# SPRINT – 1 PROJECT DOCUMENT

|  |  |
| --- | --- |
| Date | 28 October 2022 |
| Team ID | PNT2022TMID10060 |
| Project Name | Flight Delay Prediction Using Machine  Learning |

**DEVELOPMENT PHASE:**

**SPRINT-1:**

# Outline:

1. Data Pre-processing
2. EDA/Data Analysis
3. Feature Engineering
4. Model Building
5. Saving Best Model

# Required Libraries:

* Pandas - Data Pre-processing
* Numpy - Data Pre-processing, Analysis
* Matplotlib - Visualization
* Seaborn - Visualization
* Imblearn - Balancing Data
* Sklearn - Model Building
* Pickle - Model saving

# Software/Tool:

* Anaconda- Jupyter Notebook
* Used Language Python

**Data Pre-processing:**

# Data Collection:

Dataset is collected from the IBM career smartinternz portal in Guided Project.

# Dataset description:

The dataset contains 31 variables with various data types such as string, object, time, integer, float.

Data columns (total 31 columns):

# Column Non-Null Count Dtype

1. YEAR 11231 non-null int64
2. QUARTER 11231 non-null int64
3. MONTH 11231 non-null int64
4. DAY\_OF\_MONTH 11231 non-null int64
5. DAY\_OF\_WEEK 11231 non-null int64
6. UNIQUE\_CARRIER 11231 non-null object
7. TAIL\_NUM 11231 non-null object
8. FL\_NUM 11231 non-null int64
9. ORIGIN\_AIRPORT\_ID 11231 non-null int64
10. ORIGIN 11231 non-null object
11. DEST\_AIRPORT\_ID 11231 non-null int64
12. DEST 11231 non-null object
13. CRS\_DEP\_TIME 11231 non-null object
14. CRS\_DEP\_TIME.1 11231 non-null int64
15. DEP\_TIME 11124 non-null object
16. DEP\_TIME.1 11124 non-null float64
17. DEP\_DELAY 11124 non-null float64
18. DEP\_DEL15 11124 non-null float64
19. CRS\_ARR\_TIME 11231 non-null object
20. CRS\_ARR\_TIME.1 11231 non-null int64
21. ARR\_TIME 11116 non-null object
22. ARR\_TIME.1 11116 non-null float64
23. ARR\_DELAY 11043 non-null float64
24. ARR\_DEL15 11043 non-null float64
25. CANCELLED 11231 non-null int64
26. DIVERTED 11231 non-null int64
27. CRS\_ELAPSED\_TIME1 11231 non-null object
28. ACTUAL\_ELAPSED\_TIME1 11231 non-null object
29. CRS\_ELAPSED\_TIME 11231 non-null int64
30. ACTUAL\_ELAPSED\_TIME 11043 non-null float64
31. DISTANCE 11231 non-null int64 dtypes: float64(7), int64(14), object(

**Columns Description:**

Dest means Destination Airport.

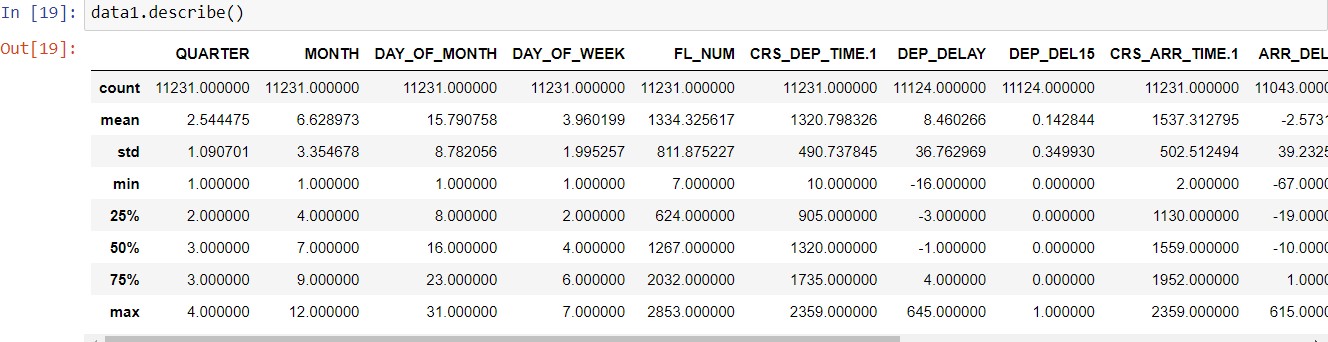
Crs\_dep\_time and crs\_arr\_time is planned departure and arrival time.

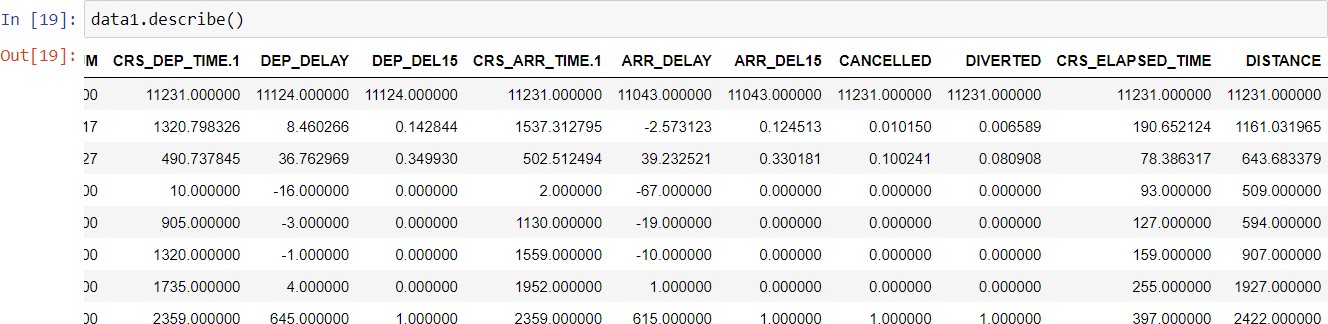
Crs\_elapsed \_time is estimated travel time as per plan. Arr\_time and dep\_time are actual arrival and departure time. Actual\_elapsed\_time is actual travelled time

To pre-process our dataset, we need to import above mentioned required libraries, then import data using pandas.

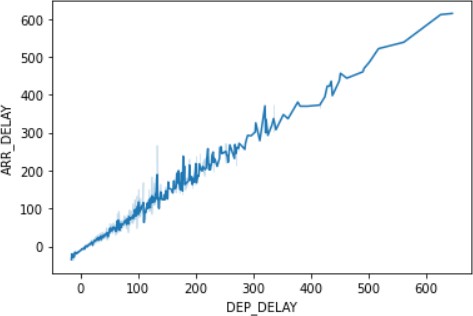
This data does not contain any duplicated values and null values except in arrival , departure time columns, because these left empty when flights are cancelled.

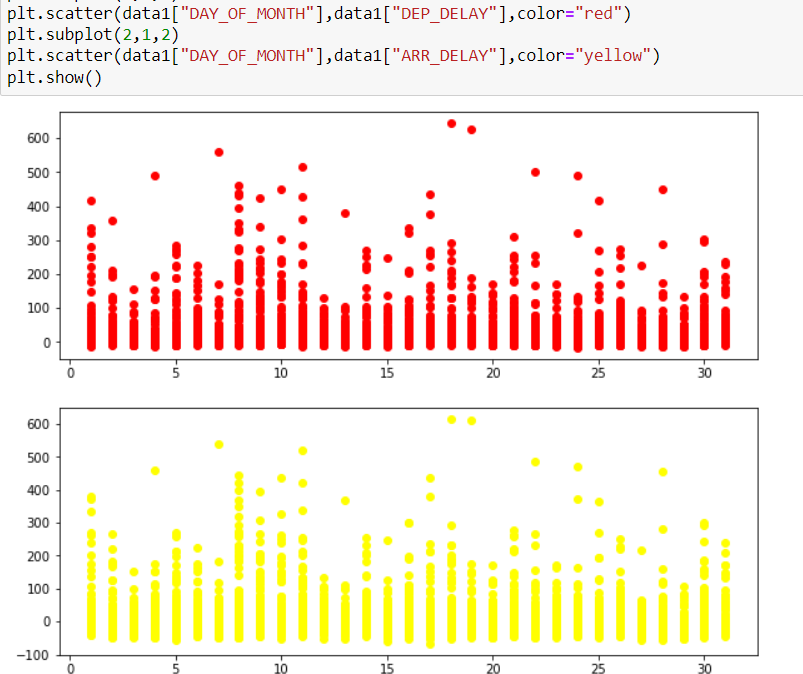
**Descriptive Analytics:**



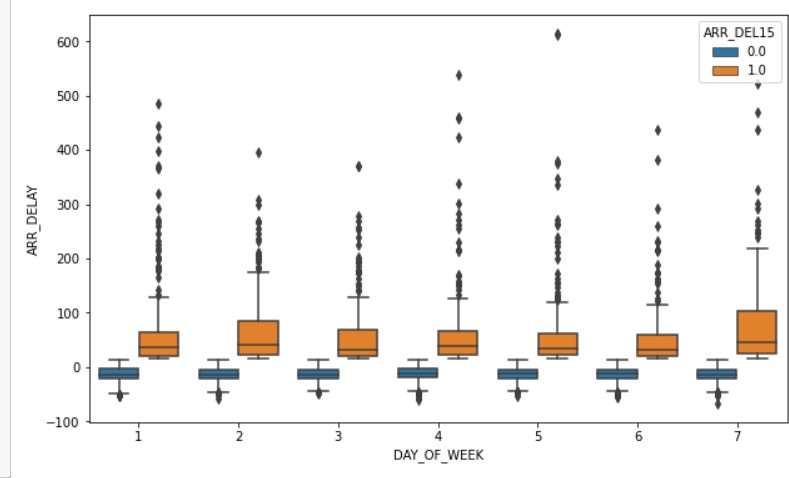


# Data Analysis And Visualization:

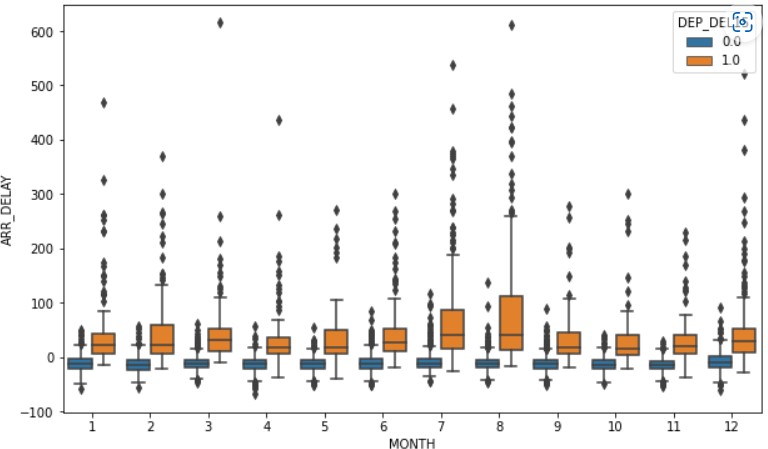
This graph shows the positive trend and strong binding between arrival and departure delay.



This above picture shows the relationship between day of month and delays.

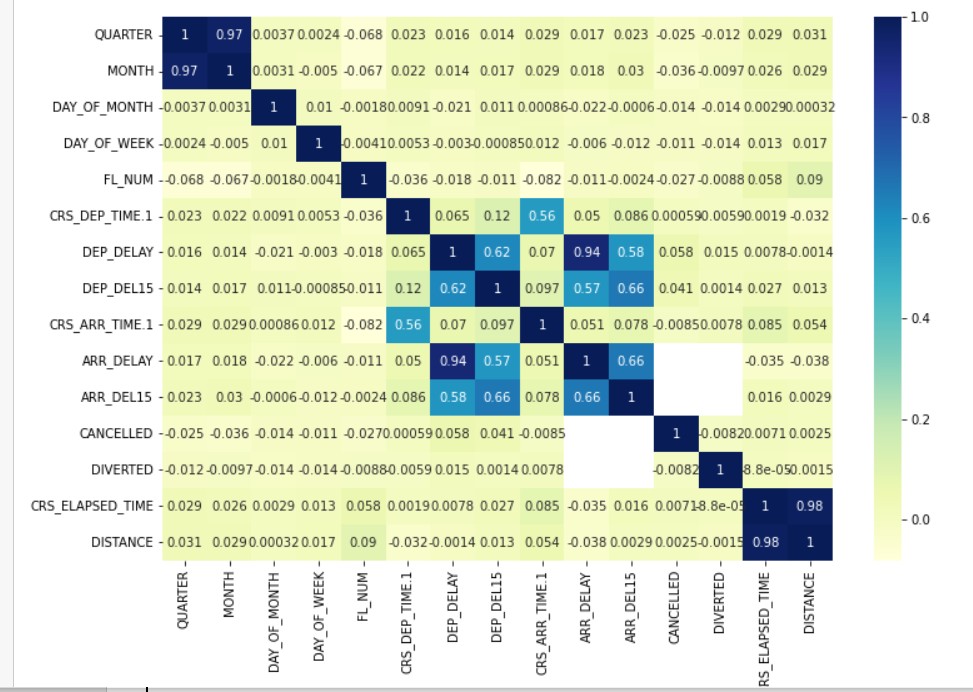


This above boxplot shows the trends of days of the week and delay, Monday and Saturday had high delays.



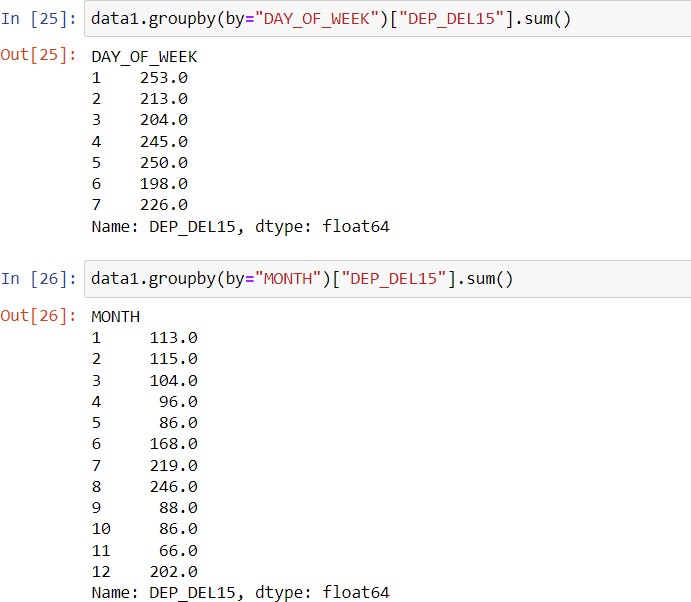
This above boxplot shows the seasonal relationship between months and delays. August had highest no of delays.

Correlation between columns:



# Feature Engineering:

We engineered Season from the month according to the analysis



Then Engineered NDELAY column from the summary of ARR\_DEL15, DEP\_DEL15, CANCELLED, DIVERTED columns.

Splitted NDELAY as dependenr column and others independent columns after removing unnecessary columns.

**Data Balancing:**

We balanced our using SMOTE technique which works based on KNN principle.



Encoding Categorical columns into numerical columns: We encoded ORGIN ,DEST into numerical columns.

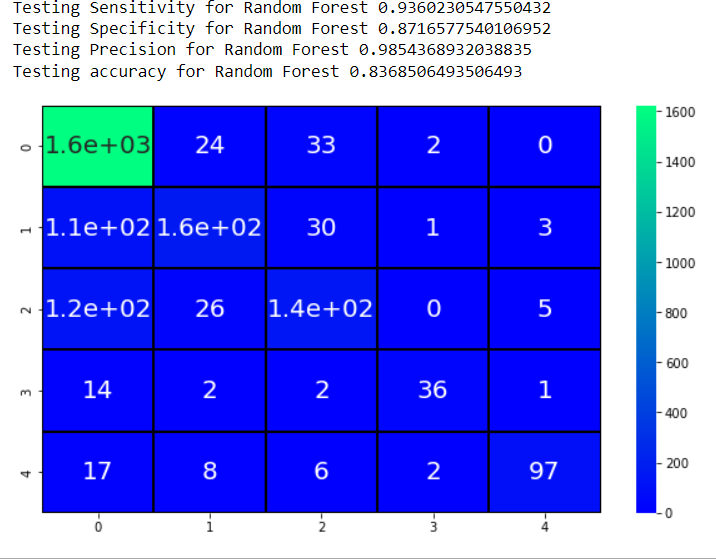
# Model Buliding:

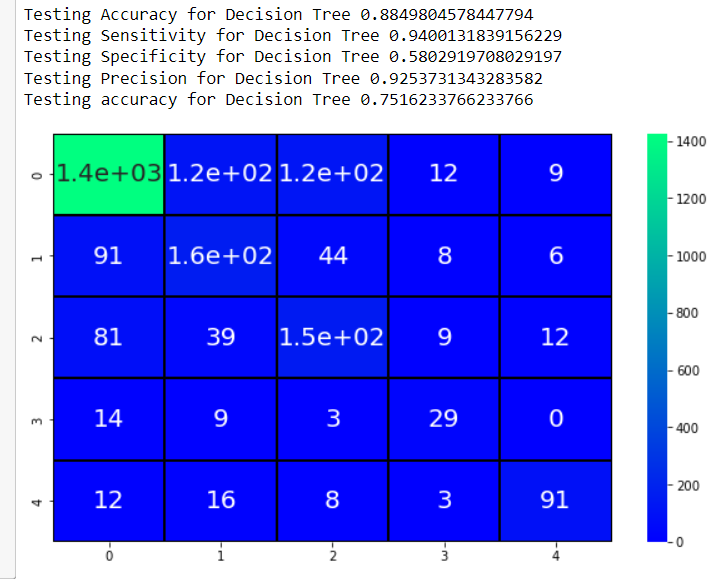
We builded

Decision Tree with 0.7536525974025974 Random Forest with 0.8368506493506493 SVM with 0.6128246753246753

KNN with 0.7280844155844156

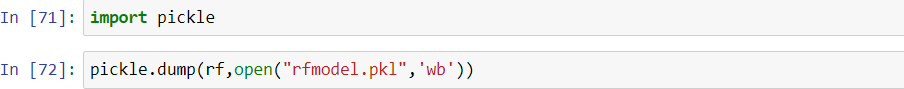
Logistic Regession with 0.6830357142857143

We will explore only Random Forest and Decision Tree which have high accuracy Random Forest:

Decision Tree:

# Model Saving:

Random Forest gives the best accuracy then others , so we save random forest model using pickle.



# Conclusion:

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