# Flight Delay Analysis

Katy Ly,

Akshat Patil,

Michelle Su,

Chieh-Hsin Wu,

Saurabh Yadav

**Table of Contents**

## Motivation

### *Background*

Travelling has always been a popular activity for most of the population, whether it is traveling for leisure, business, or other personal reasons. There is a wide selection of travel destinations, airlines, and time periods to choose from. However, depending on unexpected circumstances, flights can be delayed. Delays can occur due to weather conditions, security, aircraft maintenance, etc. Being able to predict flight delays and the reason behind them can help potential travelers plan their trips more strategically.

### *Business Problem*

It is estimated that there are anywhere between 7,782 and 8,755 commercial planes in the air on average at any given time these days. Thus, with such an abundant number of aircraft, we can see Delays in landing at a particular Airport. Not only this, but several other factors would also delay the flight. Our motivation is to provide a solution that could reduce the number of delays by predicting the factors that caused this event.

### *Significance of Problem*

From an airport or Flight Company's point of view, we can see that such delays can cost a considerable amount to the organization. Thus, predicting the reason which causes such delays would help the Air Industry save money and minimize the delays and work efficiently.

## Data Description

The original dataset includes 484,551 rows and 29 columns of data. After cleaning our data (as will be discussed next in our report, we end up having 437,902 rows and 33 columns of data. We've sourced our data second handed from https://www.kaggle.com/datasets/undersc0re/flight-delay-and-causes.

Original dataset:

| Variable Name | Data Type | Description |
|---|---|---|
| DayOfWeek | integer | 1 (Monday) - 7 (Sunday) |
| Date | object | Scheduled date |
| DepTime | object | Actual departure time (local, hhmm) |
| ArrTime | object | Actual arrival time (local, hhmm) |
| CRSArrTime | object | Scheduled arrival time (loca, hhmm) |
| UniqueCarrier | object | Unique carrier code |
| Airline | object | Airline company |
| FlightNum | integer | Flight number |
| TailNum | object | Plane tail number |
| ActualElapsedTime | integer | Actual time airplane spends in air (min) |
| CRSElapsedTime | integer | Estimated elapsed time of flight (min) |
| AirTime | integer | Flight time (min) |
| ArrDelay | integer | Difference in min between scheduled and actual arrival time |
| DepDelay | integer | Departure delay |
| Origin | object | Origin airport code |
| Org_Airport | object | Origin airport name |
| Dest | object | Destination airport code |
| Dest_Airport | object | Destination airport name |
| Distance | integer | Distance between airports (miles) |
| TaxiIn | integer | Wheels down and arrival at destination airport gate (min) |
| TaxiOut | integer | Time elapsed between departure from origin airport gate and wheels off (min) |
| Cancelled | integer | Was the flight cancelled? |
| CancellationCode | object | Reason for cancelling |
| Diverted | integer | 1 = yes, 0 = no |
| CarrierDelay | integer | Flight delay due to carrier (in min = yes, 0 = no) |
| WeatherDelay | integer | Flight delay due to weather (in min = yes, 0 = no) |

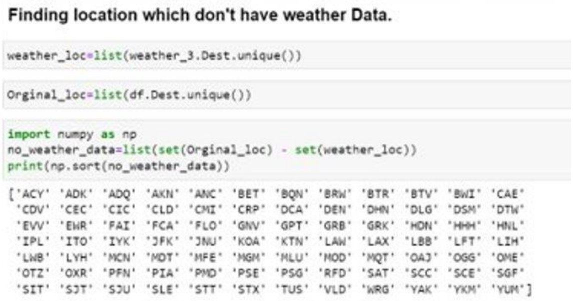
| | | |
|---|---|---|
| NASDelay | integer | Flight delay by NSA (in min = yes, 0 = no) |
| SecurityDelay | integer | Flight delay by security (in min = yes, 0 = no) |
| LateAircraftDelay | integer | Flight delay by late aircraft (in min = yes, 0 = no) |

**EDA**

*Data Cleaning*

1st EDA:

- Dropped columns and null values
  - Cancelled, Cancellation Code, and Diverted have a single value
  - Org Airport, Dest Airport gave the same info as Origin and Dest
  - Taxi In + Taxi out information is covered in Actual Elapsed Time
- Checked records to see if any of the flights arrived on time (found none)
  - Adjust project goal to predict whether a flight is delayed due to weather or not
- Create a new dataset with destination of the flight and date of the flights (use for scrapping weather records for each destination airport for the month)
- Web scrapped (we used: https://www.wunderground.com/)
  - From 274 destination airport and ~84 destination airport data was not available



**Finding location which don't have weather Data.**

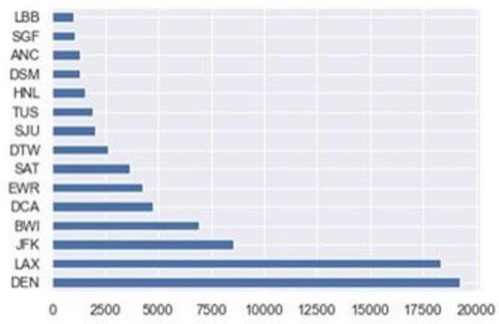
```
weather_loc=list(weather_3.Dest.unique())
```

```
Orginal_loc=list(df.Dest.unique())
```

```
import numpy as np
no_weather_data=list(set(Orginal_loc) - set(weather_loc))
print(np.sort(no_weather_data))
```

```
['ACY' 'ADK' 'ADQ' 'AKN' 'ANC' 'BET' 'BQN' 'BRW' 'BTR' 'BTV' 'BWI' 'CAE'
 'CDV' 'CEC' 'CIC' 'CLD' 'CMI' 'CRP' 'DCA' 'DEN' 'DHN' 'DLG' 'DSM' 'DTW'
 'EVV' 'EWR' 'FAI' 'FCA' 'FLO' 'GNV' 'GPT' 'GRB' 'GRK' 'HDN' 'HHH' 'HNL'
 'IPL' 'ITO' 'IYK' 'JFK' 'JNU' 'KOA' 'KTN' 'LAW' 'LAX' 'LBB' 'LFT' 'LIH'
 'LWB' 'LYH' 'MCN' 'MDT' 'MFE' 'MGM' 'MLU' 'MOD' 'MQT' 'OAJ' 'OGG' 'OME'
 'OTZ' 'OXR' 'PFN' 'PIA' 'PMD' 'PSE' 'PSG' 'RFD' 'SAT' 'SCC' 'SCE' 'SGF'
 'SIT' 'SJT' 'SJU' 'SLE' 'STT' 'STX' 'TUS' 'VLD' 'WRG' 'YAK' 'YKM' 'YUM']
```

- Performed manual scraping for airports whose occurrences are more than 2500

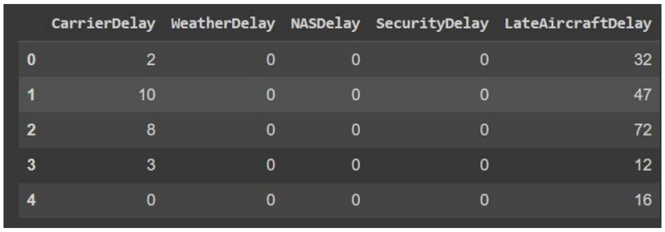


- Cleaned up weather data collected and merged it with original dataset

2$^{nd}$ EDA:

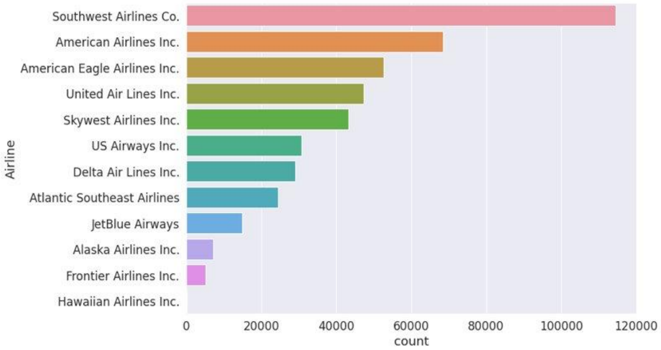
- Dropped rows having Null values for Weather Data
- Separate Date Information into Month Day Quarter Weekday etc.
- Separate Local Time (hh : mm) in Hours and Minutes
- Dropped Taxi in, Taxi out, Time, and date columns as information was getting leaked
- Created an additional column named CRSDepTime (Scheduled Departure Time)
- Finalized data set was created, which will be used for Model building

*Visualizations*

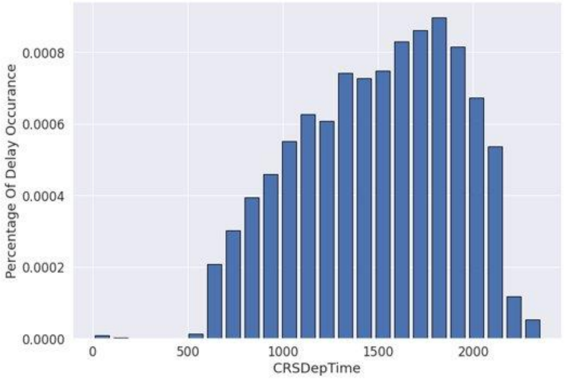


| | CarrierDelay | WeatherDelay | NASDelay | SecurityDelay | LateAircraftDelay |
|---|---|---|---|---|---|
| 0 | 2 | 0 | 0 | 0 | 32 |
| 1 | 10 | 0 | 0 | 0 | 47 |
| 2 | 8 | 0 | 0 | 0 | 72 |
| 3 | 3 | 0 | 0 | 0 | 12 |
| 4 | 0 | 0 | 0 | 0 | 16 |

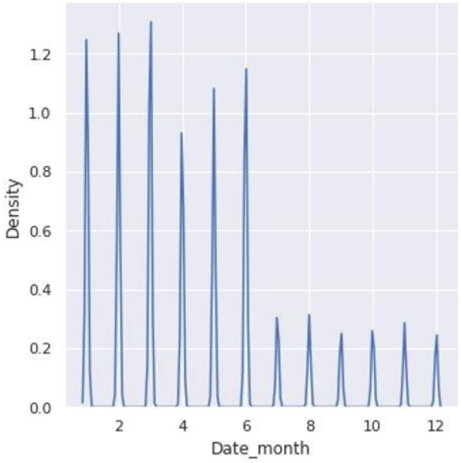
All the delay categories are tracked in minutes and all flights in the dataset have at least one of the delay types.



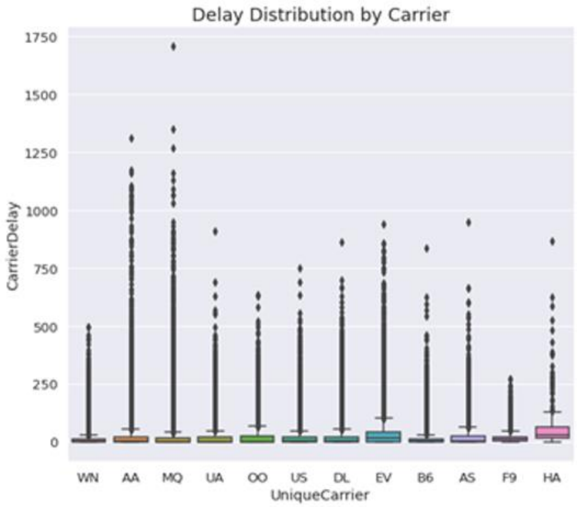
Southwest Airlines Company has the largest number of travel delays in this dataset.



A High Percentage of Delay occurs between 15:00:00 to 20:00:00 (i.e) 3 PM to 8 PM. The flights scheduled to depart at 12 AM to 5 AM: less delay



Most of the delays occurred in the month of March. We found out airlines in the US performed worse in 2019 as almost 20% of the overall flights in US were delayed.



Carriers with higher average delay generation are Hawaiian Airlines (HA) with 69.25 minutes per flight and Atlantic Southeast Airlines (EV) with 33.48 minutes per flight.

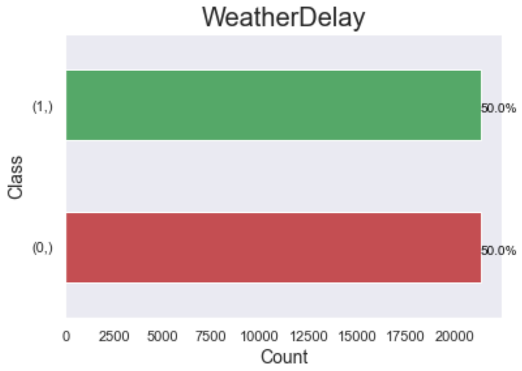
**Objective 1: Classification of Weather Delay**

**Web Scrapping:**

We initially performed Web-scrapping to collect the weather attributes for the destination location and merged it with the original data. The site used for web-scrapping is:

https://www.wunderground.com/

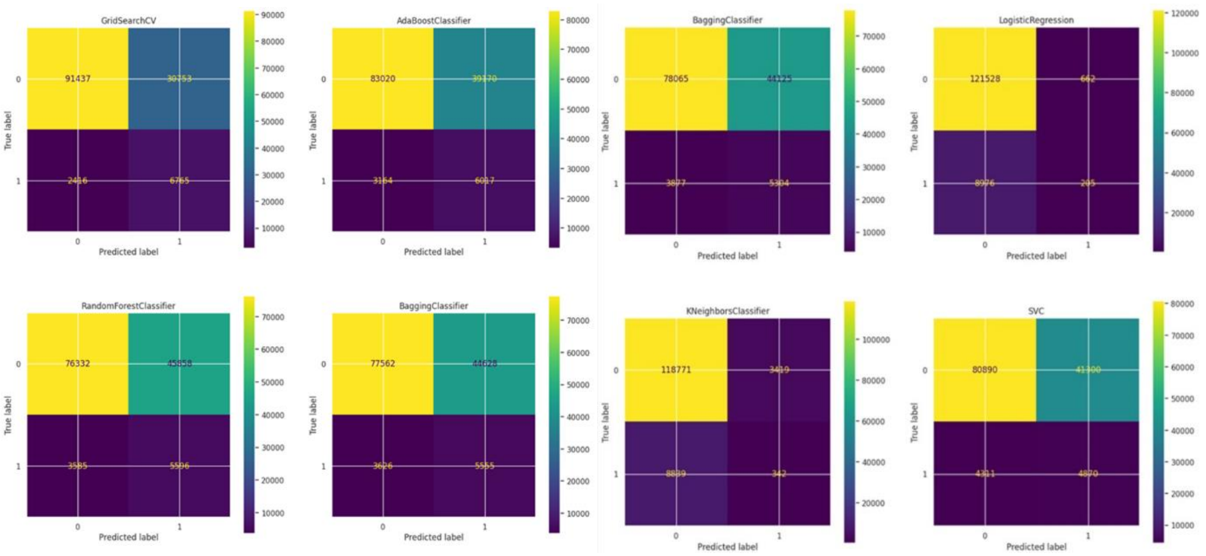
We then applied Random UnderSampling to overcome the issue of imbalance dataset.



We applied following models :

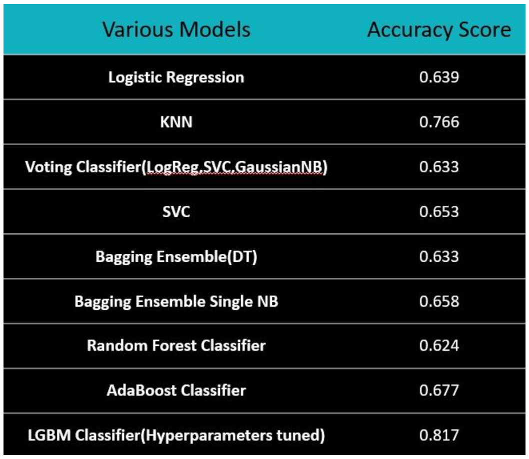
*LGBMClassifier, AdaBoostClassifier, RandomForestClassifier, BaggingClassifier, KNeighborsClassifier, SVC*

Following is picture of Confusion matrix for all the above models:



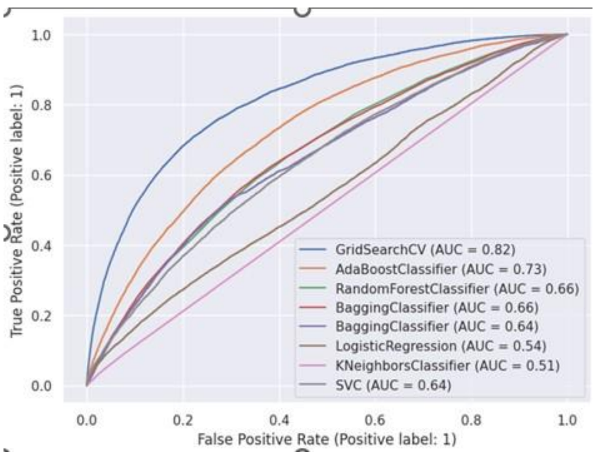
Performance Evaluation and Model Comparison:

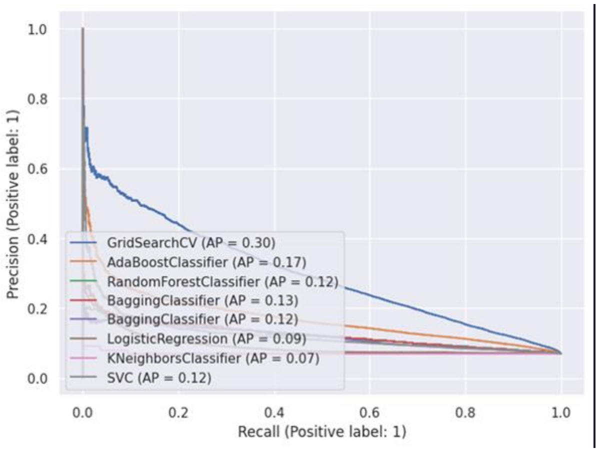
Accuracy



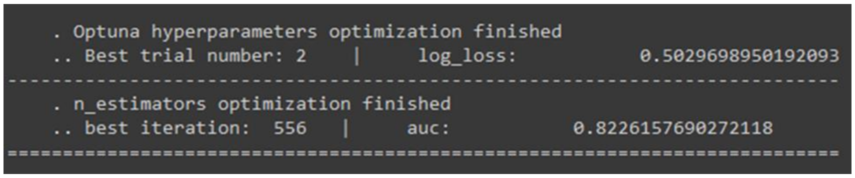
| Various Models | Accuracy Score |
|---|---|
| Logistic Regression | 0.639 |
| KNN | 0.766 |
| Voting Classifier(LogReg,SVC,GaussianNB) | 0.633 |
| SVC | 0.653 |
| Bagging Ensemble(DT) | 0.633 |
| Bagging Ensemble Single NB | 0.658 |
| Random Forest Classifier | 0.624 |
| AdaBoost Classifier | 0.677 |
| LGBM Classifier(Hyperparameters tuned) | 0.817 |

AUC: Precision-Recall Graph:

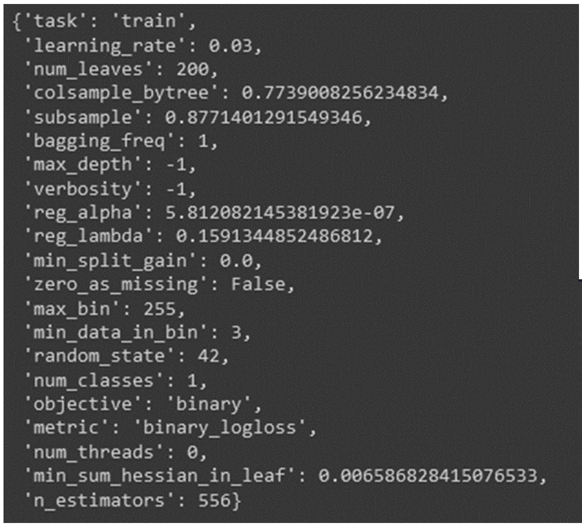




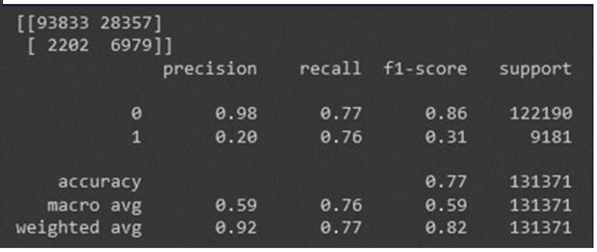
Hyperparameters: Best chosen model was LGBM Classifier and following are Hyperparameter.



```
    . Optuna hyperparameters optimization finished
    .. Best trial number: 2    |     log_loss:         0.5029698950192093
-----------------------------------------------------------------------
    . n_estimators optimization finished
    .. best iteration:  556   |     auc:          0.8226157690272118
=======================================================================
```



```
{'task': 'train',
 'learning_rate': 0.03,
 'num_leaves': 200,
 'colsample_bytree': 0.7739008256234834,
 'subsample': 0.8771401291549346,
 'bagging_freq': 1,
 'max_depth': -1,
 'verbosity': -1,
 'reg_alpha': 5.812082145381923e-07,
 'reg_lambda': 0.1591344852486812,
 'min_split_gain': 0.0,
 'zero_as_missing': False,
 'max_bin': 255,
 'min_data_in_bin': 3,
 'random_state': 42,
 'num_classes': 1,
 'objective': 'binary',
 'metric': 'binary_logloss',
 'num_threads': 0,
 'min_sum_hessian_in_leaf': 0.006586828415076533,
 'n_estimators': 556}
```

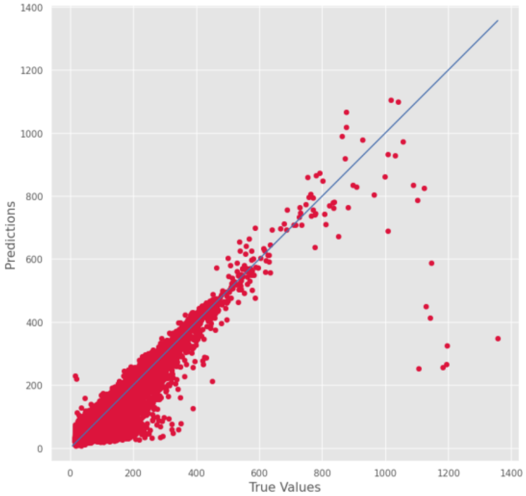


```
[[93833 28357]
 [ 2202  6979]]
              precision    recall  f1-score   support

           0       0.98      0.77      0.86    122190
           1       0.20      0.76      0.31      9181

    accuracy                           0.77    131371
   macro avg       0.59      0.76      0.59    131371
weighted avg       0.92      0.77      0.82    131371
```

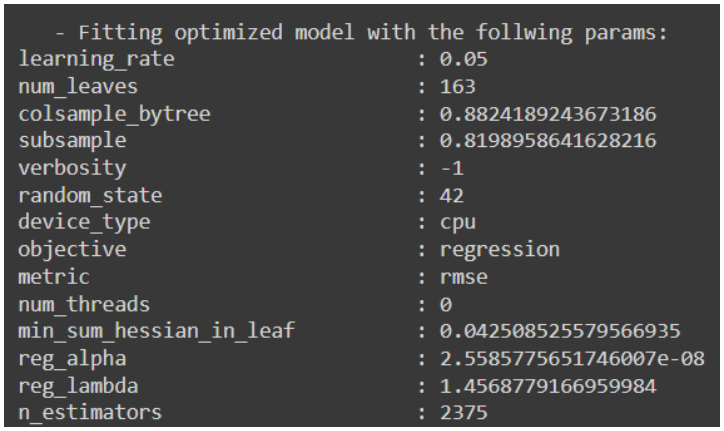
## Objective 2: Predicting Arrival Delay

*LBGM Regressor*

True values vs Predictions:



Best hyperparameters chosen:

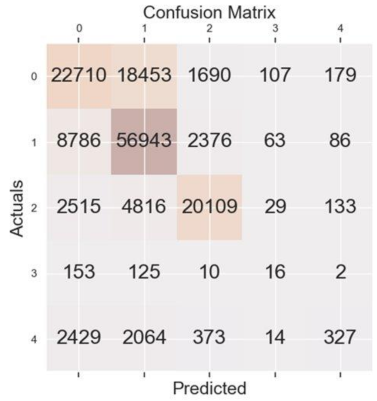


```
   - Fitting optimized model with the follwing params:
learning_rate                      : 0.05
num_leaves                         : 163
colsample_bytree                   : 0.8824189243673186
subsample                          : 0.8198958641628216
verbosity                          : -1
random_state                       : 42
device_type                        : cpu
objective                          : regression
metric                             : rmse
num_threads                        : 0
min_sum_hessian_in_leaf            : 0.042508525579566935
reg_alpha                          : 2.5585775651746007e-08
reg_lambda                         : 1.4568779166959984
n_estimators                       : 2375
```

Evaluation metric: RMSE = 17.12, R^2 = 0.91

## Objective 3: Classifying Type of Delay

*LGBM Classifier*



Confusion Matrix

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 22710 | 18453 | 1690 | 107 | 179 |
| 1 | 8786 | 56943 | 2376 | 63 | 86 |
| 2 | 2515 | 4816 | 20109 | 29 | 133 |
| 3 | 153 | 125 | 10 | 16 | 2 |
| 4 | 2429 | 2064 | 373 | 14 | 327 |

Actuals
Predicted

0- class in confusion Matrix is CarrierDelay

1- class in confusion Matrix is LateAircraftDelay

2- class in confusion Matrix is NASDelay

3- class in confusion Matrix is SecurityDelay

4- class is WeatherDelay

**Performance Evaluation**

Target variable: DelayType

Training Accuracy: 0.7128

Test Accuracy: 0.6927

F1 Score: 0.69272

**Conclusion**

In objective 1, LGBM Classifier was the top performer. With the model created for objective 1, we could help airlines by looking at the weather conditions and predicting whether the flight is going to be delayed due to weather or not. Airlines can then inform their customers about possible weather delays. In objective 2, LGBM Regressors was the top performer. With the model from objective 2, airlines can provide passengers with information about a change in the status of the flight and whether the flight is going to be delayed by more than 15 minutes. For objective 3, our model doesn't perform accurately in the current stage because we are missing variables that are needed to perform the prediction.

If we are given the opportunity to further work on this project, we would need to collect data for flights that do arrive on time. Also, we need to collect weather data for the locations we currently don't have and perform the analysis again. Lastly, predicting based on types of delays we need to train the models and re-evaluate them by comparing their scores and selecting the best performer.