Scalability Check for Machine Learning System Predicting Flight Delays

Anshumaan Chauhan
University of Massachusetts
Amherst
Department of Information and
Computer Sciences
achauhan@umass.edu

Pratyush Dubey
University of Massachusetts
Amherst
Department of Information and
Computer Sciences
pdubey@umass.edu

Sriharsha Hatwar
University of Massachusetts
Amherst
Department of Information and
Computer Sciences
shatwar@umass.edu

Abstract

Machine learning algorithms have made tremendous progress recently and have been applied to various real-world problems. One of the applications also includes the task of predicting delays in Flight timings, which is one of the serious problems faced by the Airline business. However, the problem with machine learning models involving deep learning is that they need - high computational power systems to train and store the model. Additionally, the system needs to be made end to end scalable and is ignored by the trends in recent research. This paper discusses the approach we have used to predict the delay in flight by framing this as a prediction problem. Most of the research in machine learning goes into the part of predictive modeling. However, we focus on the end-to-end aspect of the problem by using industry-standard high-performance systems such as MySQL and SparkSQL. We show that our solution can not only be used for predictive modeling but also provides an end-to-end explanation of the whole product with faster real-time predictions and scalability.

I. INTRODUCTION

According to the Federal Aviation Administration, the estimated amount in billions of dollars that was incurred to Airlines, Passengers, and others indirectly because of flight delays was 23.7, 26.6, 30.2 and 33 for the years 2016-2019 respectively. Delay in a flight can have a cascading effect of delaying other flights. Hence it is very important to capture any unexpected delay so that the passengers can accordingly reschedule their flight and the airport authorities can reorganize their tasks efficiently. This problem is also one of the main focuses of many online travel companies such as Expedia, MakeMyTrip, etc.

Initial solutions involved creating simulated environments with the same components as in a real environment to analyse and predict whether there will be any delay [1]. But this needed improvements as the computational complexity of these simulations were not scalable for a real time analysis. Some researchers started to use data analytics and statistical machine learning techniques such as Bayesian networks and other statistical methods for estimating the delay [2][3]. Still, the applications of big data analytics are limited when it comes to aviation [4].

There is still a requirement for a prediction model that takes into account multiple factors such as weather conditions, air traffic congestion and many others, and then gives a real time performance with high accuracy. This problem can be solved using 2 approaches depending on what output the user wants. One is classification, where the user just wants to know whether the flight would be late or not. In this type of classification, there are many machine learning approaches proposed using Logistic Regression Classification, Decision Trees [5], Random Forest, AdaBoost, K-Nearest Neighbors [6] and Gradient Boosting [7]. The second approach is regression where the amount of time by which the flight is delayed is predicted by the model. In this, 2 steps approaches are generally followed, where the first step is classification, and the second step actually calculates the amount of delay for the flights that were categorized as 'delayed' in the first step [8]. This can also be converted into a single step regression problem, where output can be zero if there is no delay and avoid the classification.

Esmaeilzadeh et al. [9] analysed a general flight dataset and observed some useful patterns of the relationship between feature and flight delay using data visualization methods, which gives an insight into vital features influencing delay and provides a foundation for further model construction.

There is a lot of research being done in the field of Machine Learning, but little focus has been provided to the precursor data pipeline aspect. In this project we will perform scalability tests on MySQL as a data storage system and SparkSQL as a data query system.

We have chosen Spark (SparkSQL is built on top of Spark) as the data querying system because of the following reasons:

1. Good real time performance, due to less shuffling in the data processing.

2. Lazy evaluation of Spark helps in better execution of the queries on big datasets.
3. Good compatibility with libraries such as SparkSQL, Spark MLlib, GraphX and Spark Streaming.

MySQL is used by a lot of big companies such as Uber, Netflix, Amazon, etc. Listed below are some of the advantages which makes MySQL a better data storage system for the task of data analysis:

1. It is a secure database with support of data protection features, data encryption, SSH and SSL support.
2. Flexibility given to the administrator to configure the system based on the performance requirements.
3. On-Demand Scalability - which is one the main reasons we are using it in our data pipeline.
4. 24X7 support and assurance from the company - also helps in better user experience and fast upgrades.

Our main focus will be to get interesting and meaningful insights of the Airline Delay data (this dataset is not used mostly in research - prefer to test models on AOTP or QCLCD datasets). It will include many visualizations, graphs and query processing. After performing analysis on the dataset and pre-processing it, we will train Machine Learning models and test its performances on a few metrics as time taken for training, and inference with accuracy measured along with it.

This paper is organized as follows, Section II consists of the background theory and the literature review, where a brief introduction is provided for different researches done for the Flight Delay Prediction. Section III presents the proposed approach, the dataset used for experimentation and the workflow. Experimental results were listed in Section IV and finally the paper is concluded in Section V.

II. RELATED WORK

There are several factors in different phases of a flight that can lead to its delay such as security delay - time taken for all the security clearances from the respective authorities before the flight's departure, weather delay - delay caused by unfavorable weather conditions, mechanical problems, delay from other flights and many more. The delay caused by these factors may propagate to other flights as well both at departure and the arrival airports [9].

Traditionally, statistical models were applied to predict the effect of these factors on the total delay of a flight [10-12], but they had a lot of limitations such as the assumption/estimation of priors and likelihood, which could result in highly inaccurate predictions if they were not correct [13]. To overcome this several Machine Learning algorithms such as Decision Trees, K- Nearest Neighbors and Random Forest were applied for the task of flight prediction [6]. These methods performed very well and showed reasonable accuracy, but the drawback associated with this methodology was there were some variables which will have an impact on delay of a flight that remained unexplored.

Amongst the Machine Learning techniques applied in several researches, Support Vector Machines' results were superior to the rest of the algorithms. Support Vector Machines is a popular algorithm due to its ability to work well with unbalanced and high-dimensional data and using a regularizer during training to prevent overfitting [14-18].

Hajar et al. [19] in the first step applied a Multi-layered perceptron for the flight delay prediction, later they added selective training which improved the accuracy of the model. In selective training they only trained the model on the instances that were mostly relevant and helped the model to learn about delayed flights. When tested on the dataset extracted from the Bureau of Transportation Statistics, the model initially showed an R^2 score of 0.9048, which increased to 0.956.

Maryam et al. [20] proposed an approach which used a neural network architecture inspired from stack denoising autoencoder (so that it is able to handle the noise in the flight data) along with Levenberg-Marquart algorithm for the optimal parameter value search. They also developed two other algorithms, one based on LM algorithm along with an autoencoder and the other based on stack denoising autoencoder. The proposed model performed with an accuracy of more than 90% on an imbalanced dataset.

Guan et al. [21] used Long Short Term Memory, due to the fact that they are better in pertaining information from previous nodes when given sequential data. When tested and compared along with other Machine Learning algorithms such as Random Forest, the proposed algorithm showed better performance with an accuracy score of 90.2%.

A hybrid of Random Forest Regression and Maximal Information Coefficient - RFR-MIC was presented by Guo et al. [22]. They validated and tested the model on flight data covering several routes and multiple airports. The model performed better than Linear Regression, K-Nearest Neighbors and Artificial Neural Networks.

Weather Impacted Traffic Index (WITI) - a well established toolset and metric can be used to measure the effect of weather and traffic on the delays [23]. Klein et al. classified the WITI components into more classes and then used the WITI model in an individual airport fashion for prediction to estimate the delay for a flight after training the model on historical data.

Qu et al. [24] put forward an approach that made full use of both the flight data and the meteorological data. They proposed two deep convolutional architectures based on the fusion of meteorological data - Dual-channel Convolutional Neural Network and Squeeze and Excitation-Densely Connected Convolutional Network. Firstly both the datasets are fused into the model and then models are used to extract the features automatically based on the fused data. These models performed better than state-of-the-art methods and showed an accuracy score of 92.1 and 93.19% respectively.

III. APPROACH

With increase in the success of deep learning models such as Recurrent Neural Networks with Attention layers and transformers, they have been applied for the task forecasting and have performed with high accuracy. In this project we will be focusing on the scalability of the whole data pipeline, where we will be executing queries to get some insights on the data, loading and storing of data, and training various Machine Learning algorithms for the job of prediction.

Scalability check can be performed using many techniques-

1. Response Time/Latency
2. Throughput
3. CPU Usage
4. Memory Usage
5. Network Usage

We will be testing our proposed approach on two different metrics, which will be related to two different aspects of the approach. Firstly, response time of the data systems-MySQL and SparkSQL-will be tested by running queries and processing on different sizes of data. Response time is the time taken by a system/application to send the response for a generated request. We measure the time taken by different systems components to do a particular task when provided with a different size of dataset. If the latency/response time doesn't increase exponentially, we will consider the components to be scalable, else not. Secondly, accuracy along with total time taken will be used as the performance metrics for the Machine learning algorithm. Together, these will be used to give conclusions related to the scalability of the pipeline.

Accurate prediction of flights takes a lot of features into account such as - ‘Weather’, ‘Air Speed’ and many others. Getting this type of data requires installing a lot of sensors at different airports and monitoring them frequently to get updates on the conditions. Unfortunately, processing such a dataset takes a lot of time and computational resources. The dataset we are using for testing the scalability and prediction is the Airlines Delay dataset from Kaggle [25]. The size of this dataset is 248 MB and contains the summary information on the number of on-time, delayed, canceled and diverted flights appearing in DOT's monthly Air Travel Consumer Report. In our dataset we do not have much information on weather, air traffic and other factors that may result in delay.

Main focus will be given to find meaningful relationships between the attributes using different visualization techniques, do the feature extraction and engineering based on the features that correlates with the class labels. Data storage part, where the data is stored, and some simple queries are executed on the dataset to get some insights will be performed using MySQL and later complex queries on these tables for getting better and deep insights from the data will be performed using SparkSQL (Data querying engine). Once we have some insights, for better understanding of viewers during the presentation we will use Python visualization libraries such as Matplotlib and seaborn to create diagrams for each of the results (tables). Next part is Feature extraction and engineering where we will try to find correlation amongst the variables and the output - will remove all the features that are irrelevant. Once we have all the features contributing to the predictions, we will normalize the features if needed.

Finally, various ML algorithms will be used to predict whether a flight is delayed or not. For classification of the flight delays, we are currently implementing these classifiers: *NaiveBayes, RandomForest, Adaboost [26], XGBoost [27], Gradient Boosting* [28] and *TabNet* [29] machine learning models. Additionally, we are comparing the results with not just the accuracy, but also the time required for training, inference by the model. Some of the features which have the potential to cause leakage like ArrDelay time, Arrival time etc. have been dropped. After this, for model prediction, we have divided the features into two groups. One contains the numerical features and other the categorical ones. We converted the categorical features to one-hot encoding features and added them with the numerical features making a feature rich dataset.

Figure 1 represents the data pipeline for the project, as described above. Due to the carefully made choices for the system components we expect the pipeline to be scalable, that is if there is an increase in the size of the dataset then the latency should increase linearly and not in a superlinear manner.

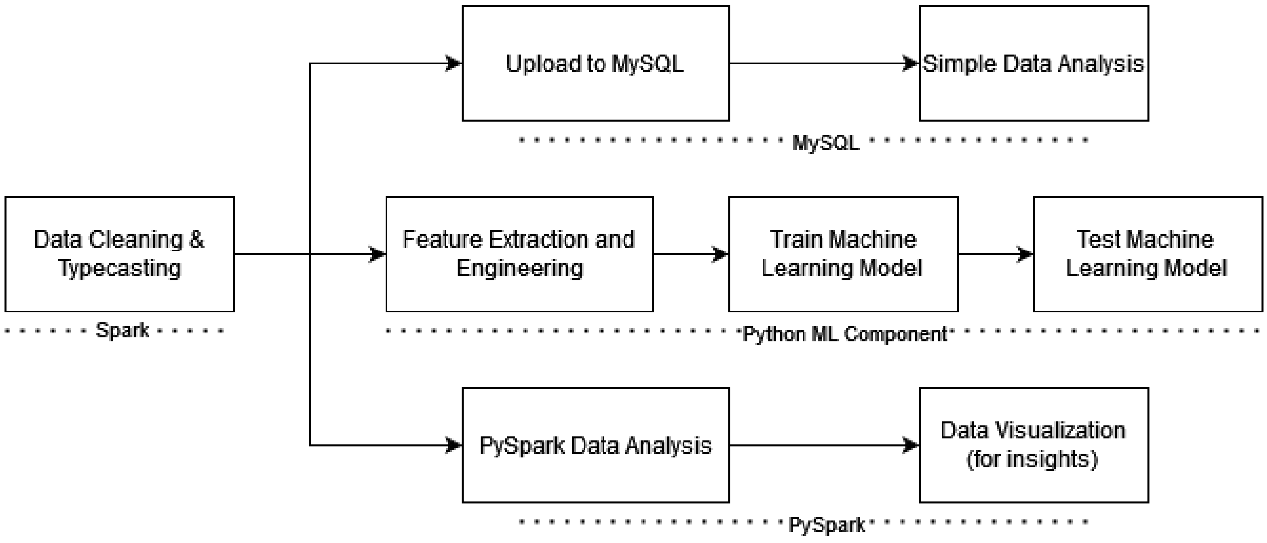

Upload to MySQL
Simple Data Analysis
MySQL
Data Cleaning & Typecasting
Spark
Feature Extraction and Engineering
Train Machine Learning Model
Test Machine Learning Model
Python ML Component
PySpark Data Analysis
Data Visualization (for insights)
PySpark

Figure 1: Proposed Approach

IV. EXPERIMENT

In this section we will state the results of MYSQL and SparkSQL queries. Initially, the whole dataset was loaded in SparkSQL's DataFrame; it has a property of having all the columns of type Text (or String) by default. Therefore we need to type cast properly in order to run successful queries in later parts. Data Types for each column could be easily inferred by analyzing all the unique entries in a particular column. After changing columns into appropriate data types, the next step was to handle missing data/null values. Usually there are 2 steps to resolve this issue, either we delete those rows or we analyze them and find a relation between those null values. Because we wanted to test the scalability of the whole data pipeline, we don't want to reduce the size of our dataset, therefore we went by the second approach, and noted down following relations:

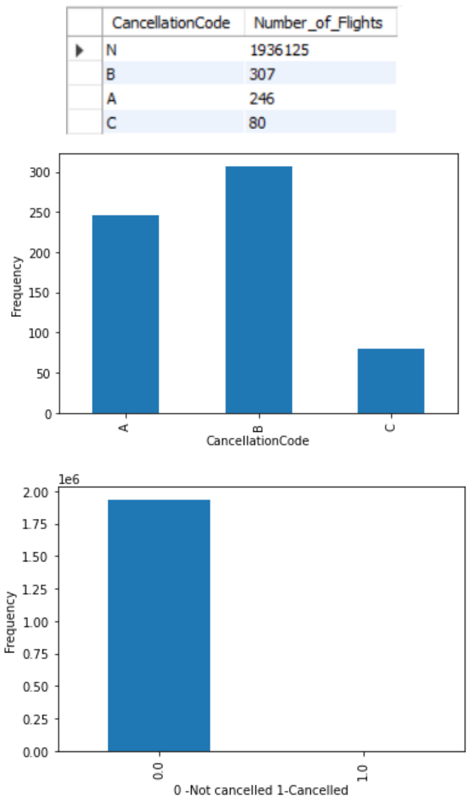
a) Flights that are diverted have null values at columns - 'CRSElapsedTime', 'ActualElapsedTime', 'ArrDelay' and 'AirTime', because the flight never landed at the correct destination.
b) Whenever a flight is cancelled or diverted, this results in null values of "ArrTime" and "TaxiIn", as these columns are specific to the destination airport and because the flight never landed there, so these columns are filled with null values.
c) Similarly, if a flight is cancelled and it never takes off, this would result in null values for the column "TaxiOut"

Now we have done some initial analysis of the dataset and have converted all the null values to 0 instead of removing them because of the above inference. Next step is to load the dataset into a data storage system - MySQL.

Due to the dull analysis nature of SQL, we performed basic analysis which are mentioned below. Figure 2 (a) represents the distribution of different cancellation codes amongst the dataset. Cancellation code represents the reason why the flight got cancelled, A - Carrier, B - Weather, C - NAS and N represents when the flight was not cancelled. We can notice that only a few flights got cancelled and most of them were either because of Weather conditions or the Carrier. Figure 2(b) represents the dissemination of the number of flights that were diverted.

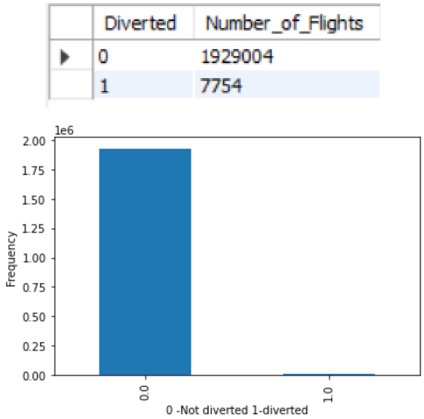
We combined the results from columns Diverted, Cancelled and ArrDelay into one column called 'label', where 0 - flights that were slightly delayed, 1- flights that were highly delayed, 2 - flights that were diverted and 3- flights that were cancelled. Figure 2 (c) illustrates a table showing the flight falling in each category.

Figure 2 (d), (e) and (f) shows the top 5 carriers that are responsible for most delayed, cancelled and diverted flights respectively. Carriers (top 5) that are having the maximum average delay for their flights is described in Figure 2 (g). Last part of analysis in MySQL is shown in Figures 2 (h) and (i), where we analyzed the airports which had their flights to have the most delay (for both arrival and destination).



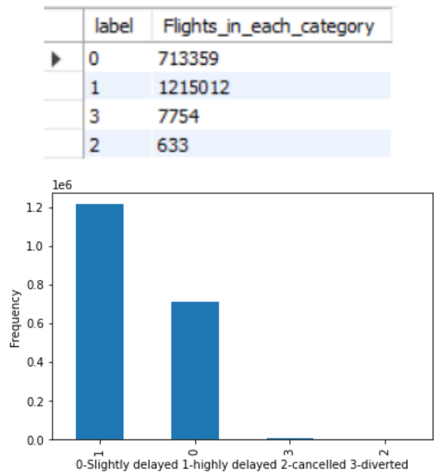
	CancellationCode	Number_of_Flights
▶	N	1936125
	B	307
	A	246
	C	80

(a)



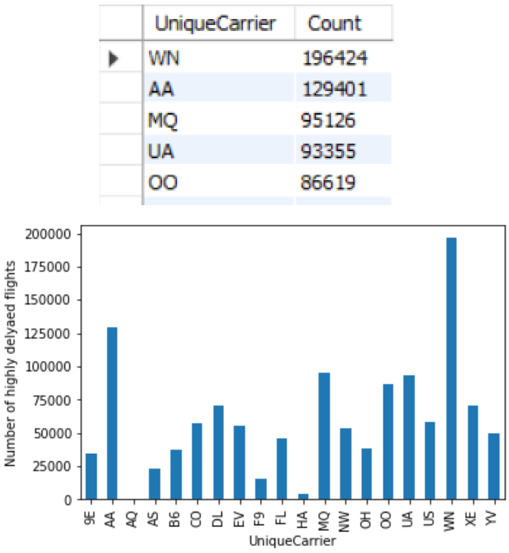
	Diverted	Number_of_Flights
▶	0	1929004
	1	7754

(b)



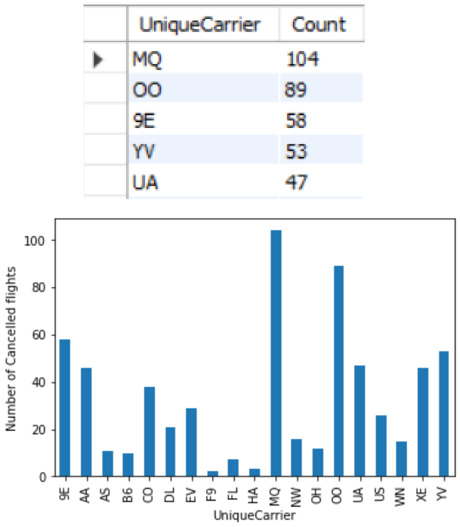
	label	Flights_in_each_category
▶	0	713359
	1	1215012
	3	7754
	2	633

(c)



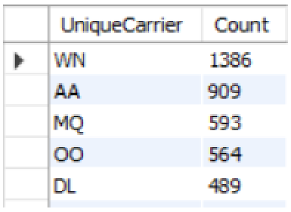
	UniqueCarrier	Count
▶	WN	196424
	AA	129401
	MQ	95126
	UA	93355
	OO	86619

(d)

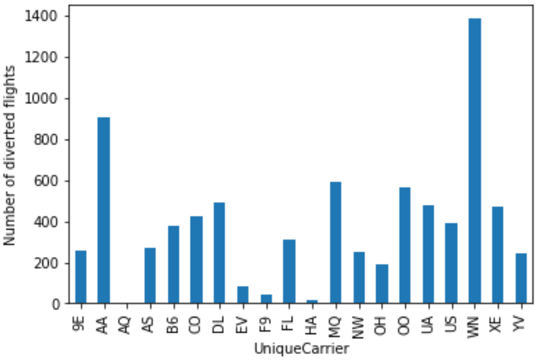


	UniqueCarrier	Count
▶	MQ	104
	OO	89
	9E	58
	YV	53
	UA	47

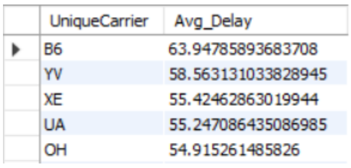
(e)



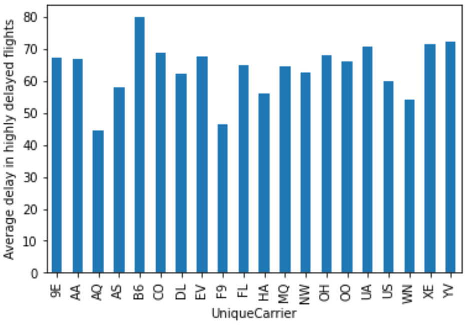
UniqueCarrier	Count
WN	1386
AA	909
MQ	593
OO	564
DL	489



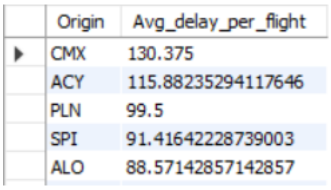
(f)



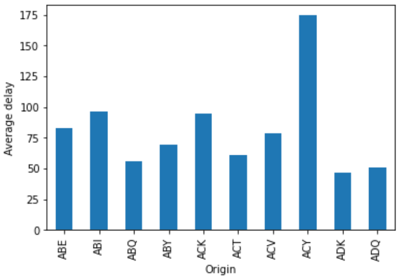
UniqueCarrier	Avg_Delay
B6	63.94785893683708
YV	58.563131033828945
XE	55.42462863019944
UA	55.247086435086985
OH	54.915261485826



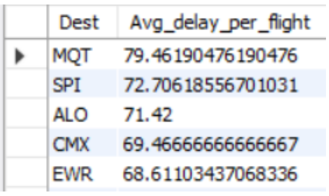
(g)



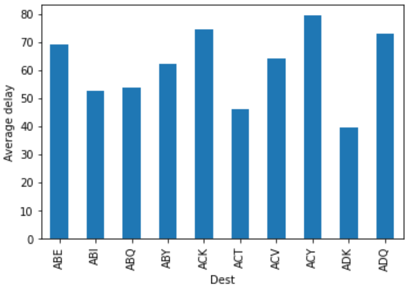
Origin	Avg_delay_per_flight
CMX	130.375
ACY	115.88235294117646
PLN	99.5
SPI	91.41642228739003
ALO	88.57142857142857



(h)



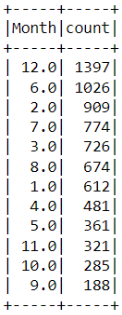
Dest	Avg_delay_per_flight
MQT	79.46190476190476
SPI	72.70618556701031
ALO	71.42
CMX	69.46666666666667
EWR	68.61103437068336



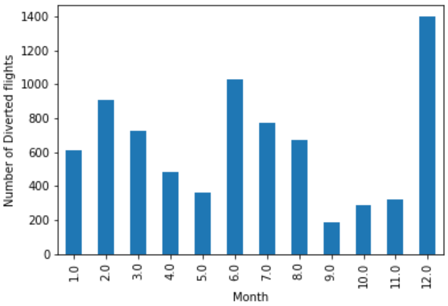
(i)

Figure 2: Results from the analysis done using MySQL queries

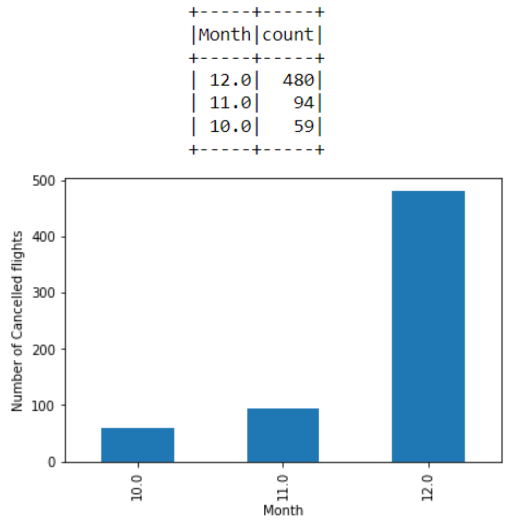
After loading the updated dataset back into Spark from MySQL, we executed some basic queries using SparkSQL, to group the number of flights that are diverted, cancelled and highly delayed based on the month (Figure 3 (a), (b) and (c) respectively), and also group them based on their departure hour and distance bucket (Figure 3 (d) and (e)). For the sake of simplicity, in results (d) and (e), the flag Delayed is added to the dataframe if ActualElapsedTime is more than CRSElapsedTime. Otherwise, the OnTime flag is added.



```
+-----+-----+
|Month|count|
+-----+-----+
| 12.0| 1397|
|  6.0| 1026|
|  2.0|  909|
|  7.0|  774|
|  3.0|  726|
|  8.0|  674|
|  1.0|  612|
|  4.0|  481|
|  5.0|  361|
| 11.0|  321|
| 10.0|  285|
|  9.0|  188|
+-----+-----+
```

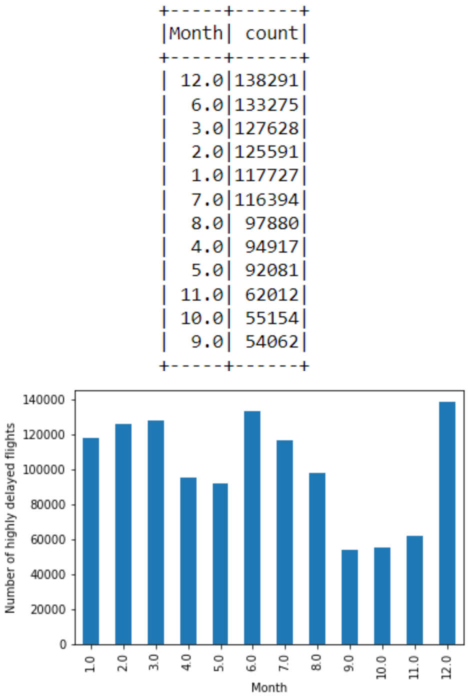


(a)



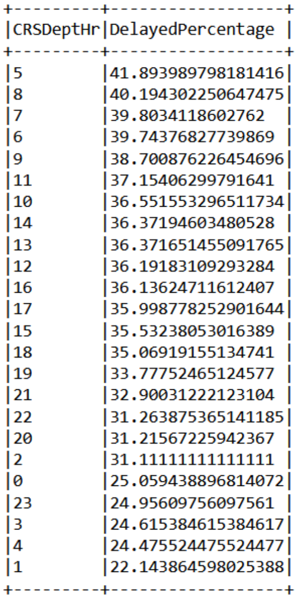
```
+-----+-----+
|Month|count|
+-----+-----+
| 12.0|  480|
| 11.0|   94|
| 10.0|   59|
+-----+-----+
```

(b)



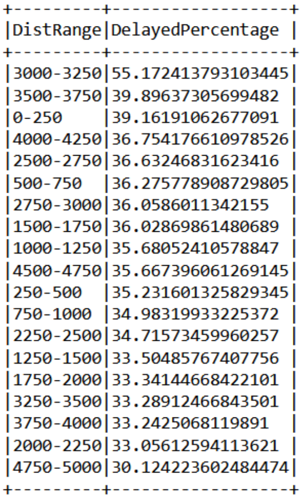
```
+-----+------+
|Month| count|
+-----+------+
| 12.0|138291|
|  6.0|133275|
|  3.0|127628|
|  2.0|125591|
|  1.0|117727|
|  7.0|116394|
|  8.0| 97880|
|  4.0| 94917|
|  5.0| 92081|
| 11.0| 62012|
| 10.0| 55154|
|  9.0| 54062|
+-----+------+
```

(c)



```
+---------+------------------+
|CRSDeptHr|DelayedPercentage |
+---------+------------------+
|5        |41.893989798181416|
|8        |40.194302250647475|
|7        |39.8034118602762  |
|6        |39.74376827739869 |
|9        |38.700876226454696|
|11       |37.15406299791641 |
|10       |36.551553296511734|
|14       |36.37194603480528 |
|13       |36.371651455091765|
|12       |36.19183109293284 |
|16       |36.13624711612407 |
|17       |35.998778252901644|
|15       |35.53238053016389 |
|18       |35.06919155134741 |
|19       |33.77752465124577 |
|21       |32.90031222123104 |
|22       |31.263875365141185|
|20       |31.21567225942367 |
|2        |31.11111111111111 |
|0        |25.059438896814072|
|23       |24.95609756097561 |
|3        |24.615384615384617|
|4        |24.475524475524477|
|1        |22.143864598025388|
+---------+------------------+
```

(d)



```
+---------+------------------+
|DistRange|DelayedPercentage |
+---------+------------------+
|3000-3250|55.172413793103445|
|3500-3750|39.89637305699482 |
|0-250    |39.16191062677091 |
|4000-4250|36.754176610978526|
|2500-2750|36.63246831623416 |
|500-750  |36.275778908729805|
|2750-3000|36.0586011342155  |
|1500-1750|36.02869861480689 |
|1000-1250|35.68052410578847 |
|4500-4750|35.667396061269145|
|250-500  |35.231601325829345|
|750-1000 |34.98319933225372 |
|2250-2500|34.71573459960257 |
|1250-1500|33.50485767407756 |
|1750-2000|33.34144668422101 |
|3250-3500|33.28912466843501 |
|3750-4000|33.2425068119891  |
|2000-2250|33.05612594113621 |
|4750-5000|30.124223602484474|
+---------+------------------+
```

(e)

Figure 3: Results from the analysis using SparkSQL

We performed the scalability test using the Response time as metrics on the first two system components that are - MySQL (data storage system) and SparkSQL (data query/processing system), Table 1 and 2 we present the scalability checks of MySQL and PySpark on a limited subset of queries we executed respectively.

Table 1: Scalability Checks for MySQL

Query	Time taken for execution of Query (s)		
	Dataset Size		
	19368	193676	1936758
Distribution of the Cancellation Code for the Cancelled flights	0.062	0.297	7.078
Frequency of highly delayed flights per Unique Carrier	0.016	0.250	7.016
Average Delay for a flight for each Unique carrier	0.031	0.359	8.406

Table 2: Scalability Checks for PySpark

Query	Time taken for execution of Query (ms)		
	Dataset Size		
	19368	193676	1936758
Inference for Null values	2.707015	4.113074	36.018239
Loading dataset into MySQL	6.028880	39.83403	388.62436
Monthwise Cancelled Flights	0.142998	0.247994	6.404154
Monthwise Diverted Flights	0.444345	0.266495	7.182363
Monthwise Highly delayed flights	0.255014	0.580966	7.249668

From table 1 and 2 we can infer that even when we increase the load, the response time increases linearly and not exponentially for both MySQL and PySpark, hence we can say the preprocessing and data-analysis part of the pipeline is scalable.

Table 3: Comparison of Accuracy and Time Taken for training of different Machine Learning models

Model	Accuracy	Time taken (s)
GaussianNB	0.65473	12.29
Random Forest Classifier	0.8513	384.57
AdaBoost Classifier	0.8491	192.93
Gradient Boosting Classifier	0.8527	677.25
XGB Classifier	0.8560	584.641
TabNet	0.9230	1440.95

The last component consists of different Machine Learning models that were trained on the preprocessed dataset. Performance of models are tested on 2 metrics - time taken for training and the accuracy scores. Table 3 consists of the initial results and the time taken for the classification tasks by several Machine Learning models.

We can infer from Table 3 that TabNet got the highest validation accuracy of 0.9239. However, in terms of time taken to train it is the slowest. Gaussian Naive Bayes performed with the least time, however, its accuracy 0.65473 is the lowest among the other machine learning models.

V. CONCLUSION

Airline delay causes losses worth billions of dollars to passengers, airline carriers, and authorities. There have been many algorithms that have been proposed to solve this issue in previous few years but only a few focused on the scalability issue. In our project, we have proposed a simple end to end machine learning pipeline which predicts airline delay with high accuracy and also provides scalability.

The contributions are made in three folds. The first and second part is composed of data analysis using MySQL and SparkSQL (these system components are used widely in the industry). We executed a bunch of queries to get some meaningful insight from the data, and the latency measures for these query executions were used as a metric to test for scalability. In the experimental results we saw that as the load was increased by a constant factor, it resulted in an linear increase in the Response time. The third and final component was the Machine Learning part, in this as expected, TabNet, Randomforest and XGboost all performed with high accuracy scores with just a little feature engineering.

Our future goal is to incorporate frameworks such as FastAPI [30] in combination with our architecture, which would enable us to build an end to end scalable machine learning system for serving the prediction of flight delay in real time.

VI. TEAM CONTRIBUTIONS

Task	**Members**
Data Acquisition	Anshumaan, Sriharsha
Data Cleansing	Anshumaan, Sriharsha
MySQL Analysis	Pratyush, Anshumaan
SparkSQL Analysis	Pratyush, Sriharsha
Data Visualization	Pratyush, Anshumaan
Feature Extraction	Sriharsha, Pratyush
Model Selection	Sriharsha, Anshumaan
Training and Testing	Sriharsha, Pratyush
Evaluation	Anshumaan, Pratyush
Project Report and Presentation	Anshumaan, Pratyush, Sriharsha

References:

[1] Kim, Young Jin, et al. "A deep learning approach to flight delay prediction." *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016.

[2] N. Xu, G. Donohue, K. B. Laskey, and C.-H. Chen, “Estimation of delay propagation in the national aviation system using bayesian networks,” in 6th USA/Europe Air Traffic Management Research and Development Seminar. Citeseer, 2005.

[3] J. J. Rebollo and H. Balakrishnan, “Characterization and prediction of air traffic delays,” Transportation Research Part C: Emerging Technologies, vol. 44, pp. 231–241, 2014.

[4] Jiang, Yushan, et al. "Applying machine learning to aviation big data for flight delay prediction." *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, 2020.

[5] V. Natarajan, S. Meenakshisundaram, G. Balasubramanian, and S. Sinha, “A novel approach: Airline delay prediction using machine learning,” in 2018 International Conference on Computational Science and Computational Intelligence (CSCI), 2018, pp. 1081–1086.

[6] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris, “Prediction of weather induced airline delays based on machine learning algorithms,” in 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 2016, pp. 1–6.

[7] N. Chakrabarty, “A data mining approach to flight arrival delay prediction for american airlines,” in 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), 2019, pp. 102–107.

[8] B. Thiagarajan, L. Srinivasan, A. V. Sharma, D. Sreekanthan, and V. Vijayaraghavan, “A machine learning approach for prediction of on time performance of flights,” in 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), 2017, pp. 1–6.

[9] Esmaeilzadeh, Ehsan, and Seyedmirsajad Mokhtarimousavi. "Machine learning approach for flight departure delay prediction and analysis." *Transportation Research Record* 2674.8 (2020): 145-159.

[10] Xu, N., L. Sherry, and K. B. Laskey. Multifactor Model for Predicting Delays at U.S. Airports. Transportation Research Record: Journal of the Transportation Research Board, 2008. 2052: 62–71.

[11] Tu, Yufeng, Michael O. Ball, and Wolfgang S. Jank. "Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern." *Journal of the American Statistical Association* 103.481 (2008): 112-125.

[12] Mueller, Eric, and Gano Chatterji. "Analysis of aircraft arrival and departure delay characteristics." *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*. 2002.

[13] Delen, D., R. Sharda, and M. Bessonov. Identifying Significant Predictors of Injury Severity in Traffic Accidents using a Series of Artificial Neural Networks. Accident Analysis & Prevention, Vol. 38, No. 3, 2006, pp. 434–444.

[14] Khaksar, Hassan, and Abdolrreza Sheikholeslami. "Airline delay prediction by machine learning algorithms." *Scientia Iranica* 26.5 (2019): 2689-2702.

[15] Huang, Chengquan, L. S. Davis, and J. R. G. Townshend. "An assessment of support vector machines for land cover classification." *International Journal of remote sensing* 23.4 (2002): 725-749.

[16] Mokhtarimousavi, Seyedmirsajad. "A time of day analysis of pedestrian-involved crashes in California: Investigation of injury severity, a logistic regression and machine learning approach using HSIS data." *Institute of Transportation Engineers. ITE Journal* 89.10 (2019): 25-33.

[17] Auria, Laura, and Rouslan A. Moro. "Support vector machines (SVM) as a technique for solvency analysis." (2008).

[18] Chen, Haiyan, Jiandong Wang, and Xuefeng Yan. "A fuzzy support vector machine with weighted margin for flight delay early warning." *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 3. IEEE, 2008.

[19] Alla, Hajar, Lahcen Moumoun, and Youssef Balouki. "A multilayer perceptron neural network with selective-data training for flight arrival delay prediction." *Scientific Programming* 2021 (2021).

[20] Yazdi, Maryam Farshchian, et al. "Flight delay prediction based on deep learning and Levenberg-Marquart algorithm." *Journal of Big Data* 7.1 (2020): 1-28.

[21] Gui, Guan, et al. "Flight delay prediction based on aviation big data and machine learning." *IEEE Transactions on Vehicular Technology* 69.1 (2019): 140-150.

[22] Guo, Zhen, et al. "A novel hybrid method for flight departure delay prediction using Random Forest Regression and Maximal Information Coefficient." *Aerospace Science and Technology* 116 (2021): 106822.

[23] Klein, Alexander, Chad Craun, and Robert S. Lee. "Airport delay prediction using weather-impacted traffic index (WITI) model." *29th Digital Avionics Systems Conference*. IEEE, 2010.

[24] Qu, Jingyi, et al. "Flight delay prediction using deep convolutional neural network based on fusion of meteorological data." *Neural Processing Letters* 52.2 (2020): 1461-1484.

[25] https://www.kaggle.com/datasets/giovamata/airlinedelaycauses

[26] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. Information Retrieval, 13(3):254–270, 2010.

[27] Chen, T. & Guestrin, C. Xgboost: a scalable tree boosting system. In Proc. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 785–794 (ACM, 2016).

[28] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232, 2001.

[29] Arik SO, Pfister T. TabNet: Attentive Interpretable Tabular Learning. arXiv preprint arXiv:1908.07442. 20 August 2019.

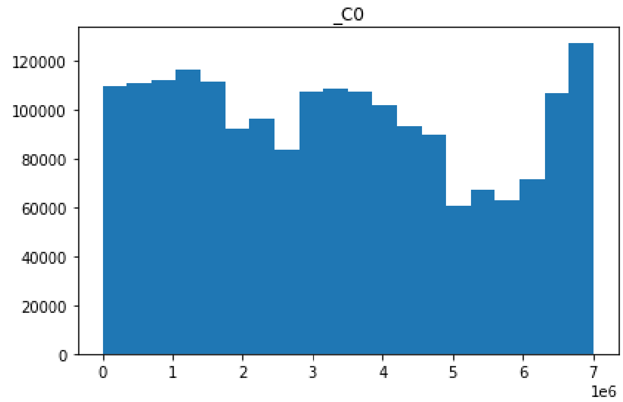
[30] github.com/tiangolo/fastapi/releases/tag/0.88.0

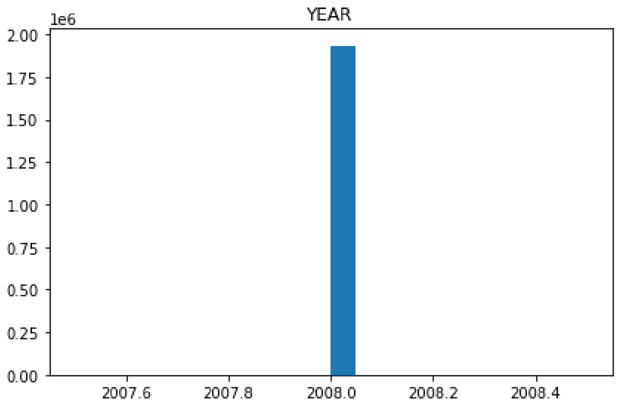
SUPPLEMENTARY MATERIAL

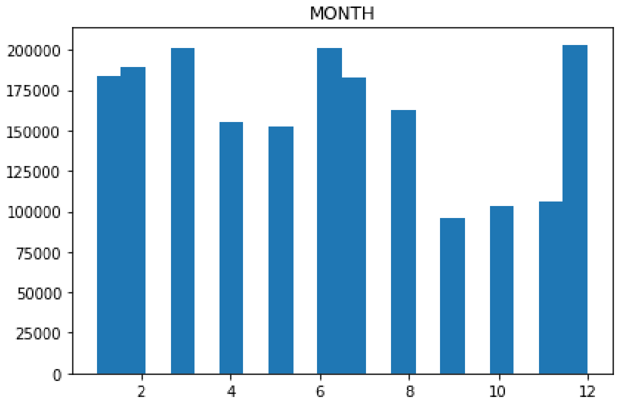
The analysis and design for any large scale system has a major dependency on the data used, that is, analyzing the classifying of the features that contributes more towards the output label has a significant impact on the results that are shown by the ML component. In this section we will be presenting the analysis we performed on the dataset apart from what we mentioned in the Experiment section.

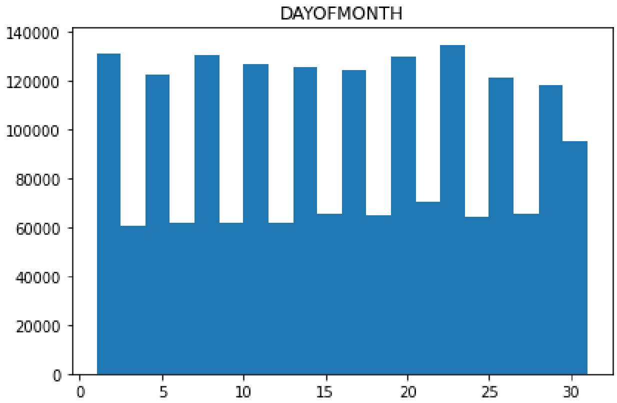
Figure 1 represents the distribution of each numerical column in the dataset. Distribution analysis gives us an insight on the values that usually a feature takes and also provides information about the outliers. If described briefly we can infer the following from the distributions:

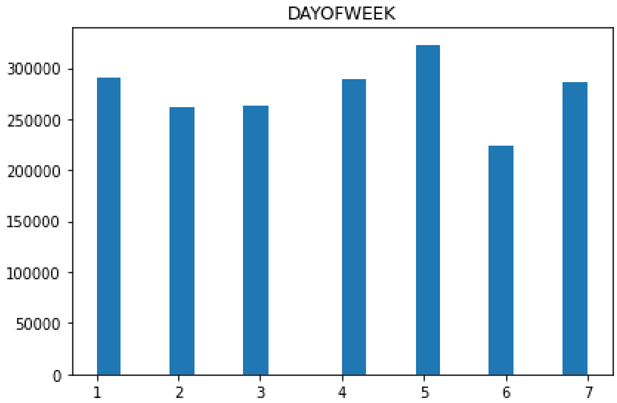
1) _C0 has a distribution same as an index variable
2) YEAR takes only one value 2008 - that is all the data was collected in the year 2008.
3) Comparable amount of flight data is provided for all the months, the days of month and the days of a week
4) Very few flights are scheduled between 12am to 5am. (Most of the flights are scheduled between 5am and 12am)
5) Most of the flights have a air time below 4 hours
6) More than 90% of the flights have a travel distance below 2500 miles.
7) A small percentage of the total flights fall into the category of cancelled or diverted.

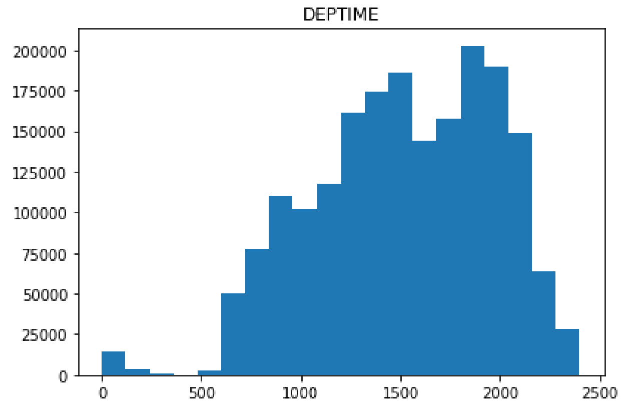


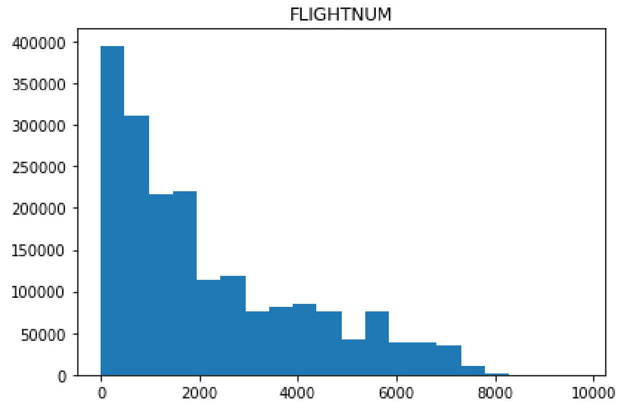


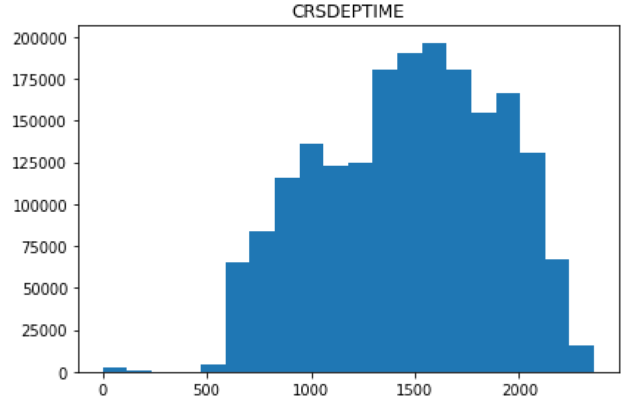


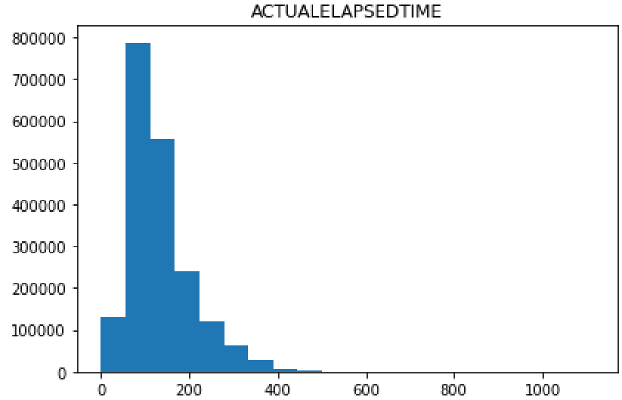


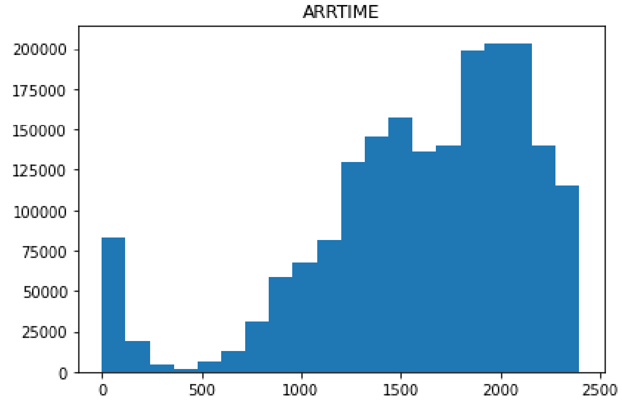


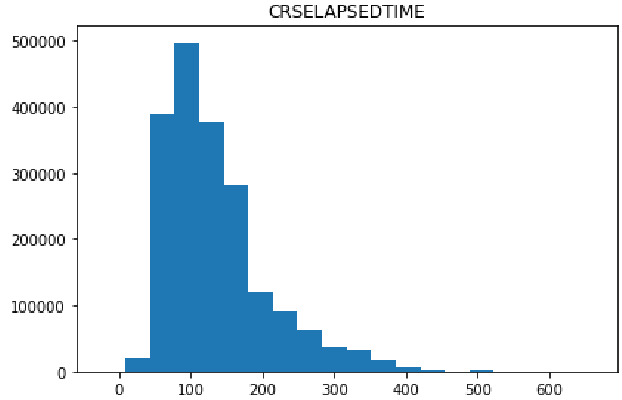

DEPTIME
0
25000
50000
75000
100000
125000
150000
175000
200000
0
500
1000
1500
2000
2500

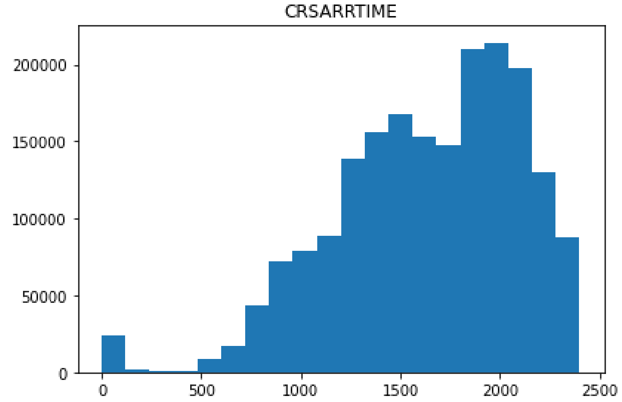

FLIGHTNUM
0
50000
100000
150000
200000
250000
300000
350000
400000
0
2000
4000
6000
8000
10000

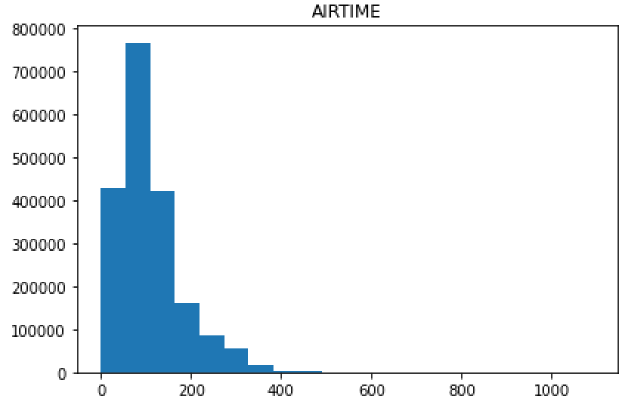

CRSDEPTIME
0
25000
50000
75000
100000
125000
150000
175000
200000
0
500
1000
1500
2000

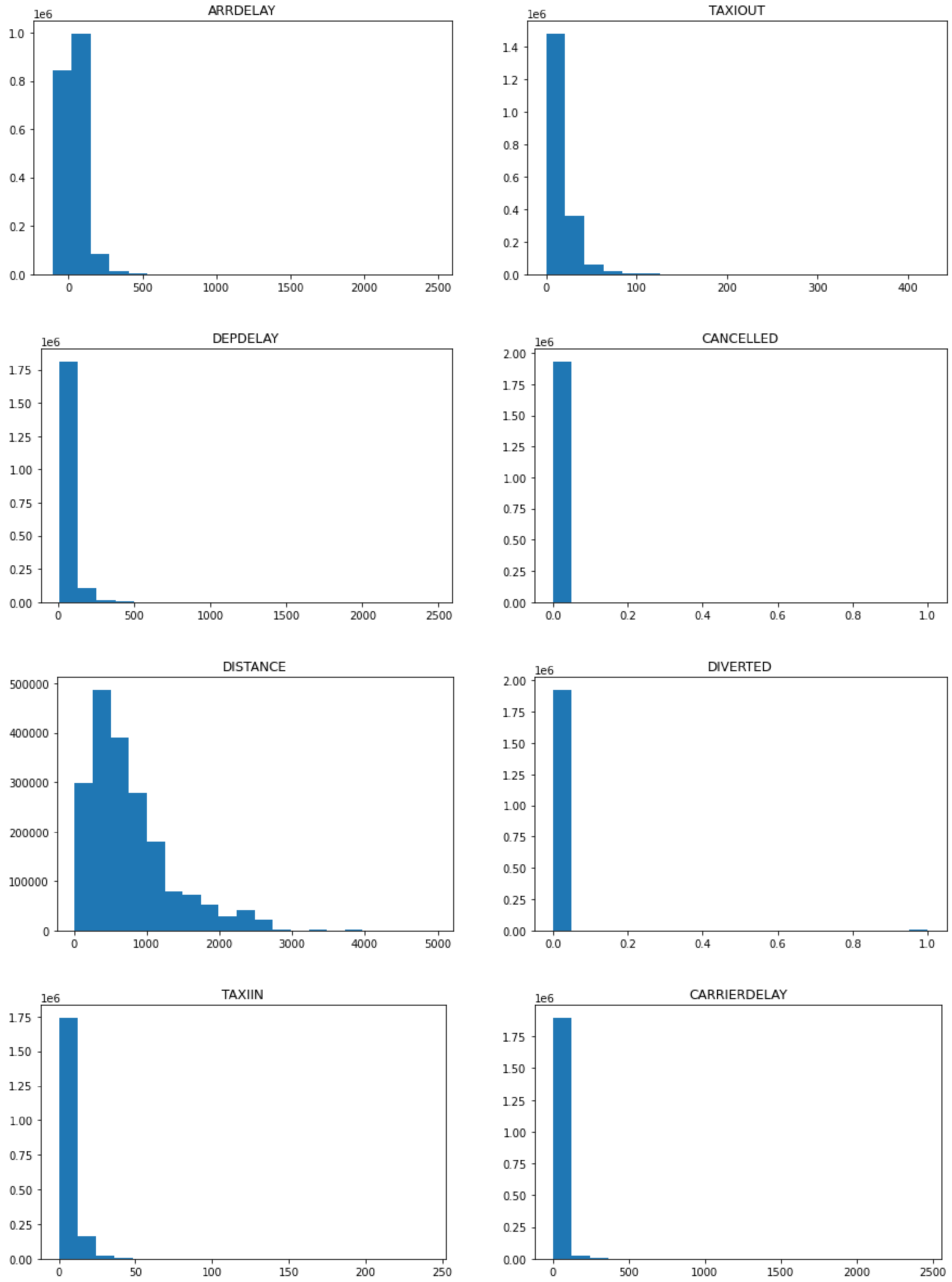

ACTUALELAPSEDTIME
0
100000
200000
300000
400000
500000
600000
700000
800000
0
200
400
600
800
1000


ARRTIME
0
25000
50000
75000
100000
125000
150000
175000
200000
0
500
1000
1500
2000
2500


CRSELAPSEDTIME
0
100000
200000
300000
400000
500000
0
100
200
300
400
500
600


CRSARRTIME
0
50000
100000
150000
200000
0
500
1000
1500
2000
2500


AIRTIME
0
100000
200000
300000
400000
500000
600000
700000
800000
0
200
400
600
800
1000


ARRDELAY
1e6
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
TAXIOUT
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
DEPDELAY
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
500
1000
1500
2000
2500
CANCELLED
1e6
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
0.8
1.0
DISTANCE
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
DIVERTED
1e6
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
0.8
1.0
TAXIIN
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
50
100
150
200
250
CARRIERDELAY
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
500
1000
1500
2000
2500

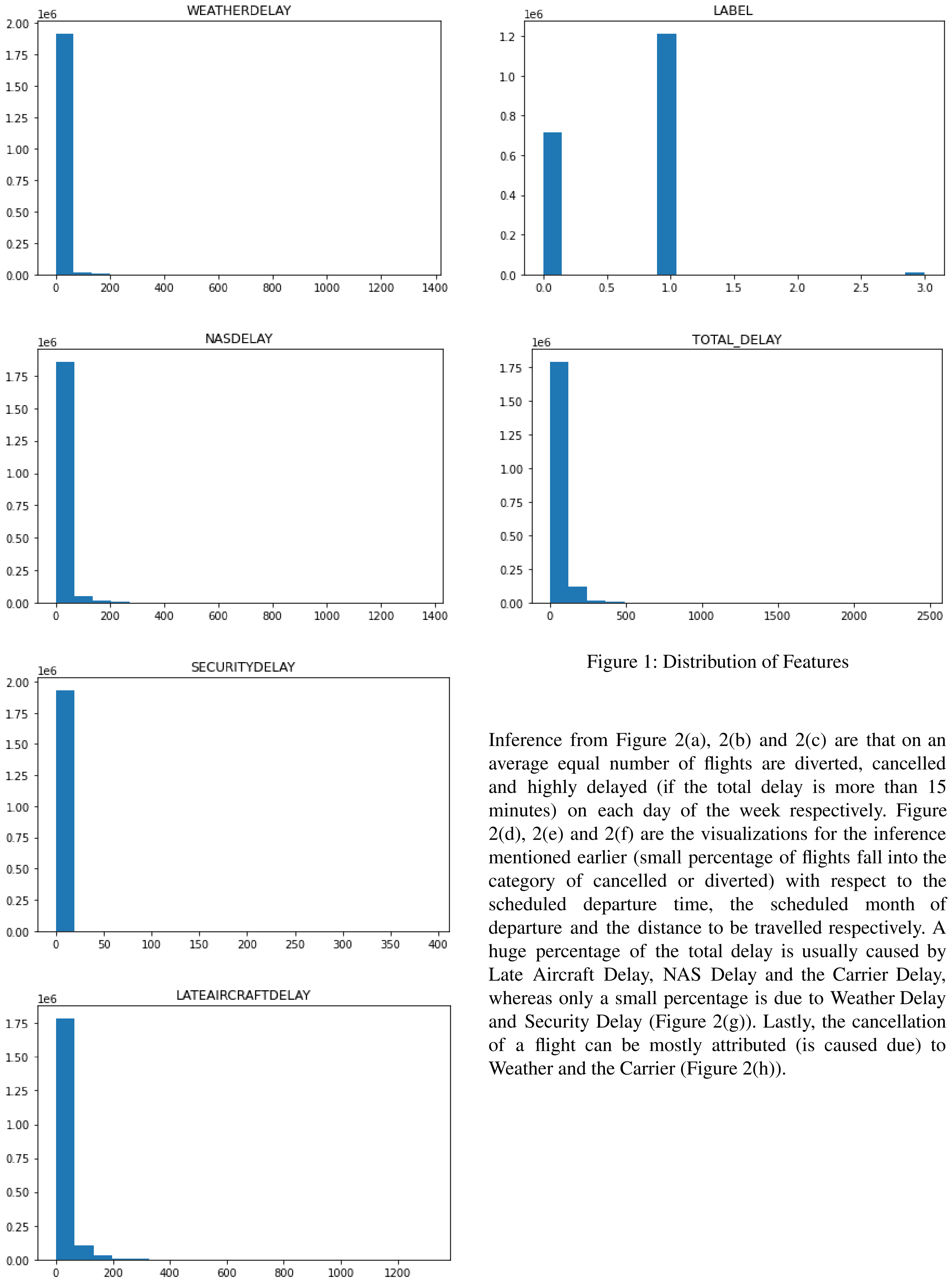
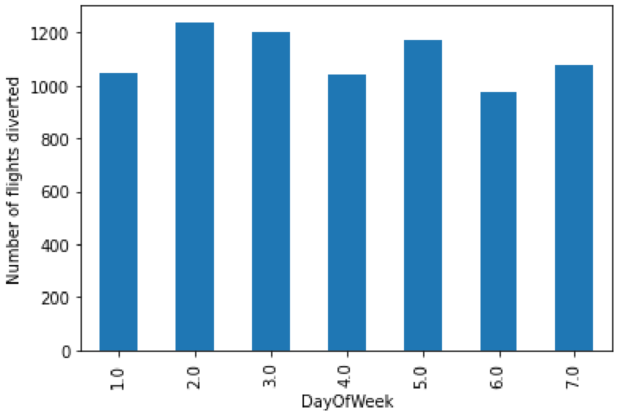
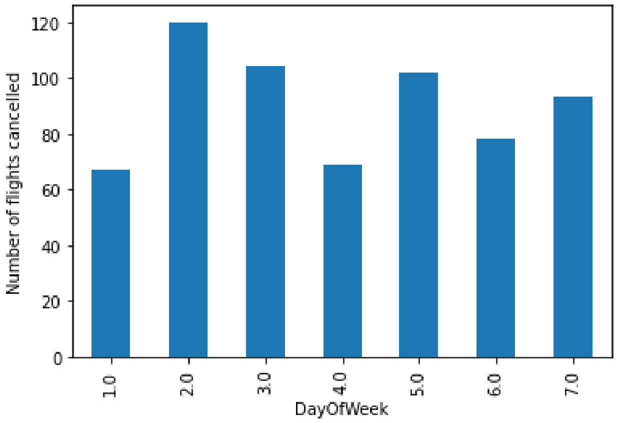


Figure 1: Distribution of Features

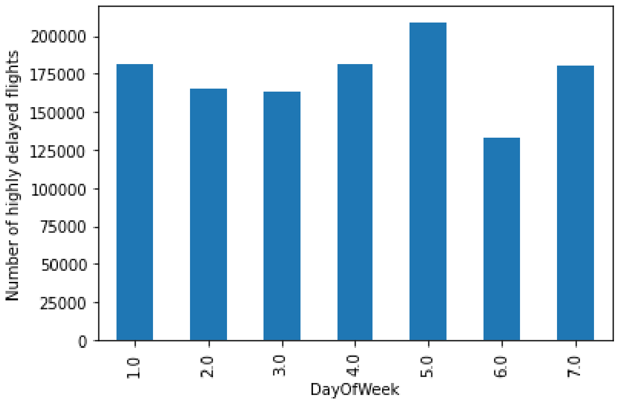
Inference from Figure 2(a), 2(b) and 2(c) are that on an average equal number of flights are diverted, cancelled and highly delayed (if the total delay is more than 15 minutes) on each day of the week respectively. Figure 2(d), 2(e) and 2(f) are the visualizations for the inference mentioned earlier (small percentage of flights fall into the category of cancelled or diverted) with respect to the scheduled departure time, the scheduled month of departure and the distance to be travelled respectively. A huge percentage of the total delay is usually caused by Late Aircraft Delay, NAS Delay and the Carrier Delay, whereas only a small percentage is due to Weather Delay and Security Delay (Figure 2(g)). Lastly, the cancellation of a flight can be mostly attributed (is caused due) to Weather and the Carrier (Figure 2(h)).


Number of flights diverted
0
200
400
600
800
1000
1200
1.0
2.0
3.0
4.0
5.0
6.0
7.0
DayOfWeek

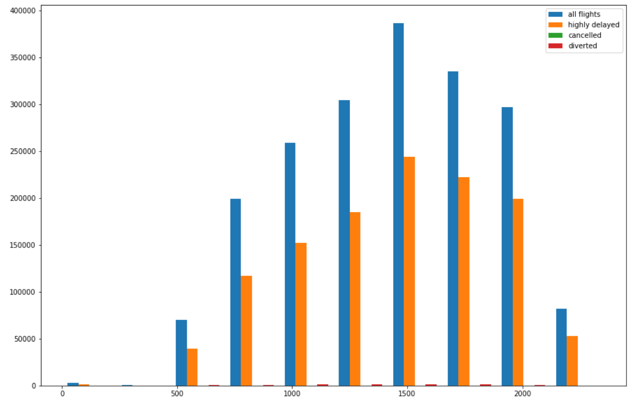
(a)


Number of flights cancelled
0
20
40
60
80
100
120
1.0
2.0
3.0
4.0
5.0
6.0
7.0
DayOfWeek

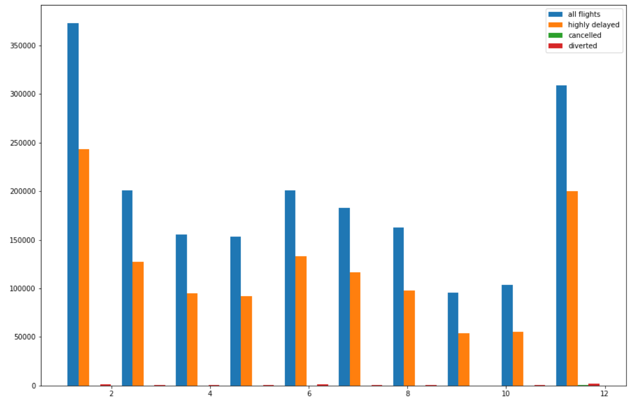
(b)


Number of highly delayed flights
0
25000
50000
75000
100000
125000
150000
175000
200000
1.0
2.0
3.0
4.0
5.0
6.0
7.0
DayOfWeek

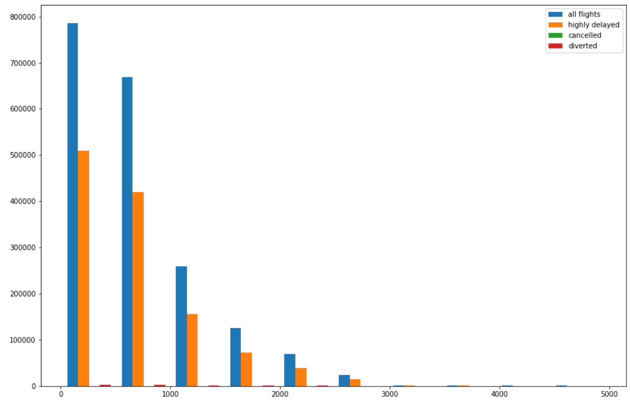
(c)


all flights
highly delayed
cancelled
diverted
0
50000
100000
150000
200000
250000
300000
350000
400000
0
500
1000
1500
2000

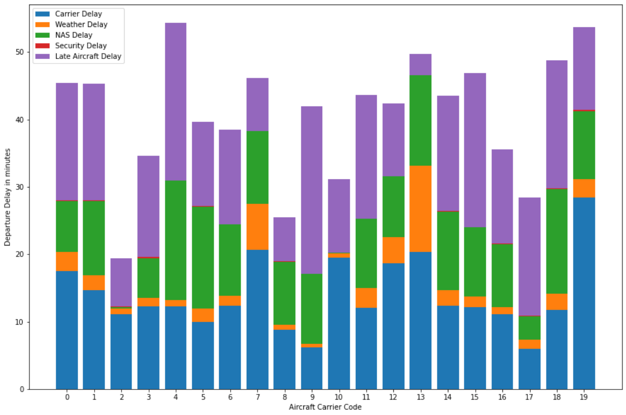
(d)


all flights
highly delayed
cancelled
diverted
0
50000
100000
150000
200000
250000
300000
350000
2
4
6
8
10
12

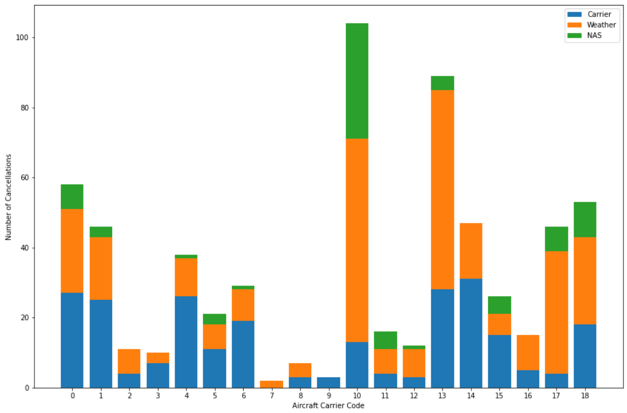
(e)


all flights
highly delayed
cancelled
diverted
0
100000
200000
300000
400000
500000
600000
700000
800000
0
1000
2000
3000
4000
5000

(f)


Carrier Delay
Weather Delay
NAS Delay
Security Delay
Late Aircraft Delay
Departure Delay in minutes
0
10
20
30
40
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Aircraft Carrier Code

(g)



(h)

Figure 2: Data Analysis using Additional Visualizations

Feature Engineering and Extraction is the process of redesigning the input features in a way using statistics and machine learning methods that helps our model learn in a better way. In order to reconstruct the features, drop some of the features or combine some features we need to have a thorough knowledge of what a feature represents and how it is related with the other features. First part is done using the exploratory data analysis we did on the dataset, now to check the correlation between different features we used a heatmap (Figure 3). We use this heatmap to perform the feature extraction on our dataset and then perform the Machine Learning experiments on this extracted dataset.

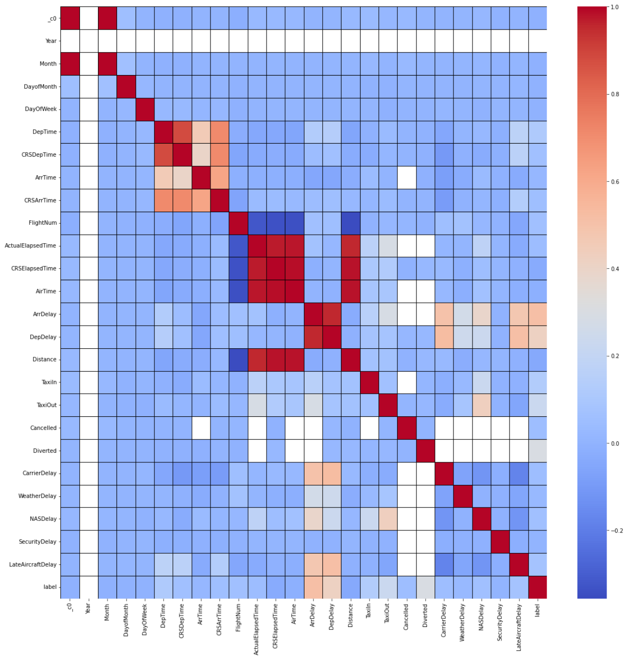


Figure 3: Correlation between different features