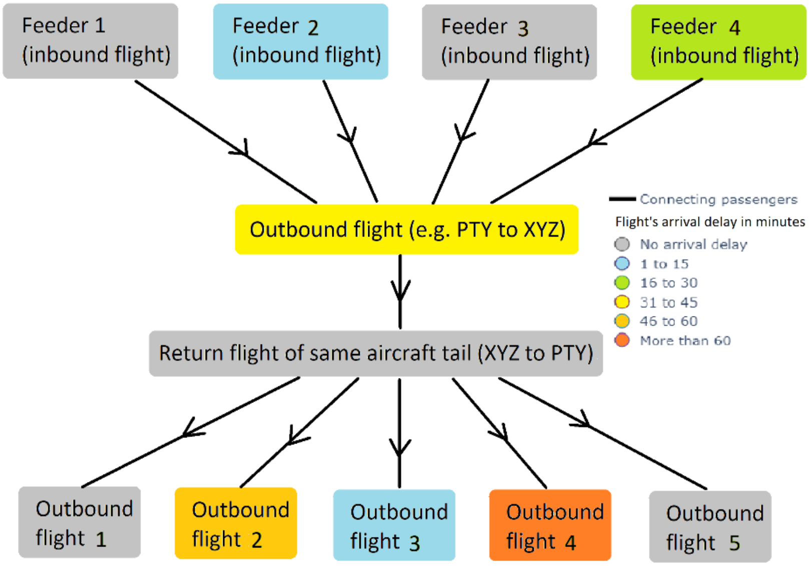
Machine Learning for Flight Delay Forecasting

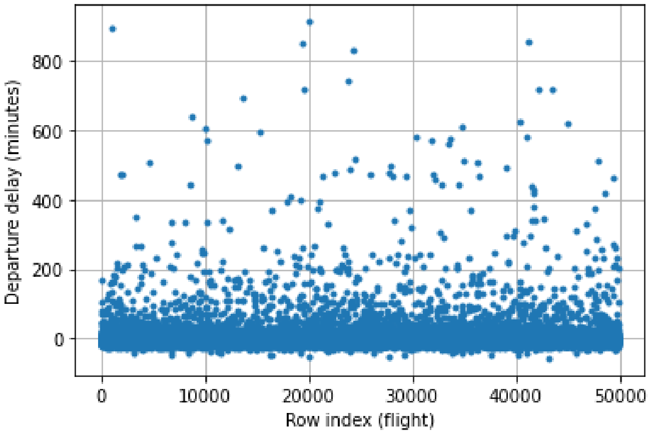
Sarthak Sharma, Abbey Centers, Michael Styron, Amanda Lovett.
CAP 5771 Data Mining, Graduate Term Project Report (3 December 2021).

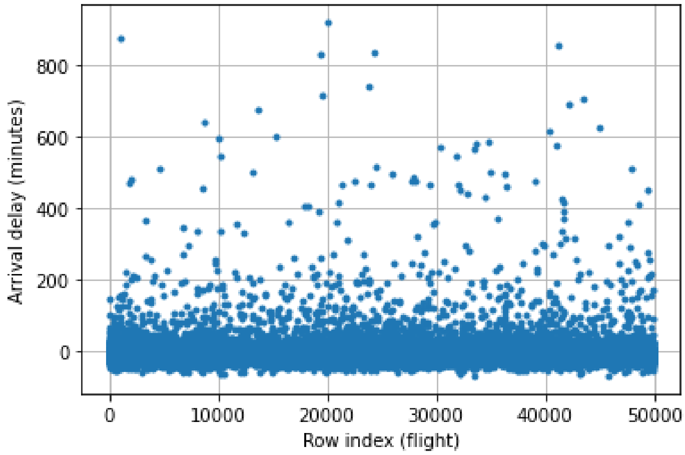
I. Problem introduction and potential applications

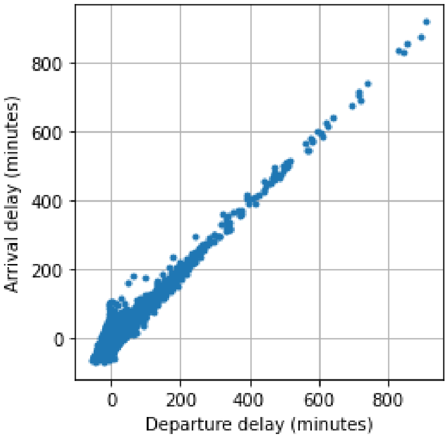
Flight delay prediction is an extremely important and popular research topic in data mining. Over the course of many years of research, sophisticated machine learning algorithms have successfully applied classification, regression, and neural network models to predict information critical to the economic growth of the airlines industry. Many airlines use the resources of artificial intelligence and machine learning to improve customer experience, reduce costs, and make proactive scheduling decisions in response to predicted delays. Such airlines include Delta, Southwest Airlines, and Air France. Certain airlines, however, do not take advantage of machine learning algorithms and the many applications they have in flight delay prediction. Copa Airlines, the national airline of the Republic of Panama, does not use machine learning for flight delay prediction and performs all flight scheduling manually. Therefore, the goal of our research was to use machine learning to forecast flight delays, improve flight planning / scheduling, improve passenger experience, and increase profits for Copa Airlines. This project was a continuation of the research performed by Sarthak Sharma while under the joint supervision of Copa Airlines and Florida State University for the Spring and Summer 2021 terms.


Feeder 1 (inbound flight)
Feeder 2 (inbound flight)
Feeder 3 (inbound flight)
Feeder 4 (inbound flight)
Outbound flight (e.g. PTY to XYZ)
Return flight of same aircraft tail (XYZ to PTY)
Outbound flight 1
Outbound flight 2
Outbound flight 3
Outbound flight 4
Outbound flight 5
Connecting passengers
Flight's arrival delay in minutes
● No arrival delay
● 1 to 15
● 16 to 30
● 31 to 45
● 46 to 60
● More than 60

The hub airport for Copa Airlines is the Panama City (PTY) airport, which has the advantageous location between North America and South America. Almost all Copa Airlines flights involve PTY airport as either origin or destination. Prediction of flights' arrival and departure delays may benefit Copa Airlines in many ways. Let there be a selection of flights inbound to PTY airport. At PTY airport, some passengers may transfer from these inbound flights to an outbound Copa Airlines flight. If 1 or more of these inbound flights has an arrival delay, then Copa Airlines staff decide (as of now, manually) whether to postpone the departure of that shared outbound flight. Conflicting interests must be weighed. Flight delays may propagate to connected flights and potentially affect a larger number of passengers later. In some cases, it may be better to delay the outbound flight so that connecting passengers of delayed feeder flights do

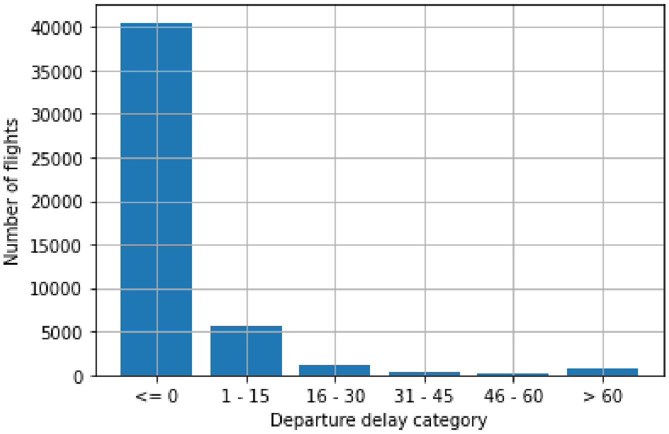


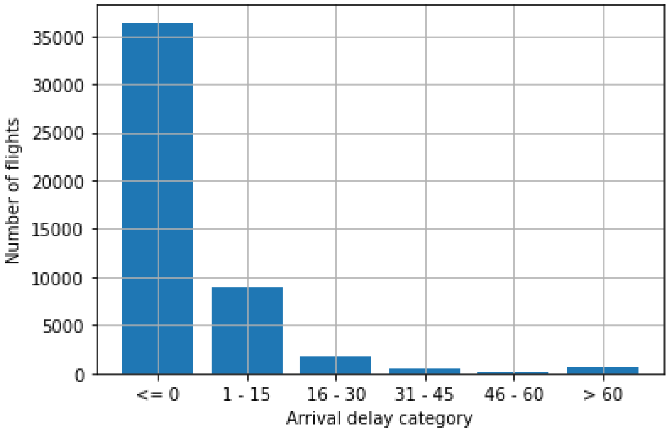




not miss it. In other cases, it may be more profitable to Copa Airlines if the outbound flight departs PTY airport on time without the delayed passengers, and these delayed passengers get seats on the next available flight. When making this decision, the added cost for passenger overnight residence in hotels must also be considered. Flight delay forecasting may help to automate this decision in a data-driven way. We carefully explored the provided data and tested it on a wide variety of machine learning models.

The dataset that we used (confidential, provided by Copa Airlines) is for Copa Airlines flights before the COVID-19 pandemic started. There are 49862 rows (1 for each flight) and 76 attributes. In addition to the features listed below, there are 9 additional columns that summarize the flight delay duration with a Boolean value. For example, one such column indicates if a flight was delayed by 15 minutes or less with a true or false value. Similarly, another column indicates if a flight was delayed by 80 minutes or less with a true or false value. These columns exist for flight delay durations of 0, 5, 14, 15, 80, and 90 minutes.





II. Literature survey

Flight delay prediction is a popular research topic, and many different algorithms and attribute correlation studies have been published and successfully applied to develop high-performance models. The study completed in [2] examines attribute correlations often observed in flight delay prediction problems, including weather patterns, scheduled departure and arrival times, destination airport, day, month, scheduled time and distance, airline, and tail number. N. Chakrabarty [3] offers a thorough overview of the many different approaches to flight arrival delay prediction in the article, "A Data Mining Approach to Flight Arrival Delay Prediction for American Airlines." Successful algorithms include gradient boosting classification, gradient boosting regression, multiple linear regression, Naïve Bayes classification, and supervised machine learning algorithms like decision tree, random forest, AdaBoost, and k-Nearest Neighbors for predicting weather influenced flight delay. Deep learning solutions like recurrent neural networks and artificial neural networks have also proved effective in flight delay forecasting. Logistic regression, decision tree regression, Bayesian ridge, random forest regression, and gradient boosting regression are tested and successfully applied to predict the presence of flight delay in [4]. For the dataset evaluated in [4], Random Forest Regressor was concluded to perform the best using the mean absolute error as a performance metric. Flight delay propagation is another strong area of research since delays often affect other flights using the same aircraft or carrying passengers that are between flights [5].

According to [1], the long short-term memory (LSTM) neural network is also a strong candidate for performing time series forecasting on certain flight attributes. This research was published for the 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology and uses deep learning to predict features like average delay time, departure delay time, arrival delay time, and delay rate. The LSTM model is trained using back-propagation and is a type of recurrent neural network that uses memory blocks over standard neurons [9]. Memory blocks are sophisticated units that use various logic gates to remember and forget past sequences of time series data, allowing each unit to remember relevant patterns observed at previous time steps and use those sequences to predict future samples [9]. Although many of these attributes were not readily available to us, we resampled the original dataset to derive features like arrival and departure delay rate and average delay time and applied this technique. This study also considers weather patterns, but we were unable to acquire the appropriate historical weather records for the scope of this project. We successfully applied similar data preparation steps and algorithms and observed good model performance without the additional weather data.

3. Methodologies

Logistic Regression Model

One of the methods we used was Logistic Regression. Logistic regression, contrary to its name, is actually a classification model. It uses a sigmoid function in order to help classify the data. It can be used to classify either binary data or just data split into certain classes. For this project we used it to classify binary data, specifically whether the flight did or did not have any delay.

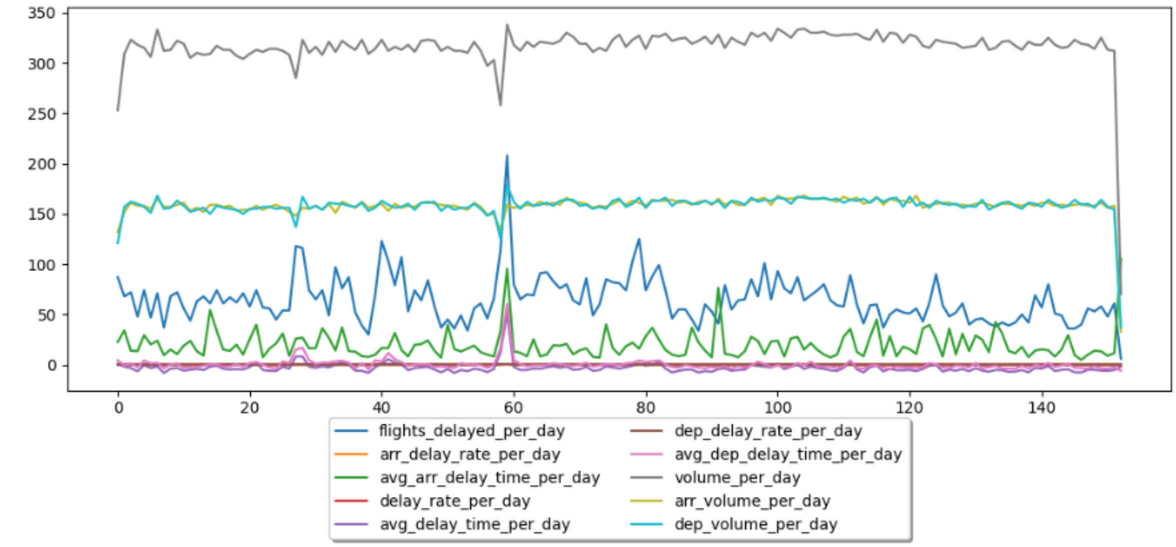
Multiple Linear Regression model

Considering that there has been proven research regarding the effectiveness of the Multiple Linear Regression model in forecasting flight delay, [3][16], we consider its implementation. Utilization of this model makes sense, given that there are several factors that determine whether a flight has delay or not, (date and time, location, distance, etc.). While a simple linear regression model, uses a bivariate model (input x is used to determine an output y), we implement a multiple linear regression model, which uses a multivariate model (inputs (x1, x2, …) are used to determine an output y).[14] Unlike logistic regression, which is a classification model, multiple linear regression can be used as a prediction model [15], and as such we utilize it in order to obtain the amount of delay (in minutes) that a flight may have.

Long Short-Term Memory Neural Network

Due to the success of time series forecasting in literature, we also decided to try various regression and neural network algorithms to evaluate our data and identify potential sample correlations with previous moments in time. To do this, we reformatted the original set to have a constant sample rate that would be appropriate for time series forecasting, For the purposes of this experiment, we first queried the original set for samples where Panama City (PTY) airport was either the origin or destination airport in order to narrow our focus on the main, central airport. We then aggregated various attributes to form new features and create a dataset with a set rate of one sample per day. These new features were created based on the research example in [1], where the long short-term memory neural network was successfully applied to such attributes. These features are listed below and the entire dataset is plotted in the following figure.

1. Total number of flights delayed
2. Arrival delay rate
3. Average arrival delay time in minutes
4. Delay rate per day
5. Average delay time per day in minutes
6. Departure delay rate
7. Average departure delay time
8. Volume per day
9. Arrival volume per day
10. Departure volume per day



In addition to the LSTM model, we also tested k nearest regression, decision tree regression, extra tree regression, and support vector machine (SVM) to determine which model could more accurately predict the attributes above for one day in the future.

Simple artificial neural network

Finally, we also decided to try a simple artificial neural network to predict the categories of departure delays and arrival delays (classification problem). We used a neural network, with linear combinations of weights for artificial synapses, and ReLU (rectified linear unit) activation functions for artificial neurons.

4. Implementations, potential problems, and assumptions if any

Logistic Regression Model

To test how accurate the logistic model was, we ran some tests using different amounts of attributes to train the model as well as having different sample sizes to train the model. All of the models were tested with the same test set that was randomized and had a size of 200. Once the models were trained, they were then tested on this test set and a score was taken to measure its accuracy. Before training, we enumerated any columns that contained string values, dropped rows with missing data, and normalized the attributes.

We trained 12 different models in total using two different variables changing, that being the number of attributes trained and the sample size of the training set. There were four different number of attributes used between 1 and 4 and there were three different sample sizes, those being 500 samples, 2000 samples, and 5000 samples (these samples sizes will be referred to as small, medium, and large from this point forward). We trained and tested each of these models a total of 10 different times, each with a different randomized test set. The results were then averaged and then plotted. In the end we also tried utilizing all the attributes in three different models of varying sample sizes to see if this would improve the scores.

Early on we ran into a problem: the attribute we were trying to predict was imbalanced. Most of its values belonged to one class, specifically around 84%. What ended up happening is that the models would come out with a score of around this percent every time, but this was since they were only predicting the majority class. This means the results these tests were giving us weren't at all useful and we needed to figure out a solution. Because of this, we decided to run an oversampling algorithm on the training set. Once this was done the algorithm started predicting both classes, not just one. From here on out we were able to trust the scores of our models again.

Multiple Linear Regression Model

For the implementation of the multiple linear regression model, we first must consider what our input values and our output value will be. There are two types of delay that are most notorious for causing issues- the arrival delay and the departure delay. Therefore, we run our multiple linear regression implementation twice, once on each kind of delay. The input components, to name a few, include the flight's origin and destination, day of the year and time in which the flight takes place, and information regarding the aircraft and flight itself (size, crew count, etc.). Additionally, we take into consideration the departure delay as an input variable for determining arrival delay, as previous research [16] indicates this may have ideal results.

As for the multiple linear regression model itself, we utilize the `LinearRegression` function from within `sklearn.linear_model` as the foundation. We set the fit intercept to true, as the data is not necessarily centered, and we want the model to calculate an intercept that more closely fits the data. We use a cleaned version of the original dataset as our initial file input. From here, for each variable that we are trying to determine (arrival vs. departure delay), we run a test where we split the initial data into an initial training and test set, then train a `LinearRegression` model upon our training set. Then, we predict the values for our test set and compare them to our test set's results using an R^2 score, or the coefficient of determination as the metric.

Initially, there were a handful of concerns that arose with the initial implementation of the multiple linear regression model. At first, the model was not only being tested upon the exact number of minutes for a flight's arrival/departure delay, but also whether a flight had any delay at all- in other words, a binary variable that would likely be better be suited for a classification method rather than a regression method. Upon further research, it was determined that binary variables are not suitable output for the multiple linear regression method, and the general assumption is that such an output variable should be continuous. [15] Therefore, the primary values that we ended up testing this model for was the actual amount of arrival and departure delay in minutes.

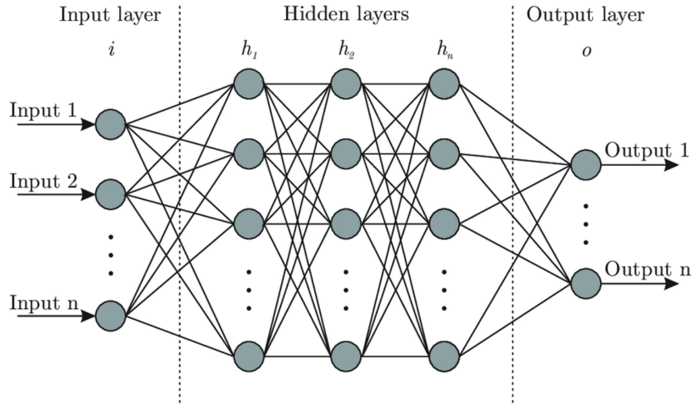
K Nearest Neighbor, Decision Tree, Extra Tree, Support Vector

The basic structure of our regression algorithm analysis was based on [11], which shows how to efficiently test a suite of regression algorithms on a multivariate dataset, perform one or multi-step time series forecasting, and analyze the results using walk-forward validation. Walk-forward validation is performed by training on a small subset of data (e.g. the first 33%) and making one-step predictions for the remaining test set. After every prediction, the model is then retrained on all available data before predicting the next step. We also made use of the `GridSearchCV` function from the `sklearn.model_selection` library to perform a grid search on the parameters for each model and find the combination that produced the best accuracy results. Such parameters include the number of neighbors in k nearest regression, the C, gamma, and kernel values for support vector machine, and the number of leaves and nodes in decision tree and extra tree regressor.

Additionally, we made use of the `series_to_supervised` [12] function from multiple hands-on exercises and tutorials that can be found on the Machine Learning Mastery website [13]. This function is used to transform a time series dataset into a supervised learning problem. For every sample, this function reads and flattens the previous *x* samples and adds them to the current record as new attributes. This allows each algorithm to use a lookback window of samples as input into the model if there is a correlation between each sample and those that came before it. We performed a grid search on this parameter for the regression methods previously listed to determine which lookback window size gives the best performance.

Simple artificial neural network

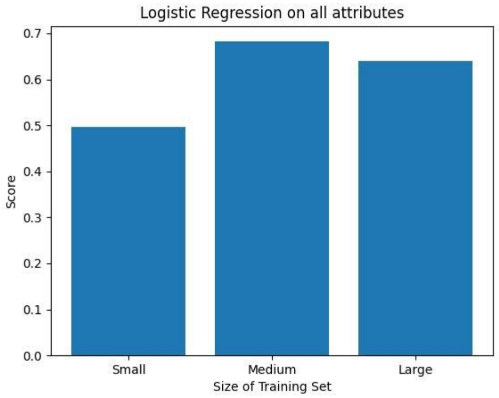
The general artificial neural network for classification into delay categories was implemented using PyTorch, with 3 hidden layers, each having 100 artificial neurons. Synapses are modeled using linear combinations of weight functions, and activation functions use ReLU. The number of input points equals the number of features selected (28), and the number of output points equals the number of delay categories (6). The selected features include scheduled times and dates of departure and arrival, origin and destination airports, aircraft tail numbers, flight numbers, available and scheduled rotation times (duration between 2 consecutive flights using the same aircraft), etc. Loss function used is the cross-entropy loss function. The neural network is trained for 10 epochs, in each type of analysis (predicting categories of departure delays and arrival delays). Training and testing are implemented as functions that are called in each epoch. A batch size of 100 is used.

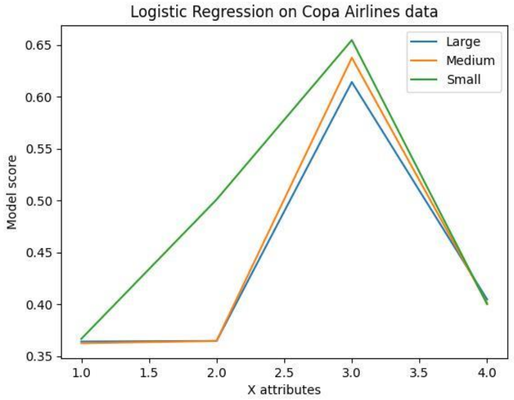


5. Results

Logisitic Regression Model

The graphs below show the results of the logistic regression tests. The best type of model seems to be the model using all the attributes as well as having a medium sample sized training set. This model had a score of around 68%. The large sampled sized variant of the model also did well, having a score right below that of the medium sample sized. For the models that didn't utilize all the attributes, the best number of attributes used was three with the best sample size being small as opposed to medium. For some reason the accuracy fell some whenever we introduced a fourth attribute. This could be because the attribute introduced may not have had a strong correlation with the attribute being tested. All in all, the logistic regression wasn't very accurate for our dataset, even at its best with an accuracy of 68%. Since this model wasn't as accurate, we decided not to use it for any predictions of the delay. Instead, we decided to try other training models and see if they had any better luck at being more accurate.

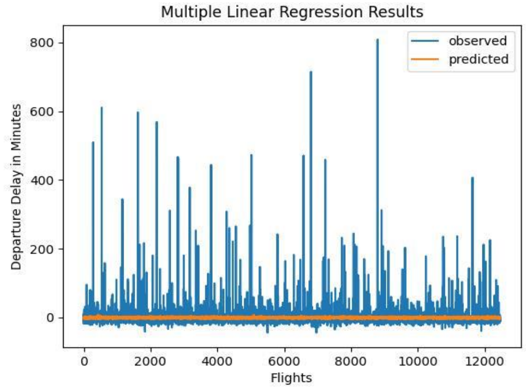


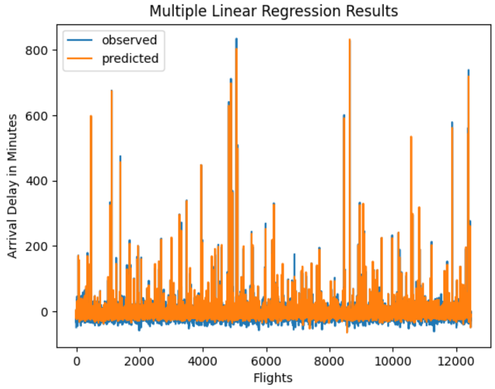


Multiple Linear Regression Model

The results for the multiple linear regression model are as follows. For the estimation of departure delay, the predictions obtained from the model do not seem to match up with the vast range of departure delays observed in the test set. With an R^2 score of 0.004, it is safe to say that based upon factors such as time/date, flight information, and location alone, we cannot accurately predict how much delay there will be for a flight's departure. On the other hand, if we take departure delay into account for determining arrival delay, we see much more fruitful results. The R^2 score for the arrival delay model was 0.884, a significant increase over the departure delay model. Given that the only difference between the two inputs was the presence of the departure delay, we can safely assume there is a strong correlation between the two.

While the results for the arrival delay are promising and reflect the research discussed in [16], the model as is does not work for determining departure delay. Therefore, alternative determining factors may be considered for departure delay, some of which may not be included in the dataset, such as weather, or may be difficult to predict, such as human error (ex. a passenger is late and the airline hold the flight). The results for arrival delay, on the other hand, have positive implications, as being aware of how late a flight may be arriving after it has determined may allow an airline to make alternative preparations depending on how late exactly the plane may be.



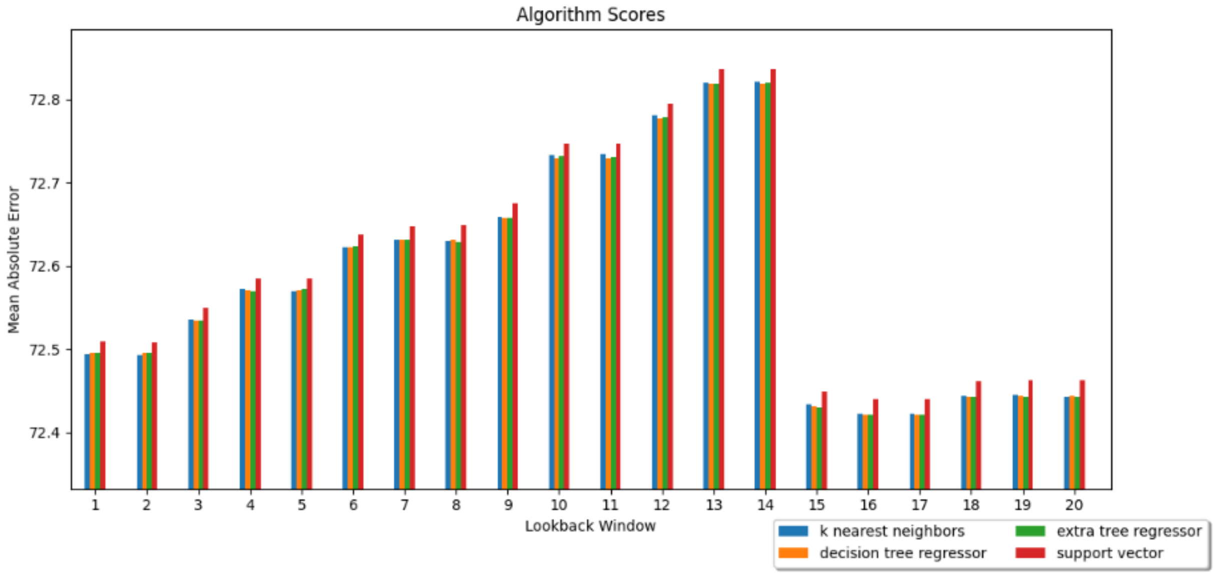


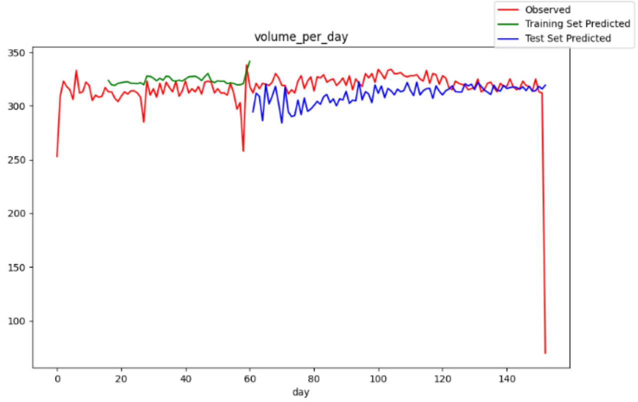
K Nearest Neighbor, Decision Tree, Extra Tree, Support Vector

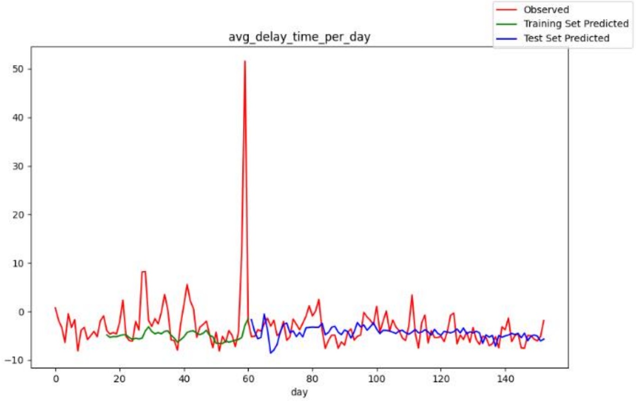
The prediction results for k nearest neighbors, extra tree regressor, decision tree regressor, and support vector machine are displayed in the figure below. After training on the first 33% of data, walk-forward validation was used to obtain the one-step prediction results on the remaining 66%. The means absolute error between these one-step predictions and the actual observed samples was then calculated. This process was repeated for each algorithm for look back windows between 1-20. The prediction mean absolute errors are displayed in the bar graph and indicate that the decision tree regressor and extra tree regressor work equally well at lookback windows of 16 and 17 with a mean absolute error of 72.4.

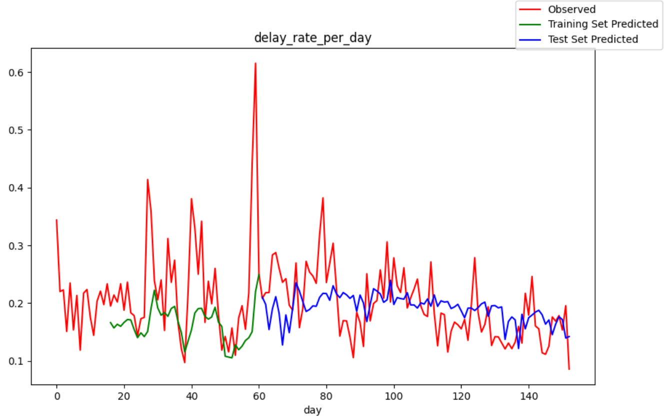
K Nearest Neighbor, Decision Tree, Extra Tree, Support Vector

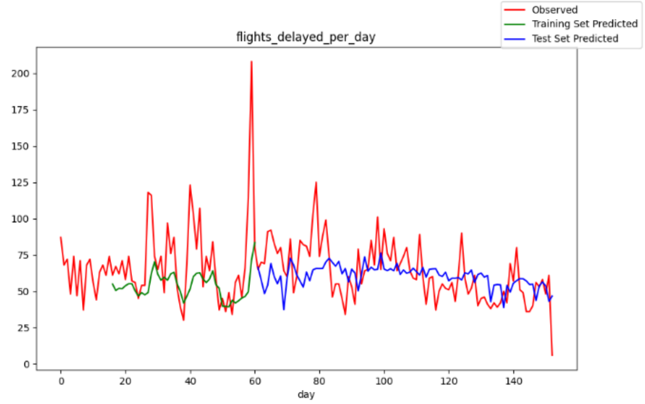
For the long short-term memory neural network, we tried many different combinations for the lookback window, number of neurons, number of epochs of training, and the dropout parameter, which is a regularization parameter used to prevent overfitting by "forgetting" the output of certain hidden nodes in the network. Using 8 neurons, a look back window size of 16, 40 epochs of training, and a dropout parameter of 0.2, we were able to obtain good one-step predictions using the LSTM model. To analyze the results, we performed walk-forward validation and obtained the mean absolute errors.



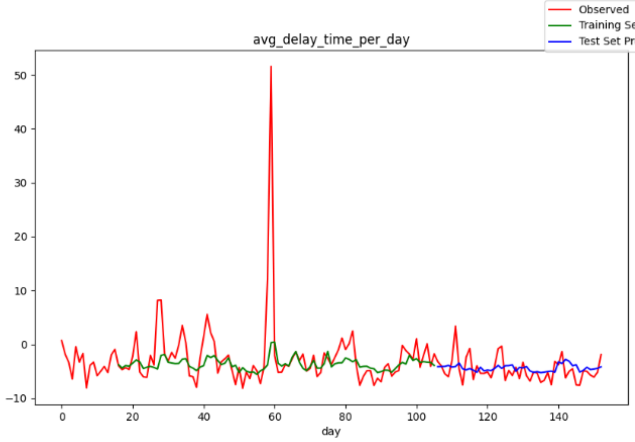


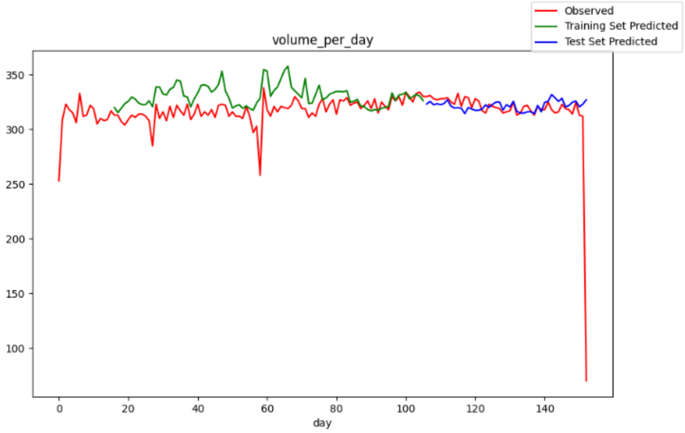


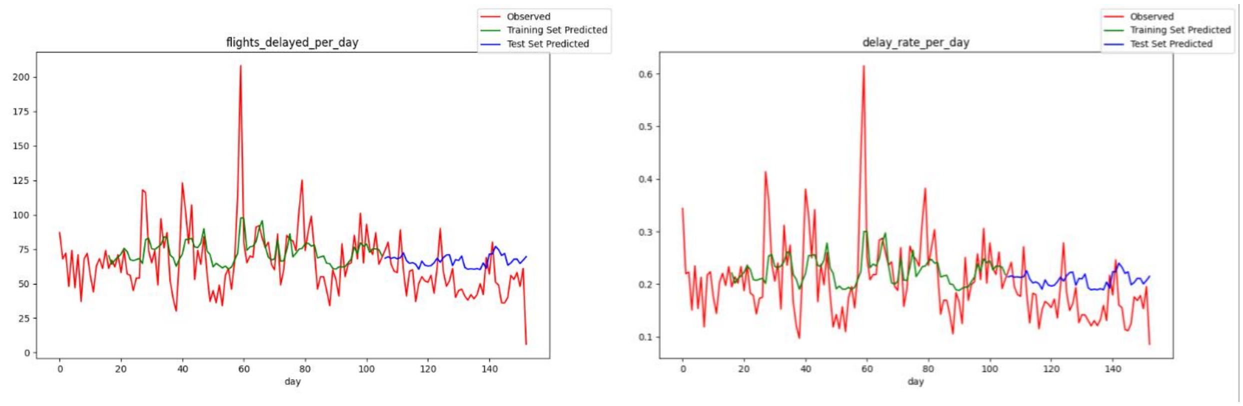


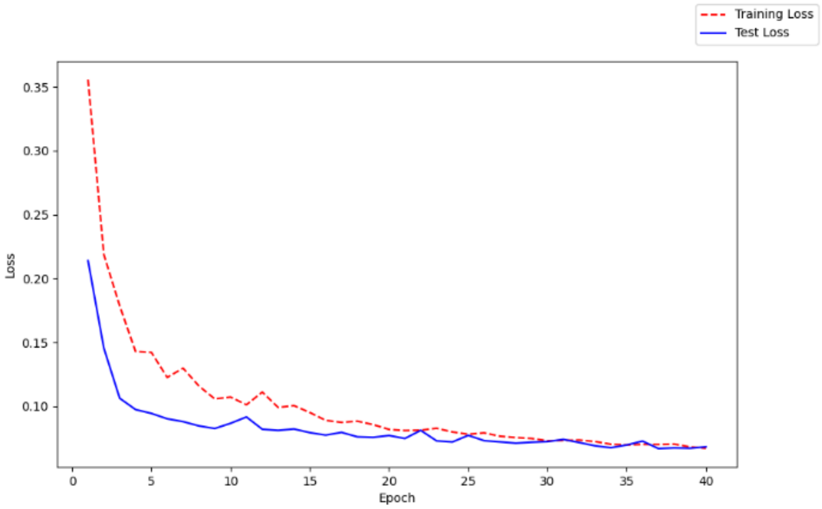


Walk-forward validation is not typically used for neural network analysis because of the time it takes to train before every one-step prediction. We had the time and resources to make this analysis, especially since data resampling drastically reduced the size of the dataset we were working with. This form of analysis also allowed us to compare these results to the regression methods that we also analyzed with walk-forward validation. Using a lookback window of 16, we computed a mean absolute error of 5.95 on the test set, which is significantly smaller than the errors calculated for any of the regression methods that were tried at the same lookback window. We also analyzed the performance of the LSTM model by training on the first 66% of data and using the resulting network weights to predict the remaining 33% without retraining at every step. We used the same network configuration and analyzed the prediction results as well as the loss of the training set and the test set. These results are included below. Since the test loss begins to increase around 40 epochs, it is clear that 40 is a good stopping point to prevent overfitting.









Simple artificial neural network

The simple artificial neural network was run for 10 epochs for each case (departure and arrival). The prediction accuracies are about *83% for departure delay categories*, and *76% for arrival delay categories*. These may seem good, but on closer examination we see that these are the fractions of flights in the largest category (no delay). So, it seems that the neural network is simply making all predictions for the "no delay" category. This seems to be a problem of case imbalance (there are 6 classes, but one class has more than half of all data samples). This may be solved by breaking up the large class into smaller classes, or some other method may have to be used.

6. Conclusion and future research directions

We successfully explored, cleaned, and prepared the provided Copa Airlines data and applied a wide range of machine learning models to predict flight delay-related attributes. Such algorithms included logistic regression, multiple linear regression, k nearest neighbor, extra tree regression, decision trees, support vector machines, the long short-term memory neural network, and artificial neural networks. For each approach, we tried multiple variations on the input parameters and compared the accuracy results. We concluded that multiple linear regression and the long short-term neural network offered the most accurate and useful results. The multiple linear regression algorithm would allow staff to make extremely accurate predictions about the arrival delay time given a particular flight's departure delay time. The LSTM neural network would allow Copa Airlines staff to use a small lookback window of data to obtain an accurate one-day forecast for several different attributes like average arrival flight delay time, average departure flight delay time, arrival delay rate, and departure delay rate for the PTY airport. This information would allow staff to proactively prepare hotels for passengers and make necessary adjustments to flight schedules.

With a few modifications, the LSTM model could also be used to predict multiple days into the future, so this would be a good research area to explore. The correlation between weather and flight delay is also an extremely popular research topic and would be worth exploring to improve the accuracy of our models. Although weather data was not readily available to us, it would be possible to extract this information from historical records online and combine those samples with our own dataset by date. We would expect to see a strong correlation between weather attributes and the arrival and departure delays.

7. References

1. Y. Jiang, J. Miao, X. Zhang and N. Le, "**A multi-index prediction method for flight delay based on long short-term memory network model**," 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT, 2020, pp. 159-163, DOI: 10.1109/ICCASIT50869.2020.9368554.

2. R. Dhanawade, M. Deo, N. Khanna and R. V. Deolekar, "**Analyzing Factors Influencing Flight Delay Prediction**," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), 2019, pp. 1003-1007.

3. N. Chakrabarty, "**A Data Mining Approach to Flight Arrival Delay Prediction for American Airlines**," 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), 2019, pp. 102-107, DOI: 10.1109/IEMECONX.2019.8876970.

4. P. Meel, M. Singhal, M. Tanwar and N. Saini, "**Predicting Flight Delays with Error Calculation using Machine Learned Classifiers**," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020, pp. 71-76, DOI: 10.1109/SPIN48934.2020.9071159.

5. R. Yao, W. Jiandong and X. Tao, "**Prediction model and algorithm of flight delay propagation based on integrated consideration of critical flight resources**," *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, 2009, pp. 98-102, DOI: 10.1109/CCCM.2009.5267970.

6. Peter Belobaba, Amedeo Odoni, Cynthia Barnhart, etc. "**The Global Airline Industry**". 2016, Wiley.

7. Nabin Kafle and Bo Zou. "**Modeling Flight Delay Propagation: A New Analytical-Econometric Approach**". *Transportation Research: Part B: Methodological*. September 2016. DOI: 10.1016/j.trb.2016.08.012.

8. Sarthak Sharma. "**Copa Airlines Research Project: Report for Directed Individual Study (Summer 2021 term)**". Department of Scientific Computing, Florida State University.

9. J. Brownlee, "Time Series prediction with LSTM recurrent neural networks in python with keras," *Machine Learning Mastery*, 27-Aug-2020. [Online]. Available: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/. [Accessed: 03-Dec-2021].

10. J. Brownlee, "Multivariate time series forecasting with lstms in Keras," *Machine Learning Mastery*, 20-Oct-2020. [Online]. Available: https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/. [Accessed: 03-Dec-2021].

11. J. Brownlee, "How to develop multivariate multi-step time series forecasting models for Air Pollution," *Machine Learning Mastery*, 27-Aug-2020. [Online]. Available: https://machinelearningmastery.com/how-to-develop-machine-learning-models-for-multivariate-multi-step-air-pollution-time-series-forecasting/. [Accessed: 03-Dec-2021].

12. J. Brownlee, "How to convert a time series to a supervised learning problem in Python," *Machine Learning Mastery*, 21-Aug-2019. [Online]. Available: https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/. [Accessed: 03-Dec-2021].

13. D. D. J. D. Scientist and K. B. S. Engineer, "Machine learning mastery," *Machine Learning Mastery*, 25-Oct-2021. [Online]. Available: https://machinelearningmastery.com/. [Accessed: 03-Dec-2021].

14. Uyanik Gülden and Güler Neşe. **"A Study on Multiple Linear Regression Analysis".** *Procedia – Social and Behavioral Sciences* December 2013. DOI: 10.1016/j.sbspro.2013.12.027

15. Tranmer, M., Murphy, J., Elliot, M., and Pampaka, M. (2020) Multiple Linear Regression (2nd Edition); Cathie Marsh Institute Working Paper 2020-01. https://hummedia.manchester.ac.uk/institutes/cmist/a rchive-publications/working-papers/2020/2020-1-multiple-linear-regression.pdf

16. Yi Ding, **"Predicting flight delay based on multiple linear regression"** 2017. IOP Conf. Ser.: Earth Environ. Sci. 81 012198