Capstone Project 1 Milestone Report

1. Updated Problem Statement:

Aerospace is the one of the biggest industries in the world. Passengers safety perception is extremely effective on the investment's fate. In order to protect the customers' health and comfort, and the companies' investments, flight safety is always priority in the market. The statistical researches on this topic was not analysed through the view of a data scientists lately. In the researches, simpler models used with basic statistical approach. Searching for evidence and identifying the real issues is not that easy if we are talking about flight safety. It is a culture for the aviators. So, it is very similar to analyze texts with NLP to investigate a real safety issue. My original goal at the beginning was to contribute to aircraft and aviation safety management systems by analyzing and predicting one of the following questions precisely:

- By providing the conditions (such as pilot experience, weather, aircraft and mission type, phase of flight, location) under the classification of consequences and factors of the issue, can we predict how likely to have a flight safety issue?
- Which parts of the flight have more risk in the terms of safety issue?
- Is there a seasonal pattern on safety incidents depending on date, location or weather?

The results can contribute to pilot training, aircraft design, risk assessment of operations in airline companies. A proactive approach can be developed for air traffic control system as well as a reasonable allocation of resources can be conducted for safety purposes. But after having further analysis; it turns out that the dataset I have is not the best fit for that questions. The first step should be to understand what are the basic categories for safety issues and what are the root causes to that undesired safety incidents. Having said that, the problem statement after data cleaning and data wrangling processes will be modified to the following statement:

- Can we predict the reason of air conflict safety issues by analyzing crew statements?

And the updated title of the project will be;

- A classification model on causes of air conflicts by air crew narratives.

2. Obtaining Data:

Aviation Safety Network is providing up-to-date, complete and reliable authoritative information on airliner incidents and safety issues to everyone. The Aviation Safety Reporting System (ASRS) database is the world's largest repository of voluntary, confidential safety information provided by aviation's frontline personnel, including pilots, controllers, mechanics, flight attendants, and dispatchers. The database provides a foundation for specific products and subsequent research addressing a variety of aviation safety issues.

ASRS's database includes the narratives submitted by reporters (after they have been sanitized for identifying details). These narratives provide an exceptionally rich source of information for research. The database also contains coded information by expert analysts from the original report which is used for data retrieval and analyses.

https://asrs.arc.nasa.gov/overview/database.html

https://asrs.arc.nasa.gov/search/database.html

Following variables are listed in the database:

Report number (ACN), Month of incident, Location, State, Flight Conditions, Weather, Reporter Organization, Reporter Function, Flight Plan, Mission, Flight Phase, Aircraft Model, Airspace Class, Event Type, Anomaly Issue, Anomaly Procedure, Detector, Primary Problem, Contributing Factors, Human Factors, Result, Narrative, Synopsis.

One missing but pretty important variable would be the aging of aircraft and engine. Due to confidentially the data set is not including these variables. But partially maintenance condition is included.

I downloaded a data with approximately 200.000 observations in 100 columns in which has texts in some of them. I am following the explained steps below:

- The website had a limitation of 5000 observation for each download. In the query page, I chose date criteria for all data to be downloaded. The first file was 2019. Then I chose January-June in the first chunk, and July-December in the second chunk each year. As a result, except 2019, 1999, 1995 and 1988, all years have two files with approximately 3500 observations. 1999 and 1995 have three files. In total 64 csv files obtained with exact same formatting.
- All files were combined by using following commands with a for loop and can be executed like;

Read from 64 csv files & Form a Dataframe

In [122]:
```
filenames=glob('database/*.csv')
data=[]
for f in filenames:
    df=pd.read_csv(f, header=[0,1], delimiter=',')
    data.append(df)
```

In [123]:
```
final_df = pd.concat(data, axis=0, ignore_index=True)
final_df.shape
```

Out[123]: (202525, 97)

3. Data Cleaning & Data Wrangling:

- I got rid of the duplicates and unnecessary columns with the following command;

```
df_nodup=final_df.drop_duplicates()
df_nodup.shape
df1=df_nodup.drop(columns=['Work Environment Factor','RVR.Single Value','Aircraft Zone', 'Maintenance Status.Maintenance Deferred','Maintenance Status.Records Complete', 'Maintenance Status.Released For Service','Maintenance Status.Required / Correct Doc On Board','Maintenance Status.Maintenance Type','Maintenance Status.Maintenance Items Involved', 'Cabin Lighting','Crew Size Flight Attendant.Number Of Crew','Callback','When Detected'], level=1)

df1=df_nodup.drop(columns=['Report 2','Unnamed'], level=0)
```

- In order to classify the data frame according to “Events” column, I created multiple data frames. The classification of the data is as follows:
    a. Airspace conflict issues,
    b. Abnormal equipment/activity due to emergency situation,
    c. Flight cabin event,
    d. Ground issues.

  In events column 55 different explanation word taxonomy used. Each taxonomy with different combination falls under one of the classifications above. I wrote a function for this categorization as below and applied it for Anomaly column:

```
Types=['Ground','Emergency','Conflict','Cabin']
def categorize(dataframe,column_name,new_column='Type'):
    dataframe[new_column]='Not Categorized'
    dataframe.index.fillna('Empty')
    dataframe.fillna('Empty')
    for index, row in dataframe.iterrows():
        if 'Equipment' in dataframe[column_name][index]:
            dataframe[new_column][index]='Emergency'
        elif 'Ground' in dataframe[column_name][index]:
            dataframe[new_column][index]='Ground'
        elif 'Clearence' in dataframe[column_name][index]:
            dataframe[new_column][index]='Conflict'
        elif 'Conflict' in dataframe[column_name][index]:
            dataframe[new_column][index]='Conflict'
        elif 'Deviation' in dataframe[column_name][index]:
            dataframe[new_column][index]='Conflict'
        elif 'Cabin' in dataframe[column_name][index]:
            dataframe[new_column][index]='Cabin'
        else:
            dataframe[new_column][index]='Not Categorized'
```

- I created a new dataframe with only “conflict” kind of safety issues which are including air traffic conflicts.

Conflict 91669
Emergency 65320
Ground 23084
Not Categorized 20544
Cabin 1908

- In order to get rid of two level column names and get more useful column names, I applied the following code:

```
column_names=list(df_c1.columns)
del column_names[-1]
del column_names[-1]
df_new={}
for l1,l2 in column_names:
   if df_c1[l1,l2].dtype=='object':
      df_t=df_c1[l1,l2].fillna('None')
      column_t=str(l1)+'_'+str(l2)
      df_new[column_t]=df_t
   if df_c1[l1,l2].dtype=='float64' or df_c1[l1,l2].dtype=='int64':
      df_t=df_c1[l1,l2].fillna(0)
      column_t=str(l1)+'_'+str(l2)
      df_new[column_t]=df_t
len(df_new)
column_new_names=['_No','Month', 'Time', 'Place_refer', 'State', 'Radial', 'Distance', 'AGL', 'MSL', 'MCondition', 'Visibility','Light', 'Ceiling', 'AC1_ATC', 'AC1_Operator', 'AC1_Model', 'AC1_Crew', 'AC1_Rule', 'AC1_FP', 'AC1_Mission', 'AC1_Nav', 'AC1_Phase', 'AC1_Route', 'AC1_Airspace', 'AC1_Seats','AC1_Passengers','AC2_ATC','AC2_Operator', 'AC2_Model', 'AC2_Crew', 'AC2_Rule', 'AC2_FP', 'AC2_Mission', 'AC2_Nav', 'AC2_Phase', 'AC2_Route','AC2_Airspace','AC2_Seats', 'AC2_Passengers', 'P1_Loc', 'P1_Org', 'P1_Func', 'P1_Qual','P1_Experience','P1_HumaFactor','P2_Loc','P2_Org', 'P2_Func', 'P2_Qual', 'P2_Experience', 'Anomaly', 'Detector', 'Result', 'Other_Factors', 'Pri_Problem', 'Narrative', 'Synopsis']
data_clean.columns=column_new_names
data_clean.head()
```

- For missing values; I used the “categorize” function that I defined in 6th bullet. If the Dtype of the column is object then the empty cells will be filled with “None”. It the column is float or integer the empty cells are filled with 0. The concerning lines are as follows:

```
if df_c1[l1,l2].dtype=='object':
      df_t=df_c1[l1,l2].fillna('None')
      column_t=str(l1)+'_'+str(l2)
      df_new[column_t]=df_t
   if df_c1[l1,l2].dtype=='float64' or df_c1[l1,l2].dtype=='int64':
      df_t=df_c1[l1,l2].fillna(0)
      column_t=str(l1)+'_'+str(l2)
      df_new[column_t]=df_t
```

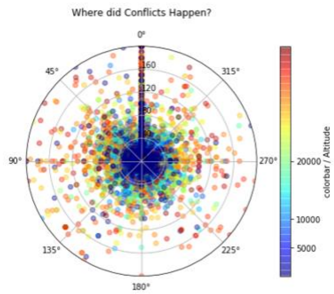
- For each column I applied different codes to clean data. Outliers of each column treated differently. If there is an error, it needs to be checked individually. Examples of some columns are as follows:

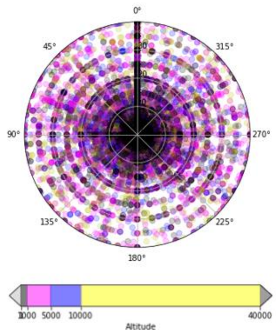
```
empty_time=['None','ZZZ']
for item in empty_time:
   for index,row in df.iterrows():
      if df.Time[index]==item:
         if df.Light[index]=='Daylight':
            df.Time[index]=='0601-1200'
         elif df.Light[index]=='Night':
            df.Time[index]=='0001-0600'
……………….
df.AGL=pd.to_numeric(df.AGL,errors='coerce')
df.MSL=pd.to_numeric(df.MSL,errors='coerce')
……………….
df.State=df.State.str.upper()
df[df.State=='US.AIRPORT']['State']
df.loc[40984, 'State']='US'
……………….
position=df[~((df.Distance==0) & (df.Radial==0))][['Distance','Radial','Altitude']]
……………….
Phase_df=df.AC1_Phase.str.split(';')
Phase_df=Phase_df.str.get(0)
df['AC1_Phase']=Phase_df
df.AC1_Phase.value_counts().head(11)
…………………..
Airspace=df[~(df.AC1_Airspace=='None')]['AC1_Airspace'].value_counts()
#Airspace.index=Airspace.index.str.split('')
Airspace.index=Airspace.index.str.get(6)
Airspace=Airspace.groupby(Airspace.index).sum().sort_values(ascending=False).head(6)
Airspace.index='Class '+Airspace.index

    Airspace= Airspace.sort_index()
```

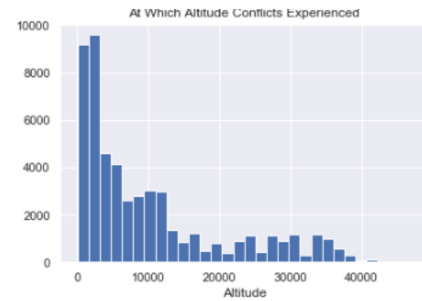
4. Initial Findings:

- After classifying the safety incidents in four categories, air conflict issues were being chosen and the columns which has less than 10% information dropped. As a result, the new data set shape is 91670 x 58.
- Then I started to research on altitude, location (radial, distance columns), time, light, phase, airspace class, month, year, state, primary cause and narrative (synopsis) variables.
- Between all those variables there were inconsistencies because of having empty cells in each questionnaire. The tendency of the crew was using narrative and synopsis sections but arbitrary inputs for the remaining cells. For that reason, most of the variables has a big chunk of “None”, “0” or “NaN” inputs.
- Radial, distance and altitude show us where the exact location of conflict issue. At least one of this three variables might have a “0” value since it was empty in original input file.

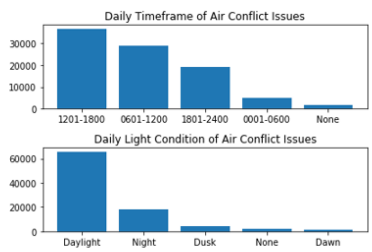
I tried to visualize those three on a polar chart. “0” radial, “0” distance or “0” altitude has more data then the others. It creates a kind of unreliability on those data.



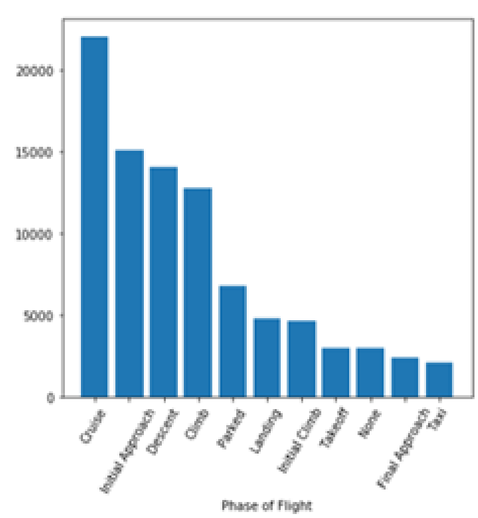




- One other interaction analysis between the variables was time and light variables. Those two has similar characteristics by nature. The effects on the conflict issues should be same and this makes it redundant.



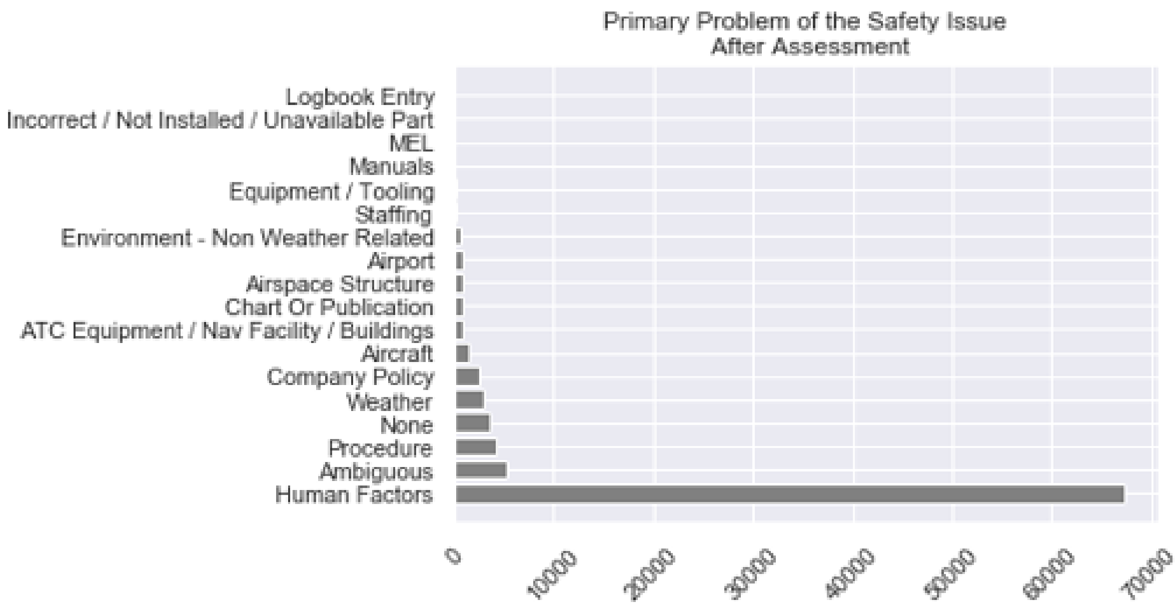
- One big problem for this dataset is all information belongs to conflict issues. And we do not have the information of other flights which has no records of safety issues. So, for dependant column all records should be “1” and obviously this is not an appropriate target variable. To predict the safety issues won’t be possible with the current database. Then, I focused on “Phase”, “Anomaly” and “Pri_Problem” columns as the dependant variables.



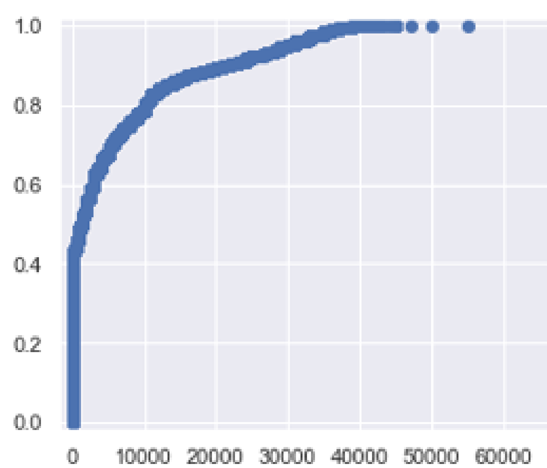
a. Phases: This variable represents the phase of flight such as “take-off”, “taxi”, “cruise” or “landing” and is highly inter-related with the other location variables and has no relationship with time, light, meteorology… etc. variables. Additionally, to predict this variable by location variables won’t provide any kind of value-added to the customer. Because location variables are showing the parameters relative to airports. It is a different way to define phase.
b. Anomaly: During the data wrangling part, the anomaly variable was used to categorize the data and between four categories “air

conflict" issues were selected. And now the target column become more homogenous. Also, other groups of variables are not strongly related with "Anomaly" column.

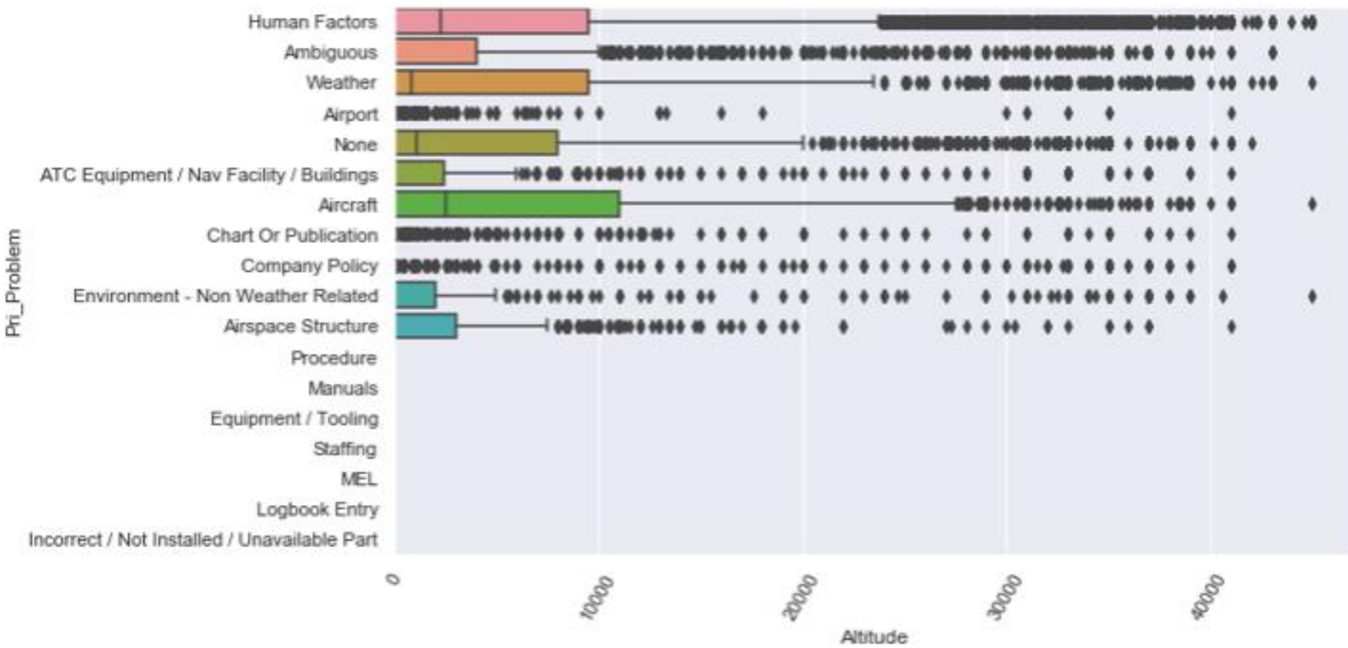
c. Pri_Problem: This variable represents the root cause of the conflict issue. The biggest chunk of data is Human Factors. This variable is the best candidate for target column. Because, firstly there is a huge value-added in this research to understand the safety issue's reason. Second, the data in this column is pretty consistent and pure analyzed through human experiences. The next step should be to understand the relations with other variables.

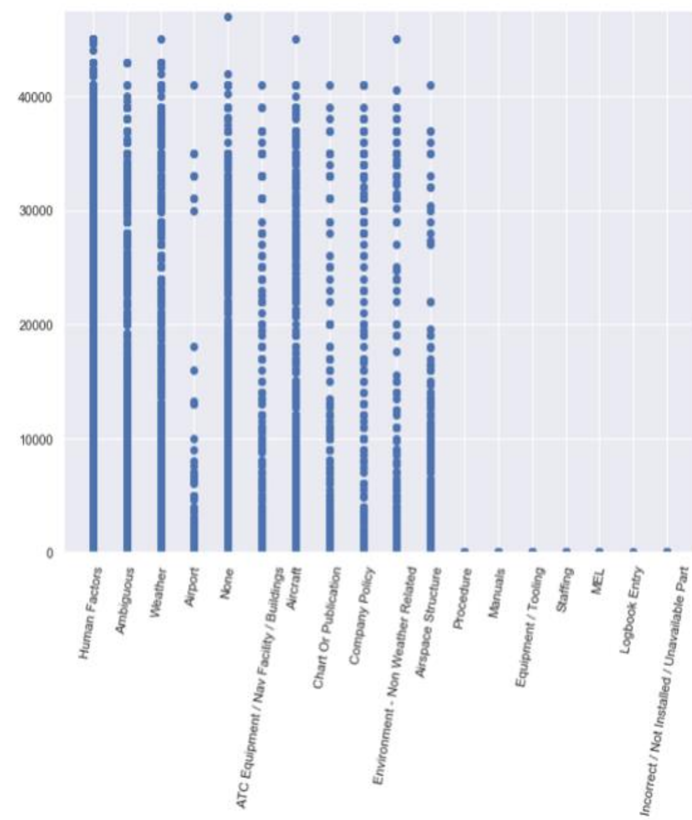


- If we analyse the relationship between independent variables and Pri_Problem column, the findings were not so strong. As an example, I will show "Altitude" variable distribution on target column.
- Emprical Cumulative Distribution Function of "Altitude" column is depicted below:



- The boxplot and scatterplot of Altitude and Pri_Problem column is as follows:





As a result; the most suitable model for this project would provide the root cause analysis for aviation safety issues. Pri_Problem column will be the target (dependant) variable. But still there is not a very powerful connection with independent variable and dependant variable. Further research needed for a better machine learning model. Other options can be “Narrative” and “Synopsis” columns as the independent variables. But those columns are not numerical and the data is formed by texts. My decision will be going on those variables to find a decoding as numerical and formulate a relationship with the target column.

The further stage for this study will be NLP model on Narrative and Synopsis columns to predict primary cause of Air Conflict Safety Issues.