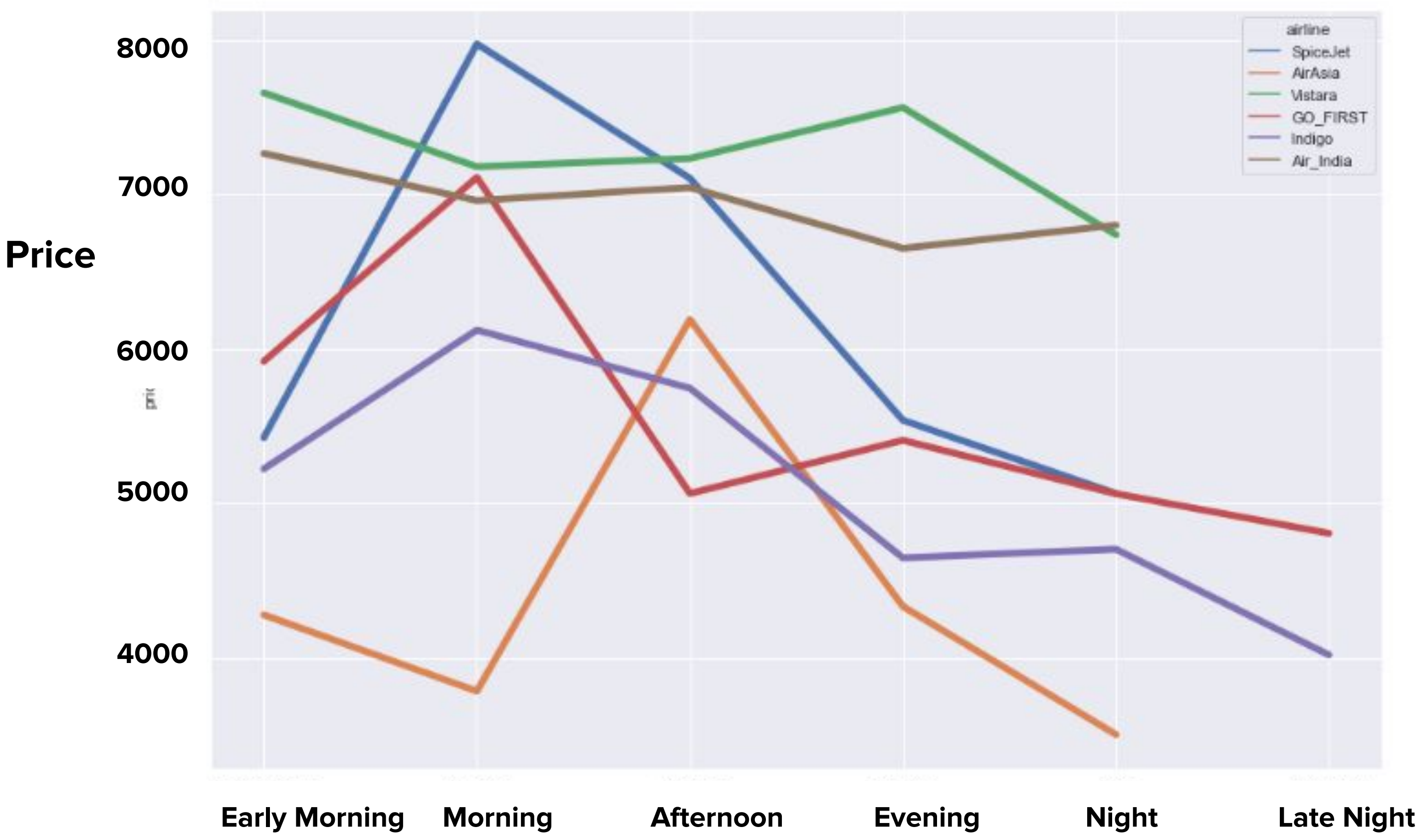
- Price Difference Among Airlines
- Time of day vs Prices
- Days Left vs Price
- Number of Stops vs Price

Price Difference among Airlines



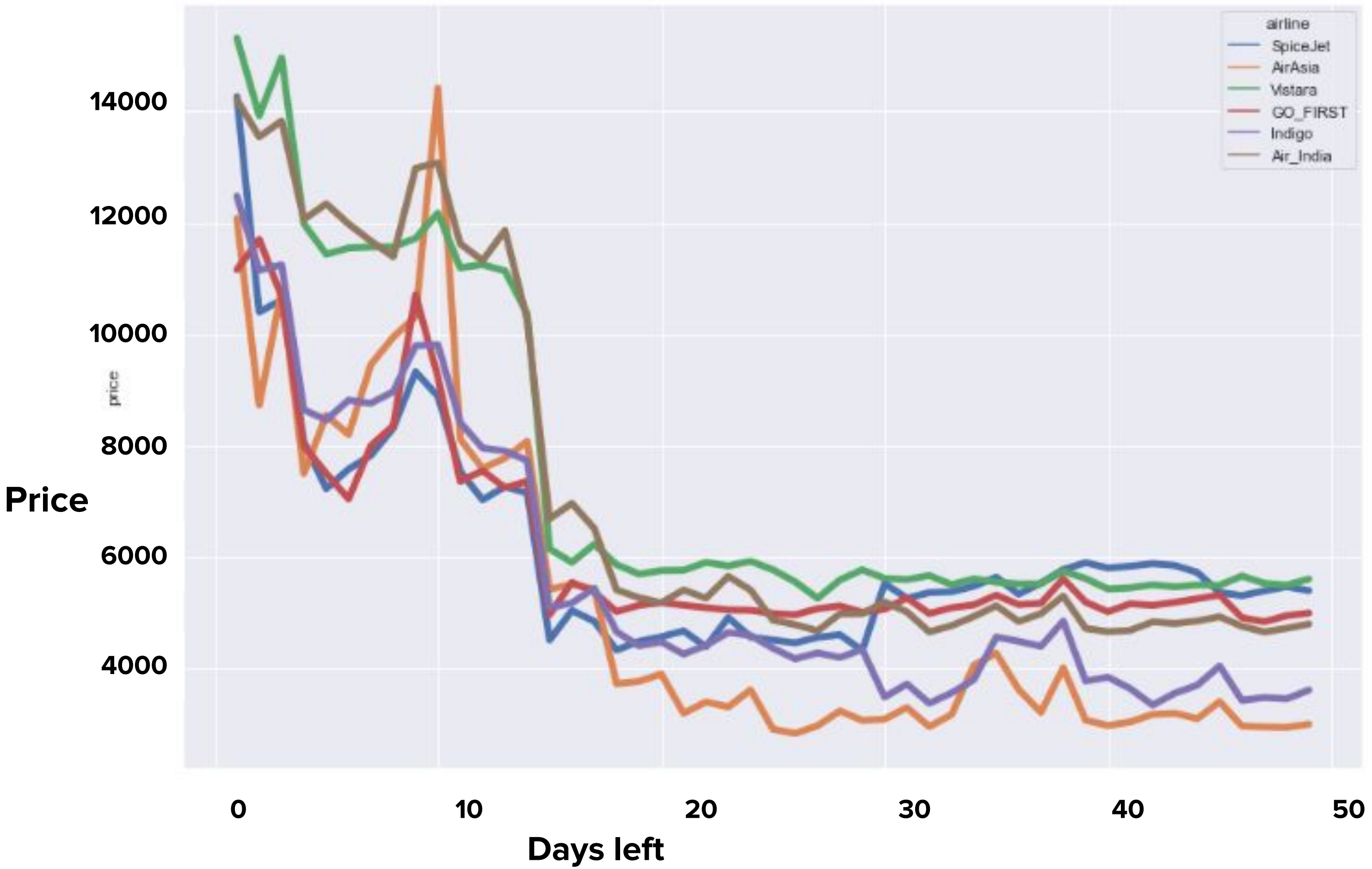
- Air India has the most outliers.
- Go First has the least.
- Median prices for the most airlines are at the 6000 mark.

Time of day vs Price



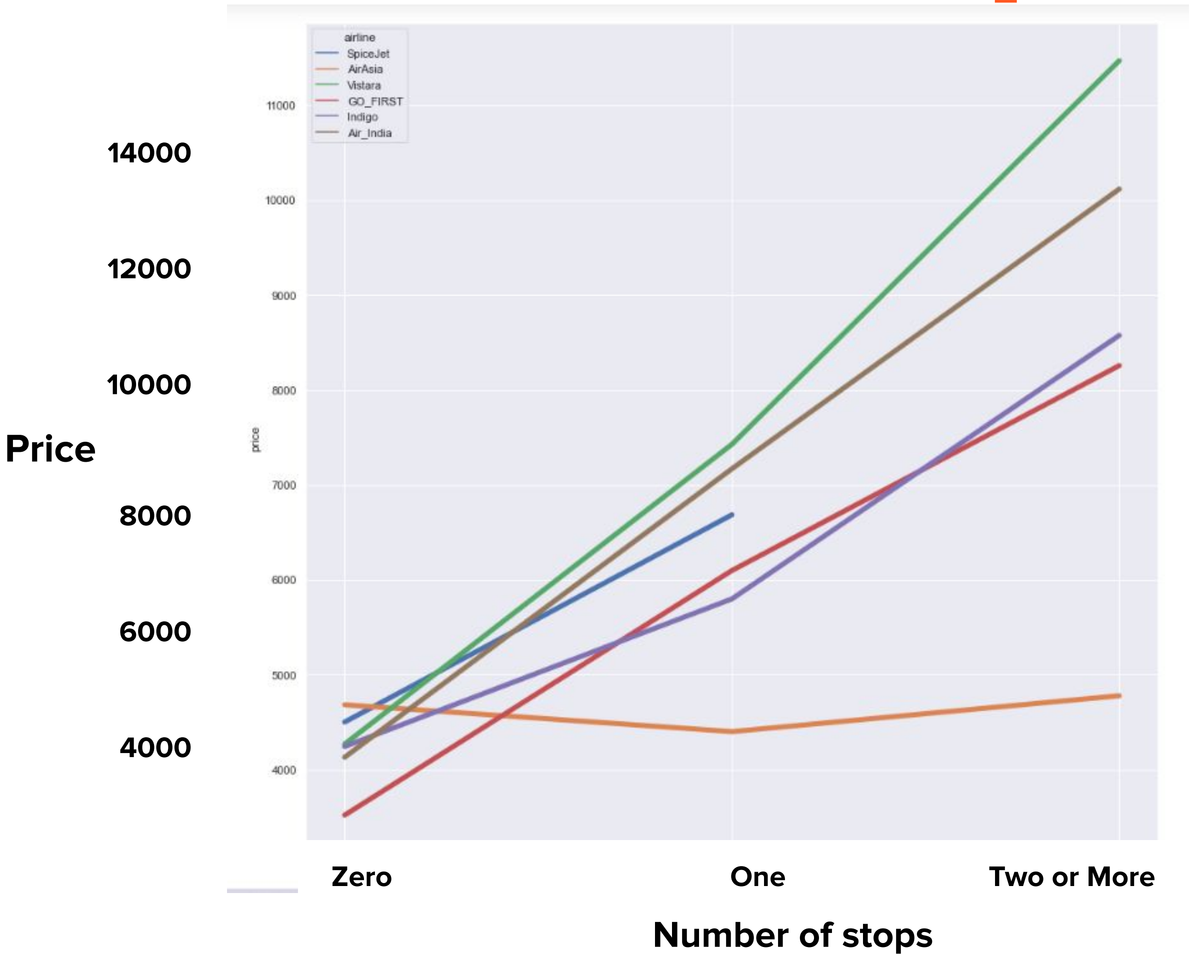
Noticeable price spikes and drops throughout different times of the day

Days Left vs Price



Noticeable Negative relationship between Days left and Price Variables

Number of Stops vs Price



Increase in Price
with more stops

Machine Learning

Machine Learning

Check for Additional Redundancies	Fitting into the Decision Tree • Encoding • Confusion Matrix	Fitting into the Random Forest • Confusion Matrix

Machine Learning

- Removed source city and arrival time

```
#drop redundacy column
flightDataDelhi.drop(['source_city'],inplace = True,axis=1)
flightDataDelhi.drop(['arrival_time'],inplace = True,axis=1)
```

- Encode categorical variables into an array of integer columns

```
# Import the encoder from sklearn
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder()

# OneHotEncoding of categorical, only required for airline, departure_time, stops and destination city
flightDataDelhi_cat = flightDataDelhi[['airline', 'stops', 'destination_city']]
ohe.fit(flightDataDelhi_cat)
#transform the category into columns
flightDataDelhi_cat_ohe = pd.DataFrame(ohe.transform(flightDataDelhi_cat).toarray(),
                                   columns=ohe.get_feature_names_out(flightDataDelhi_cat.columns))

# print out the result
flightDataDelhi_cat_ohe.info()
```

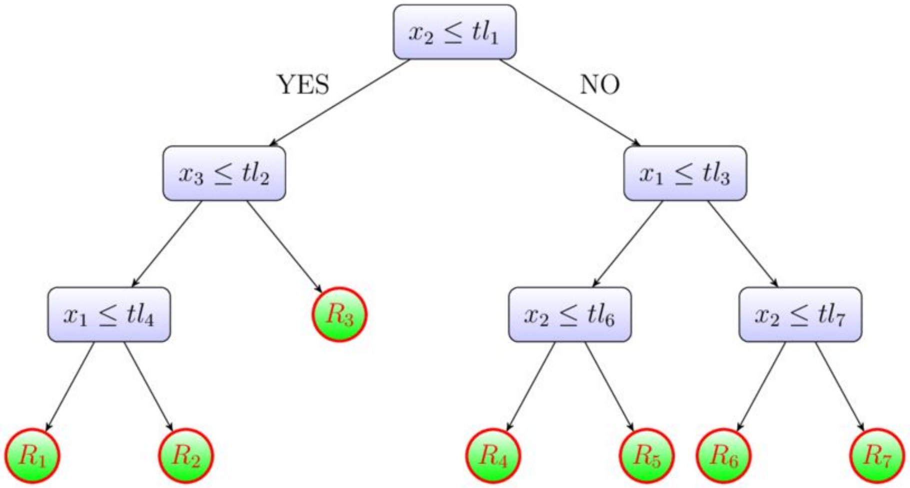
Why Decision Tree?

Pros:

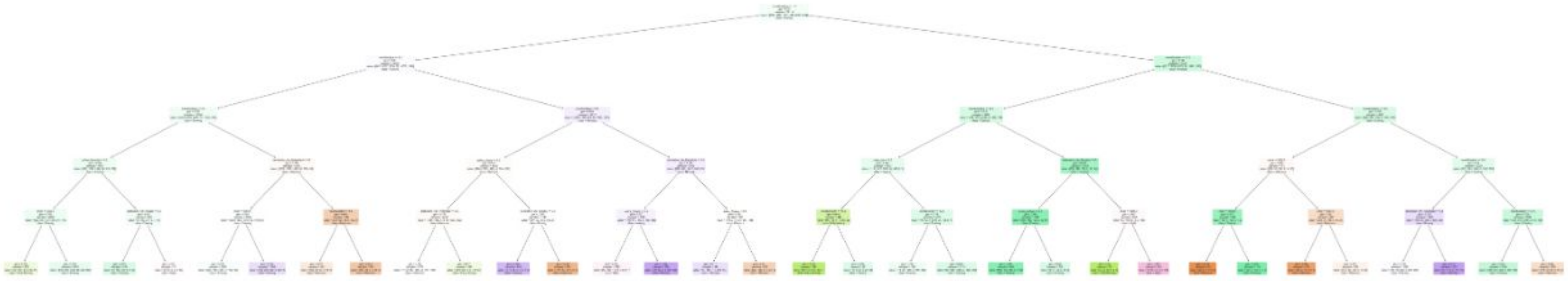
- Scale invariance
- Robust to irrelevant features
- Easily Interpretable

Cons:

- Tend to overfit



Decision Tree



Decision tree is used to compare different predictors variables to a categorical response variable (departure time).

Based on the classification accuracy, it is approximately 60%.

Additionally, the false positive rate falls below 50%

Decision Tree (Confusion Matrix)

Train Data


```
Train Data
Accuracy  :      0.6198286738827279

TPR Train :      0.6589979246961162
TNR Train :      0.5699841545310496

FPR Train :      0.43001584546895044
FNR Train :      0.34100207530388377
```

Test Data

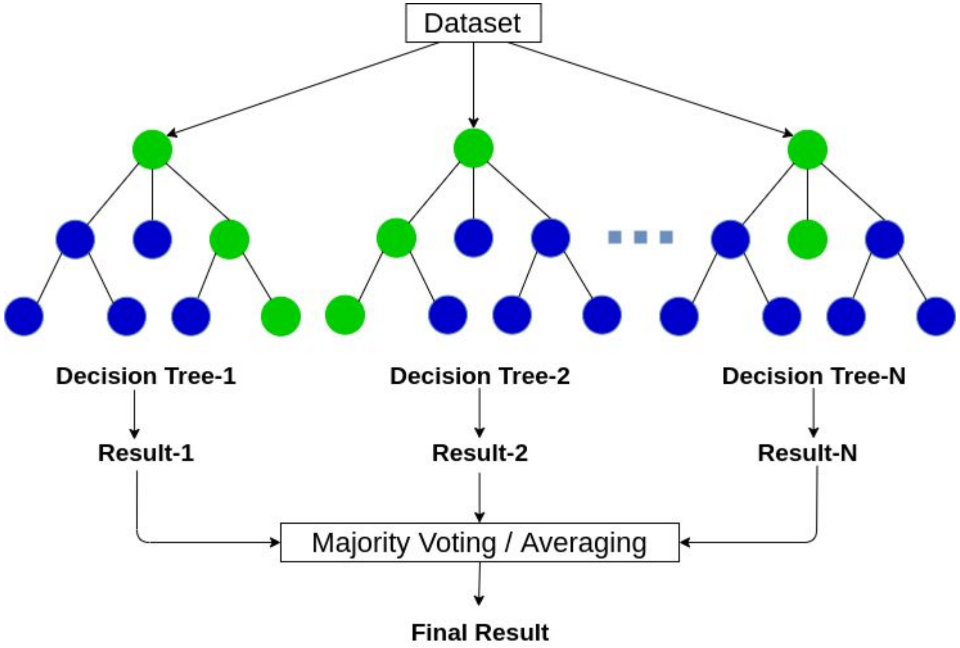

```
Test Data
Accuracy   :      0.6241381981563251

TPR Test  :      0.6623467112597548
TNR Test  :      0.5763125763125763

FPR Test  :      0.4236874236874237
FNR Test  :      0.33765328874024525
```

Why Random Forest?

- Group of Decision trees working together
- 100s/1000s of Decision trees forms a Random Forest

Pros:

- Dilutes the overfitting issue
- Less variance compared to single Decision Tree



Random Forest (Confusion Matrix)

Train Data


```
Train Data
Accuracy  :      0.8657945414702172

TPR Train :      0.9010562749895566
TNR Train :      0.8215702417483721

FPR Train :      0.17842975825162788
FNR Train :      0.09894372501044339
```

Test Data


```
Test Data
Accuracy   :     0.7679913238825625

TPR Test :       0.813563975837452
TNR Test :       0.7089777777777778

FPR Test :       0.29102222222222224
FNR Test :       0.18643602416254806
```

Results

Based on the ML:

- Increase of Accuracy from 60% to 70/80%
- True Positive Rate increased from 60% to 80/90%
- False Positive rate decreased from 40% to 20/10%

Outcome

Based on the ML:

For Ticket purchasers/ passengers

- Able to purchase cheaper tickets

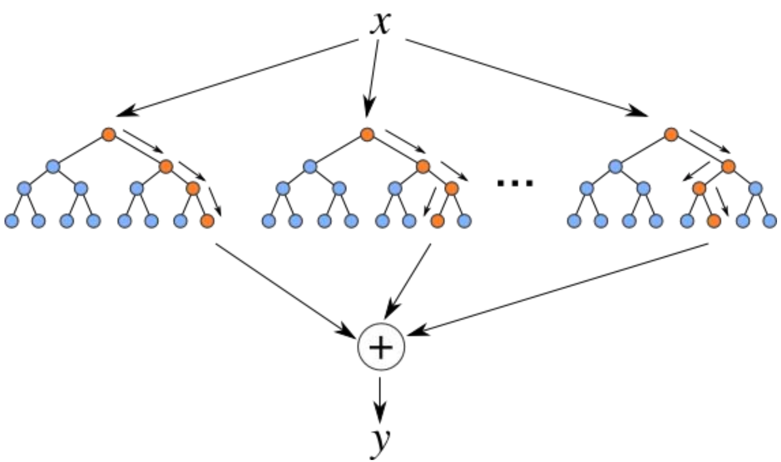
for Airlines

- Able to plan ticket pricing accordingly
- Able to plan better placed promotions taking into consideration dates and time of the day

Outcome

What we have learned

- Line Plot EDA
- Usage on OneHotEncoder from sklearn.preprocessing to encode our categorical variables into an array of integer columns
- Handling and categorising datasets for easier computation and visualization
- RandomForestClassifier from scikit-learn



THANK YOU!