SPRINT – 1 PROJECT DOCUMENT

Date	16 November 2022
Team ID	PNT2022TMID46686
Project Name	Developing a Flight Delay Prediction using Machine Learning

DEVELOPMENT PHASE:

SPRINT-1:

Outline:

- Data Pre-processing
- Data Analysis
- Feature Engineering
- Model Building
- Random Forest Classification

Required Libraries:

- Pandas - Data Pre-processing
- Numpy - Data Pre-processing, Analysis
- Matplotlib - Visualization
- Seaborn - Visualization
- Imblearn - Balancing Data
- Sklearn - Model Buildling

Software/Tool:

- Anaconda- Jupyter Notebook
- Used Language Python

Data Pre-processing:

Data Collection:

Dataset is collected from the available website.

Dataset description:

```
Out[9]: YEAR                    0
        MONTH                   0
        DAY                     0
        DAY_OF_WEEK             0
        AIRLINE                 0
        FLIGHT_NUMBER           0
        TAIL_NUMBER           277
        ORIGIN_AIRPORT          0
        DESTINATION_AIRPORT     0
        SCHEDULED_DEPARTURE     0
        DEPARTURE_TIME       1477
        DEPARTURE_DELAY      1477
        TAXI_OUT             1538
        WHEELS_OFF           1538
        SCHEDULED_TIME          1
        ELAPSED_TIME         1803
        AIR_TIME             1803
        DISTANCE                0
        WHEELS_ON            1596
        TAXI_IN              1596
        SCHEDULED_ARRIVAL       0
        ARRIVAL_TIME         1596
        ARRIVAL_DELAY        1803
        DIVERTED                0
        CANCELLED               0
        CANCELLATION_REASON  98449
        AIR_SYSTEM_DELAY     81864
        SECURITY_DELAY       81864
        AIRLINE_DELAY        81864
        LATE_AIRCRAFT_DELAY  81864
        WEATHER_DELAY        81864
        dtype: int64
```

Data Analytics:

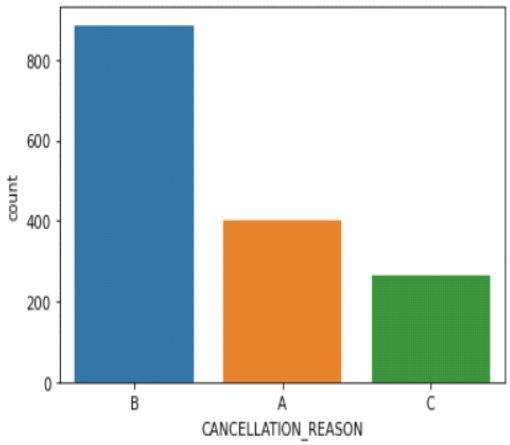
	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTUR
5126156	2015	11	17	2	MQ	3125	N646MQ	BMI	ORD	6
301985	2015	1	20	2	AA	2482	N505AA	DFW	AUS	18
4886973	2015	11	1	7	EV	2828	N629AE	DFW	LAW	21
1589374	2015	4	12	7	WN	1371	N368SW	BNA	CLE	14
4545902	2015	10	11	7	DL	1370	N917DN	10397	14576	9

5 rows × 31 columns

Data Analysis And Visualization:

```
sns.countplot(x='CANCELLATION_REASON',data=flights)
```

```
<AxesSubplot:xlabel='CANCELLATION_REASON', ylabel='count'>
```

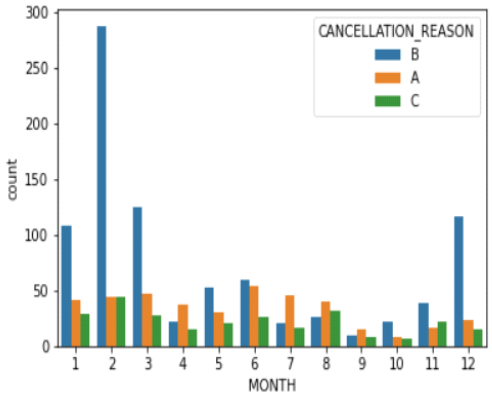


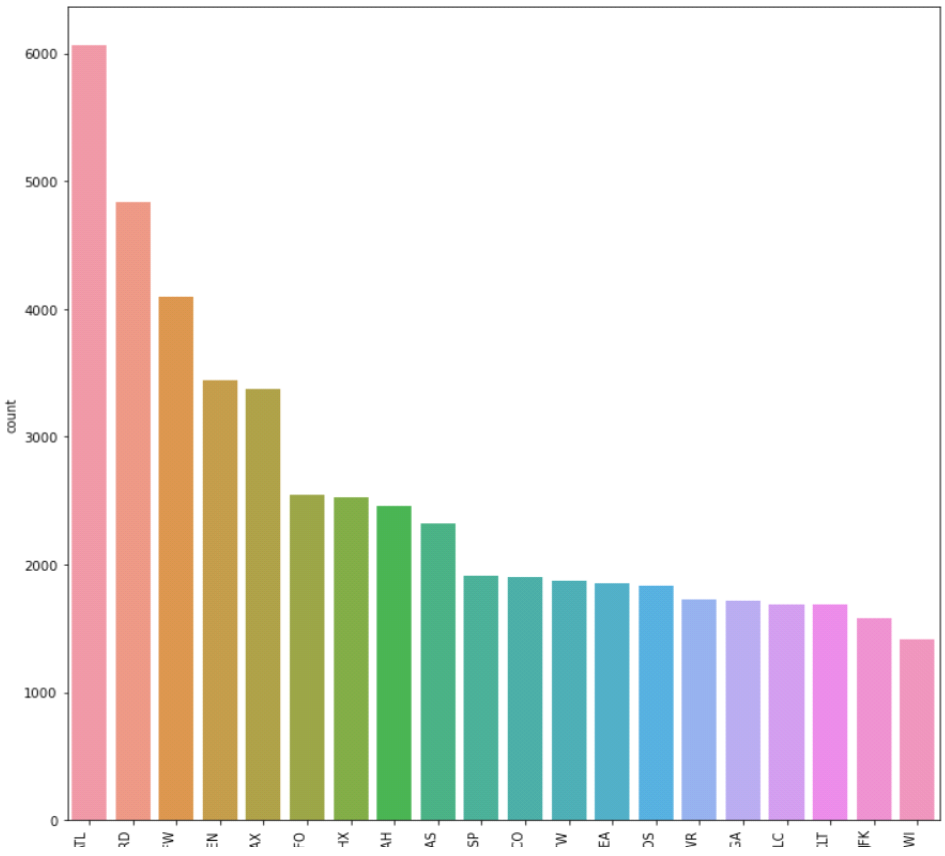
Reason for Cancellation of flight: A - Airline/Carrier; B - Weather; C - National Air System; D - Security

We can observe from graph easily that mostly weather is responsible for delays of flight.

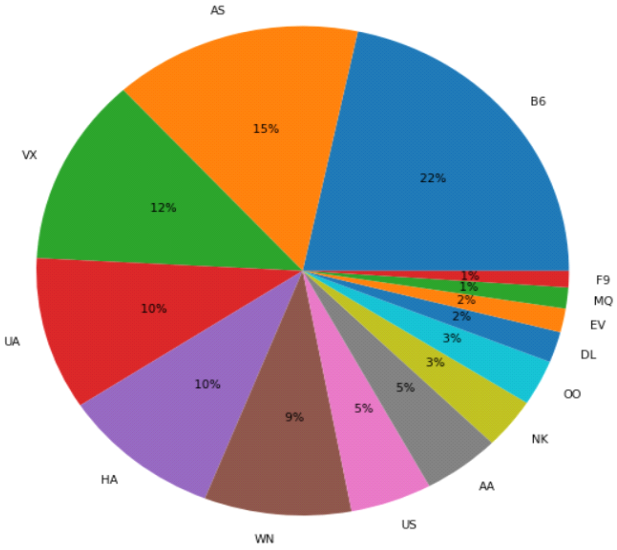
```
sns.countplot(x="MONTH",hue="CANCELLATION_REASON",data=flights)
```

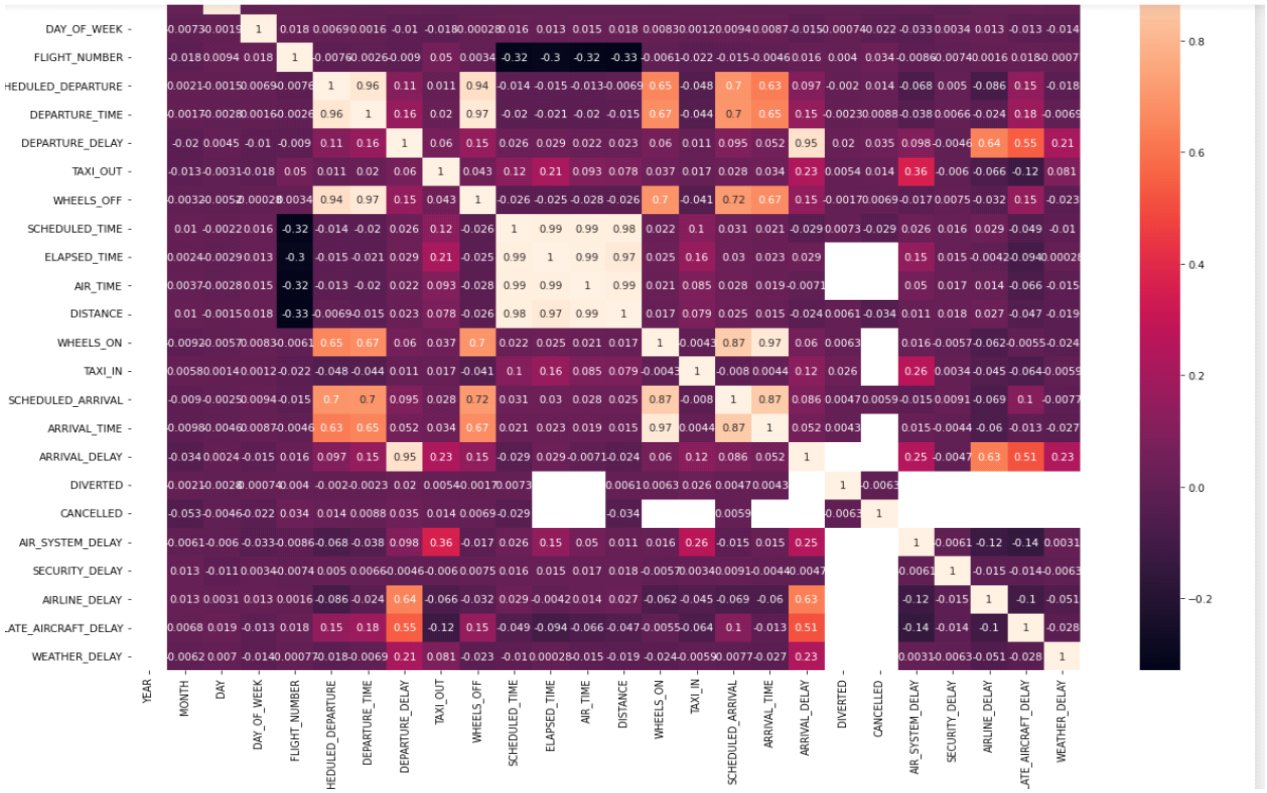
```
<AxesSubplot:xlabel='MONTH', ylabel='count'>
```




count
6000
5000
4000
3000
2000
1000
0

In [13]:
```
axis = plt.subplots(figsize=(10,14))
Name = flights["AIRLINE"].unique()
size = flights["AIRLINE"].value_counts()
plt.pie(size,labels=Name,autopct='%5.0f%%')
plt.show()
```





Feature Engineering:

Very High Correlation Between Arrival Delay and Departure Delay
It shows that maximum of the Arrival Delays are due to the Departure Delays.

```
df=pd.DataFrame(flights)
df['DAY_OF_WEEK']= df['DAY_OF_WEEK'].apply(str)
df["DAY_OF_WEEK"].replace({"1":"SUNDAY", "2": "MONDAY", "3": "TUESDAY", "4":"WEDNESDAY", "5":"THURSDAY", "6":"FRIDAY", "7":"SATU
flights
```

	MONTH	DAY	DAY_OF_WEEK	AIRLINE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY	DISTANCE	ARR
4330572	9	27	SATURDAY	B6	BOS	PIT	1515	7.0	496	
2153991	5	17	SATURDAY	B6	LAX	FLL	1430	4.0	2343	
2268611	5	24	SATURDAY	AS	SEA	SNA	1655	-9.0	978	
5344954	12	1	MONDAY	VX	LAX	FLL	1025	53.0	2343	
1728777	4	21	MONDAY	UA	MCO	EWR	800	-8.0	937	
...	...	...	...	...	...	...	...	...	...	
3542391	8	8	FRIDAY	AS	LAX	SEA	1955	82.0	954	
3777973	8	23	SATURDAY	OO	SLC	BUR	838	-1.0	574	
4002231	9	6	SATURDAY	WN	LAS	PIT	1010	1.0	1910	
1143520	3	16	SUNDAY	DL	SFO	ATL	730	-2.0	2139	
5414693	12	5	FRIDAY	AA	CLT	DFW	1855	-7.0	936	

98197 rows × 10 columns

Data Balancing:

```
print(flights.ORIGIN_AIRPORT.nunique())
print(flights.DESTINATION_AIRPORT.nunique())
print(flights.AIRLINE.nunique())
```

```
321
320
14
```

```
flights=flights.dropna()
flights
```

Model Buliding:

```
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
X=final_data.drop("DEPARTURE_DELAY",axis=1)
Y=final_data.DEPARTURE_DELAY
```

```
X
```

	MONTH	DAY	SCHEDULED_DEPARTURE	DISTANCE	ARRIVAL_DELAY	AIRLINE_AS	AIRLINE_B6	AIRLINE_DL	AIRLINE_EV	AIRLINE_F9	...	DESTINA
3194391	7	19	1133	1008	-32.0	0	0	0	1	0	...	
3403025	7	31	1500	3417	-10.0	0	0	1	0	0	...	
1986115	5	7	630	888	-16.0	0	0	0	0	0	...	
435889	1	29	1525	642	25.0	0	0	0	0	0	...	
2612747	6	14	1525	971	-22.0	0	0	0	0	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3366975	7	29	1345	140	-7.0	0	0	0	1	0	...	
627702	2	11	1310	594	-16.0	0	0	0	0	0	...	
530792	2	5	615	1065	-24.0	0	1	0	0	0	...	
3161417	7	17	1020	406	-15.0	0	0	0	0	0	...	
1560546	4	10	1529	1605	-10.0	0	0	0	0	0	...	


```
pp=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
pp
```

	Actual	Predicted
5648606	5.0	-0.08
1190313	89.0	78.74
177785	-3.0	-1.34
225285	0.0	1.88
2814995	-16.0	-0.17
...	...	...
3071475	-5.0	-5.51
378775	-7.0	-3.29
2913785	8.0	31.97
3023908	-4.0	-3.49
1468738	-5.0	-2.38

12000 rows × 2 columns

Random Forest Classification:

```
# Random search of parameters, using 5 fold cross validation,search across 100 different combinations
rf_random = RandomizedSearchCV(estimator = reg_rf, param_distributions = random_grid,scoring='neg_mean_squared_error', n_iter = 1
```

```
rf_random.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=148; total time=   4.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=148; total time=   4.8s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=148; total time=   4.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=148; total time=   4.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=148; total time=   5.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=182; total time=  10.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=182; total time=   9.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=182; total time=   8.7s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=182; total time=   9.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=182; total time=   9.0s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=44; total time=  38.5s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=44; total time=  36.6s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=44; total time=  36.7s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=44; total time=  37.7s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=44; total time=  38.7s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=61; total time=  55.5s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=61; total time=  53.8s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=61; total time=  53.8s
```


Conclusion:

In this sprint , we builded our model , evaluated and saved. In next sprint, we deploy ourmodel IBM cloud using IBM Watson and building Dashboard.