An Analysis of the Average Gate Arrival Delay Using Machine Learning

Dhruvil Dalwadi, Sarah Denlinger, Carol Kingori, and Masoud Soroush
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250

(Dated: April 30, 2021)

Abstract

A large portion of metrics used for airport analyses is formed by various kinds of average flight delays. In this paper, we demonstrate how an average metric in an airport analysis can be predicted by the knowledge of other average metrics, without the need to refer to individual flight delays. Our case study analyzes the data associated with the Chicago O'Hare International Airport (ORD) for the period 3/1/2019 - 5/31/2019 obtained from the Federal Aviation Administration (FAA) database. Using various standard supervised machine learning algorithms, we analyze the average gate arrival delay at ORD and identify the contributing factors among the average metrics used for airport analyses. After identifying the contributing variables to the average gate arrival delay, we first employ standard regression techniques to predict the amount of the average gate arrival delay at ORD. Second, using several supervised classification techniques, we offer a classification analysis of the average gate arrival delay at ORD.

I. INTRODUCTION

The issue of flight delays is one that has been around for a long time and still proves to be costly for airlines and the Federal Aviation Administration (FAA) today. According to the Bureau of Transportation Statistics, in 2019, 18.02% of flights were delayed [1]. These delays can be costly and drive customers away from certain airlines. The FAA used a FAA-sponsored study from 2007 to estimate the cost of delays for the years of 2016 to 2019. The FAA estimated that the total cost of delays in 2019 was 33 billion dollars [2].

Flight delays can also cause negative emotions in passengers. A study done by Kim and Park [3] found that service delays significantly influenced anger in passengers and anger was found to have a negative influence on repurchase intention and even have a positive influence on negative word-of-mouth. This study showed that airline service delays can cause passengers to feel angry and decide not to fly with certain airlines again and speak poorly of the airline to others.

In addition to severe financial and emotional negative impacts, flight delays can significantly affect the rankings of airports as well. According to FAA Aviation System Performance Metrics (ASPM) for airport analysis [4], various kinds of flight delays are considered among the metrics to evaluate the efficiency of an airport (for a partial list of different kinds of flight delays considered by FAA see table (TABLE I)). However, what is relevant for airport analyses is the notion of *average delay* rather than the individual delays caused by individual flights. The primary method of calculating average flight delays – relevant for airport analyses – is to cumulate the data for all individual incoming and outgoing flights at the airport in consideration, and to take the average of the individual delays. Undoubtedly, this approach is costly, as it requires an immense amount of data to collect, maintain, and process.

In this paper, we propose an alternative approach towards average flight delays. Instead of tracking all individual flights to and from an airport, we predict a specific average flight delay (*e.g.* the average gate arrival delay) by the knowledge of other average flight delays at the airport in consideration. In other words, in order to predict a specific average flight delay, we do not trace the individual flights and merely rely on the average metrics used for the airport analysis. To demonstrate the feasibility of this proposal, we perform a case study in this paper.

To conduct our case study, we choose the Chicago O'Hare International Airport, referred to as ORD (*i.e.* its FAA airport code) throughout this paper. In 2018, ORD was ranked as the world's sixth busiest airport, having 83.2 million passengers and handling approximately 904,000 aircraft movements [5]. In 2001, Chicago announced the O'Hare Modernization Program (OMP) by Chicago Aviation Department, a program that was envisioned to modernize the airport and improve efficiency, capacity, and safety [6]. Since then, ORD has gone through a number of substantial modernization projects ranging from a relocated cargo facility featuring the largest airport green roof in the US, to initiatives for increasing runways and expanding terminals. In 2017, ORD was ranked 7th in the world of mega airports with an on-time performance (OTP) of 79.85% [7]. These considerations all point toward the fact that ORD is functioning well under normal circumstances, and one is not anticipating for massive anomalies and irregularities.

To narrow down our case study even further, we focus on a dataset that represents the ASPM metrics for airport analysis for ORD during the period March 1, 2019 through May 31, 2019. This time period is after the completion of the modernization project (OMP), and well before the time COVID-19 restrictions were put in place. Moreover, the months March to May are more representative of a normal travel season.

The analysis carried out in this paper is categorized un-

der the class of supervised learning. Our aim is to predict the average gate arrival delay at ORD during the above mentioned time period. Although other types of average delays among ASPM metrics for airport analysis can be declared as the target variable, we will carry out our supervised analysis with the average gate arrival delay as the target. Our motivation for this choice relies on the expectation that the average gate arrival delay exhibits the least stochasticity in its behavior among the average metrics used for the airport analysis. Nonetheless, we expect our methodology to be applicable to other choices of the target variable.

In order to analyze the chosen target variable (*i.e* average gate arrival delay at ORD), we employ a series of standard machine learning algorithms to select the relevant ASPM features among metrics used for airport analysis. We will then implement various regression analyses to predict the amount of the average gate arrival delay. The second part of our analysis employs various supervised classification techniques to predict the class associated with the average gate arrival delay at ORD.

This paper is organized as follows. In section (II) we briefly review some of the relevant studies found in the literature. In section (III), we demonstrate how the dataset used in this study is retrieved from the FAA database. Section (IV) describes the methodology of the analysis carried out in this paper. Section (V) forms the bulk of the analysis conducted in this study, and consists of two parts. Section (V A) presents the results of various regression analyses conducted in this paper to predict the average gate arrival delay at ORD. Section (V B) summarizes the results of various classification algorithms to predict the binary class associated with the average gate arrival delay at ORD. We devote section (VI) to conclusions and possible future directions.

II. LITERATURE REVIEW

The problem of flight delays is not new, there has been much research done on creating accurate and effective models to predict delays. A variety of approaches have been used to achieve this goal. Thiagarajan et al. [8] used historical flight and weather data in order to create a two stage predictive model to predict if a flight would be delayed and the amount of time it would be delayed. Both classification and regression algorithms were used to accomplish this task. A Gradient Boosting Classifier, Random Forest Classifier, Extra-Trees Classifier, and Adaboost Classifier were used for binary classification, while an Extra-Trees Regressor, Random Forest Regressor, Gradient Boosting Regressor, and Multilayer Perceptron were used to predict how delayed a flight would be. Deep Neural Network was also employed in both the classification and regression stages. For both arrival and departure delay prediction the Gradient Boosting Classifiers and the Extra-Trees Regressors performed the best. The Deep Neural Network did not perform as well as the other models for both the binary classification and regression stages, Thiagarajan et al. [8] inferred that this could be due to not having enough data to properly train the model.

Yu et al. [9] also used a deep learning approach to predict flight delays using data from January 2017 and March 2018 of domestic flights from Beijing Capital International Airport, PEK, to Hangzhou International Airport, HGH. For this study a DBN-SVR model, Deep Belief Network-Support Vector Regression, was used as a combination of an unsupervised method in combination with a supervised method to be able to perform regression and classification. The model performed well and when compared to k-Nearest Neighbors, SVM, and LR models, it performed the best out of the four.

Both supervised and unsupervised algorithms were used by Truong [10] in order to predict the risk of a flight in the National Airspace System, NAS, being delayed. The unsupervised algorithm used was the Tabu search method and the supervised algorithm used was the Bayesian Network augmented Naive Bayes, BNAN, classifier. The Tabu search algorithm proved to be useful in uncovering relationships between variables without needing a target variable and the BNAN classifiers predictive power was found to be very good.

Chakrabarty et al. [11] also compared the performance of Gradient Boost Classifier, Support Vector Machine, Random Forest and K-Nearest Neighbors in predicting flight arrival delay. Data from the Bureau of Transport Services (BTS) was used for the year 2015 to 2106. Data for American Airline flights in 5 major flights was analyzed in this study. The area under the Receiver Operating Characteristic (ROC) curve was used as a measure of accuracy. In addition, the F1score, precision and recall were also used to measure performance. Overall, in this case Gradient Boost Classifier was the best performing model.

A Gradient Boosted Decision Tree as a regression model was used by Manna et al. [12] to predict flight delays as well. On-time flight performance data for 70 of the busiest airports in the U.S. from TransStats data from the U.S. Department of Transportation was used for the study. The correlation coefficient between Departure Delay and Arrival Delay was found to be high and the day of the week and airport activity were also found to play a role in whether a flight will be delayed. The model was found to accurately predict flight delay patterns. The model was limited however, as it could only predict delay patterns for airports it was trained with so the model would need to be trained with historical data from an airport before the model could predict delays.

Using the Aviation Systems Performance Metrics data from June through August 2015 and June through August of 2016, ensemble, ordinary least-squared, and penalized machine learning algorithms were compared by Diana [13] based on their ability to predict taxi-out time. After comparing the results, it was found that different models performed better than others based on different

selection criteria. It was also found that even though ensemble learning models can be very sophisticated, traditional regression models can still perform well and in some cases better than these models. Diana (2018) recommended finding a model that gives the best balance between bias and variance.

Balakrishna et al. [14] took a reinforcement learning approach to predicting taxi-out times for flights, using Aviation System Performance Metric data from Tampa International Airport, TPA, from June 1, 2007 to August 25, 2007. A reinforcement learning, RL, estimator was used to predict the taxi-out time and performed well, proving its ability to adjust to flight departures being inherently stochastic and its ability to perform well when predicting taxi out times.

III. DATASET

We have retrieved the dataset used in this analysis from the FAA database [15] which is publicly available. For this purpose, we chose the time period of 3/1/2019 to 5/31/2019 to conduct our analysis on because it was before the COVID-19 restrictions put in place in 2020. The months of March to May are also more representative of a normal travel season, as this period is not considered a high travel season, such as the summer, or a low travel season, such as the winter.

By selecting the period 3/1/2019 - 5/31/2019 for Dates, ORD - Chicago for the Airport, and grouping the result by the fields Airport, Local Hour and Date under the Aviation Performance System Metrics (ASPM) [15], one obtains a table which possesses the features listed in the left column of table (TABLE I), in addition to Hour, Date, and Facility columns.

Feature Name	**Symbol**
% On-Time Gate Departures	$x[1]$
% On-Time Airport Departures	$x[2]$
% On-Time Gate Arrivals	$x[3]$
Average Gate Departure Delay	$x[4]$
Average Taxi Out Time	$x[5]$
Average Taxi Out Delay	$x[6]$
Average Airport Departure Delay	$x[7]$
Average Airborne Delay	$x[8]$
Average Taxi In Delay	$x[9]$
Average Block Delay	$x[10]$
Average Gate Arrival Delay	$x[11]$

TABLE I: The above table represents the symbol used for each variable throughout this paper.

For the ease of notation, we represent the above features by the symbols $x[1]$-$x[11]$ throughout this article. Table (TABLE I) represents the symbol associated with each variable throughout the analysis carried out in this paper. Figure (FIG. 1) displays a sample of five rows of the corresponding data frame. The total number of instances in this data frame amounts to 2207 observations (*i.e.* rows).

In this data frame, $x[1]$, $x[2]$, and $x[3]$ are expressed as percentages while the rest of the features, $x[4]$-$x[11]$, are expressed in minutes. Moreover, each day has been divided into 24 one-hour segments (specified by the Hour column with values $0-23$), and the features are averaged at ORD airport for each one-hour segment of the day.

IV. METHODOLOGY

The aim of the analysis carried out in this section is to identify the main contributing factors to the average gate arrival delay at ORD airport. We employ several supervised machine learning techniques to analyze the average gate arrival delay. Our analysis consists of two key parts. In the first part, the average gate arrival delay, $x[11]$, is declared as the continuous target/response variable of the analysis. After identifying the relevant features (that will be outlined in this section), the main goal in the first part of the analysis will be predicting the amount of the average gate arrival delay at ORD airport in a normal season. As will be outlined in next section (V A), we will take the advantage of linear and polynomial regression techniques to predict the amount of the average gate arrival delay.

In the second part of our analysis, we first transform the target variable (*i.e.* the average gate arrival delay), $x[11]$, to a categorical variable with two classes, namely class [0] corresponding to No Delay in the average gate arrival time, and class [1] corresponding to the Delayed ones. In defining these two classes, we follow the FAA convention [4] in which a gate arrival delay less than 15 minutes is regarded as No Delay and a gate arrival delay of 15 minutes or more is regarded as Delayed for individual flights. As will be described in a greater detail in the next section (V B), we then employ several supervised classification techniques to classify the delay class associated with gate arrivals.

We devote the rest of the current section to an elaboration on the extraction of the relevant features for the analyses that will be carried out in the next section (V).

A. Two Features

In constructing our first model to predict the average gate arrival delay, we merely rely on the definitions of the variables, $x[1] - x[11]$, that are provided by FAA. The FAA definitions [4, 16] of the existing variables in

1	Hour	Date	Facility	ScheduledDepartures	ScheduledArrivals	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]	x[10]	x[11]
865	9	2019-04-08	ORD	69.0	73.0	91.78	86.30	95.77	4.51	17.34	6.07	7.88	1.01	6.75	1.76	1.61
1454	15	2019-05-15	ORD	86.0	47.0	90.59	80.00	89.36	5.81	18.93	7.52	10.85	1.47	6.19	1.68	3.09
1672	18	2019-03-18	ORD	110.0	100.0	87.96	62.04	96.97	4.86	24.46	13.02	15.06	2.05	6.64	2.26	2.74
433	4	2019-05-06	ORD	0.0	5.0	0.00	0.00	60.00	0.00	0.00	0.00	0.00	0.00	1.46	6.20	11.00
1925	20	2019-05-26	ORD	61.0	66.0	90.16	85.25	86.96	2.20	17.90	6.64	6.62	3.88	11.89	3.94	7.28

FIG. 1: The above figure displays a sample of five rows of the data frame used for the analysis in this article.

our retrieved dataset (III) divulge that the block delay and the taxi in delay are directly contributing to the gate arrival delay for *individual flights*. Therefore, upon taking the average, it is expected that the average block delay, $x[10]$, and the average taxi in delay, $x[9]$, to contribute to the average gate arrival delay, $x[11]$. However, it is vital to notice that due to the stochastic nature of the averaging process, other variables will contribute to the average gate arrival delay. In fact, as it will be outlined in section (V A), linear and polynomial regressions in the presence of only $x[9]$ and $x[10]$ as the explanatory variables of the model do not acquire the desired accuracy for predicting the average gate arrival delay. In order to achieve a higher accuracy, one has to inevitably expand the set of explanatory variables that would contribute to the average gate arrival delay at ORD airport.

B. More Features

As highlighted in section (IV A), one has to include more explanatory variables in order to achieve a higher accuracy in predicting the target variable $x[11]$. In this section, we present our reasoning behind including new explanatory variables in our models.

Our strategy to identify the most relevant explanatory variables among $x[1] - x[10]$ is twofold. First, we calculate the correlation matrix based on Pearson's correlation coefficients for a sample of n observations. Given a series of n measurements between variables x_i and x_j (we will use $x[i]$ and x_i interchangeably throughout this paper), the sample correlation coefficient $r_{x_i x_j}$ between x_i and x_j is given by

$$r_{x_i x_j} = \frac{\sum_{\alpha=1}^{n}(x_{i,\alpha} - \bar{x}_i)(x_{j,\alpha} - \bar{x}_j)}{\sqrt{\sum_{\alpha=1}^{n}(x_{i,\alpha} - \bar{x}_i)^2 \cdot \sum_{\alpha=1}^{n}(x_{j,\alpha} - \bar{x}_j)^2}}, \quad (1)$$

where $\bar{x}_i$ and $\bar{x}_j$ represent the arithmetic means of x_i and x_j, respectively. Using (1), it is straightforward to calculate the correlation coefficients among variables $x[1] - x[11]$. Figure (FIG. 2) represents the the correlation matrix of variables $x[1] - x[11]$.

The last row of the above matrix (FIG. 2) represents the

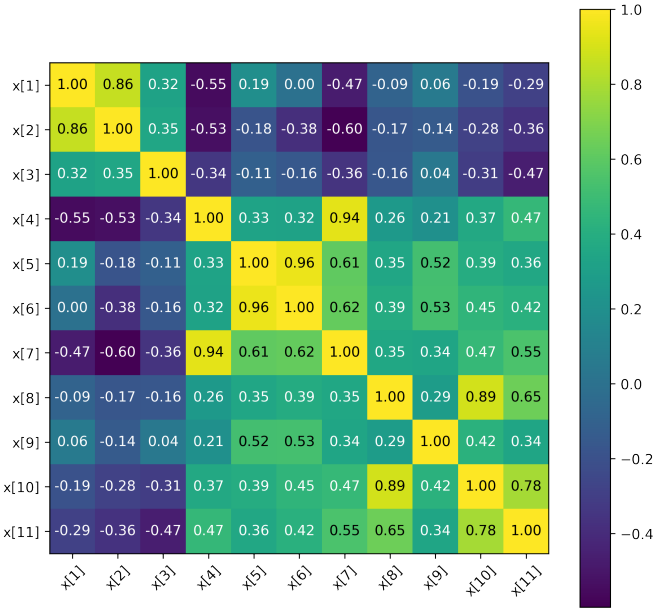

FIG. 2: The above figure displays the correlation matrix associated with the variables used in this analysis.

correlation coefficients of all variables with the average gate arrival delay, $x[11]$. It is evident that the average block delay, $x[10]$, has the highest correlation with the average gate arrival delay, $x[11]$, in accordance with our anticipation based on the FAA definitions as outlined in section (IV A). After the average block delay, $x[10]$, the average airborne delay, $x[8]$, and the average airport departure delay, $x[7]$ have the highest correlation with the average gate arrival delay, $x[11]$. In fact, from the standpoint of Pearson's correlation, the effect of the average taxi in delay on the average gate arrival delay is not very significant. It is also worth noticing that the correlations of $x[1]$ (% On-Time Gate Departures), $x[2]$ (% On-Time Airport Departures), and $x[3]$ (% On-Time Gate Arrivals) with the average gate arrival delay, $x[11]$, are negative. This is of course in accordance with the intuition that the higher the percentages of on-time flight arrivals and departures are, the less average gate arrival delay is expected. Not surprisingly, among $x[1]$, $x[2]$, and $x[3]$, the percentage of on-time gate arrivals, $x[3]$, has the strongest negative correlation with the average gate arrival delay.

Our second approach in identifying the most relevant features for the analysis of the average gate arrival delay, $x[11]$, is based on the notion of Gini impurity. Using a standard ensemble learning technique, we employ a random forest regressor to train 10000 bootstrap aggregated decision trees. Each tree will find a pair (x_i, t_{x_i}) of a single variable x_i ($1 \leq i \leq 10$, for the 10 predictors) with a corresponding threshold t_{x_i} for splitting the train set into two subsets left and right. The goal of each decision tree is to choose (x_i, t_{x_i}) so that the CART cost function

$$J(x_i, t_{x_i}) = \frac{n_\ell}{n} MSE_\ell + \frac{n_r}{n} MSE_r \qquad (2)$$

is minimized. In eq. (2), the mean squared error for each node is defined by

$$MSE_{\text{node}} = \sum_{\alpha \in \text{node}} (x_{11,\alpha} - \hat{x}_{11,\text{node}})^2 \ , \qquad (3)$$

where $\hat{x}_{11,\text{node}}$ is the mean value of the target variable x_{11} at the node in consideration. Moreover, n_ℓ and n_r in eq. (2) refer to the number of instances in the left and right nodes, respectively. The feature importance f_{x_i} for predictor x_i ($1 \leq i \leq 10$) in each tree is then defined by

$$FI_{x_i} = \sum_{j \in \Gamma_i} \frac{n_j}{10} \Delta I_j^i \ , \qquad (4)$$

where n_j denotes the number of instances reaching the node j, and ΔI_j^i is the impurity reduction achieved by the predictor x_i at node j. In eq. (4), Γ_i stands for the set of all nodes that split on the predictor x_i. In the random forest setup, the final feature importance for each predictor must be averaged over all trees.

Using the scikit learn library, we have implemented the above outlined random forest regressor, and extracted the feature importance of the predictors x_i. Figure (FIG. 3) represents the relative feature importance when the average gate arrival delay, x_{11}, is regarded as the target variable.

Comparing the most relevant predictors (for the gate arrival delay) extracted from (FIG. 2) and (FIG. 3), it is evident that there is a large overlap between the sets of most important features extracted from the two different approaches outlined in this section. In particular, it is noticeable that the average block delay, x_{10}, is identified as the most contributing factor to the average gate arrival delay in *both approaches*.

Considering the outcomes of the correlation coefficient analysis (FIG. 2) and of the random forest feature importance (FIG. 3), we will expand the set of predictors in our regression and classification analyses in section (V) in two steps. First, we enlarge the set of predictors from 2 variables (*i.e.* the average block delay, $x[10]$, and the average taxi in delay, $x[9]$) to 4 variables by including the average airborne delay, $x[8]$, and the average airport departure delay, $x[7]$. As it will be outlined in the next section, the performance of our models in the presence of these 4 predictors is improved. To improve the performance even further, we include two more variables,

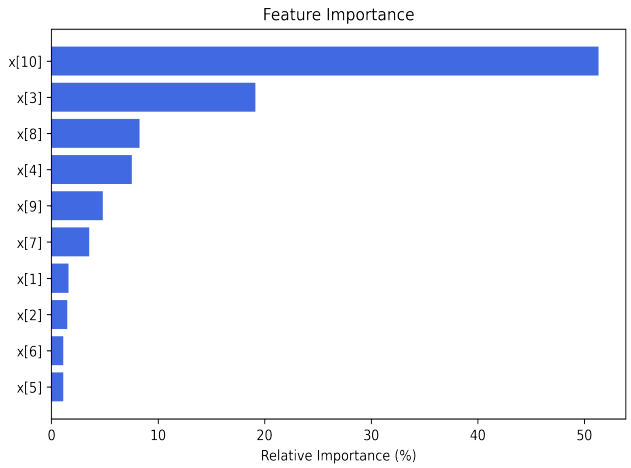

FIG. 3: Using a random forest regressor, and regarding the average gate arrival delay, $x[11]$, as the continuous target variable, the relative importance of features is depicted in above diagram.

namely the % on-time gate arrivals, $x[3]$, and the average gate departure delay, $x[4]$, in the set of predictors in the second step. The following table summarizes the list of the involved predictors in our models in each step.

Setps	**Involved Predictors**
(I): Two features	$\vec{\mathcal{X}}_I = \langle x_{10}, x_9 \rangle$
(II): Four features	$\vec{\mathcal{X}}_{II} = \langle x_{10}, x_9, x_8, x_7 \rangle$
(III): Six features	$\vec{\mathcal{X}}_{III} = \langle x_{10}, x_9, x_8, x_7, x_3, x_4 \rangle$

TABLE II: The above table shows predictors included at each step in the regression and classification models.

It should be pointed out that involving any further predictor beyond what was listed in the above table will not improve the performance of the models in any meaningful manner. This is of course in accordance with the results of (FIG. 2) and (FIG. 3) that, as far as the average gate arrival delay is considered, the effect of the last few predictors on the performance of the models is insignificant.

Another common method for feature selection – when the target variable is continuous – is the Lasso regression (*i.e.* regression with $L1$-type regularization). We have considered this latter method as well, but as it will be outlined in section (V A), this method proves not to be advantageous in the context of the current problem.

C. The Effect of Prior Hours Delays

So far, we have considered the effect of the explanatory variables on the target variable when all variables are measured at the same time. In other words, in sections (IV A) and (IV B), the target variable is explained by the features that are all measured at one specific Hour, namely the Hour at which the target variable is observed. It is however plausible to anticipate that delays in prior hours to partially cause delays in later hours. Therefore, a natural question in this context is to what extent prior hours delays affect the delays in later hours.

To address the effect of prior hours delays on the average gate arrival delay, we must collect the delays among the identified explanatory variables in section (IV B) during the hours prior to the observation point. A quick inspection of the dataset (FIG. 1) reveals that there are no delays greater than 3 hours. Based on this observation and other administrative considerations regarding airport operation procedures, for each delay among the identified explanatory variables in section (IV B), we collect delays one hour prior to the observation time, and two hours prior to the observation time in addition to the delay at current time (*i.e.* the observation time). As will be outlined in section (V A 4), the effect of prior hours delays will lead to an improvement of the accuracy of the regression models.

V. ANALYSIS

Our analysis of the average gate arrival delay consists of two vital parts. First, we treat the average gate arrival delay, $x[11]$, as the continuous target variable of our models and perform several regression analyses to predict the amount of the average gate arrival delay at ORD airport. Second, we convert the average gate arrival delay, $x[11]$, to a categorical target variable, and perform various supervised classification analyses.

A. Regression Problem

We will apply linear and polynomial regression techniques to predict the target variable x_{11} (average gate arrival delay). We will perform the regression tasks by the virtue of the scikit-learn; the standard machine learning library of Python. We will perform regression in four different stages, as was outlined in section (IV).

Before we report the results of our regression analysis, we should mention an important remark regarding the issue of regularization used for regression analyses. In regression analyses, $L1$- and $L2$-regularizations are commonly used to set a better balance between the bias and the variance. In particular, $L1$-regularization can often be used as a feature selection procedure. This is possible when an appropriately chosen $L1$-type penalty would set the coefficients of many predictors to zero. In our analysis in this paper, we have examined both $L1$- and $L2$-regularizations (commonly known by Lasso and Ridge regressions), as well as the ElasticNet (*i.e.* a combination of $L1$- and $L2$-type regularizations). We found that in all three cases (Lasso, Ridge, and ElasticNet), there is no significantly meaningful difference between our results, as the best performance in each case is obtained when the considered penalty is extremely small. This is of course not surprising, as all three types of regression in the presence of a negligible penalty tend to the usual regression (*i.e.* regression with no penalty). The fact that all three regularization types lead the same results implies that we cannot employ the Lasso regression as a feature selection method in this context.

1. (I): Two Features

As outlined in section (IV), our first model consists of only two features, namely the average block delay, x_{10} and the average taxi in delay, x_9. The average gate arrival delay, x_{11}, will be the continuous target variable. In linear regression, a linear relation between the predictors – represented by the two-dimensional vector $\vec{\mathcal{X}}_I$ (see table (TABLE II))– and the target variable is assumed. This linear relation is given by

$$x_{11} = \vec{w}_I \cdot \vec{\mathcal{X}}_I + w_{I0} \ , \tag{5}$$

where $\vec{w}_I = \langle w_1, w_2 \rangle \in \mathbb{R}^2$ is a two-dimensional vector.

For the polynomial regression, we assume that the relation between the predictors and the target variable is governed by a polynomial of degree two. To perform the quadratic regression, we first need to construct all quadratic monomials from the predictors. In general, if there exist N variables (say z_i with $1 \leq i \leq N$), the number of all independent quadratic monomials

$$\{z_i z_j \,|\, 1 \leq i \leq j \leq N\} \tag{6}$$

is given by

$$\#\text{ of quadratic monomials} = \frac{N(N+1)}{2} \ . \tag{7}$$

In the present case, the number of predictors is $N = 2$. Hence, there will be three quadratic monomial, namely

$$\{x_{10}^2 \, , \, x_9^2 \, , \, x_{10}x_9\} \ . \tag{8}$$

We now define the new vector $\vec{\mathcal{X}}_I^{\,Q} \in \mathbb{R}^5$ which includes the quadratic monomials (8) in addition to the linear ones

$$\vec{\mathcal{X}}_I^{\,Q} = \langle x_{10} \, , \, x_9 \, , x_{10}^2 \, , \, x_9^2 \, , \, x_{10}x_9 \rangle \ . \tag{9}$$

To perform the quadratic regression, a quadratic relation between the predictors and the target variable is assumed. The latter relation is explicitly given by

$$x_{11} = \vec{w}_I^{\,Q} \cdot \vec{\mathcal{X}}_I^{\,Q} + w_{I0}^{Q} \ , \tag{10}$$

where $\vec{w}_I^Q \in \mathbb{R}^5$ and constant w_{I0}^Q are to be determined.

In liner and polynomial regressions, the goal is to find the constant vector $\vec{w}_I$ and the constant w_{I0}, known as the intercept, in the linear case and constant vector $\vec{w}_I^Q$ and the constant w_{I0}^Q in the quadratic case. This is done by minimizing the mean squared error function. The mean squared error function, MSE, for a sample of n instances is given by

$$MSE = \frac{1}{n}\sum_{\alpha=1}^{n}(x_{11,\alpha} - \hat{x}_{11,\alpha})^2 \ , \tag{11}$$

where $\hat{x}_{11,\alpha}$ is the predicted value for the target variable obtained from eq. (5).

Splitting the data into train and test subsets (with the train subset having 0.7 of the entire number of instances in the dataset), the best fit by linear regression acquires the following scores.

Regression	Subset	Accuracy Score
linear	train	0.5606
	test	0.6739
quadratic	train	0.6230
	test	0.6741

TABLE III: This table summarizes the accuracies of the regression models when two predictors are considered.

The vector $\vec{w}_I$ and the constant w_0 are found to be

$$\vec{w}_I = \langle 1.9506\,, 0.0839\rangle \ , \quad w_{I0} = 1.5908 \ . \tag{12}$$

In the above equation, one must note that while vector $\vec{w}$ is dimensionless, the constant w_0 carries the minute unit. As it is observed, the accuracy of the model is not at a satisfactory level, and we desire to have a better performance. One reason for expanding the set of predictors is in fact to achieve a higher accuracy.

2. (II): Four Features

In this part of the analysis, we include two more predictors in accordance to our logic in section (IV). As indicated in table (TABLE II), the set of predictors $\vec{\mathcal{X}}_{II} = \langle x_{10}, x_9, x_8, x_7\rangle$ forms a vector in $\mathbb{R}^4$. We perform a linear regression as well as a polynomial regression, and compare the results. For the linear regression, a linear relation of the form

$$x_{11} = \vec{w}_{II} \cdot \vec{\mathcal{X}}_{II} + w_{II0} \ , \tag{13}$$

between the target variable and the predictors is assumed. In this case, $\vec{w}_{II} = \langle w_1, w_2, w_3, w_4\rangle$ is a vector in $\mathbb{R}^4$.

In the current situation, we have four predictors ($N = 4$). Hence, according to (7), there will be 10 quadratic monomials to be considered in the polynomial regression. For this purpose, we introduce a new vector $\vec{\mathcal{X}}_{II}^Q$ where in addition to the 4 predictors of $\vec{\mathcal{X}}_{II}$, we have included the 10 quadratic monomial constructed from the 4 predictors of $\vec{\mathcal{X}}_{II}$. The relationship between the target variable, x_{11}, and the predictors is given by the following quadratic equation

$$x_{11} = \vec{w}_{II}^Q \cdot \vec{\mathcal{X}}_{II}^Q + w_{II0}^Q \ , \tag{14}$$

in which $\vec{w}_{II}^Q$ and $\vec{\mathcal{X}}_{II}^Q$ are vectors in $\mathbb{R}^{14}$, and w_{II0}^Q is the intercept constant.

As section (V A 1), the goal is to find the best fit for $\vec{w}_{II}$, w_0, $\vec{w}_{II}^Q$, and w_{II0}^Q by minimizing the mean squared error function (11). Splitting the data to train and test subsets and performing the regression analysis parallel to section (V A 1), the results are summarized in the following table.

Regression	Subset	Accuracy Score
linear	train	0.6239
	test	0.7084
quadratic	train	0.6908
	test	0.6293

TABLE IV: This table summarizes the accuracies of the regression models when four predictors are considered.

Comparing the accuracy scores between the two-feature and the four-feature cases summarized in tables (TABLE III) and (TABLE IV), it is clear that the accuracy of the the train subset has raised from 0.56 to 0.62. Moreover, the quadratic regression, raises the accuracy score to 0.69 which is an extra desirable improvement on top of the linear regression. We avoid an explicit presentation of the regression coefficients $\vec{w}_{II}$, $\vec{w}_{II}^Q$ and the intercepts w_{II0}, w_{II0}^Q, as they are not particularly illuminating.

3. (III): Six Features

Parallel to the treatment in the previous subsections (V A 1) and (V A 2), we briefly report the results for part III when six predictors are included. In this case, the set of predictors are described by the vector $\vec{\mathcal{X}}_{III} = \langle x_{10}, x_9, x_8, x_7, x_3, x_4\rangle$ in $\mathbb{R}^6$, in accordance with table (TABLE II). In case of the linear regression, the target

variable, x_{11} and the predictors are related as follows

$$x_{11} = \vec{w}_{III} \cdot \vec{\mathcal{X}}_{III} + w_{III0} \ , \tag{15}$$

where $\vec{w}_{III}$ is a constant vector in $\mathbb{R}^6$ and w_{III0} is the intercept.

Using eq. (7), we have to construct 21 quadratic monomials to perform the quadratic regression. Let $\vec{\mathcal{X}}^{Q}_{III} \in \mathbb{R}^{27}$ denote the set of 6 predictors together with their 21 corresponding quadratic monomials. Then, in the case of quadratic regression, the following relation between the target variable and the predictors is assumed.

$$x_{11} = \vec{w}^{Q}_{III} \cdot \vec{\mathcal{X}}^{Q}_{III} + w^{Q}_{III0} \ . \tag{16}$$

The following table summarizes the accuracies acquired by the linear and quadratic regression model in the presence of the 6 predictors represented by $\vec{\mathcal{X}}_{III}$.

Regression	**Subset**	**Accuracy Score**
linear	train	0.6603
	test	0.7329
quadratic	train	0.8262
	test	0.7756

TABLE V: This table summarizes the accuracies of the regression models when six predictors are considered.

Comparing the above accuracies in table (TABLE V) with the corresponding ones summarized in table (TABLE IV), it is evident that the performance of the model in both linear and quadratic situations is ameliorated. However, as we discussed in section (IV), the performance of the model will not improve further by considering more predictors – measured at the same time as the target variable – because the effect of the rest of the variables is insignificant. The only possibility for further improvements may come from the effect of prior hours delays that will be explored in the next subsection.

4. (IV): Effect of Prior Hours Delays

In this section, we will explore the effect of the prior hours delays on the average gate arrival delay at ORD. An analysis of prior hours delays in general settings will be beyond the scope of this paper. In our treatment in here, we will regard the prior hours delays as new predictors involved in the regression analysis. This approach would make sense in the context of the analysis followed in this paper, as there are no delays in our dataset that go beyond 3 hours. In fact, average delays beyond one hour are sparse in the dataset considered in this analysis. It is therefore justified to consider for each component of $\vec{\mathcal{X}}_{III}$ – except for x_3 which is not a delay – delays one hour and two hours prior to the observation hour. This would increase the number of predictors from 6 to 16. After dropping the zero columns (there are 3 of them), we find a new set of predictors represented by vector $\vec{\mathcal{X}}_{IV} \in \mathbb{R}^{13}$.

To conduct the linear regression algorithm, a linear relation between the new set of predictors and the target variable is assumed.

$$x_{11} = \vec{w}_{IV} \cdot \vec{\mathcal{X}}_{IV} + w_{IV0} \ . \tag{17}$$

In eq. (17), $\vec{w}_{IV} \in \mathbb{R}^{13}$ represents the regression coefficients, and w_{IV0} is the intercept.

Using eq. (7), there will be 91 quadratic monomials to consider for the quadratic regression. Including the quadratic monomials into the set of predictors, we find the new vector $\vec{\mathcal{X}}^{Q}_{IV} \in \mathbb{R}^{104}$. To conduct the quadratic regression, the following relation between the predictors and the target variable is assumed

$$x_{11} = \vec{w}^{Q}_{IV} \cdot \vec{\mathcal{X}}^{Q}_{IV} + w^{Q}_{IV0} \ , \tag{18}$$

in which $\vec{w}^{Q}_{IV} \in \mathbb{R}^{104}$ and $w^{Q}_{IV0} \in \mathbb{R}$. The following table summarizes the accuracies of the linear and quadratic regressions when the effect of prior hours delays is taken into account.

Regression	**Subset**	**Accuracy Score**
linear	train	0.7222
	test	0.6607
quadratic	train	0.8676
	test	0.7245

TABLE VI: This table summarizes the accuracies of the regression models when the effect of prior hour delays is included.

Comparing the table (TABLE VI) with table (TABLE V), it is evident that the performance of the model in both linear and quadratic regressions has improved. We declare the accuracies achieved by the quadratic regression in this section as our best model reported in this paper.

Figure (FIG. 4) summarizes the accuracy of linear and quadratics regression models for the train subset.

B. Classification Problem

In this section, we carry out various supervised classification analyses and briefly report our results. As outlined

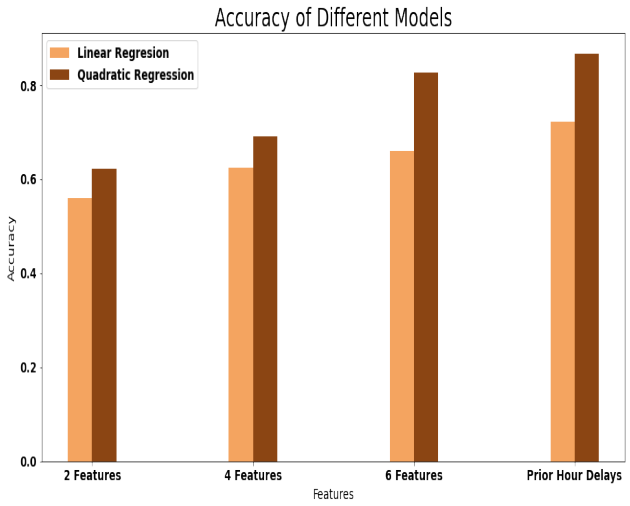

FIG. 4: In the above figure, the accuracy of linear and quadratic regression models are depicted for the train subset.

in section (IV), we convert the target variable, x_{11}, into a binary categorical variable, y, with classes [0] and [1] corresponding to No Delay and Delayed, respectively. Our treatment in this section is in parallel with the analysis carried out in section (V A) where we increased the number of predictors at each step. As will be seen, the best performances are obtained at step (IV) when the predictors are given by $\vec{\mathcal{X}}_{IV}$ (see table (TABLE (II)) for the definition of $\vec{\mathcal{X}}_{IV}$). We proceed in this section as follows. We perform logistic regression and support vector machines for the classification of the binary categorical target variable, y in steps (I)-(III). In step (IV), we perform random forest classification and the gradient boosting in addition to the logistic regression and support vector machines. At the end, we will take the advantage of an ensemble learning technique, and will perform stacking of the 4 previous classification models applied in step (IV).

To ease the notation and in order to avoid repetition, we define set $\mathcal{S}$ to be

$$\mathcal{S} = \{I,\, II,\, III,\, IV\}\ , \tag{19}$$

where elements of set $\mathcal{S}$ refer to the four different steps introduced in table (TABLE II) and used in section (V A).

We employ logistic regression as our first supervised classification algorithm used to analyze the binary target variable y. Logistic regression is a prominent classification algorithm which is widely used, especially when the categorical target variable is binary. In logistic regression, the probability of observing the (i)-th instance with features $\vec{\mathcal{X}}_\Lambda^{(i)}$ in the class $y^{(i)}$ (note that $y^{(i)}$ is either 0 or 1) is given by

$$P\big(y^{(i)}\,|\,\vec{\mathcal{X}}_\Lambda^{(i)}\big) = \phi(z_\Lambda^{(i)})^{y^{(i)}}\big(1-\phi(z_\Lambda^{(i)})\big)^{1-y^{(i)}}\ , \tag{20}$$

where $\phi : \mathbb{R} \to (0,1)$ is the sigmoid function and is given by

$$\phi(z) = \frac{1}{1+e^{-z}}\ . \tag{21}$$

In eq. (20), $\Lambda \in \mathcal{S}$, and $z_\Lambda^{(i)}$ is given by

$$z_\Lambda^{(i)} = \vec{v}_\Lambda \cdot \vec{\mathcal{X}}_\Lambda^{(i)} + v_{\Lambda 0}\ , \tag{22}$$

where $\vec{v}_\Lambda$ is a constant vector and $v_{\Lambda 0}$ is a constant.

In logistic regression, it is assumed that the n instances are independent, and hence, the total probability function is given by

$$\mathcal{P}_\Lambda = \prod_{i=1}^{n} P\big(y^{(i)}\,|\,\vec{\mathcal{X}}_\Lambda^{(i)}\big)\ . \tag{23}$$

The goal in logistic regression is to find $\vec{v}_\Lambda$ and $v_{\Lambda 0}$ by maximizing the total probability function $\mathcal{P}_\Lambda$ (or equivalently, by minimizing the cost function which is defined as minus the logarithm of $\mathcal{P}_\Lambda$).

We have implemented logistic regression in all 4 steps and have summarized the results in table (TABLE VII). As is observed in this table, the logistic regression models perform with a high level of accuracy in all 4 steps (Even with only 2 features (step (I)), the accuracy of the model is 0.92). Looking at $F1$-scores achieved by logistic regression models, it is evident from table (TABLE VII) that $F1$-scores for class [0] are all above 0.95. However, for class [1], the $F1$-scores are comparatively poorer in all 4 steps.

As our next supervised classification algorithm, we employ the support vector machines. The idea behind the support vector classifier (SVC) is straightforward. The model predicts class $\hat{y}^{(i)}$ for the (i)-th instance according to the following rule

$$\hat{y}^{(i)} = \begin{cases} 0\ , & \text{if } z_\Lambda^{(i)} < 0\ , \\ \\ 1\ , & \text{if } z_\Lambda^{(i)} \geq 0\ , \end{cases} \tag{24}$$

where $z_\Lambda^{(i)}$ is given by eq. (22). Unlike logistic regression, SVC directly assigns a class to each instance without calculating any probability for the event. In other words, SVC separates the n observations by a hyperplane into two classes (assuming the instances are linearly separable).

The goal of SVC is to find $\vec{v}_\Lambda$ and $v_{\Lambda 0}$ by minimizing the following cost function

$$J(\vec{v}_\Lambda, v_{\Lambda 0}, \zeta) = \frac{1}{2}\vec{v}_\Lambda \cdot \vec{v}_\Lambda + C\sum_{i=1}^{n}\zeta^{(i)} \tag{25}$$

which is subject to the following constrains

$$\begin{cases} \lambda^{(i)}\big(\vec{v}_\Lambda \cdot \vec{\mathcal{X}}_\Lambda^{(i)} + v_{\Lambda 0}\big) \geq 1 - \zeta^{(i)}\ , \\ \\ \zeta^{(i)} \geq 0\ , \end{cases} \qquad \text{for } i = 1, \cdots, n\ .$$

In eq. (25), $\zeta^{(i)}$ is the so called the (i)-th slack variable which indicates how much the (i)-th instance is allowed to violate the margin. Also, C in eq. (25) is a hyperparameter which allows to define trade-off between the two competing terms in eq. (25). Finally, $\lambda^{(i)}$ is a constant (either $+1$ or -1 depending on the predicted class $\hat{y}^{(i)}$ of the (i)-th instance).

Support vector machines (SVM) – a more general variant of SVC that allows to encounter non-linear terms to define the decision regions (instead of simple hyperplanes defined by eq. (24)) – are also employed in our analysis to enhance the performance of the models.

In each step given by an element of $\mathcal{S}$, we have examined several different SVM models, and have chosen the one with the best accuracy as the representative of the SVM calculation for that step. Table (TABLE VII) summarizes the results obtained from implementing SVM through scikit-learn library of Python. As is observed from table (TABLE VII), in each step, the results obtained from SVM models are very similar to of those obtained from the logistic regression models. In fact, the two models perform very similar at each stage, and no significant distinction between their results is observed.

For the rest of this section, we employ some of the prominent ensemble learning techniques in order to fortify our classification models (at stage IV) even further.

Random forest classifier is a common and effective bagging method of an ensemble of decision trees. We employ a random forest classifier to train 2000 bootstrap aggregated decision trees. The details of the results of the applied random forest model has been summarized in table (TABLE VII). The performance of the random forest model is closely similar to those of the logistic regression and the SVM. However, the random forest classifier allows to measure the importance of the features. Notice that at stage (IV), we have included the effect of prior hours delays in our model. Therefore, one natural question in this context is that among all earlier hours delays which ones are more significant. This question can be answered by calculating the feature importance of the predictors using the notion of Gini impurity. The result of the feature importance calculation has been presented in figure (FIG. 5). First, we note that in figure (FIG. 5), $x[i][-1]$ and $x[i][-2]$ refer to feature $x[i]$ measured one hour and two hours prior to the observation time, respectively. There are three comments in order. First, it is interesting to observe that for the classification models, the role of the percentage of on-time gate arrivals, $x[3]$, is even more significant than the role of the average block delay, $x[10]$ (compare with figure (FIG. 3)). Second, it is evident from figure (FIG. 5) that the features measured at the observation time all have a greater importance than of those measured in prior hours. Third, among the features measured at prior hours, the average airport departure delay measured one hour prior to the observation time, $x[7][-1]$, is the dominant feature among all features measured at prior times. After $x[7][-1]$, the average gate departure delay measured one hour prior to the observation time, $x[4][-1]$ is the most relevant feature measured at prior times. As is clear from figure (FIG. 5), one may forget about the rest of the features measured at prior hours as their role in classification of the class of the target variable is insignificant.

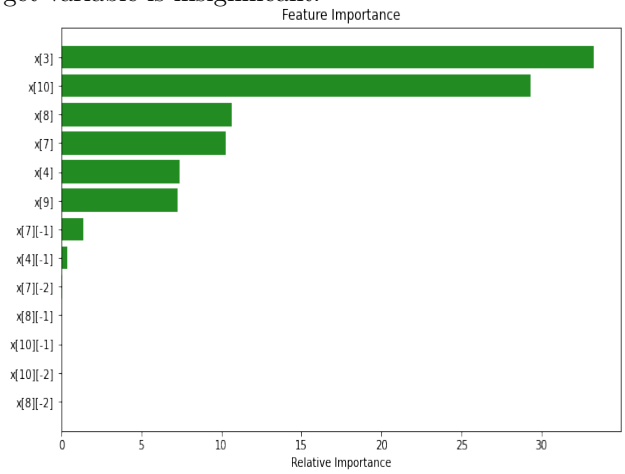

FIG. 5: In above figure, the relative feature importance of the predictors involved in $\vec{\mathcal{X}}_{IV}$ is depicted.

The next ensemble learning method that we will use to fortify the performance of our classification model is the boosting technique. More specifically, we employ the gradient boosting algorithm that sequentially adds predictors to an ensemble. Each one improves its predecessor and transforms weak learners to stronger ones. In the current situation, the gradient boosting trains 5000 decision trees each with maximum depth 3. The results of the gradient boosting model has been summarized in table (TABLE (VII)). Let us highlight a couple of interesting facts about the results of the gradient boosting. First, as is observed from table (TABLE VII), gradient boosting improves the accuracy of the model even further, compared to logistic regression and SVM models. Second, gradient boosting considerably improves the $F1$-score of the model on the class [1]. This is particularly important ,as all previous models were incapable of achieving such high $F1$-score for class [1].

In our last analysis in this section, we employ another ensemble learning algorithm, namely the stacking method. Stacking uses the prediction of the previous models (in this case the logistic regression, SVM, random forest classifier, and the gradient boosting) as inputs for a second layer learning algorithm. The second layer optimally combines the predictions of the previous models to form a new set of predictions. We have implemented the stacking algorithm through the scikit-learn library of Python, and the last row of table (TABLE VII) represents the obtained results. As is observed in table (TABLE VII), the stacking model slightly improves on the accuracy of the model. Therefore, we declare the stacking model as our best classification model in the analysis carried out in this section.

Figure (FIG. 6) represents the confusion matrix of the

Features	Model	Accuracy Score	Class	Precision	Recall	F1-Score
(I): two features	logistic regression	0.9216	[0]	0.93	0.98	0.96
			[1]	0.77	0.45	0.57
	SVC	0.9186	[0]	0.94	0.97	0.95
			[1]	0.71	0.49	0.58
(II): four features	logistic regression	0.9231	[0]	0.94	0.98	0.96
			[1]	0.75	0.50	0.60
	linear kernel SVM	0.9231	[0]	0.93	0.98	0.96
			[1]	0.79	0.45	0.57
(III): six features	logistic regression	0.9306	[0]	0.95	0.97	0.96
			[1]	0.75	0.59	0.66
	polynomial (degree=2) kernel SVM	0.9367	[0]	0.95	0.98	0.96
			[1]	0.81	0.58	0.68
(IV): prior hours delays	logistic regression	0.9396	[0]	0.95	0.99	0.97
			[1]	0.88	0.60	0.71
	polynomial (degree=2) kernel SVM	0.9381	[0]	0.94	0.99	0.97
			[1]	0.92	0.55	0.69
	random forest classifier	0.9396	[0]	0.94	1.00	0.97
			[1]	0.96	0.54	0.69
	gradient boosting classifier	0.9547	[0]	0.96	0.99	0.97
			[1]	0.90	0.72	0.80
	stacking the above 4 models	0.9577	[0]	0.96	1.00	0.98
			[1]	0.97	0.69	0.80

TABLE VII: The above table summarizes the results of various supervised classification algorithms applied to predict the average gate arrival delay class at ORD airport.

stacking model for a test subset of size 662 instances.

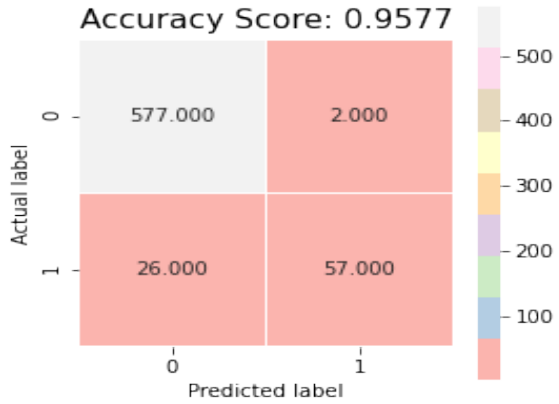

FIG. 6: The above figure represents the confusion matrix of the stacking model.

We will close this section by presenting the Receiver Operating Characteristic (ROC) curve of the stacking classifier. Figure (FIG. 7) plots the sensitivity (*i.e.* recall or the true positive rate) of the stacking model versus its specificity (*i.e.* false positive rate). As a measure to evaluate the performance of the classifier, one calculates the area under the curve (AUC) from the ROC plot. The perfect classification is associated with AUC = 1. As indicated in figure (FIG. 7), the AUC of the stacking classifier happens to be 0.96. This provides another verification of the fact that our final classifier (the stacking model) is close to a perfect classifier.

VI. CONCLUSIONS

In this article, we have shown how standard supervised machine learning algorithms can effectively be employed to analyze the average delays in airport analyses. Although our analysis was focused on the average gate arrival delay, the methodology presented in this paper can be applied to analyze other types of average delays used in airport metrics. The dataset used for the analysis was

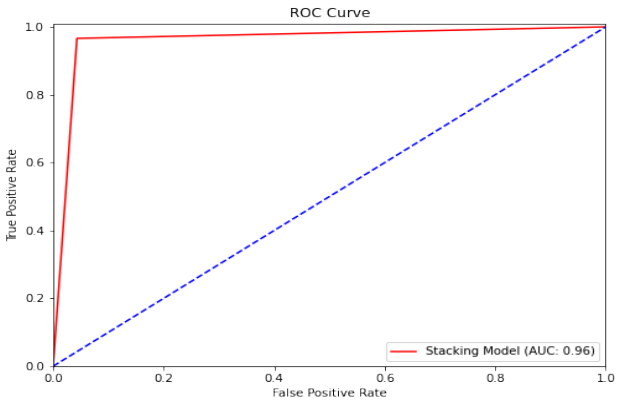

FIG. 7: In the above figure, the Receiver Operating Characteristic (ROC) curve of the stacking model has been plotted.

retrieved from the FAA database [15]. In this dataset, all variables are *averaged* variables commonly used to quantify the performance of an airport. One way to predict these average metrics is to carefully analyze all individual incoming and outgoing flights at the airport in consideration and take the average of individual metrics to obtain the average metrics. Among many other subtleties, this approach requires an immense amount of data to process. An alternative approach to the problem is to understand the average metrics by the average metrics. In other words, one explores relations and correlations among the average metrics and aims to predict a small subset of the average metrics by the knowledge of the other relevant average metrics. The latter approach is patently less costly and can be implemented in an accelerated manner. In this paper, we have taken this latter approach to analyze the average gate arrival delay at ORD airport for the period 3/1/2019 - 5/31/2019.

In order to identify the most relevant contributing factors to the average gate arrival delay, we have used two complementary approaches in this paper. First, we used the notion of Pearson's correlation to compute the correlation matrix between different (averaged) variables involved in this analysis. Our second approach to extract the most relevant contributing variables to the average gate arrival delay is based on the feature importance obtained by using a random forest regressor. We have observed that there exists a large overlap between the most relevant features extracted from the two applied methods. In both approaches, the average block delay was identified as most relevant predictor for the average gate arrival delay. As an interesting side remark, it is worth noticing that, in the feature importance obtained from the random forest regressor, the percentage of on-time gate arrivals has been recognized as the second most relevant predictor for the average gate arrival delay. A low percentage of on-time gate arrivals among airport performance metrics is often associated with an inclement weather. An inclement weather impacts gate arrivals in a negative manner – as reflected by the negative correlation between the percentage of on-time gate arrivals and the average gate arrival delays in the correlation matrix – by causing delays. This is an interesting observation since the dataset used in this paper does not include any variable which directly reflects the weather condition under which the ORD airport has been operating. Nonetheless, the percentage of the on-time gate arrivals strongly depends on weather conditions, and hence, the effect of weather conditions has been implicitly considered in the analysis carried out in this paper[1].

Upon identifying the most relevant predictors, we performed a regression analysis to predict the average gate arrival delay at ORD airport. The regression analysis has been carried out at several stages by including more predictors at each step. In the first step where only two predictors were included, the accuracies of the linear and (quadratic) polynomial regression models were not at the desired level, as many relevant variables were left out. To improve the performance of the regression models, we included first four and then six predictors in our analysis. While the accuracy of the regression models has amended, we realized that inclusion of further predictors does not improve the performance of the model, as the effect of the remaining predictors on the average gate arrival delay is insignificant. In our last attempt to fortify the performance of the regression models, we considered the effect of the prior hours delays on the average gate arrival delay. While a systematic analysis of the prior hours delays is beyond the scope of this paper, we incorporated the delays prior to the observation time in our analysis in a minimal but justified manner. For each of the predictors considered in our regression models, we included the delays one hour and two hours prior to the hour of observation as well. In this context, the best performance for predicting the average gate arrival delay was obtained from the quadratic regression model in the presence of six predictors with the effect of the prior hours delays included. It would be interesting to observe if an even stronger accuracy for predicting the average gate arrival delay can be achieved by applying the deep learning methods [17].

In the last part of the analysis carried out in this paper, we applied several paramount supervised classification algorithms to predict the class associated with the average gate arrival delay. An inspection of the distribution plot of the average gate arrival delay at ORD airport reveals that the majority of delays are less than 30 minutes. In fact, average gate arrival delays greater than 30 minutes are sparse. Hence, it would be most suitable to define only two classes for the average gate arrival delay, namely class [0] corresponding to No Delay when the average gate arrival delay is less than 15 minutes and class [1] corresponding to Delayed when the average gate arrival delay is 15 minutes or greater. We have employed the logistic regression, support vector machines, the random forest classifier, and the gradient boosting algorithms to address this binary classification problem.

[1] We thank Dr. Tony Diana for drawing our attention to this fact.

For the latest model where the effect of prior hours delays is included, we have fortified the model's performance by applying an ensemble learning technique, namely the stacking method. Although the ensemble learning techniques do improve the results of the classification models slightly, we have realized that the performance of any of the individual applied algorithms is already very strong.

The proposed methodology in this paper can be applied to similar situations where predictions and classifications of averaged metrics are concerned. First, it would be interesting to conduct similar analyses of other types of average airport delays. The average gate arrival delay is expected to possess the least stochastic nature among the average delays considered in this paper. It is anticipated that applying our methodology to variables such as the average gate departure delay or the average airport departure delay would offer less accuracy due to the more stochastic nature of these variables. Second, it is desirable to carry out similar analyses for other periods than what was considered in this study. We anticipate that the models to perform with less accuracy for the periods that are considered as high season of travels for the airline industry.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Tony Diana for sharing his expertise throughout this project, and for his comments on this manuscript.

[1] Bureau of Transportation Statistics, "*On-Time Performance - Reporting Operating Carrier Flight Delays at a Glance*", Retrieved from https://www.transtats.bts.gov/HomeDrillChart.asp, (2021).

[2] Federal Aviation Administration, "*Cost of Delay Estimates 2019*". Retrieved from https://www.faa.gov/data_research/aviation_data_statistics/media/cost_delay_estimates.pdf, (2020, July 8)

[3] Kim, N., and Park, J. "*A study on the impact of airline service delays on emotional reactions and customer behavior*" Journal of Air Transport Management, 57, 19-25, (2016).

[4] Federal Aviation Administration, "*ASPM Airport Analysis: Definitions of Variables*", https://aspmhelp.faa.gov/index/ASPM_Airport_Analysis__Definitions_of_Variables.html.

[5] Hetter, K. (2019, September 16). "*This is the world's busiest airport.*" CNN. Retrieved March 28, 2021, from https://www.cnn.com/travel/article/worlds-busiest-airports-2018/index.html

[6] Chicago Department of Aviation.(n.d.). "*OHare History.*" Retrieved March 02, 2021 from https://www.flychicago.com/business/CDA/Pages/OHare.aspx

[7] OAG, "*On-time Performance for Airlines and Airports and Top 20 Busiest Routes*," (2018) https://www.oag.com/hubfs/Free_Reports/Punctuality_League/2018/PunctualityReport2018.pdf

[8] Thiagarajan, B., Srinivasan, L., Sharma, A. V., Sreekanthan, D., and Vijayaraghavan, V. "*A machine learning approach for prediction of on-time performance of flights.*" IEEE Xplore, (2017). doi:10.1109/DASC.2017.8102138 https://ieeexplore.ieee.org/document/8102138

[9] Yu, B., Guo, Z., Asian, S., Wang, H., and Chen, G. "*Flight delay prediction for commercial air transport: A deep learning approach.*" Transportation Research Part E: Logistics and Transportation Review, 125, 203-221, (2019). doi:10.1016/j.tre.2019.03.013 https://www.sciencedirect.com/science/article/pii/S1366554518311979?via%3Dihub

[10] Truong, D. "*Using causal machine learning for predicting the risk of flight delays in air transportation.*" Journal of Air Transport Management, 91, (2021). doi:10.1016/j.jairtraman.2020.101993 https://www.sciencedirect.com/science/article/pii/S0969699720305755?via%3Dihub

[11] Chakrabarty N., Kundu T., Dandapat S., Sarkar A., Kole D.K. "*Flight Arrival Delay Prediction Using Gradient Boosting Classifier.*" In: Abraham A., Dutta P., Mandal J., Bhattacharya A., Dutta S. (eds) Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, vol 813, pp 651-659. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1498-8_57

[12] Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P., and Barman, S. "*A statistical approach to predict flight delay using gradient boosted decision tree.*" IEEE Xplore, (2018). doi:10.1109/ICCIDS.2017.8272656 https://ieeexplore.ieee.org/document/8272656

[13] Diana, T. "*Can machines learn how to forecast taxi-out time? A comparison of predictive models applied to the case of Seattle/Tacoma International Airport.*" Transportation Research Part E: Logistics and Transportation Review, 119, 149-164, (2018). doi:10.1016/j.tre.2018.10.003 https://www.sciencedirect.com/science/article/pii/S136655451830543X

[14] Balakrishna, P., Ganesan, R., and Sherry, L. "*Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures.*" Transportation Research Part C: Emerging Technologies, 18(6), 950-962, (2010). doi:10.1016/j.trc.2010.03.003 https://www.sciencedirect.com/science/article/pii/S0968090X1000029X

[15] FAA Operations & Performance Data, Aviation System Performance Metrics (ASPM), https://aspm.faa.gov/Default.asp

[16] Federal Aviation Administration, "*ASPM: Individual Flights: Definitions of Variables*", https://aspmhelp.faa.gov/index/ASPM__Individual_Flights__Definitions_of_Variables.html.

[17] *Work in progress.*