FLIGHT_FARE_PREDICT

INTRODUCTION

- This Dataset is related to Flight Fare Prediction
- This Dataset is collected from Makemytrip.com
- The size of this Dataset is rows=2121, columns=10

- 1.flight= flight Brand name
- 2.flight_code= it is model no. of Flight
- 3.FROM = Airport name, where flight will be take-off from
- 4. TO = Airport name, where flight will be landing
- 5. duration= Flight journey time in hourly format
- 6. CLASS = economy class, premium class & Business class Flight seats
- 7.departure_time= in 24 hour format
- 8. arrival-time
- 9. price = Flights fare

DATA SOURCES & THEIR FORMAT

- I collected this data from 'Makemytrip.com' through web scraping in CSV format

Below shown Data description

```
df.head()
```

	flight	flight_code	FROM	TO	duration	Date	CLASS	price	departure_time	arival_time
0	SpiceJet	SG 8251	New Delhi	Kolkata	2.10	2022-09-24	economy	5948	18.5	21.00
1	SpiceJet	SG 8254	New Delhi	Kolkata	2.25	2022-09-24	economy	5948	20.1	22.35
2	IndiGo	6E 5219	New Delhi	Kolkata	2.10	2022-09-24	economy	5954	7.1	9.20
3	IndiGo	6E 6005	New Delhi	Kolkata	1.55	2022-09-24	economy	5954	22.2	0.15
4	Air India	AI 401	New Delhi	Kolkata	2.10	2022-09-24	economy	5955	6.5	9.00

```
# let's check the dataset shape
df.shape
```

```
(2121, 10)
```

```
df.isnull().sum() # checking null value in dataset
```

```
df.info()      # brief summary about dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2134 entries, 0 to 2133
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   flight          2134 non-null   object 
 1   flight_code     2134 non-null   object 
 2   FROM            2134 non-null   object 
 3   TO              2134 non-null   object 
 4   duration        2134 non-null   float64
 5   Date            2134 non-null   object 
 6   CLASS           2134 non-null   object 
 7   price           2134 non-null   int64  
 8   departure_time  2134 non-null   float64
 9   arival_time     2134 non-null   float64
dtypes: float64(3), int64(1), object(6)
memory usage: 166.8+ KB
```

Business Problem Framing

- The data given in this Dataset is important to predict the fare of Flights
- According to this data we are going to predict the fare of the Flight
- At this time there are many companies that do online ticket booking Flights, so you can go to this company's web portal and find out the Flight Fare for a few days ahead, in this way this dataset is connected to real world

Review of Literature

- I have taken this Dataset from ‘MakeMyTrip.com’ through web scrapping and have taken most of the factors which matter more to predict the fare of the Flights in this dataset so that the Fare can be predicted with more accuracy
- If we collect this data date wise and at the same time we increased the data according to the time, then the accuracy of prediction might be good because we cannot predict for a long time by collecting more data at the same time, because the Fare of the Flights varies from time to time

ANALYTICAL PROBLEM FRAMING

Mathematical/ Analytical Modeling of the Problem

- First used some methods to get Statistical information of this Dataset such as we used Describe function of PANDAS then
- we get Statistical information of all attributes and the range of distribution of all continuous attributes means how much value is distributed in which range, almost Outliers are also known, from this method
- I got the possibility of an attribute being skewed, then checking skewness and Outliers of that Attributes then that Attributes were skewed
- we analyzing the correlation of the Dataset through plotting correlation heatmap then conclusion was

CLASS attribute more positive correlated with price and duration, flight these attributes are less positive correlated with price

If the Dataset has a Target continuous value and positively correlated attributes are more in such a Dataset, then we are more likely to get more accuracy.

DATA PREPROCESSING DONE

- First load the dataset on Jupiter notebook and print it then clean the dataset

- I started looking at randomly some rows and columns of dataset and tried to understand and after check the Null value in this dataset but in this Dataset, there was no any Null value or anything to do like data cleaning, so were moved ahead
- I described the Dataset for Statistical analysis and after that there were some categorical Variables on which value_counts() used to how many types of values are in this variable
- ACCORDING TO EDA
- I visualize the data using Matplotlib & seaborn
- we can use label-encode on Flight Attributes according visualization(barplot)
- maximum fare price of Business class seat in Flights
- New Delhi, Mumbai, Chennai, Bengaluru, Ahemadabad from these international airports most of Flight take off
- to another domestic airport
- There are only two categorical columns in this Dataset on which label-encoding can be used, Flight and CLASS
- 'flight_code' is categorical attribute and in this 1317 different value if we use one-hot-encoding or get_dummies() method on this attribute than dataset attributes size more than 1320 and we could build a model with keeping its and after again remove this attribute from dataset and then build a model so in r2_score no shown any differences
- I analyzing correlation of this dataset, outliers and skewness and last process is Scaling the dataset

Data Inputs- Logic- Output Relationships

- After using EDA, I found that the two categorical attributes (‘flight’, ‘CLASS’) are related with Target(‘price) and a ‘duration’ attribute’ less correlated with target, These attributes are important to predicting the target
- If the Target is continuous attributes, then the more positively correlated the input columns with the target, then the accuracy of the predicted value will also be accurate

HARDWARE AND SOFTWARE REQUIREMENT TOOLS

- we have done this entire project through (sklearn, pandas numpy, matplotlib, Seaborn, scipy,) Load the dataset, and all data cleaning & analyzing part done by Pandas.
- we have done Visualization the dataset through matplotlib, & seaborn. All Machine Learning model have been created in this project through Sklearn , xgboost

MODEL DEVELOPMENT AND EVALUATION

Identification of possible problem-solving approaches (methods)

- Before making a model on this dataset, we used some methods on it like (data-cleaning, EDA, data-preprocessing and Train & Test the model)
- we had to do Data cleaning because this data was obtained through web scrapping.
- After Data cleaning, I did EDA on this dataset so that each Attribute of the data can be understood well and which attribute is more important to predict the fare and as everyone knows that EDA is a very important process for understanding the Dataset
- In Data Preprocessing, there was no preprocess of much data because the dataset had nine attributes in which there were four categorical and four continuous type attributes, we also checking distribution of the dataset and used the describe function on the dataset to get the Statistical information and analyzed on its
- I checked Outliers only one column showed outliers but there was not much outliers. after that reduced the skewness of the data and last Scaling the input data.
- Now we started the process of model building

Testing of Identified Approaches (Algorithms)

- We used five types of algorithms for building the model

KNN-regressor, RandomForest-Regressor, AdaboostBoost, SGD-Regressor, GradientBoosting

- we get maximum r2_score with RandomForest, GradientBoosting in testing the model and from GradientBoosting we get maximum cross_validation_score

- when we used Hyperparameter tuning with GridseachCV on GradientBoosting & RandomForest then we get same r2_score from both

Run and Evaluate selected models

- On this Dataset KNN regressor could not perform well

Below shown snapshot of KNN model

```
## LET'S TRAIN THE MODEL WITH KNeighborsRegressor

from sklearn.neighbors import KNeighborsRegressor
knn= KNeighborsRegressor(n_neighbors=6,p=4)
knn.fit(x_train,y_train)
k_pred= knn.predict(x_test)
print('r2_score=',r2_score(y_test,k_pred))
print('mean_absolute_error=',mean_absolute_error(y_test,k_pred))
print('mean_squared_error=',mean_squared_error(y_test,k_pred))
```

```
r2_score= 0.804809207593372
mean_absolute_error= 2576.00431372549
mean_squared_error= 21162182.311960787
```

- Random Forest score more than KNN regressor because this dataset because the distribution of this Dataset was in such a way that on this algorithm like SVM, KNN would not be able to perform well

Below shown snapshot of this

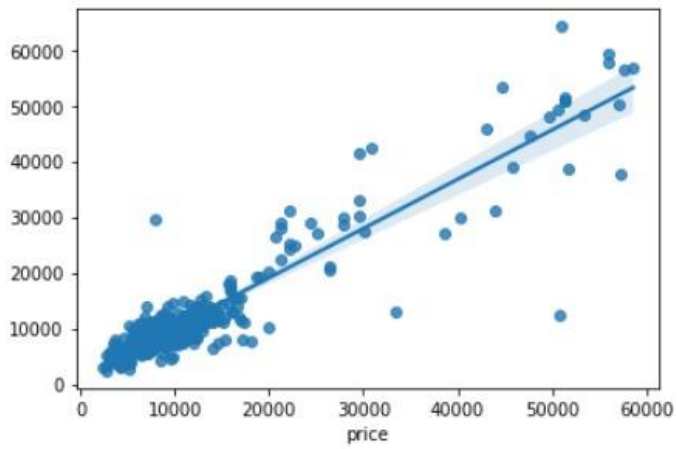
```
## # LET'S TRAIN THE MODEL WITH RandomForestRegressor

from sklearn.ensemble import RandomForestRegressor
RF = RandomForestRegressor()
RF.fit(x_train,y_train)
r_pred= RF.predict(x_test)
print('r2_score=',r2_score(y_test, r_pred))
print('mean_absolute_error=',mean_absolute_error(y_test,r_pred))
print('mean_squared_error=',mean_squared_error(y_test,r_pred))
#print('root_mean_absolute_error=',mean_squared_error(y_test,r_pred))
```

```
r2_score= 0.8655786223103847
mean_absolute_error= 2074.782938515406
mean_squared_error= 14573687.960477708
```

```
# plotting regplot with y_test vs r_pred(randomForest predicted value)
sns.regplot(y_test,r_pred)
```

```
<AxesSubplot:xlabel='price'>
```



In this plot, some points which are visible at a distance from the line because 2 percent data points were Outliers in this Dataset

- GradientBoosting also performed like RandomForest model
- We can see in this plot

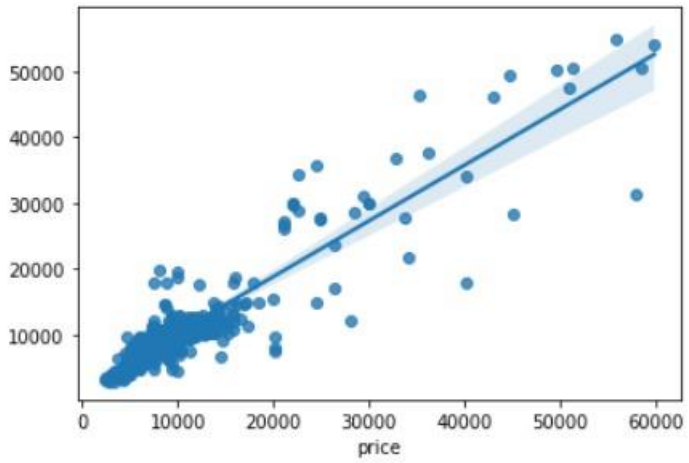
```
### LET'S TRAIN THE MODEL WITH GRADIENTBOOSTING Regressor

from sklearn.ensemble import GradientBoostingRegressor
GB = GradientBoostingRegressor()
GB.fit(x_train,y_train)
GB_pred= GB.predict(x_test)
print('r2_score=',r2_score(y_test, GB_pred))
print('mean_absolute_error=',mean_absolute_error(y_test,GB_pred))
print('mean_squared_error=',mean_squared_error(y_test,GB_pred))
```

```
r2_score= 0.8250537388024314
mean_absolute_error= 2154.8091143267784
mean_squared_error= 12858247.302264411
```

```
# plotting regplot of predicted value of GradientBoostingRegressor
sns.regplot(y_test, GB_pred)
```

```
<AxesSubplot:xlabel='price'>
```



- I used different types of metrics to find which one model is perform better on this dataset
- r2_score, mean_squared_error and mean_absolute_error these metrics list that we used in building the model for analyzing its accuracy
- We also using cross_validation score with k-fold all model
- We used Hyperparameter tuning with GridSearchCV on RandomForest and GradientBoosting

VISUALIZATION

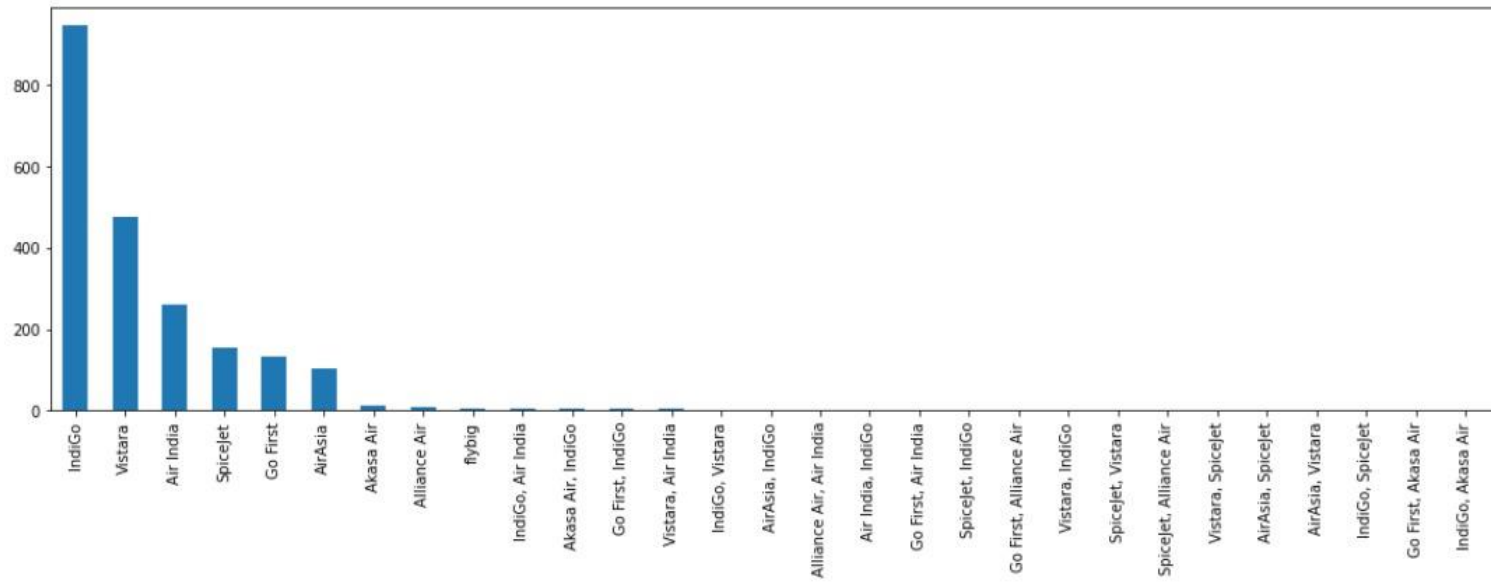
- This plot is of ‘Flight’ attribute and show which Flight brand has most of time in this attribute

```
## LET'S CHECK WHICH COMPANY'S FLIGHT HAVE MORE TAKE-OFF FROM ALL THE AIRPORT

df['flight'].value_counts().plot(kind='bar', figsize=(18,5))

# INDIGO,VISTARA, AIR-INDIA,SPICEJET AND AIRASIA THESE ARE MOST TIME TAKEOFF FROM ALL THE AIRPORT
# INDIGO FIRST POSITION, VISTARA SECOND POSITION,AIR-INDIA THIRD POSITION
```

```
<AxesSubplot:>
```



let's check in 'economy' value of 'CLASS' attribute to which flight fare price has highest

We can see that through this visualization method

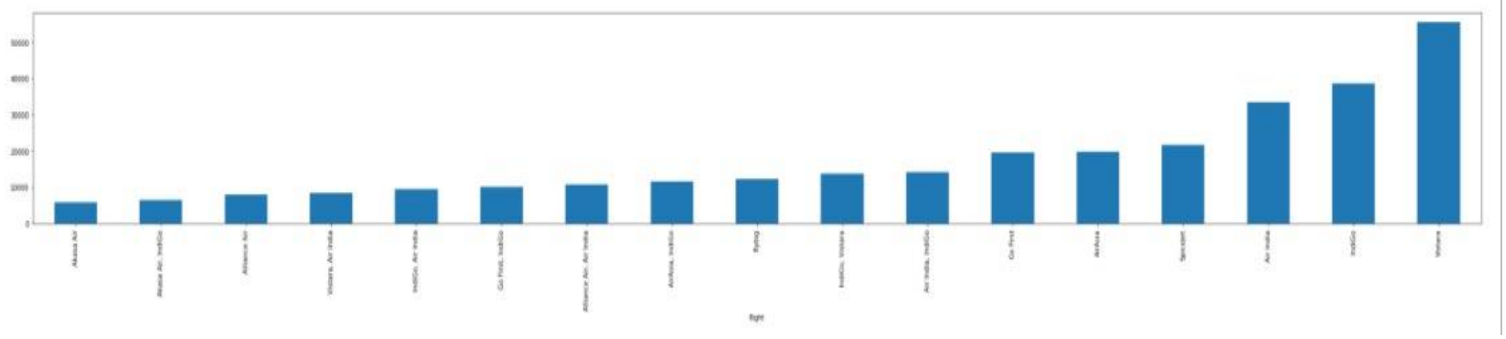
. Vistara flight fare has highest compare to another flight and akash-air flight brand has minimum fare

- According this visualization we change this attribute value into continuous type

```
#
# let's check in 'economy' value of 'CLASS' attribute to which flight fare price has highest

df[df['CLASS']=='economy'].groupby(['flight'])['price'].max().sort_values().plot(kind='bar',figsize=(44,5))
```

```
<AxesSubplot:xlabel='flight'>
```



CONCLUSION

Learning Outcomes of the Study in respect of Data Science

- we get important information from visualization to which Flight fare price has highest in each journey
- we visualize barplot using groupby method with (departure_airport, arival_airport, FLight_name) then we get best information about relationship of these attributes. If we don't use visualization on this dataset then some information was such type of that we don't understand it without visualization
- In EDA data Visualization power is most important point to we get information in understandable way