IDS 575 Machine Learning Statistics Final Project Report

Project on Flight Delay Prediction

Professor: Dr. Theja Tulabandhula

Date of Submission:17/04/2022

Group Members: Anand Sabbineni, Srinivas Gundluri ,Shivani Erigineni, Keerthan Devineni.

**Aim**

This project is aimed at developing a model to predict weather delays in airports which in turn result in flight delays. In this project we used past data to train the model itself using supervised learning algorithms. The model is scaled to improve the accuracy. Then we used StackingCVRegressor to select the best prediction from the individual outcomes of models.

**Data**

We have collected 3 years of flight data and weather data and joined both data sets to get the final data set containing weather attributes and weather delay.

**Attributes of Dataset**

Independent variables/Explanatory variables

1. Temperature
2. Humidity
3. Dew point
4. Wind speed
5. Precipitation
6. Pressure
7. Month and hours

Dependent variable

1. Weather Delay: target variable which gives approximate flight delay due to weather conditions given the independent variables.

Sample screenshots of data
Final data.csv (flight data+weather data)

Final dataset:

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Temperatu | Dew Point | Humidity | Wind Spee | Pressure | Precipitation | month | hours | WEATHER_DELAY | | | |
| 2 | 0 | 25 | 8 | 49 | 12 | 29.36 | 0 | 1 | 18 | 0 | | | |
| 3 | 1 | 19 | 5 | 54 | 13 | 29.27 | 0 | 1 | 7 | 0 | | | |
| 4 | 2 | 28 | 11 | 49 | 9 | 29.33 | 0 | 1 | 15 | 0 | | | |
| 5 | 3 | 27 | 10 | 49 | 13 | 29.35 | 0 | 1 | 17 | 0 | | | |
| 6 | 4 | 19 | 6 | 57 | 15 | 29.36 | 0 | 1 | 10 | 0 | | | |
| 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0 | | | |
| 8 | 6 | 19 | 6 | 57 | 15 | 29.36 | 0 | 1 | 10 | 0 | | | |
| 9 | 7 | 23 | 8 | 53 | 13 | 29.34 | 0 | 1 | 13 | 0 | | | |
| 10 | 8 | 20 | 7 | 57 | 10 | 29.37 | 0 | 1 | 11 | 0 | | | |
| 11 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0 | | | |
| 12 | 10 | 24 | 8 | 51 | 14 | 29.38 | 0 | 1 | 19 | 0 | | | |
| 13 | 11 | 25 | 9 | 50 | 12 | 29.34 | 0 | 1 | 14 | 0 | | | |
| 14 | 12 | 27 | 10 | 49 | 13 | 29.35 | 0 | 1 | 17 | 0 | | | |
| 15 | 13 | 21 | 6 | 53 | 17 | 29.23 | 0 | 1 | 6 | 0 | | | |
| 16 | 14 | 28 | 11 | 49 | 9 | 29.33 | 0 | 1 | 15 | 0 | | | |
| 17 | 15 | 19 | 6 | 57 | 15 | 29.36 | 0 | 1 | 10 | 68 | | | |
| 18 | 16 | 19 | 6 | 57 | 9 | 29.39 | 0 | 1 | 12 | 0 | | | |
| 19 | 17 | 28 | 11 | 49 | 9 | 29.33 | 0 | 1 | 15 | 0 | | | |
| 20 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0 | | | |
| 21 | 19 | 21 | 7 | 55 | 13 | 29.39 | 0 | 1 | 22 | 0 | | | |
| 22 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0 | | | |
| 23 | 21 | 21 | 7 | 55 | 13 | 29.39 | 0 | 1 | 22 | 0 | | | |
| 24 | 22 | 21 | 7 | 55 | 13 | 29.39 | 0 | 1 | 22 | 0 | | | |

Weather dataset:

| A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | Temperatu | Dew Point | Humidity | Wind Spee | Wind Gust | Pressure | Precipitatic | Wind | Condition | Date |
| 0 | 2:22 AM | 45 | 38 | 76 | 15 | 0 | 29.06 | 0 | WNW | Cloudy | 01-01-2016 |
| 1 | 3:22 AM | 43 | 37 | 80 | 14 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 |
| 2 | 4:22 AM | 42 | 36 | 79 | 14 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 |
| 3 | 5:22 AM | 42 | 35 | 76 | 13 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 |
| 4 | 6:22 AM | 41 | 35 | 79 | 15 | 0 | 29.05 | 0 | NNW | Cloudy | 01-01-2016 |
| 5 | 6:22 AM | 41 | 35 | 79 | 15 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 |
| 6 | 7:22 AM | 40 | 34 | 79 | 15 | 0 | 29.07 | 0 | NW | Cloudy | 01-01-2016 |
| 7 | 8:22 AM | 40 | 34 | 79 | 14 | 0 | 29.09 | 0 | NW | Mostly Clou | 01-01-2016 |
| 8 | 9:22 AM | 40 | 33 | 77 | 13 | 0 | 29.11 | 0 | NW | Cloudy | 01-01-2016 |
| 9 | 10:22 AM | 41 | 33 | 73 | 13 | 0 | 29.12 | 0 | NW | Mostly Clou | 01-01-2016 |
| 10 | 11:22 AM | 45 | 34 | 65 | 8 | 0 | 29.11 | 0 | NW | Cloudy | 01-01-2016 |
| 11 | 12:22 PM | 45 | 34 | 65 | 10 | 0 | 29.12 | 0 | WNW | Cloudy | 01-01-2016 |
| 12 | 1:22 PM | 45 | 34 | 65 | 14 | 0 | 29.1 | 0 | NW | Cloudy | 01-01-2016 |

| A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unnamed: | Time | Temperatu | Dew Point | Humidity | Wind Spee | Wind Gust | Pressure | Precipitatic | Wind | Condition | Date | time |
| 0 | 0 | 2:22 AM | 45 | 38 | 76 | 15 | 0 | 29.06 | 0 | WNW | Cloudy | 01-01-2016 | 01-01-2016 02:22 |
| 1 | 1 | 3:22 AM | 43 | 37 | 80 | 14 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 03:22 |
| 2 | 2 | 4:22 AM | 42 | 36 | 79 | 14 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 04:22 |
| 3 | 3 | 5:22 AM | 42 | 35 | 76 | 13 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 05:22 |
| 4 | 4 | 6:22 AM | 41 | 35 | 79 | 15 | 0 | 29.05 | 0 | NNW | Cloudy | 01-01-2016 | 01-01-2016 06:22 |
| 5 | 5 | 6:22 AM | 41 | 35 | 79 | 15 | 0 | 29.05 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 06:22 |
| 6 | 6 | 7:22 AM | 40 | 34 | 79 | 15 | 0 | 29.07 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 07:22 |
| 7 | 7 | 8:22 AM | 40 | 34 | 79 | 14 | 0 | 29.09 | 0 | NW | Mostly Clou | 01-01-2016 | 01-01-2016 08:22 |
| 8 | 8 | 9:22 AM | 40 | 33 | 77 | 13 | 0 | 29.11 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 09:22 |
| 9 | 9 | 10:22 AM | 41 | 33 | 73 | 13 | 0 | 29.12 | 0 | NW | Mostly Clou | 01-01-2016 | 01-01-2016 10:22 |
| 10 | 10 | 11:22 AM | 45 | 34 | 65 | 8 | 0 | 29.11 | 0 | NW | Cloudy | 01-01-2016 | 01-01-2016 11:22 |

Flight Dataset:

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FL_DATE | ORIGIN | DEST | CRS_DEP_T | DEP_TIME | DEP_DELAY | WHEELS_O | WHEELS_O | TAXI_IN | CRS_ARR_T | ARR_DELAY | CANCELLED | DIVERTED | CRS_ELAPS | ACTUAL_EL |
| 0 | 01-01-2016 | LAX | ATL | 2255 | 2256 | 1 | 2315 | 542 | 5 | 600 | -13 | 0 | 0 | 245 | 231 |
| 1 | 01-01-2016 | SLC | ATL | 1656 | 1700 | 4 | 1712 | 2205 | 8 | 2229 | -16 | 0 | 0 | 213 | 193 |
| 2 | 01-01-2016 | BNA | ATL | 1320 | 1446 | 86 | 1501 | 1638 | 6 | 1530 | 74 | 0 | 0 | 70 | 58 |
| 3 | 01-01-2016 | JAX | ATL | 1145 | 1144 | -1 | 1156 | 1239 | 8 | 1302 | -15 | 0 | 0 | 77 | 63 |
| 4 | 01-01-2016 | MDT | ATL | 1737 | 1727 | -10 | 1738 | 1923 | 11 | 1949 | -15 | 0 | 0 | 132 | 127 |
| 5 | 01-01-2016 | SAV | ATL | 1408 | 1403 | -5 | 1418 | 1459 | 6 | 1523 | -18 | 0 | 0 | 75 | 62 |
| 6 | 01-01-2016 | BUF | ATL | 615 | 612 | -3 | 650 | 844 | 5 | 843 | 6 | 0 | 0 | 148 | 157 |
| 7 | 01-01-2016 | PNS | ATL | 1435 | 1431 | -4 | 1441 | 1622 | 6 | 1648 | -20 | 0 | 0 | 73 | 57 |
| 8 | 01-01-2016 | CMH | ATL | 1123 | 1117 | -6 | 1130 | 1244 | 7 | 1302 | -11 | 0 | 0 | 99 | 94 |
| 9 | 01-01-2016 | DAL | ATL | 1005 | 1016 | 11 | 1026 | 1252 | 8 | 1304 | -4 | 0 | 0 | 119 | 104 |
| 10 | 01-01-2016 | MEM | ATL | 1600 | 1555 | -5 | 1603 | 1751 | 7 | 1826 | -28 | 0 | 0 | 86 | 63 |
| 11 | 01-01-2016 | SEA | ATL | 745 | 742 | -3 | 802 | 1518 | 6 | 1527 | -3 | 0 | 0 | 282 | 282 |
| 12 | 01-01-2016 | RDU | ATL | 1100 | 1104 | 4 | 1115 | 1216 | 5 | 1224 | -3 | 0 | 0 | 84 | 77 |

| A | Q | R | S | T | U | V | W | X | Y | Z | AA | AB | AC | AD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AIR_TIME | DISTANCE | CARRIER_D | WEATHER_ | LATE_AIRC | Temperatu | Dew Point | Humidity | Wind Spee | Wind Gust | Pressure | Precipitatic | Condition | time |
| 0 | 207 | 1947 | | | | 42 | 27 | 55 | 16 | 0 | 29.14 | 0 | Cloudy | 01-01-2016 22:56 |
| 1 | 173 | 1590 | | | | 45 | 31 | 58 | 15 | 0 | 29.09 | 0 | Mostly Clou | 01-01-2016 17:00 |
| 2 | 37 | 214 | 3 | 0 | 71 | 46 | 34 | 63 | 12 | 0 | 29.07 | 0 | Cloudy | 01-01-2016 14:46 |
| 3 | 43 | 270 | | | | 45 | 34 | 65 | 8 | 0 | 29.11 | 0 | Cloudy | 01-01-2016 11:44 |
| 4 | 105 | 620 | | | | 45 | 31 | 58 | 15 | 0 | 29.09 | 0 | Mostly Clou | 01-01-2016 17:27 |
| 5 | 41 | 214 | | | | 46 | 34 | 63 | 12 | 0 | 29.07 | 0 | Cloudy | 01-01-2016 14:03 |
| 6 | 114 | 712 | | | | 41 | 35 | 79 | 15 | 0 | 29.05 | 0 | Cloudy | 01-01-2016 06:12 |
| 7 | 41 | 271 | | | | 46 | 34 | 63 | 12 | 0 | 29.07 | 0 | Cloudy | 01-01-2016 14:31 |
| 8 | 74 | 447 | | | | 45 | 34 | 65 | 8 | 0 | 29.11 | 0 | Cloudy | 01-01-2016 11:17 |
| 9 | 86 | 721 | | | | 41 | 33 | 73 | 13 | 0 | 29.12 | 0 | Mostly Clou | 01-01-2016 10:16 |
| 10 | 48 | 332 | | | | 45 | 33 | 63 | 13 | 0 | 29.05 | 0 | Cloudy | 01-01-2016 15:55 |
| 11 | 256 | 2182 | | | | 40 | 34 | 79 | 15 | 0 | 29.07 | 0 | Cloudy | 01-01-2016 07:42 |
| 12 | 61 | 356 | | | | 45 | 34 | 65 | 8 | 0 | 29.11 | 0 | Cloudy | 01-01-2016 11:04 |

**Data Pre-Processing:**

Changing the time format:

```
def preprocess(year):
    for i in range(1,13):
    if(i<10):
        month = "0"+str(i)
    else:
        month = str(i)
    #print(month)
    if((i==1) | (i==3) | (i==5) | (i==7) | (i==8) | (i==10) | (i==12)):
        for j in range(1,32):
            if(j<10):
                date = "0"+str(j)
            else:
                date = str(j)
            try:
                df= pd.read_csv(string1+"-"+month+"-"+date+'.csv')
                df['time'] = df['Time'].map(ampm)
                df['time'] = df['time']+ " " + df['Date']
                df['time'] = pd.to_datetime(df['time'],format = "%H:%M %Y-%m-%d")
                df.to_csv(string2+"-"+month+"-"+date+'.csv')
            except:
                print(date,month)
    elif(i==2):
        for j in range(1,28):
            if(j<10):
                date = "0"+str(j)
            else:
                date = str(j)
            try:
                df= pd.read_csv(string1+"-"+month+"-"+date+'.csv')
                df['time'] = df['Time'].map(ampm)
                df['time'] = df['time']+ " " + df['Date']
                df['time'] = pd.to_datetime(df['time'],format = "%H:%M %Y-%m-%d")
                df.to_csv(string2+"-"+month+"-"+date+'.csv')
            except:
                print(date,month)
    else:
        for j in range(1,31):
            if(j<10):
                date = "0"+str(j)
            else:
                date = str(j)
            try:
                df= pd.read_csv(string1+"-"+month+"-"+date+'.csv')
                df['time'] = df['Time'].map(ampm)
                df['time'] = df['time']+ " " + df['Date']
                df['time'] = pd.to_datetime(df['time'],format = "%H:%M %Y-%m-%d")
                df.to_csv(string2+"-"+month+"-"+date+'.csv')
            except:
                print(date,month)
    return 1
```

Merging the data:

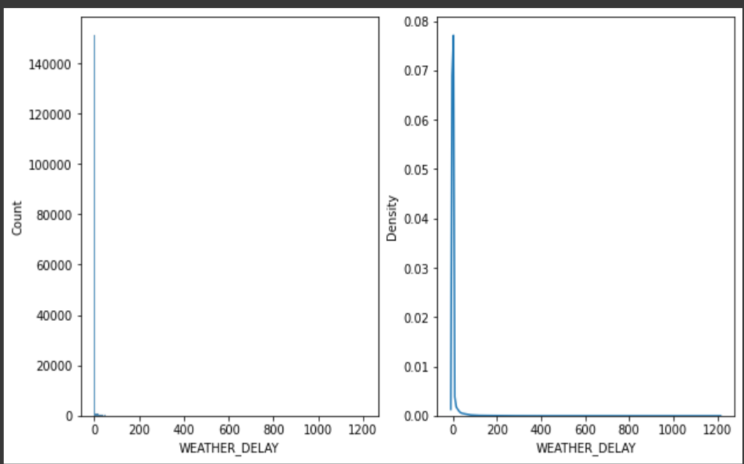
```
def merge(year):
    string1 = './'+year+".csv"
    month,date="",""
    cnt=0
    for i in range(1,13):
        if(i<10):
            month = "0"+str(i)
        else:
            month = str(i)
        #print(month)
        if((i==1) | (i==3) | (i==5) | (i==7) | (i==8) | (i==10) | (i==12)):
            for j in range(1,32):
                if(j<10):
                    date = "0"+str(j)
                else:
                    date = str(j)
                df= pd.read_csv(string1+"-"+month+"-"+date+'.csv')
                df.drop(['Time','Date'],axis=1,inplace=True)
                cnt=cnt+1
                if((i==1) & (cnt==1)):
                    df1= df
                else:
                    df1= pd.concat([df1,df])
                #print(date)
        elif(i==2):
            for j in range(1,28):
                if(j<10):
                    date = "0"+str(j)
                else:
                    date = str(j)
                df= pd.read_csv(string1+"-"+month+"-"+date+'.csv')
                df.drop(['Time','Date'],axis=1,inplace=True)
                df1= pd.concat([df1,df])
        else:
            for j in range(1,31):
                if(j<10):
                    date = "0"+str(j)
                else:
                    date = str(j)
                df= pd.read_csv(string1+"-"+month+"-"+date+'.csv')
                df.drop(['Time','Date'],axis=1,inplace=True)
                df1= pd.concat([df1,df])
    df1.drop(['Unnamed: 0','Unnamed: 0.1'],axis=1,inplace=True)
    df1.reset_index(inplace=True)
    df1.drop('index',axis=1,inplace=True)
    df1.to_csv('./merge'+year+'.csv')
    return 1
```

```
[ ] years = [2016,2017,2018]
    for i in years:
        merge(i)
```

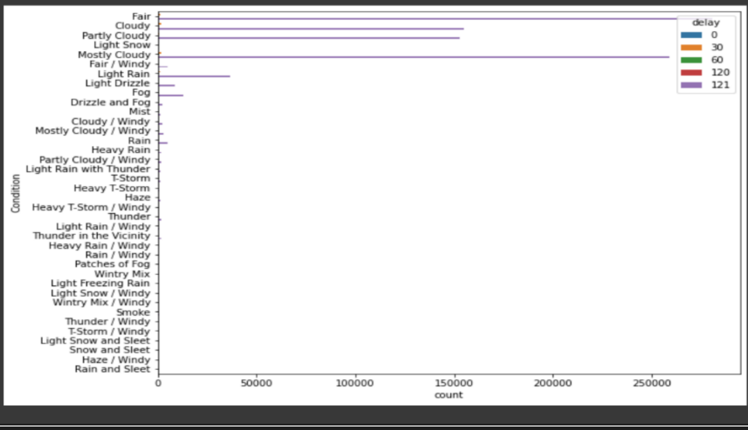
**EDA:**

We performed EDA to find the correlation between the variables. As part of EDA various graphs were plotted as below

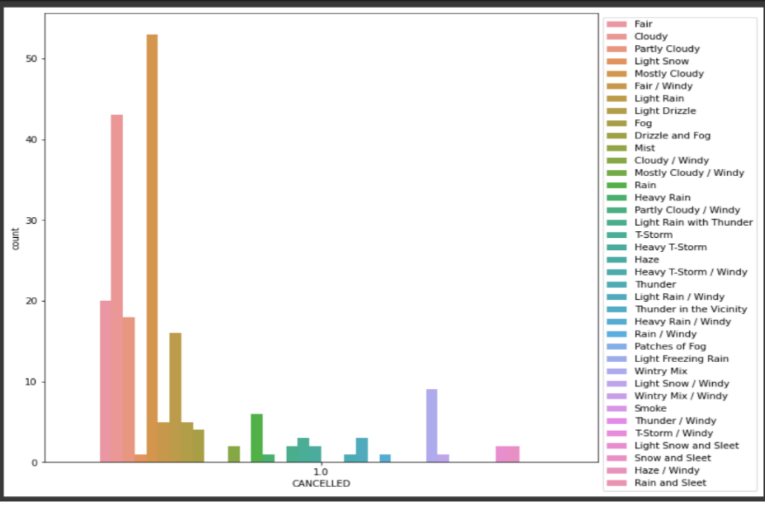
```
[ ] fig,axes = plt.subplots(ncols=2,figsize=(10,6))
    sns.histplot(ax=axes[0],x=df['WEATHER_DELAY'],bins=500)
    sns.kdeplot(ax=axes[1],x=df['WEATHER_DELAY'])
```



```
plt.figure(figsize=(10,8))
sns.countplot(y=df[df['delay']>0]['Condition'],hue=df['delay'])
```



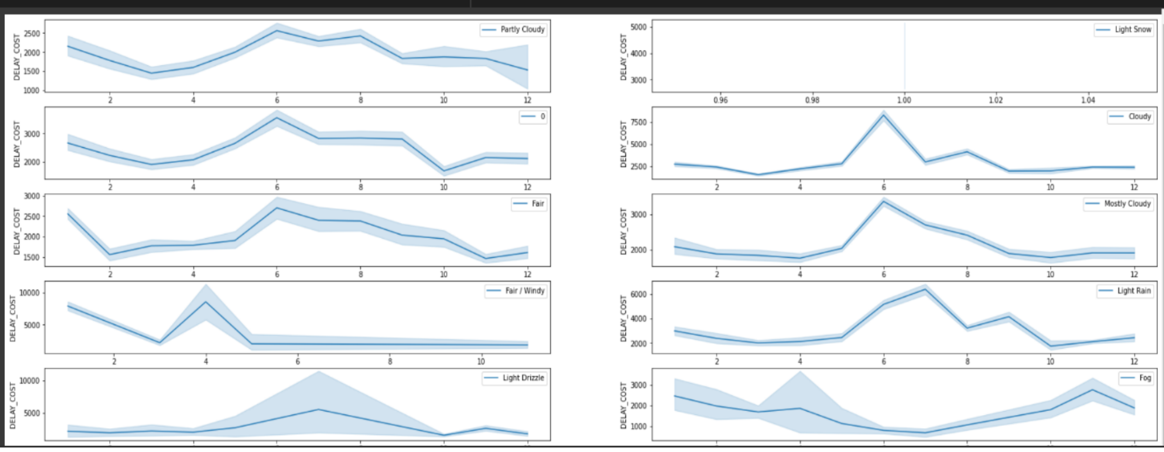

```
plt.figure(figsize=(10,10))
sns.countplot(x=df[df['CANCELLED']==1]['CANCELLED'],hue=df['Condition'])
plt.legend(bbox_to_anchor=(1,1))
```



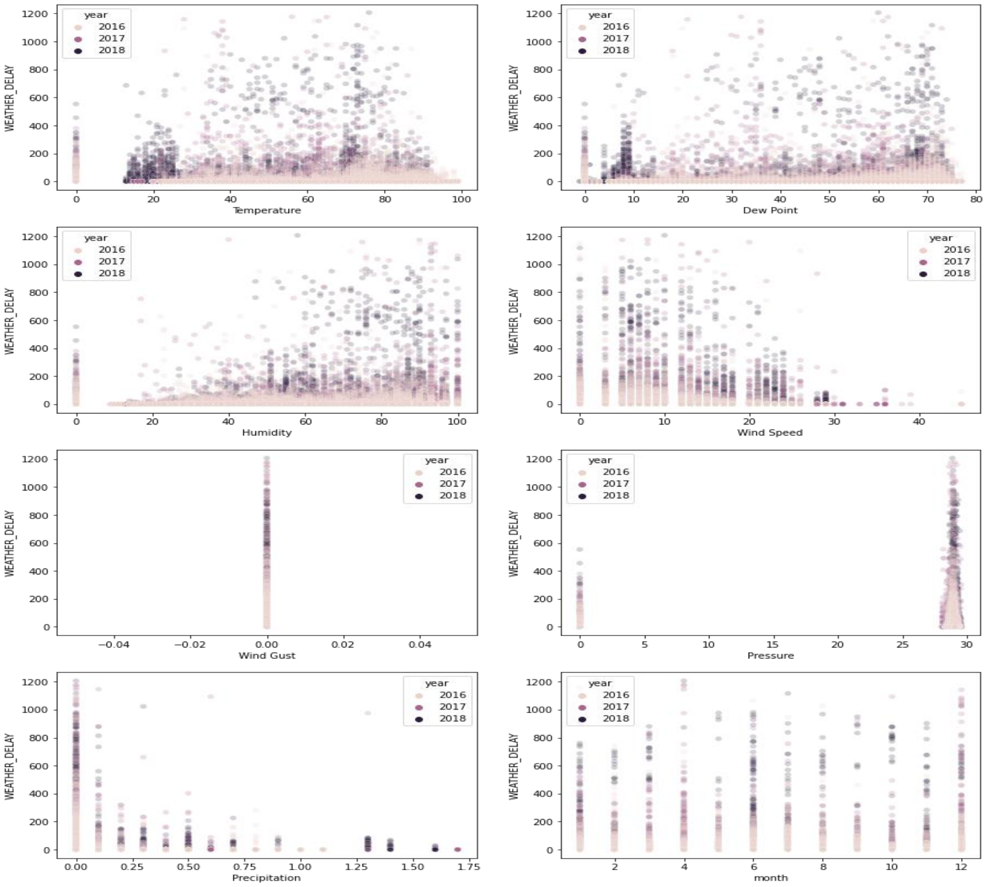

From the above plot, we can analyse that the delay tends to rise in months around June, July and August due to rain and tend to be high in December and January due to snowfall.

Line Plots of the weather conditions

```
fig,axes = plt.subplots(nrows=14,ncols = 2,figsize= (30,30))
cnt= 0
for i in range(0,14):
    for j in range(0,2):
        sns.lineplot(ax=axes[i][j],x=df[df['Condition']==conditions[cnt]]['time'].dt.month,y=df[df['Condition']==conditions[cnt]]['DELAY_COST'],label= conditions[cnt])
        cnt = cnt+1
```



```python
fig,axes = plt.subplots(nrows=4,ncols=2,figsize=(15,20))
#['Temperature','Dew Point','Humidity','Wind Speed','Wind Gust','Pressure','Precipitation','month']
sns.scatterplot(ax=axes[0][0],x=df['Temperature'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[0][1],x=df['Dew Point'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[1][0],x=df['Humidity'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[1][1],x=df['Wind Speed'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[2][0],x=df['Wind Gust'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[2][1],x=df['Pressure'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[3][0],x=df['Precipitation'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
sns.scatterplot(ax=axes[3][1],x=df['month'],y=df['WEATHER_DELAY'],alpha=0.2,hue=df['year'])
```




## Model and Methodology:

An ensemble model is used where multiple models with different algorithms are used to build a single best model to predict the weather delay, which would give information about the flight delays.

Different types of models used are discussed in detail below:

Lasso Model:

Lasso model is like the linear regression model but would allow regularizing coefficients to avoid overfitting. Linear regression chooses the best model to minimize the RSS and to it, the lasso regression, adds a penalizing factor to the least square, and this way the regularized model may have a slightly high bias but less variance for future prediction than linear regression

To find the best values for Lamba (penalizing factor), a module of Sklearn model_selection package GridSearchCV is used for hyperparameter tuning. GridSearchCV goes through various values and combinations of the hyperparameters and fits the model on the training dataset

```
# lasso model
alphas = 10**np.arange(-7,0,0.1)
params = {"alpha":alphas}
lassocv = GridSearchCV(Lasso(max_iter=1e7),param_grid=params,verbose = 5)
lassocv.fit(x_train,ytrain)
lassomodel = Lasso(alpha = lassocv.best_params_['alpha'],max_iter=1e7)
```

Random Forest Regressor:

The next ensemble learning technique used is Random Forest Regressor. This technique uses prediction based on the trees as it is more accurate since it takes many predictions because it uses the average value. This also guarantees stability as any changes in the dataset wouldn't impact the forest of trees but only to one tree where the change is made.

Random forest regressor parameters included in our dataset are n_estimators and max_depth and it is used to get the less biased and low variance result with the use of the ensembling method.

In our code we employed it as follows:

RandomForestRegressor(n_estimators=200 , max_depth=15)

```
#random forest regressor
rfc = RandomForestRegressor()
rfc.fit(x_train,ytrain)
rfcpred = rfc.predict(x_test)
```

```
# xgb random forest regression
xgrmodel = XGBRFRegressor(gamma=10)
xgrmodel.fit(x_train,ytrain)
xgrpred = xgrmodel.predict(x_test)
```

Extreme Gradient Boost Regressor:

XGboost regressor is similar to the random forest, in this the model makes various decision trees to predict and the learning rate is based on the descent, it adjusts itself in each boosting round. In order to find the best values for learning rate, a module of Sklearn model_selection package GridSearchCV is used for hyperparameter tuning.

GridCVSearch is used for finding the best learning rate for the algorithm to work.

```
#xgb regressor
lrate = 10**(np.arange(-3,0.2,0.01))
cvxg = GridSearchCV(XGBRegressor(n_estimators=150),param_grid={"learning_rate":lrate},verbose=5).fit(x_train,ytrain)
xgbmodel = XGBRegressor(n_estimators=150,learning_rate=cvxg.best_params_['learning_rate'])
xgbmodel.fit(x_train,ytrain)
xgbpred = xgbmodel.predict(x_test)
```

**Final Model:**

To get at our solution, we had to go beyond linearity. Using the StackingCVRegressor, we employed the ensembling / stacking approach to acquire the best results.

```
stack = StackingCVRegressor(regressors=(lassomodel, rfc, xgrmodel, xgbmodel),
                            meta_regressor=xgbmodel, cv=10,
                            use_features_in_secondary=True,
                            store_train_meta_features=True,
                            shuffle=False,
                            random_state=42)
```

The final model we utilized was a mixture of all the forecasts, and the final projection was the average of all the weather delays projected.

**Further Improvements in development of the current project**

We aimed at making our model as flexible as possible and in doing so we avoided including various other factors such as security delay, airline maintenance delay etc. that contribute to airline delays at airport. Model can be improved by considering those other factors and training the model. While we tried regression algorithms, we tried to get the best fit by tuning the parameters.

**Group Member Contributions:**

Data Preparation was performed by Anand Sabineni, Keerthan Devineni

EDA by Shivani Erigineni , Srinivas Gundluri

Model Building, Evaluation and Analysis by Anand Sabineni, Keerthan Devineni, Shivani Erigineni, Srinivas Gundluri