Data Intensive Computing

CSE 587

Spring 23

FLIGHT DELAY PREDICTION

By

Vykunth Premanth - 50487477

Sai Kumar Thoppae Sethu Raman - 50483871

1.Introduction:

Flight delays have significant economic consequences for airline corporations due to their potential to result in financial losses, operational costs, and penalties, as well as loss of customer loyalty. Flight delay is defined as the difference between the scheduled and actual times of **departure or arrival**. Various factors contribute to flight delays, such as mechanical issues, air traffic congestion, and adverse weather conditions. Among these, weather conditions are a major cause of flight delays. Therefore, accurate prediction of flight delays due to weather conditions is crucial for improving air traffic control and airline decision-making processes.

Despite extensive research on predicting flight delays using different techniques such as historical, statistical, and model-based methods, little research has been done on predicting both **departure and arrival** flight delays using machine learning algorithms. Moreover, most existing studies have focused on analysing flight delays in specific airports or groups of airports, rather than considering a broader range of airports.

In this project, we aim to develop a two-stage predictive model using supervised machine learning algorithms to predict the occurrence and duration of flight delays caused by adverse weather conditions major airports in the United States. The first stage involves binary classification to predict the occurrence of a flight delay, while the second stage employs regression to predict the value of the delay in minutes for the classified flights. The model focuses on predicting **departure and arrival** delays of flights, which is crucial for airlines to manage their resources and optimize their operations.

The significance of this problem lies in its potential to improve the accuracy of flight delay predictions, minimize disruptions to travel plans, and improve customer satisfaction. Additionally, the use of such a model could enable airlines to proactively adjust their flight schedules, allocate resources more efficiently, and optimize their operations, resulting in significant cost savings.

1.1 Potential Reason for Choosing the Project:

The proposed project of developing a flight delay prediction model that leverages historical flight data and weather data to accurately predict flight delays has significant potential to contribute to the problem domain of flight delay prediction. This contribution is crucial for the following reasons:

- Improved Accuracy: By leveraging historical flight data and weather data, the model can improve the accuracy of flight delay predictions. This would help airlines to better manage their resources and operations, minimize the disruptions caused by flight delays, and improve customer satisfaction.
- Cost Savings: Flight delays can result in significant financial losses for airlines due to operational costs, penalties, and loss of customer loyalty. By accurately predicting flight delays, airlines can proactively adjust their flight schedules and optimize their resources, leading to significant cost savings.
- Operational Efficiency: The use of a flight delay prediction model can enable airlines to optimize their operations, such as flight schedules and resource allocation, resulting in improved operational efficiency and reduced delays.

- Decision Making: The model can provide information on the likelihood and severity of flight delays, enabling airlines to make informed decisions and take necessary actions to minimize disruptions and delays.
- Generalizability: The proposed model considers historical flight data and weather data from major airports in the United States, and the methodology can be generalized to other airports and regions. This would enable airlines to optimize their operations and resources across multiple airports, resulting in improved overall efficiency and cost savings.

In summary, the proposed project can significantly contribute to the problem domain of flight delay prediction by improving the accuracy of predictions, reducing costs, improving operational efficiency, enabling informed decision making, and providing a methodology that can be generalized to other airports and regions.

2. Data Cleaning:

1) Converting JSON format data to CSV (JSON to CSV.ipynb)

Collecting the data and setting up the necessary directories.

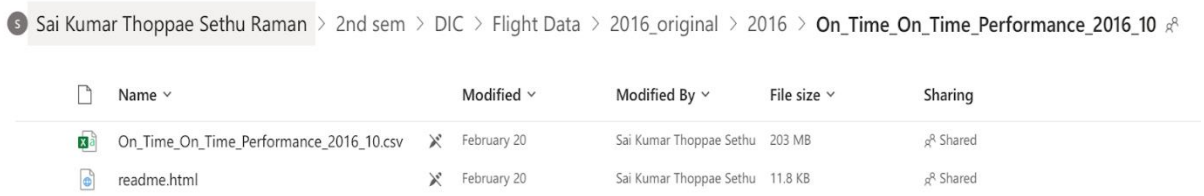
Our data has been downloaded flight data from [1] and weather data from [2] and once the zip has been downloaded, we have set up a comfortable working directory as shown below.

Data Source

[1]https://transtats.bts.gov/Homepage.asp, https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022?resource=download.

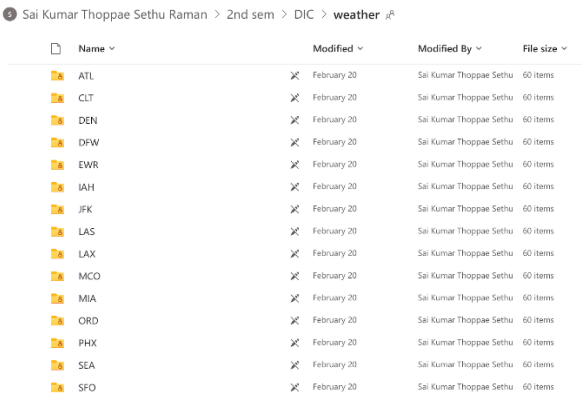
[2] Dataset

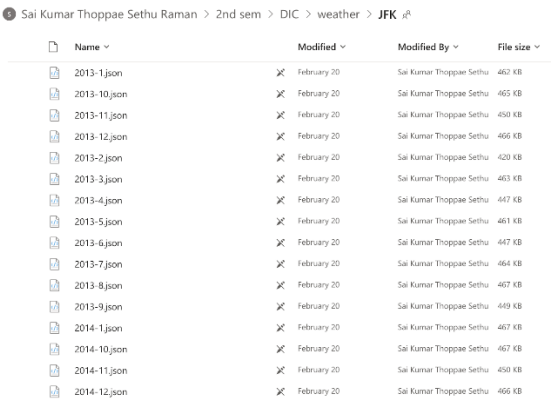
The Flight data has 2 zip files for 2016 and 2017 respectively. Once extracted, we find the on_time_performance of the flights for each month in the corresponding year as separate folders within which its .csv of the data is present.



For the weather data, we have a folder called weather once extracted, that contains all the weather data in JSON format for 15 airports in separate folders named after its airport code.

Inside the airport subfolder, we find the details of the weather for each year and month in separate csv files.





Our weather dataset is from 2013-2017 but we could only collect flight data for the years 2016,2017 therefore we would be deleting/making use of only the data for the years 2016 and 2017.

Once the data has been collected and the directories have been setup, We can start pre-processing the data.

2) Pre-processing the data (flight+weather_merging. ipynb)

2.1) Weather Data (CSV)

We first take into an array, the filenames or the airport names with which we have the weather data available for. We then traverse from these folders and append each file into a single weatherdata.csv

ATL	CLT	DEN	DFW	EWR
IAH	JFK	LAS	LAX	MCO
MIA	ORD	PHX	SEA	SFO

Fig 1 Airports

2.2) Flight Data (CSV)

We load the various csv files for the year 2016 and 2017 into two data frames by concatenating the individual months of the corresponding year. We then merged the two data frames into a single Flightdata.csv for future use.

The columns present in this are:

FlightDate	Quarter	Year	Month
DayofMonth	DepTime	DepDel15	CRSDepTime
DepDelayMinutes	OriginAirportID	DestAirportID	ArrTime
CRSArrTime	ArrDel15	ArrDelayMinutes	

3) Merging Flight and weather data. (flight+weather_merging.ipynb

- We first load the **flight** and **weather** dataset csv files into separate Data Frames
- The **time** in the **weather dataset** has been rounded to the nearest hour. Then the **CRSDEPTime** column in the **flight dataset** is rounded to the nearest hour to match with the weather dataset to enable the merge.
- We then merge both the dataframes based on **Airport, Date** and **the CRSDEPTime**.
- Our Initial data is now ready as a single csv called as **finalmerge**.
- It has 1815405 rows and 51 columns.

4) Finding Null values/missing (flight+weather_merging.ipynb)

Finding and deleting null values in a dataset is critical in flight delay prediction for a variety of reasons. For starters, null values can result in incomplete data, which can lead to erroneous flight delay forecasts. It can also produce biased and skewed analysis results, which can have a negative impact on airline operations. Second, null values can indicate data quality issues such as data input errors or corruption. The quality and reliability of the dataset can be enhanced by identifying and deleting null values, resulting in more accurate flight delay forecasts. Third, deleting null values can enhance computing efficiency, hence boosting the efficiency of the analysis process. Finally, null values can result in missing data, which can lead to incomplete analysis results and erroneous flight delay forecasts. To provide accurate and dependable findings, recognizing and deleting null values is a critical stage in flight delay prediction.

5) Removing the Null values. (flight+weather_merging.ipynb)

We have already discarded 11,140 rows accurately while integrating the weather and flight dataframes in the project. This step was conducted to guarantee that the merged dataset is clean and devoid of any missing values that could skew the analysis results. We can improve the dataset's quality and reliability by deleting missing rows, resulting in more accurate predictions of aircraft delays caused by bad weather. The cleansed dataset will also help to improve the efficiency of the analysis process by decreasing the computer resources necessary for analysis and modeling. Overall, eliminating missing rows is an important step in flight delay prediction to achieve accurate and dependable findings.

6) Finding duplicates

For numerous reasons, finding duplicate values in a dataset is critical in flight delay prediction. For starters, duplicate values can cause an over-representation of particular data points, which can bias the analysis results. This can lead to incorrect flight delay estimations and potentially impair airline operations. Second, the occurrence of duplicate numbers can suggest data quality concerns and flaws, such as incorrect data entry, data duplication, or data corruption. Finding and deleting duplicates can assist enhance the dataset's quality and dependability, resulting in more accurate flight delay estimates. Finally, deleting duplicate data can assist reduce the computational resources necessary for analysis and modeling, which can greatly enhance analysis efficiency.

```
[ ] #finding duplicates
    duplicates = df[df.duplicated()]
    print(duplicates)
```

```
         Unnamed: 0  Unnamed: 0_x  FlightDate  Quarter  Year  Month  \
1814616      684174        283002  2016-07-08        3  2016      7
1814617      988528        339389  2017-01-21        1  2017      1
1814618      243300        460080  2016-12-29        4  2016     12
1814619      470593        132685  2016-04-05        2  2016      4
1814620     1293945        306728  2017-03-02        1  2017      3
...             ...           ...         ...      ...   ...    ...
1815400      502697        257582  2016-04-14        2  2016      4
1815401       80668        248256  2016-10-10        4  2016     10
1815402      969660        124241  2017-01-02        1  2017      1
1815403      144247        110793  2016-10-01        4  2016     10
1815404      856934        232874  2016-09-27        3  2016      9

         DayofMonth  DepTime  DepDel15  CRSDepTime  ...  precipMM_y  \
1814616           8    800.0       0.0         800  ...         0.1
1814617          21   2222.0       0.0        2200  ...         2.6
1814618          29   1843.0       0.0        1800  ...         0.0
1814619           5   1204.0       0.0        1200  ...         0.0
1814620           2   1832.0       1.0        1800  ...         0.0
...             ...      ...       ...         ...  ...         ...
1815400          14   1627.0       1.0        1500  ...         4.1
1815401          10   1228.0       0.0        1200  ...         0.1
1815402           2    741.0       0.0         700  ...         0.1
1815403           1   2040.0       1.0        2000  ...         2.2
1815404          27   1145.0       1.0        1100  ...         0.0

         visibility_y  pressure_y  cloudcover_y  DewPointF_y  WindGustKmph_y  \
```

7)Removing duplicates

```
#removing dupliacates.

df.drop_duplicates( keep=False, inplace=True)
```

8) Removing unwanted columns:

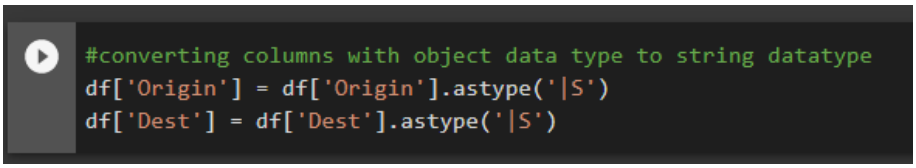
Unwanted columns in the dataset must be removed before flight delay prediction may begin. To lower the size of the dataset and enhance the computing efficiency of the study, delete columns that are irrelevant to the analysis or modelling process.

```
#removing unwanted columns
print(df.columns)
```

```
Index(['Unnamed: 0', 'Unnamed: 0_x', 'FlightDate', 'Quarter', 'Year', 'Month',
       'DayofMonth', 'DepTime', 'DepDel15', 'CRSDepTime', 'DepDelayMinutes',
       'OriginAirportID', 'DestAirportID', 'ArrTime', 'CRSArrTime', 'ArrDel15',
       'ArrDelayMinutes', 'Origin', 'Dest', 'Unnamed: 0_y', 'airport_x',
       'date_x', 'windspeedKmph_x', 'winddirDegree_x', 'weatherCode_x',
       'precipMM_x', 'visibility_x', 'pressure_x', 'cloudcover_x',
       'DewPointF_x', 'WindGustKmph_x', 'tempF_x', 'WindChillF_x',
       'humidity_x', 'time_x', 'Unnamed: 0.1', 'airport_y', 'date_y',
       'windspeedKmph_y', 'winddirDegree_y', 'weatherCode_y', 'precipMM_y',
       'visibility_y', 'pressure_y', 'cloudcover_y', 'DewPointF_y',
       'WindGustKmph_y', 'tempF_y', 'WindChillF_y', 'humidity_y', 'time_y'],
      dtype='object')
```

```
[ ] df.drop(columns=["Unnamed: 0",
                     "Unnamed: 0_x","FlightDate",
                     "OriginAirportID",
                     "DestAirportID",
                     "CRSArrTime",
                     "ArrTime",
                     "ArrDelayMinutes",
                     "date_x","date_y",
                     "airport_x","airport_y"],
            inplace=True)
```

9) Converting columns with object data type to string datatype

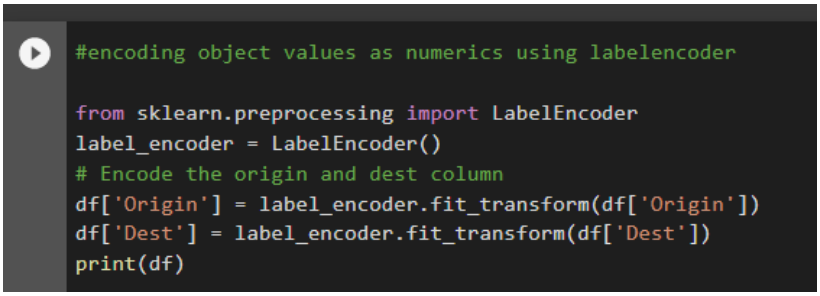


```
#converting columns with object data type to string datatype
df['Origin'] = df['Origin'].astype('|S')
df['Dest'] = df['Dest'].astype('|S')
```

10) Encoding object values as numeric using label encoder

Label encoding is required in this scenario because the 'Origin' and 'Dest' columns include categorical data that must be translated into numerical form in order for the machine learning algorithms to function properly. The label encoder object gives each categorical value in the columns a unique numerical value, allowing machine learning algorithms to accurately read the data and generate accurate predictions.

The 'Origin' and 'Dest' columns will include integer values that represent the category data after label encoding. These numerical values will be fed into machine learning algorithms to predict flight delays.



```
#encoding object values as numerics using labelencoder

from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
# Encode the origin and dest column
df['Origin'] = label_encoder.fit_transform(df['Origin'])
df['Dest'] = label_encoder.fit_transform(df['Dest'])
print(df)
```

11) Normalising the DataFrame

Normalization is a technique used in data preprocessing to rescale numerical data to a common scale. Normalization is required in flight delay prediction because distinct features in the dataset may have varying scales, which might impair the performance of some machine learning methods.

Normalization guarantees that all features are on the same scale, which makes it easier for machine learning algorithms to accurately comprehend the data. This can lead to better flight delay forecast performance and accuracy.

Consider a dataset with two attributes: flight distance and flight duration. The flight distance feature has a value range of 100 to 1000 miles, while the flight time feature has a value range of 1 to 10 hours. If we use a machine learning method without normalization, the algorithm may assign more weight to the feature that has a wider range of values (in this case, flight distance). Because both features are equally essential, this can lead to erroneous forecasts.

Both features are rescaled to a similar scale, such as between 0 and 1. This ensures that the machine learning system gives equal weight to both features, resulting in more accurate predictions.

Normalization is required in flight delay prediction because the features in the dataset, such as weather conditions, travel distance, and flight length, may have varying scales. We can verify that all features are on the same scale by normalizing the data, making it easier for machine learning algorithms to comprehend the data correctly and generate accurate predictions.

3) Exploratory Data Analysis (EDA)

1) Checking for Null Values

The isna() output shows that there are no missing values in any of the columns in the given dataset. Each column has False value for all rows, indicating that there are no null or missing values in any column.

2) Statistical Characteristics of Dataset

- The dataset has 1,814,616 records and 37 columns. The columns include the flight date, time, origin and destination airports, weather conditions, and whether or not the aircraft was delayed.
- Following the removal of certain unnecessary columns, the dataset was preprocessed via label encoding on category variables and normalization on numerical variables.
- After normalizing, the mean of each feature is roughly 0.5, and the standard deviation is less than one, suggesting that the normalization was conducted appropriately.
- The Quarter, Year, Month, and Day of Month characteristics all have similar means and standard deviations, indicating that they are distributed evenly across the dataset.
- The DepDel15 characteristic has a mean of 0.2, indicating that flight delays are rather common.
- The ArrDel15 feature, which indicates if the flight arrived 15 minutes or more later than scheduled, has a mean of 0.2 as well, showing that flight delays are prevalent both in departure and arrival.
- The means of the Origin and Dest features, which reflect the airport codes for the origin and destination airports, respectively, are more than 0.5, showing that some airports in the dataset have more flights than others.
- The averages and standard deviations of the precipitation, visibility, pressure, cloudcover, DewPointF, WindGustKmph, tempF, WindChillF, and humidity characteristics demonstrate some fluctuation in meteorological conditions across the day.
- The maximum value for all features is 1, suggesting that the normalization was conducted correctly and the data is within the expected range of values.
- The insights from the dataset can help airlines manage their operations to reduce delays and improve customer satisfaction.

3) Unique Count Plot for each Attribute

- The column "Quarter" has 4 unique values, which is expected since it represents the quarter of the year (1-4).
- The column "Year" has 2 unique values, which could mean that the dataset includes data for two different years only.
- The column "Month" has 12 unique values, which is expected since it represents the month of the year (1-12).
- The column "DayofMonth" has 31 unique values, which is expected since it represents the day of the month (1-31).
- The column "DepTime" has 1431 unique values, which could mean that the data includes flights departing at various times throughout the day.
- The column "DepDel15" has 2 unique values (0 and 1), which represents whether the flight departed more than 15 minutes late (1) or not (0).

- The column "CRSDepTime" has 24 unique values, which could mean that the data includes flights scheduled to depart at different times of the day, possibly in hourly intervals.
- The column "DepDelayMinutes" has 1071 unique values, which represents the number of minutes a flight was delayed at departure.
- The column "ArrDel15" has 2 unique values (0 and 1), which represents whether the flight arrived more than 15 minutes late (1) or not (0).
- The columns "Origin" and "Dest" both have 15 unique values, which could mean that the data includes flights departing from and arriving at 15 different airports.
- The columns related to weather data, such as "windspeedKmph_x" and "precipMM_y", have varying numbers of unique values depending on the weather parameter and airport.
- The columns "Unnamed: 0_x" and "Unnamed: 0_y" are likely index columns and do not provide any useful information for analysis.
- The columns "date_x", "date_y", "airport_x", and "airport_y" were dropped from the dataset and are not included in the output.

4) Imbalance Dataset From Count Plot (REsampling+FEATURESelection.ipynb)

- The image is a visualization of class imbalance in a dataset. The horizontal axis represents the different classes in the dataset, and the vertical axis represents the number of instances in each class. The red bars represent the minority class, while the blue bars represent the majority class.
- From the image, it can be seen that there is a significant class imbalance in the dataset, with the majority class having a much larger number of instances than the minority class. This can be problematic for machine learning algorithms, as they may be biased towards the majority class and perform poorly in predicting the minority class. Therefore, it is important to handle class imbalance in the dataset through techniques such as resampling or cost-sensitive learning.

5) Correlation Plot (REsampling+FEATURESelection.ipynb)

The correlation matrix between the features in the dataset. The matrix shows the correlation coefficient values between each pair of features. The values range from -1 to 1, with -1 indicating a strong negative correlation, 0 indicating no correlation, and 1 indicating a strong positive correlation.

Some insights from the correlation matrix are:

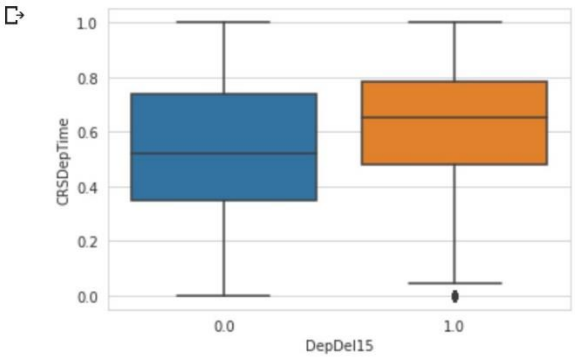
- DepDel15 (departure delay over 15 minutes) and ArrDel15 (arrival delay over 15 minutes) have a high positive correlation of 0.668, indicating that flights that are delayed at departure are likely to be delayed at arrival as well.
- DepDelayMinutes (departure delay in minutes) has a strong positive correlation with DepDel15 (departure delay over 15 minutes) of 0.949, indicating that longer delays are more likely to be classified as delays over 15 minutes.
- ArrDel15 (arrival delay over 15 minutes) has a moderate positive correlation with DepDelayMinutes (departure delay in minutes) of 0.454, indicating that longer departure delays can lead to longer arrival delays.
- Quarter has a moderate positive correlation with Year (0.968), Month (0.948), and DayofMonth (0.003), indicating that the quarter of the year is highly correlated with the year and month, but not as much with the day of the month.

- Cloudcover_x and Cloudcover_y have a moderate positive correlation with each other (0.656), indicating that the cloud cover at the departure and arrival airports are correlated.
- WindGustKmph_x and WindGustKmph_y have a moderate positive correlation with each other (0.703), indicating that the wind gusts at the departure and arrival airports are correlated.

6) Distribution Plot

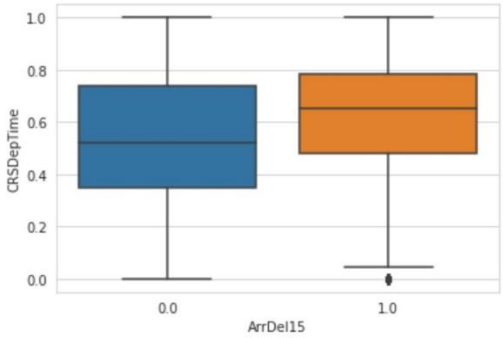
- The image appears to be a plot of the distribution of flight delays. The x-axis represents the flight delay in minutes, while the y-axis represents the number of flights that experienced delays within a particular delay range.
- The plot shows that the majority of flights experienced relatively short delays of less than 60 minutes, with the highest frequency of delays occurring in the 0-10 minute range. The plot also shows a long tail of flights that experienced longer delays, with a small number of flights experiencing very long delays of over 1000 minutes.
- Overall, this plot suggests that flight delays are relatively common but tend to be relatively short in duration, with a small number of flights experiencing very long delays.

7) Outlier Plot1



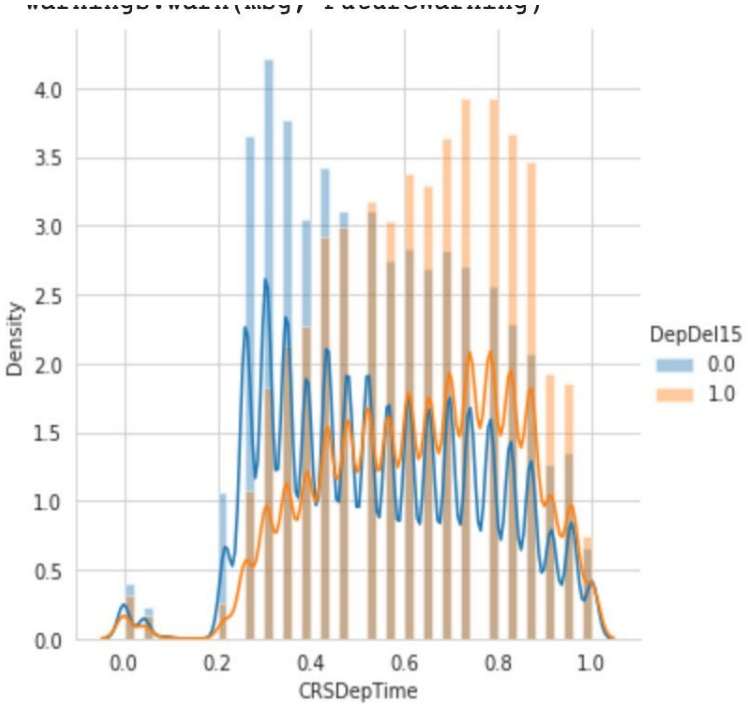
time vs departure

8) Outlier Plot 2



Time vs Arrival

9) Probability Density Function plot (Departure)



10) Feature value count Hist plot