BIG DATA ANALYSIS OF FLIGHT DATA

Muhammad Osama, Mohamed Faadil, Serikzhan Sadakbayev

Contents

Overview

The dataset is an air flight status dataset and has been chosen from Kaggle (https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022). The dataset has 61 columns and a few million rows. The dataset contains all flight information including cancellation and delays by different airlines.

Objectives

1. Combine all the datasets for the years 2018, 2019, 2020, 2021 & 2022 into a single dataset.
2. Take a sample from the combined dataset to create a small dataset on which the PySpark script can be run locally.
3. Perform data wrangling (data cleaning, feature engineering).
4. Perform visualizations to get a good understanding of the dataset.
5. Perform statistical analysis on the dataset.
6. Perform feature selection on the dataset.
7. Perform Machine Learning techniques to predict delay status of the flights.

All the above tasks were performed on the small dataset and then google cloud was used to run the PySpark script on the big dataset and get the results.

Data Wrangling

The dataset was downloaded from Kaggle for the years 2018, 2019, 2020, 2021 & 2022. All the datasets were then read into Spark and then merged into one big dataset. Then, sampling was performed on the big dataset to create a small dataset on which the PySpark script could be run locally. The following steps were taken to clean and transform the dataset:

1. The columns that were redundant or did not add any value to predicting the delay status were removed. These columns included: 'FlightDate', 'CRSDepTime', 'DepDelayMinutes', 'ArrDelayMinutes', 'CRSElapsedTime', 'ActualElapsedTime', 'Marketing_Airline_Network', 'Operated_or_Branded_Code_Share_Partners', 'DOT_ID_Marketing_Airline', 'IATA_Code_Marketing_Airline', 'Flight_Number_Marketing_Airline', 'Operating_Airline', 'DOT_ID_Operating_Airline', 'IATA_Code_Operating_Airline', 'Tail_Number', 'Flight_Number_Operating_Airline', 'Flight_Number_Operating_Airline', 'OriginAirportID', 'OriginAirportSeqID', 'OriginCityMarketID', 'OriginStateFips', 'OriginStateName', 'OriginWac', 'DestAirportID', 'DestAirportSeqID', 'DestCityMarketID', 'DestStateFips', 'DestStateName', 'DestWac', 'DepDel15', 'DepartureDelayGroups', 'DepTimeBlk', 'TaxiOut', 'WheelsOff', 'WheelsOn', 'TaxiIn', 'CRSArrTime', 'ArrDel15', 'ArrivalDelayGroups', 'ArrTimeBlk', 'DistanceGroup'.
2. Removed the flights which had been cancelled and then removed the ‘Cancelled’ column.
3. Removed all rows having null values in them.
4. Created a target variable ‘Delay_Status’. All the flights which had delayed arrival and departure times were considered as having been delayed and the rest were not delayed. This variable was made using the data from 'DepDelay' and 'ArrDelay' columns. Therefore, after creating the ‘Delay_Status’ column, these two columns were removed.
5. The columns ‘OriginCityName’ and ‘DestCityName’ were split into two, one having the city name and the other having the name of the state in which the city was located.
6. The columns ‘ArrTime’ and ‘DepTime’ were converted to date time format and then split into hours and minutes column. Subsequently, the ‘ArrTime’ and ‘DepTime’ columns were then removed.

Data Visualization

The cleaned dataset was then used to perform different data visualizations on the dataset to get a better understanding of the dataset. The following are the data visualizations performed:

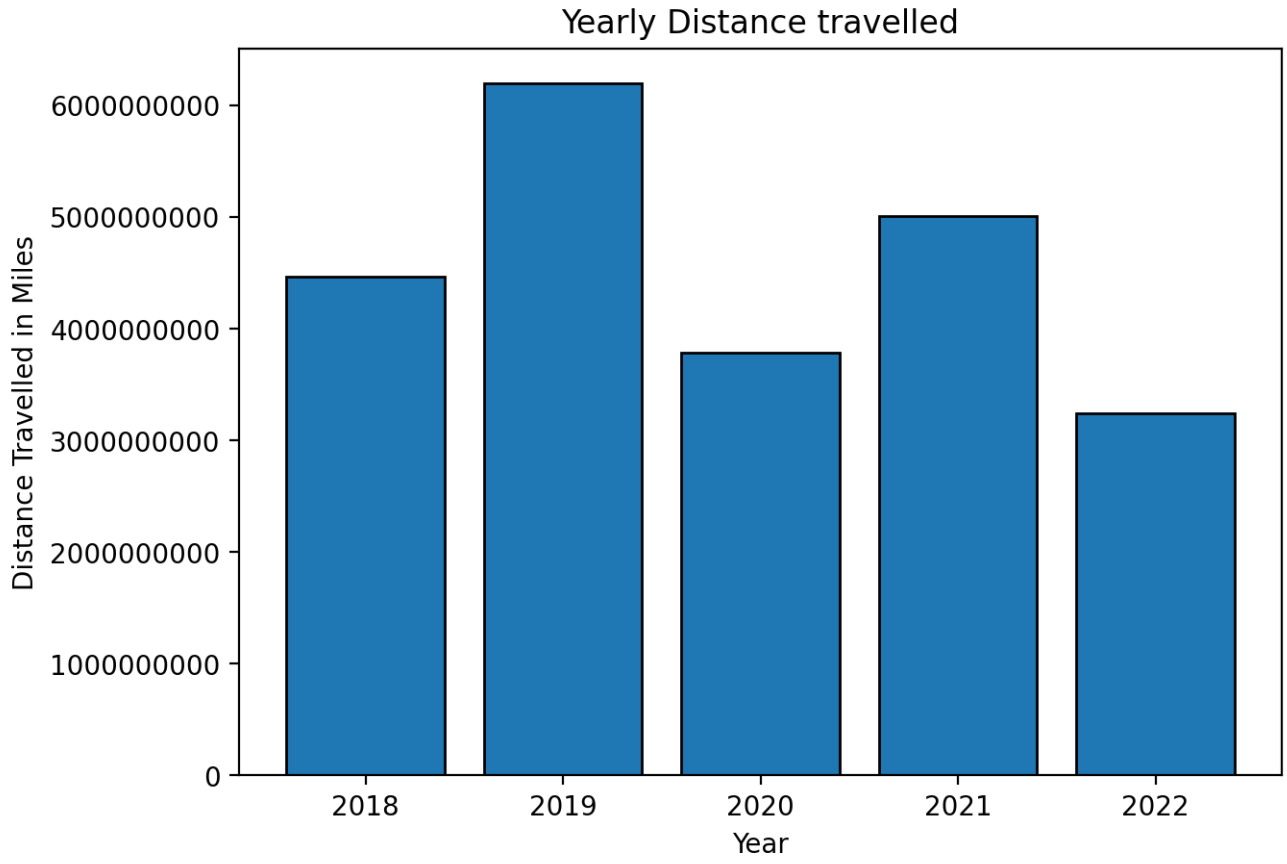


Figure 1: Distance Travelled by each Year

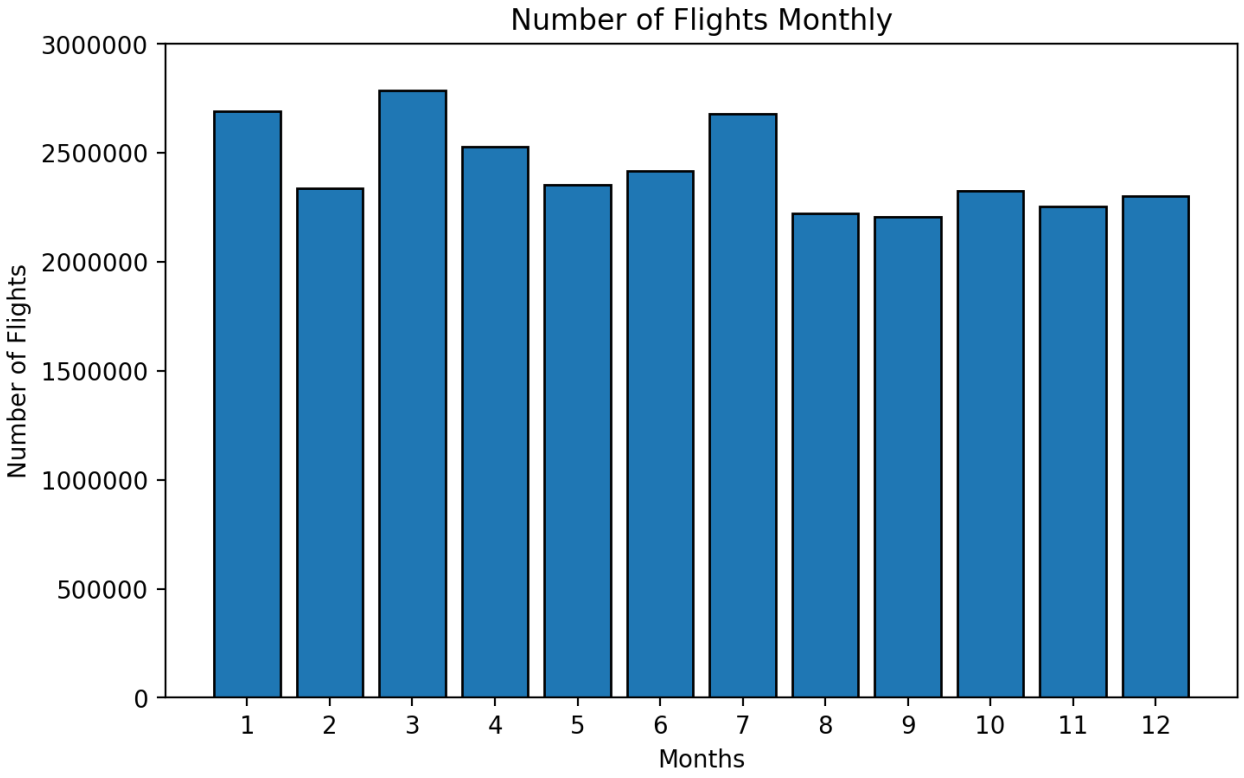


Figure 2: Number of Flights each Month

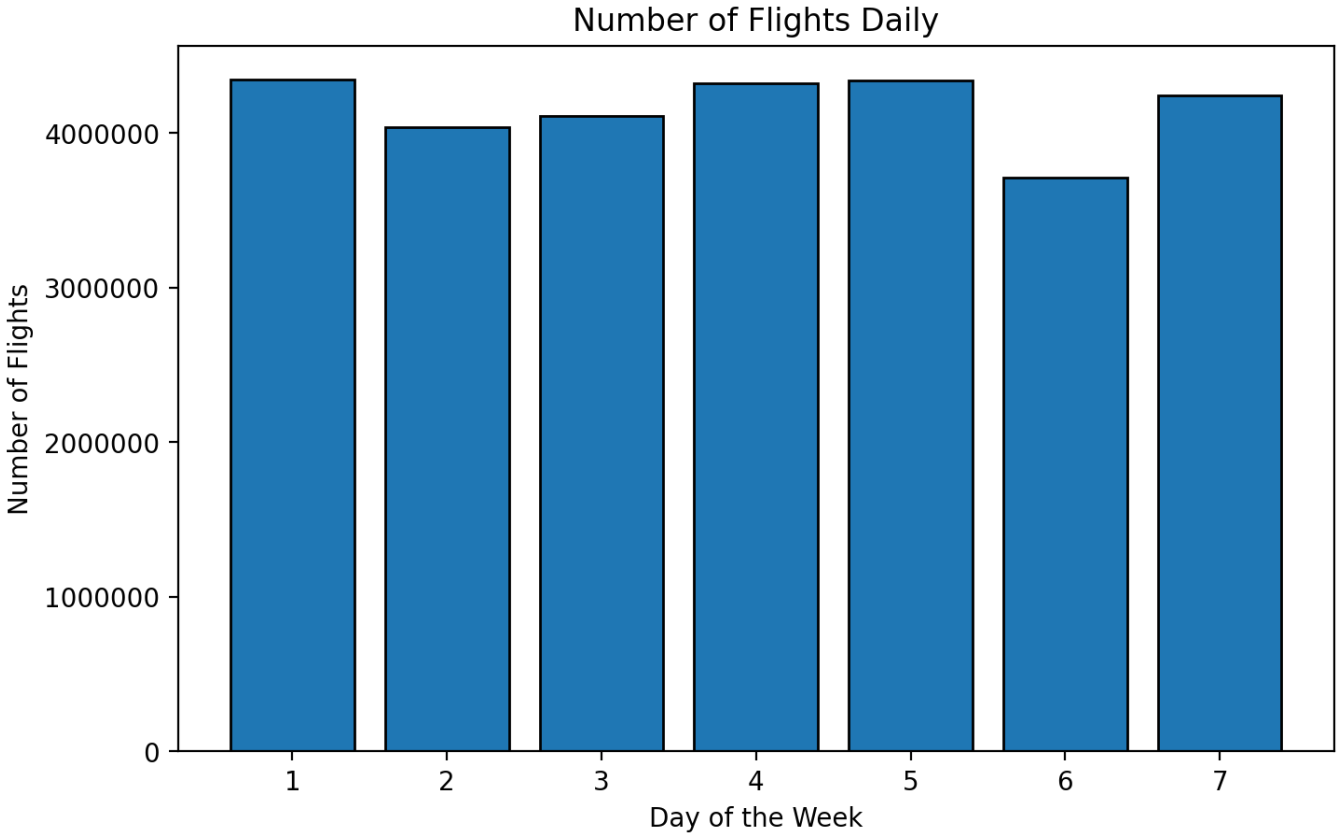


Figure 3: Number of Flights each Day of Week

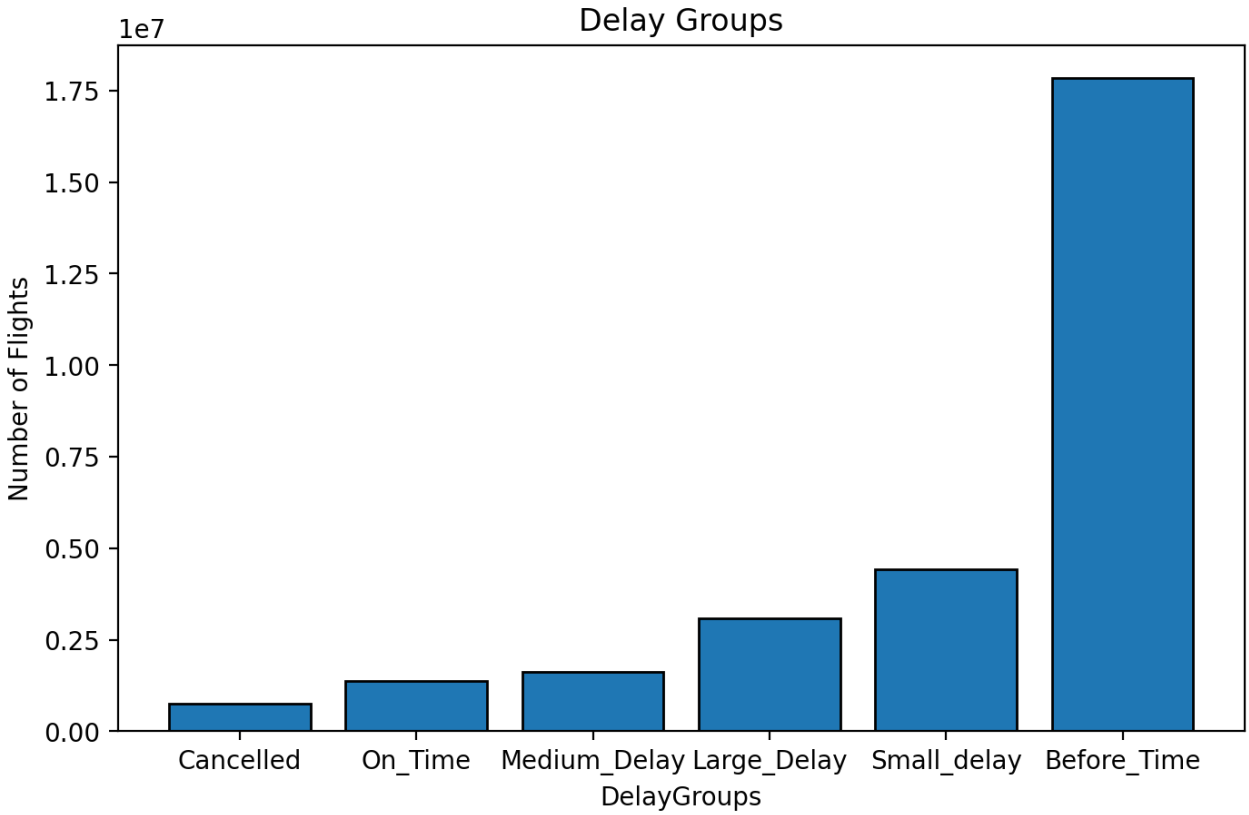


Figure 4: Number of Flights in each Delay Group

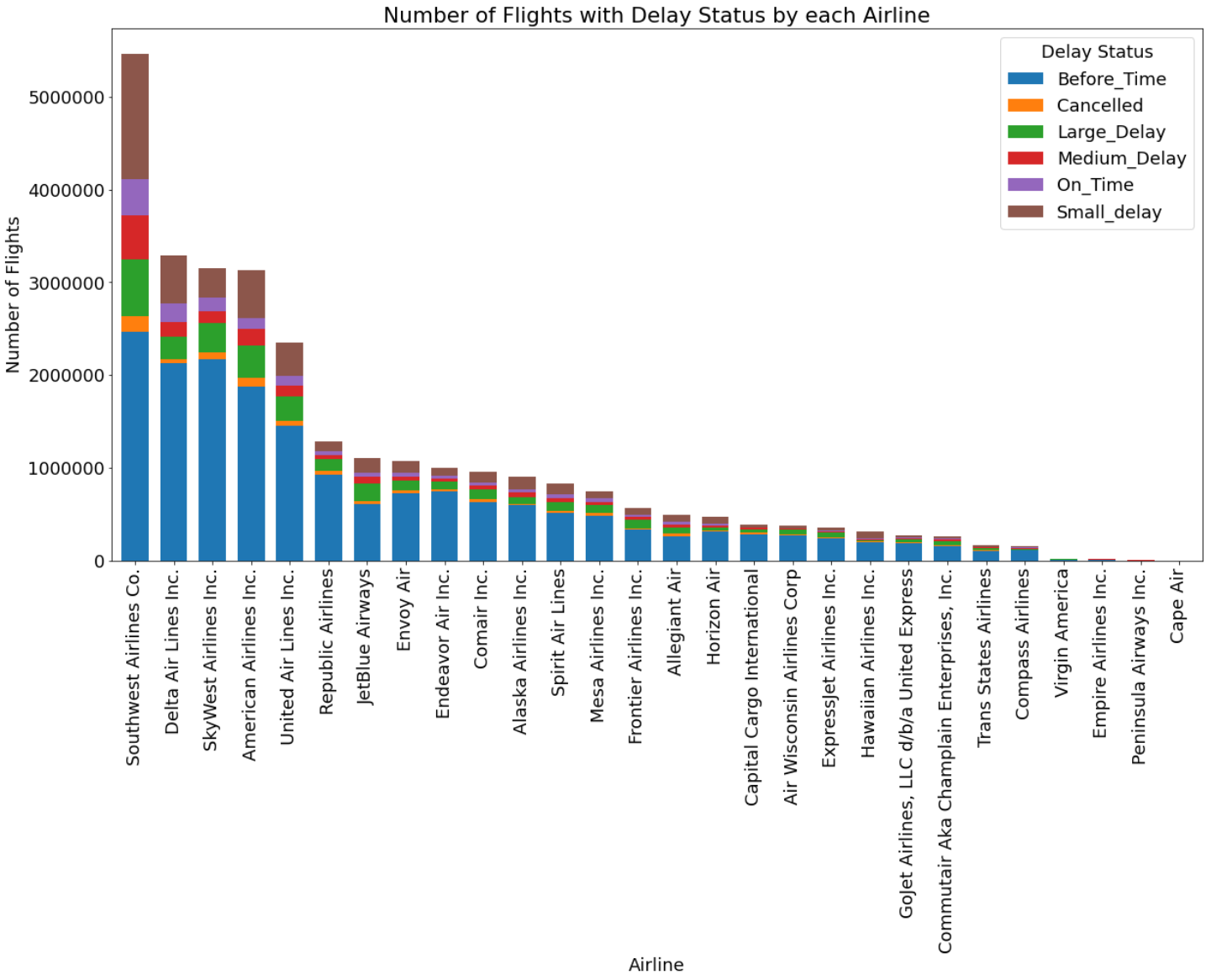


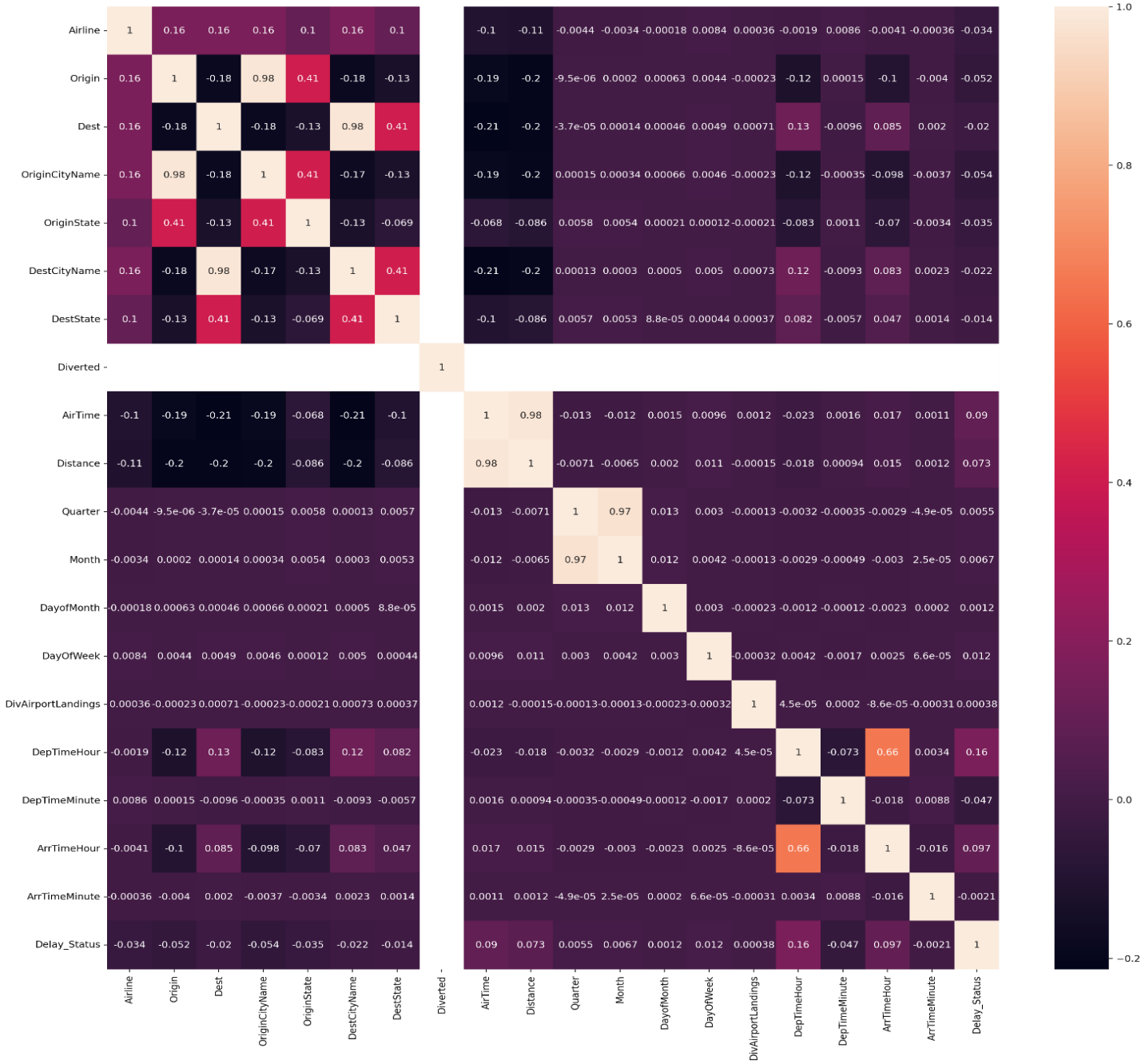
Figure 5: Number of Flights by Delay Status of Each Airline

Statistical Analysis

We performed two types of Statistical Analysis on the dataset, namely Correlation and Hypothesis Testing.

Correlation

We wanted to check the correlation among the variables of the dataset to check if there were variables which were having extremely high correlation between them or if there were variables having zero variance. From the cleaned dataset, we first converted the categorical features to numerical using string indexing in Spark ML. Once all the data was numeric, we used Spark ML to create a correlation matrix between all the variables in the dataset. The correlation matrix is shown below in the form of a heat map.



Looking at the correlation matrix we can see that the column 'Diverted' has zero variance due to which there is no value for the correlations. Therefore, we removed this column as well as this does not add any new

information to the dataset. We also removed the columns which were having extremely high correlation. These columns included 'Origin', 'Dest', 'Distance' and 'Quarter'.

Hypothesis Testing (Chi Squared)

We performed Chi Squared hypothesis testing to check for independence between the categorical variables and the target variable. We performed the test at the 95% confidence interval.

H_0: The categorical variable and the target variable are independent.

H_1: The categorical variable changes with changes in the target variable.

The table below shows the chi squared statistics.

features	pValue	degreesOfFreedom	statistics
Airline	0	27	447560.449
OriginCityName	0	370	274566.1791
OriginState	0	52	160846.3607
DestCityName	0	370	191978.2396
DestState	0	52	129005.1561

We can clearly see from the above table that the values of the statistics are extremely high and thus the p values are exceedingly small. Therefore, we reject the null hypothesis and conclude that the categorical variable shown above are not independent of the target variable.

Feature Selection

We performed two types of feature selection methods, Univariate feature selection and Variance threshold feature selection.

Univariate Feature Selection

We used univariate feature selection to select the 10 best features that can be used to predict 'Delay Status' based on ANOVA for numerical features and chi squared for categorical variables. For numerical features, the features having a high f score will be selected as they will be better in discriminating between the different 'Delay Status'. For categorical features, the features have a lower chi squared value will be selected as a higher chi square value means that the features and the target variable are not independent. The selected variables from the method are:

Categorical selected features
Airline
OriginCityName
OriginState

Numerical selected features
AirTime
Month
DayOfWeek
DepTimeHour
DepTimeMinute
ArrTimeHour
ArrTimeMinute

Variance Threshold Feature Selection

We used variance threshold selection to select the best 5 best numerical features from the dataset. Variance does not make sense for categorical variables; therefore, it was only used for numerical features. The features having the highest variance are selected using this method. The selected numerical features are shown below.

Numerical selected features
AirTime
DayofMonth
DepTimeHour
DepTimeMinute
ArrTimeHour
ArrTimeMinute

Machine Learning Models

We used different machine learning models to predict the delay status of airlines for the year 2022. The data for years 2018, 2019, 2020 and 2021 was used as the training set and the data for year 2022 served as the test set. We built a Spark pipeline to do all the transformations and the estimations. The following were the stages of the pipeline built:

1. String indexing the categorical variables.
2. One hot encoding the categorical variables.
3. Scaling the numerical variables.
4. Creating a vector of all the features.
5. Using different machine learning techniques.

The pipeline was then fit on the training set and transformation was performed on the test set to get the predictions. We used parameter grid in Spark to help tune the models by experimenting with different hyperparameters for each model. The models were then run on the full data and feature selected data and the best model was used to make the predictions on the test data. For evaluation of the models, we used ROC curve, true positive rate, false positive rate, precision, f1 score and accuracy.

Logistic Regression

We experimented with different hyperparameters for tuning the Logistic Regression Model. The values for the hyperparameters are as follows:

- Regularization parameter: 0.1, 0.01, 0.05
- Elastic net parameter: 0.0, 0.5, 1.0

The best model was then used for the predictions. The results are as follows:

Dataset	Regularization	Elastic Net	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	0.01	0	0.656039476	0.593202238	0.89375326	0.728685394	0.567773459	0.548272581
Univariate Selection Data	0.01	0	0.652770453	0.589991777	0.89366895	0.735243936	0.565550053	0.543691541
Variance Selection Data	0.01	0	0.654407972	0.595941713	0.87788078	0.706012671	0.571129899	0.557145518

Dataset	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	0.655747582	0.595022	0.889215675	0.720056866	0.56944683	0.55233964
Univariate Selection Data	0.652417523	0.591687	0.888943626	0.726672105	0.56712812	0.54769377
Variance Selection Data	0.654125174	0.597552	0.872881938	0.69732377	0.57276357	0.560843332

Logistic Regression with Cross Validation

We performed 10-fold cross validation with different hyperparameters for tuning the Logistic Regression Model. The values for the hyperparameters are as follows:

- Regularization parameter: 0.1, 0.01, 0.05
- Elastic net parameter: 0.0, 0.5, 1.0

The best model was then used for the predictions. The results are as follows:

Dataset	Regularization	Elastic Net	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	0.01	0	0.656039909	0.593202238	0.89375326	0.728685394	0.567773459	0.548272581
Univariate Selection Data	0.01	0	0.65277133	0.589991777	0.89366895	0.735243936	0.565550053	0.543691541
Variance Selection Data	0.01	0	0.654408298	0.595941713	0.87788078	0.706012671	0.571129899	0.557145518

Decision Tree

We experimented with different hyperparameters for tuning the Decision Tree Model. The values for the hyperparameters are as follows:

- Impurity method: 'gini', 'entropy'
- Maximum depth of tree: 5, 10, 15, 20

The best model was then used for the predictions. The results are as follows:

Dataset	Impurity	Max Depth	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	gini	5	0.506795918	0.584808143	0.93673213	0.792099511	0.558800489	0.519392795
Univariate Selection Data	gini	5	0.506795918	0.584808143	0.93673213	0.792099511	0.558800489	0.519392795
Variance Selection Data	entropy	20	0.51483903	0.630194407	0.83678027	0.591057325	0.602581607	0.611682717

Random Forest

We experimented with different hyperparameters for tuning the Random Forest Model. The values for the hyperparameters are as follows:

- Impurity method: 'gini', 'entropy'
- Maximum depth of tree: 5, 10, 15, 20
- Number of trees: 5, 10, 15, 20

The best model was then used for the predictions. The results are as follows:

Dataset	Impurity	Max Depth	Trees	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	entropy	20	20	0.676653293	0.56562193	0.976765036	0.8747089	0.544616007	0.466841
Univariate Selection Data	entropy	20	20	0.675597154	0.57787187	0.962111631	0.833645695	0.552779391	0.496214
Variance Selection Data	gini	20	20	0.67450657	0.56820779	0.970833873	0.86300138	0.54644672	0.47488

Linear Support Vector Machine

We experimented with different hyperparameters for tuning the Linear Support Vector Machine Model. The values for the hyperparameters are as follows:

- Regularization parameter: 0.0, 0.3, 0.5, 1.0, 2.0

The best model was then used for the predictions. The results are as follows:

Dataset	Regularization	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	2	0.649856	0.517139528	1	1	0.517139528	0.352549369
Univariate Selection Data	1	0.644262	0.517139528	1	1	0.517139528	0.352549369
Variance Selection Data	0.5	0.63937	0.517143584	0.999998529	0.99999003	0.517141652	0.352559779

Naïve Bayes

We experimented with different hyperparameters for tuning the Naïve Bayes Model. The values for the hyperparameters are as follows:

- Smoothing parameter: 0.0, 0.3, 0.5, 0.7, 1.0

The best model was then used for the predictions. The results are as follows:

Dataset	Smoothing	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	0.7	0.54881	0.523294032	0.213265512	0.14466801	0.612226605	0.469756788
Univariate Selection Data	0.3	0.544818	0.515180805	0.172385582	0.11768838	0.610705436	0.446797939
Variance Selection Data	1	0.54867	0.520655193	0.195104388	0.13068262	0.615229451	0.460570079

Gradient Boosted Trees

We experimented with different hyperparameters for tuning the Gradient Boosted Trees Model. The values for the hyperparameters are as follows:

- Maximum depth of tree: 5, 10, 15, 20

The best model was then used for the predictions. The results are as follows:

Dataset	Max Depth	ROC Area	Accuracy	True Positive Rate	False Positive Rate	Precision	F1 Score
Full Data	20	0.719628	0.651046309	0.822280782	0.53234438	0.623252444	0.639078279
Univariate Selection Data	20	0.717509	0.652255713	0.809282221	0.51591836	0.626863416	0.642244736
Variance Selection Data	20	0.736442	0.665025314	0.819137756	0.50002782	0.6369553	0.655688332