Flight Delay Prediction

Project

Group 7

Ali Mohamed Ali Salman Yusuf

Chang Liu

Wesley Knights

Project Overview and Objective

Transportation plays an important role in a person's daily routine. Most daily activities require individuals to adhere to their set schedule so that they can work efficiently and effectively. When it comes to traveling by airplane, having access to various flights greatly shortens the time it takes to get from your starting point to your final destination. Problems arise when flights become delayed due to unforeseen circumstances. One example of a variable that can cause flight delays is the weather. Therefore, to provide a better service to passengers who have stricter time commitments, we would like to work on a flight dataset that contains multiple variables such as the timing of arrival, flight departure, and other flight operational information from the 2019 US Department of Transportation with the objective of creating a model that can accurately predict a flight's estimated delay time and if a delay will happen. This project will go through the following stages: (1) Data cleaning and preparation; (2) Data exploration and analysis; and (3) using machine learning techniques to find the best model to predict flight delays.

Dataset Overview

The dataset (LINK) we are using was found on Kaggle, the data has been recorded, compiled, and originally distributed by the US Department of Transportation for flights in April 2019. The original dataset contains 1,048,576 rows and 27 columns. However, only part of the dataset has been used in our project due to the size of the dataset. Accordingly, to prepare the dataset we generated a 20% simple random sampling of the data to be used for our analysis. The dataset was then stored in a Spark dataframe and also another copy of the dataset was stored in a Pandas dataframe, both to be used for our analysis and data exploration in the later stages of the project. The final step before going into data cleaning was to prepare two columns that include flight's arrival delay and departure delay.

Data Cleaning

In the process of cleaning the data, the first step was to drop several columns that were not valuable for our analysis, these columns included delay reason columns which were in Boolean type indicating the reason such as carrier, weather, security etc., other dropped columns were flight number, flight cancellation code and diversion details. All of this data was not helpful to our analysis as they do not intersect with the project scope of predicting flight delay time.

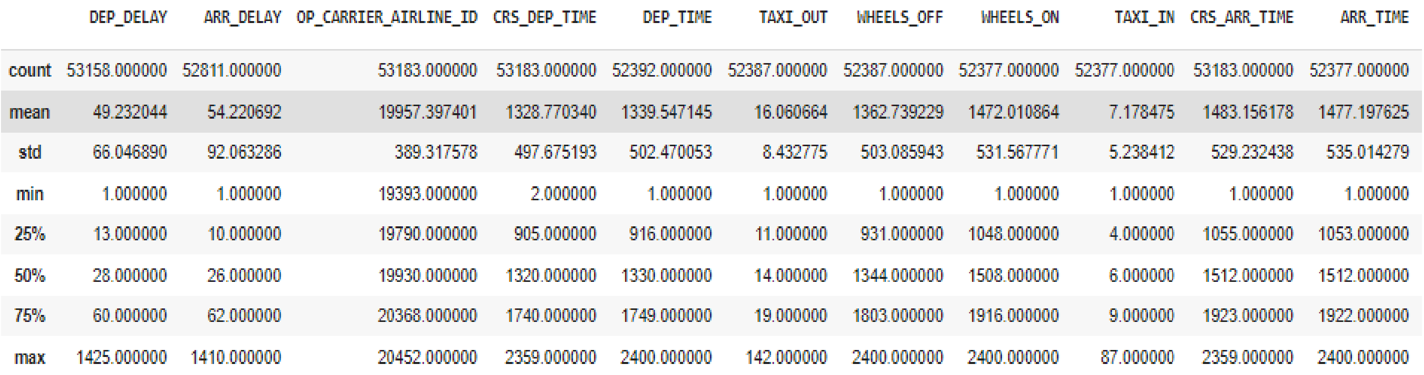
Further, there are columns that we focused on as they represent the relevant data for our project, these include, CRS_DEP_TIME (Scheduled Departure Time), DEP_TIME (Actual Departure Time), TAXI_OUT (taxi out time), TAXI_IN (taxi in time), CRS_ARR_TIME (Scheduled Arrival Time), ARR_TIME (Actual Arrival Time), ORIGIN (Departure State of Flight), and DEST (Flight Destination). The data variables that have been chosen were then studied to see if they conformed to a type of variable that can be used in a regression analysis along with the time-of-flight delay in minutes. Accordingly, the categorical and string variables that did not conform to the data type required (numeric) were changed using a One-Hot-Encoder to convert each categorical value into a new categorical column and assign a binary value of 1/0 to those columns. Further, a StringIndexer was used as a label indexer to map string columns to a machine learning column of label indices.

As part of data cleaning and preparation, we added a column for arrival and departure delay time, however, it we had an issue with the time formats from other columns needed to derive these variables which are the actual departure and arrival times and the scheduled departure and arrival times as they were formatted as integers. Accordingly, we wrote a code that extract the timestamp from the integers in order to be usefully stored in minutes. After sorting out the columns, we decided to assume that the flight arrivals happen within the same day unless the time is after midnight 24:00, this was needed because the dataset did not include the arrival date and just had the departure date. For example, if a flight departure on April 1, 2019 (10:00) and arrived on April 2, 2019 (12:00) we are assuming it was on April 1, 2019 and not April 2. This is a fair assumption that will not affect our analysis as it is very unlikely that a flight will take more than 24 hours.

Dataset Exploration

The first step to look at the data was to understand the basic summary statistic of the columns in the dataset, as shown in Table 1 below:

Table1. Summary Statistic:



	DEP_DELAY	ARR_DELAY	OP_CARRIER_AIRLINE_ID	CRS_DEP_TIME	DEP_TIME	TAXI_OUT	WHEELS_OFF	WHEELS_ON	TAXI_IN	CRS_ARR_TIME	ARR_TIME
count	53158.000000	52811.000000	53183.000000	53183.000000	52392.000000	52387.000000	52387.000000	52377.000000	52377.000000	53183.000000	52377.000000
mean	49.232044	54.220692	19957.397401	1328.770340	1339.547145	16.060664	1362.739229	1472.010864	7.178475	1483.156178	1477.197625
std	66.046890	92.063286	389.317578	497.675193	502.470053	8.432775	503.085943	531.567771	5.238412	529.232438	535.014279
min	1.000000	1.000000	19393.000000	2.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	13.000000	10.000000	19790.000000	905.000000	916.000000	11.000000	931.000000	1048.000000	4.000000	1055.000000	1053.000000
50%	28.000000	26.000000	19930.000000	1320.000000	1330.000000	14.000000	1344.000000	1508.000000	6.000000	1512.000000	1512.000000
75%	60.000000	62.000000	20368.000000	1740.000000	1749.000000	19.000000	1803.000000	1916.000000	9.000000	1923.000000	1922.000000
max	1425.000000	1410.000000	20452.000000	2359.000000	2400.000000	142.000000	2400.000000	2400.000000	87.000000	2359.000000	2400.000000

Within the first step, while looking at the statistical summary we looked as some visual statistics of data to see if there is any abnormality in data, skewness and formation that we need to consider. However, our data was not skewed or formed in any way that would conform to our analysis. Below are some of the visuals explored:

Fig1. Origin of Flight Departure:

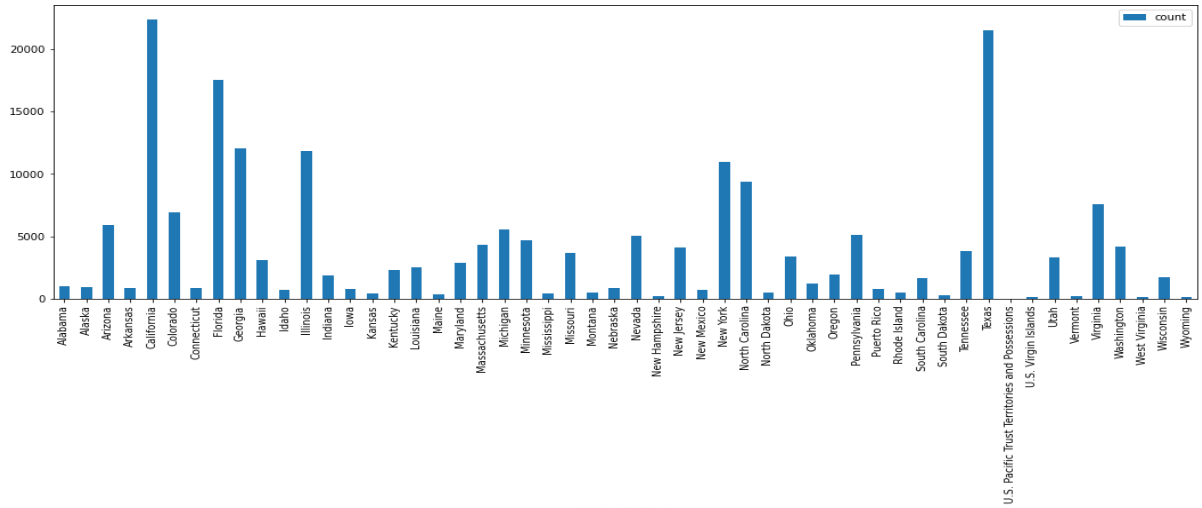
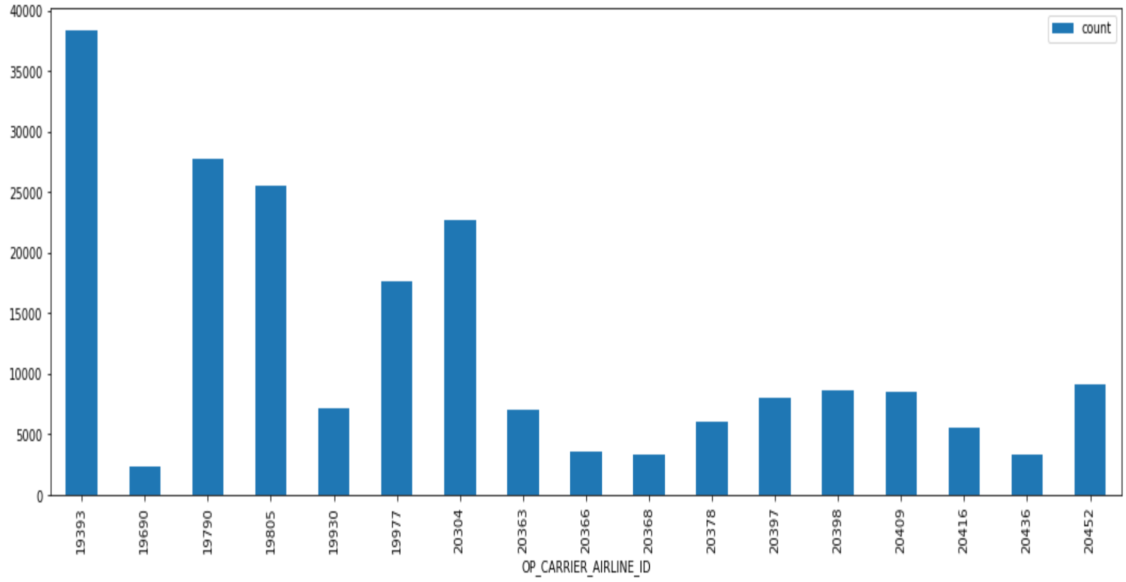


Fig2. Carrier Airline and the ID showing which Airline it is:



Airline Name	Airline ID	Carrier code	Number flights	% flights
Southwest Airlines	19393	WN	77148	19.93%
Delta Air Lines	19790	DL	58353	15.07%
American Airlines	19805	AA	39603	10.23%
ExpressJet Airlines	20366	EV	39091	10.10%
SkyWest Airlines	20304	OO	36961	9.55%
United Air Lines	19977	UA	33410	8.63%
US Airways	20355	US	30594	7.90%
Envoy Air	20398	MQ	22880	5.91%
JetBlue Airways	20409	B6	16397	4.24%
Alaska Airlines	19930	AS	9027	2.33%
Spirit Airlines	20416	NK	6978	1.80%
Hawaiian Airlines	19690	HA	6211	1.60%
Frontier Airlines	20436	F9	5868	1.52%
Virgin America	21171	VX	4601	1.19%

The second step in data exploration was to generate a correlation matrixes to study the correlation between variables and arrival time and arrival delay, accordingly, these factors with the highest correlation were the ones included in our training models and analysis.

Fig3. Correlation Matrix – Heatmap:

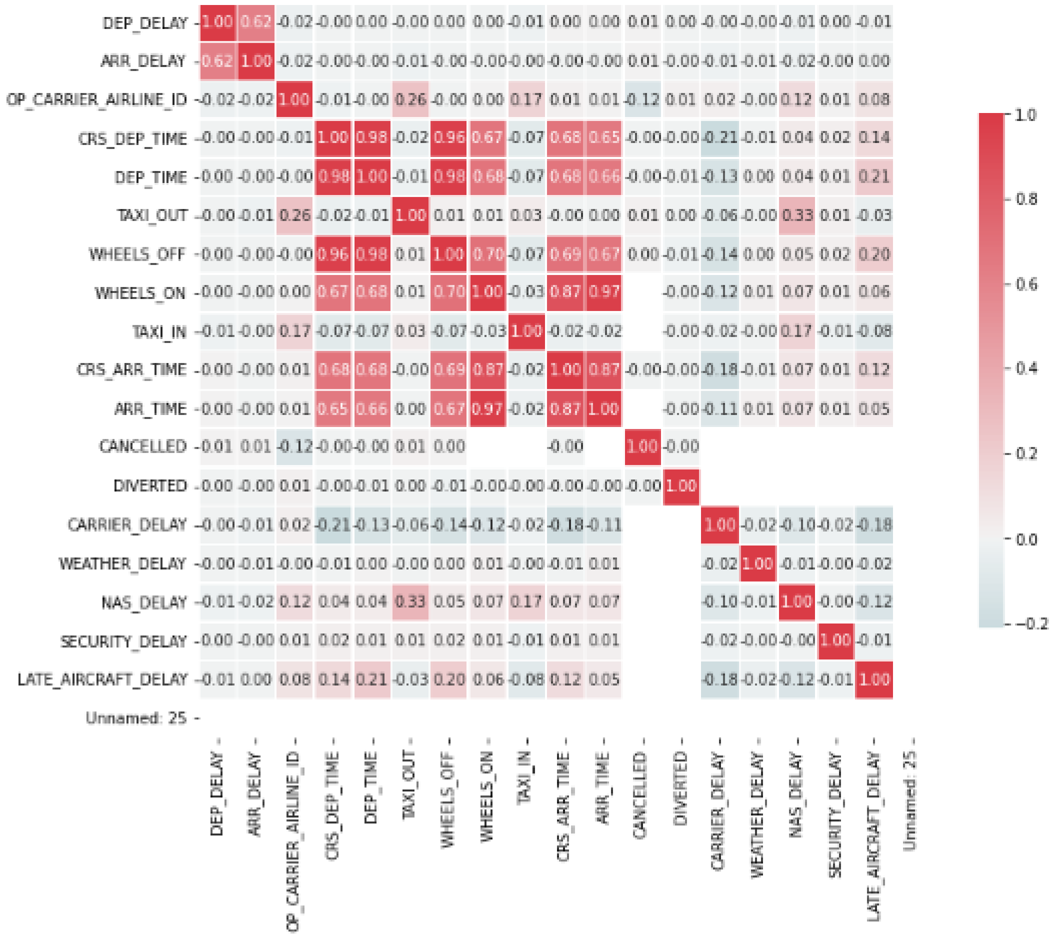
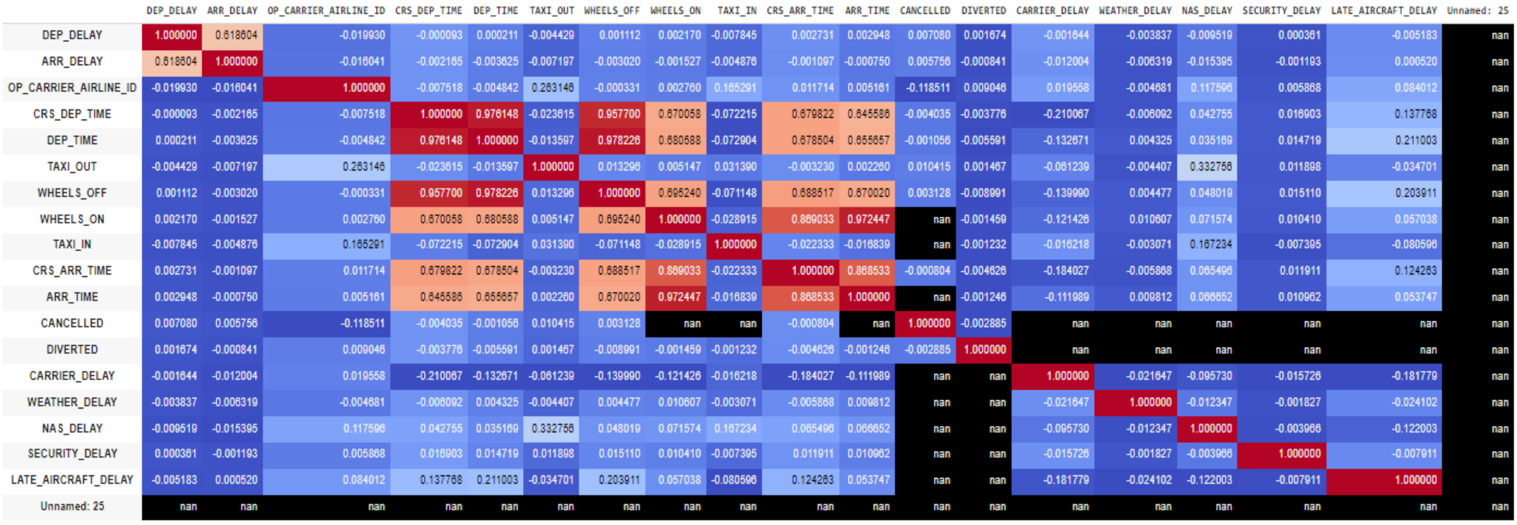


__Fig3. Correlation Matrix – Gradient, cmap=Coolwarm:__



	DEP_DELAY	ARR_DELAY	OP_CARRIER_AIRLINE_ID	CRS_DEP_TIME	DEP_TIME	TAXI_OUT	WHEELS_OFF	WHEELS_ON	TAXI_IN	CRS_ARR_TIME	ARR_TIME	CANCELLED	DIVERTED	CARRIER_DELAY	WEATHER_DELAY	NAS_DELAY	SECURITY_DELAY	LATE_AIRCRAFT_DELAY	Unnamed: 25
DEP_DELAY	1.000000	0.618604	-0.019930	-0.000093	0.000211	-0.004429	0.001112	0.002170	-0.007845	0.002731	0.002948	0.007080	0.001674	-0.001644	-0.003837	-0.009519	0.000361	-0.005183	nan
ARR_DELAY	0.618604	1.000000	-0.016041	-0.002165	-0.003625	-0.007197	-0.003020	-0.001527	-0.004876	-0.001097	-0.000750	0.005756	-0.000841	-0.012004	-0.006319	-0.015395	-0.001193	0.000520	nan
OP_CARRIER_AIRLINE_ID	-0.019930	-0.016041	1.000000	-0.007518	-0.004842	0.263146	-0.000331	0.002760	0.165291	0.011714	0.005161	-0.118511	0.009046	0.019558	-0.004681	0.117596	0.005868	0.084012	nan
CRS_DEP_TIME	-0.000093	-0.002165	-0.007518	1.000000	0.976148	-0.023615	0.957700	0.670058	-0.072215	0.679822	0.645586	-0.004035	-0.003776	-0.210067	-0.006092	0.042755	0.016903	0.137768	nan
DEP_TIME	0.000211	-0.003625	-0.004842	0.976148	1.000000	-0.013597	0.978226	0.680588	-0.072904	0.678504	0.655657	-0.001056	-0.005591	-0.132671	0.004325	0.035169	0.014719	0.211003	nan
TAXI_OUT	-0.004429	-0.007197	0.263146	-0.023615	-0.013597	1.000000	0.013296	0.005147	0.031390	-0.003230	0.002260	0.010415	0.001467	-0.061239	-0.004407	0.332756	0.011898	-0.034701	nan
WHEELS_OFF	0.001112	-0.003020	-0.000331	0.957700	0.978226	0.013296	1.000000	0.695240	-0.071148	0.688517	0.670020	0.003128	-0.008991	-0.139990	0.004477	0.048019	0.015110	0.203911	nan
WHEELS_ON	0.002170	-0.001527	0.002760	0.670058	0.680588	0.005147	0.695240	1.000000	-0.028915	0.869033	0.972447	nan	-0.001459	-0.121426	0.010607	0.071574	0.010410	0.057038	nan
TAXI_IN	-0.007845	-0.004876	0.165291	-0.072215	-0.072904	0.031390	-0.071148	-0.028915	1.000000	-0.022333	-0.016839	nan	-0.001232	-0.016218	-0.003071	0.167234	-0.007395	-0.080596	nan
CRS_ARR_TIME	0.002731	-0.001097	0.011714	0.679822	0.678504	-0.003230	0.688517	0.869033	-0.022333	1.000000	0.868533	-0.000804	-0.004626	-0.184027	-0.005868	0.065496	0.011911	0.124263	nan
ARR_TIME	0.002948	-0.000750	0.005161	0.645586	0.655657	0.002260	0.670020	0.972447	-0.016839	0.868533	1.000000	nan	-0.001246	-0.111989	0.009812	0.066652	0.010962	0.053747	nan
CANCELLED	0.007080	0.005756	-0.118511	-0.004035	-0.001056	0.010415	0.003128	nan	nan	-0.000804	nan	1.000000	-0.002885	nan	nan	nan	nan	nan	nan
DIVERTED	0.001674	-0.000841	0.009046	-0.003776	-0.005591	0.001467	-0.008991	-0.001459	-0.001232	-0.004626	-0.001246	-0.002885	1.000000	nan	nan	nan	nan	nan	nan
CARRIER_DELAY	-0.001644	-0.012004	0.019558	-0.210067	-0.132671	-0.061239	-0.139990	-0.121426	-0.016218	-0.184027	-0.111989	nan	nan	1.000000	-0.021647	-0.095730	-0.015726	-0.181779	nan
WEATHER_DELAY	-0.003837	-0.006319	-0.004681	-0.006092	0.004325	-0.004407	0.004477	0.010607	-0.003071	-0.005868	0.009812	nan	nan	-0.021647	1.000000	-0.012347	-0.001827	-0.024102	nan
NAS_DELAY	-0.009519	-0.015395	0.117596	0.042755	0.035169	0.332756	0.048019	0.071574	0.167234	0.065496	0.066652	nan	nan	-0.095730	-0.012347	1.000000	-0.003966	-0.122003	nan
SECURITY_DELAY	0.000361	-0.001193	0.005868	0.016903	0.014719	0.011898	0.015110	0.010410	-0.007395	0.011911	0.010962	nan	nan	-0.015726	-0.001827	-0.003966	1.000000	-0.007911	nan
LATE_AIRCRAFT_DELAY	-0.005183	0.000520	0.084012	0.137768	0.211003	-0.034701	0.203911	0.057038	-0.080596	0.124263	0.053747	nan	nan	-0.181779	-0.024102	-0.122003	-0.007911	1.000000	nan
Unnamed: 25	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan

After studying the correlations in the matrix, we found out that the most important variables include, schedule departure time, taxi out, departure time, wheels off, wheels on, scheduled arrival time and arrival time. Therefore, we went into further details to visualize the correlation in a scatter plot that helps show and confirm these correlations as shown below:

__Fig4. Correlation Scatter Plot – (Taxi-out and Arrival Time):__

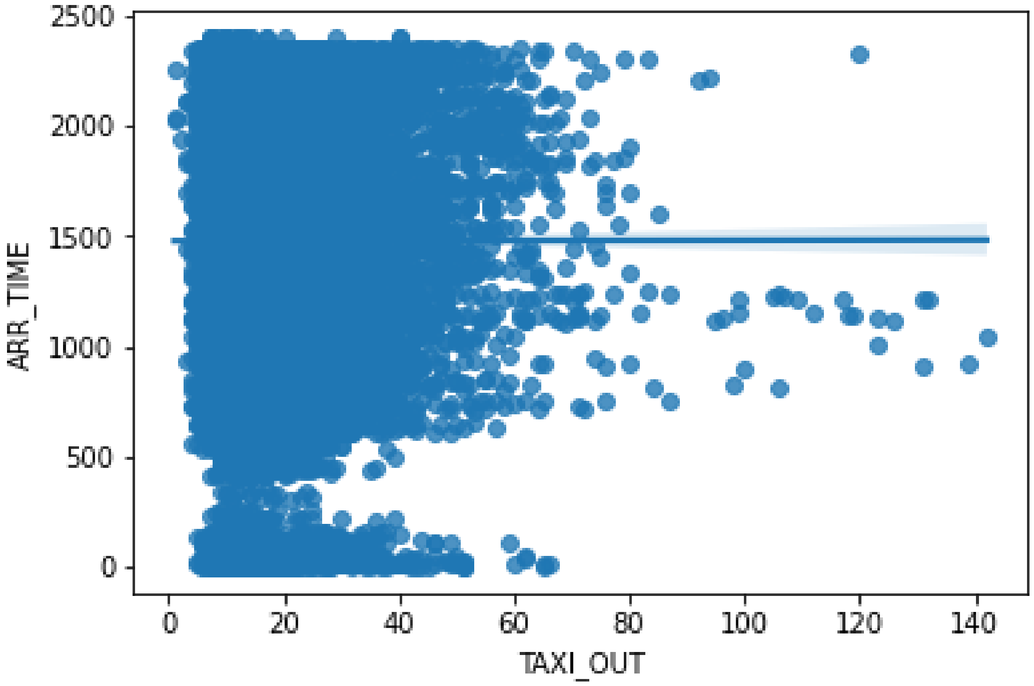


Fig5. Correlation Scatter Plot – (Arrival Delay and Departure Delay):

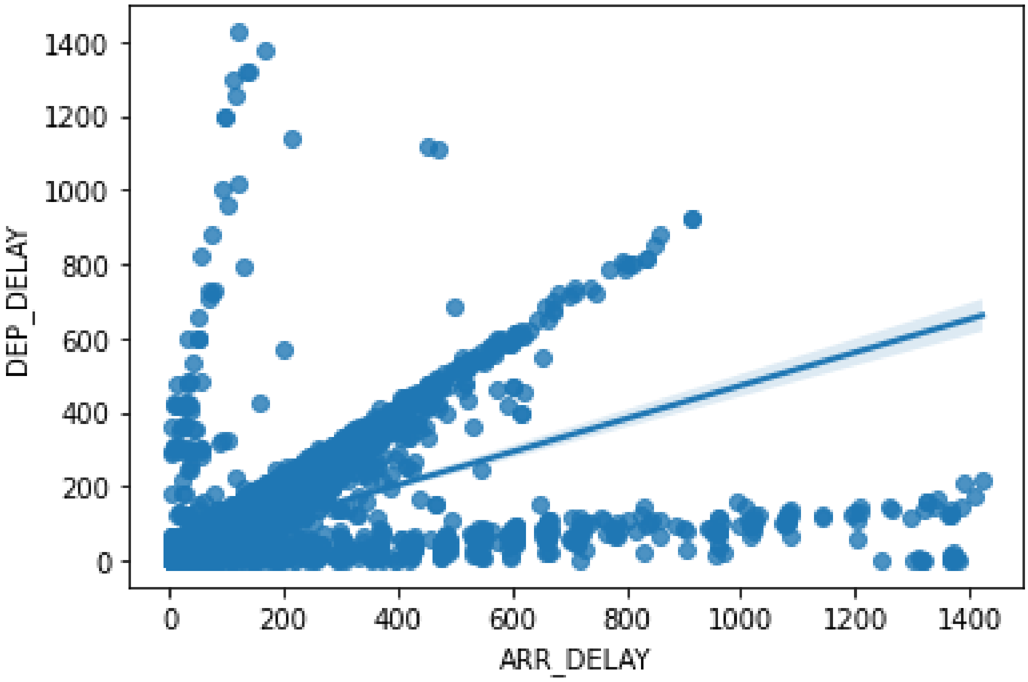
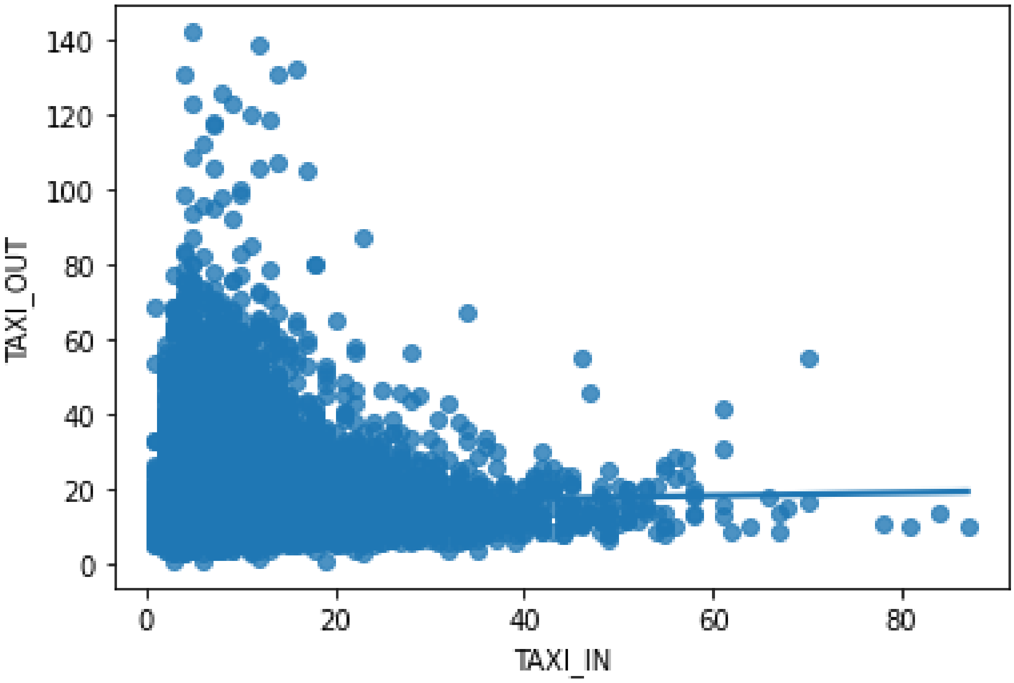


Fig6. Correlation Scatter Plot – (Taxi_in and Taxi_out):



In regard to the packages that we used for the project, as planned we used non-spark packages like Pandas, Seaborn, Matplotlib, and graphic packages.

Project Model and Inference

Following the exploration of the data, we started preparing the models and training them according to the prepared data and the chosen variables. For the purpose of our project, we have chosen 4 algorithms as models to be used to predict flight delays. Those algorithms were used as suitable, first as regression models to predict the flight delay in minutes and then as classification models to predict if a flight delay will happen. After training and testing, the regression models have been tested according to a Regression Prediction metric that is the Mean Squared Error (MSE), while the classification models have been tested and evaluated using a Classification Prediction metric of Accuracy (AUC).

Before indulging into the models, we did some feature engineering to better fit the data in our models. As mentioned earlier, we used the One-Hot-Encoder and the StringIndexer which were included in the pipeline along with a VectorAssembler which had the needed input columns that will be used for the analysis and training the model (these variables that were found to be most significant according to our regression analysis during the data exploration).

Finally, the following models have been developed, trained and tested:

1. **Linear Regression:** This regression model used a maxIteration of 10, a regParam of 0.3 and an elasticNetParam of 0.8. The following criteria and parameters were chosen to aim at a good fit for the model, they were selected after research, team experience, judgment, and testing of the model.

2. **Decision Tree:** This regressor uses MSE to decide to split a node into sub-nodes and we thought it will be very suitable for this analysis and could lead to a good prediction model. We left the default Params for this model as we wanted to see how the default model performs without enhancements.

3. **Random Forest:** This is a meta estimator, the idea behind the random forest regressor is to fit classified decision trees on sub-samples of datasets. This model should be giving us a better prediction than the decision tree regressor, therefore, we also left the default Params in order to see how it performs in comparison to the default Params of a decision tree regressor. Further, we expect that the Random Forest classifier perform with higher accuracy, but we are in doubt over the level of accuracy compared to the linear regression as the model does not have many categorical variables.

4. **Gradient Boosting:** The regressor fits regression trees on a negative gradient of loss function, this builds an optimized model in a forward-stage-wise fashion. Accordingly, we assumed that this regressor will also give us strong prediction and a probably the lowest MSE. We only limited the maxItr to 10 in order to limit over-fitting and also save memory space.

Project Results and Conclusion

The following Table 2 and Table 3, summarizes the results after testing of the models and the prediction results represented as metrics.

Table2. Regression Predication Score Results

	Regression Prediction	
Model	**Train MSE**	**Test MSE**
Linear Regression	675.67	2013.92
Decision Tree	7555.55	7740.36
Random Forest	7409.40	7737.85
Gradient Boosting	6313.25	7016.01

Table3. Classification Predication Score Results

	Classification Prediction	
Model	**Train Accuracy**	**Test Accuracy**
Linear Regression	0.9997	0.9776
Decision Tree	0.7100	0.7050
Random Forest	0.7602	0.7579
Gradient Boosting	0.8392	0.8307

Accordingly, the Linear Regression gave us the lowest test MSE (2013.92) as a regression model. Also, when used as a classification model, the Linear Regression gave the top testing accuracy of 0.9776. The Linear Regression model seems to fit the best as most variables are numeric. Therefore, the model to be chosen is the Linear Regression for both regression and classification. Further, for future research purposes the other models could be enhanced, if given the computation capacity, by adding further Params and then testing. This might enhance the models and change the results.