Jet Lagged

Nhan Hoang
Virginia Tech
nhan21@vt.edu

Vanessa Eichensehr
Virginia Tech
veichens@vt.edu

Jackson Livanec
Virginia Tech
jlivanec@vt.edu

Abstract

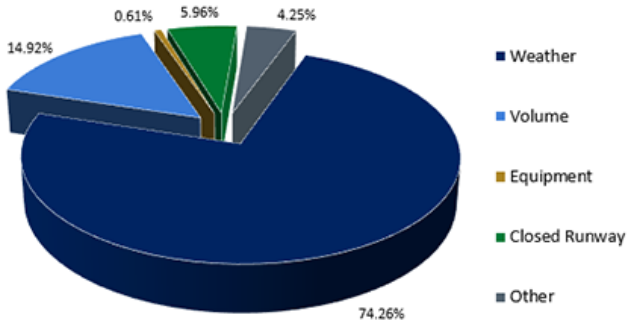
In today's world, air travel is a fundamental type of transportation to connect the world together through a network of airports. As aviation continues to grow, flight delays are also becoming more frequent. The primary objective of this research is to develop a machine learning model using the historical flight data to predict potential flight delays given real time weather data. In order for the model to learn the historical data, the flights on time performance dataset was collected through Airline Data Inc, a service that provides processed airline data sourcing from the Department of Transportation. In addition, the historical weather data was collected from Meteostat, an open source platform that stores the weather database sourcing from various public sources including National Oceanic and Atmospheric Administration (NOAA). The real-time weather data comes from using the NOAA API. Multiple algorithms were evaluated ranging from Decision Tree, Logistic Regression, Random Forest, XGBoost, One-Class SVM, and Neural Network. Using metrics like precision, recall and F1-score to validate a model's performance, notable findings are Random Forest were able to receive an accuracy of 0.66 and XGBoost with an accuracy of 0.60. This work highlights the potential for a machine learning tool that can help air traffic controllers and airports to provide announcements in a timely manner that will minimize the disturbance it causes to the customers and airlines.

In: M. Meder, A. Rapp, T. Plumbaum, and F. Hopfgartner (eds.): Proceedings of the Data-Driven Gamification Design Workshop, Tampere, Finland, 20-September-2017, published at http://ceur-ws.org

1 Introduction

detail what the problem is, what you did, and your contributions Currently, travel by air is the fastest way and most popular mode of transportation over long distances. Whether a passenger is traveling for tourism, visiting business partners, or reuniting with family, air transit all play a critical role in providing a fast and safe service. Unfortunately, air travel is not always as straightforward due to potential delays from air carriers, runway issues, and weather conditions. As the weather state is interconnected to whether an aircraft can operate safely, flight delays are sensitive to any changes in the weather whether it is a gust of wind or precipitation. Through the latest update from the Federal Aviation Administration's Next Generation Air Transportation System, a research program with an effort to modernize the current National Airspace System, the largest cause of delay is weather accounting for roughly 74% of delays greater than 15 minutes [1].

Figure 1: Causes of air traffic delay by FAA [1]



Throughout the years MITRE has been working closely with FAA to provide tools that can help air

traffic controller making informed decisions on air traffic in real-time. Dating back to 2001, MITRE developed a model to visualize how the queues of planes are affected based on a delay due to weather conditions [2]. Then, in 2013, MITRE developed another tool that integrates the weather data into Air Traffic Management to aid the air traffic controllers in real-time. Up to today, a lot of the decision-making is still in the hands of humans. While these existing tools can be helpful, they are built within the predefined rules. Meanwhile, weather conditions and flight operations are quite difficult and do not always neatly fit in with the rule-based system. Currently, the FAA hasn't taken new initiatives into the machine learning approach. However, many researchers have already taken actions to determine if machine learning can be helpful. Therefore, the goal of

2 Related Work

Numerous studies have explored machine learning techniques for predicting flight delays. Published in 2016, a study by Choi et al. predicted weather-induced airline delays using machine learning algorithms [2]. The researchers combined US domestic flight data for the 45 largest airports and weather data from 2005 to 2015 [2]. Then, they used the SMOTE sampling technique to synthetically generate data to balance their dataset [2]. They used the following features: Quarter of the Year, Month, Day of Week, Departure Time, Arrival Time, Wind direction angle, Wind speed rate, visibility, precipitation, snow depth, snow accumulation, and five other coded descriptors for the weather conditions [2]. After normalizing their data, the authors then trained and tested four machine learning models: Decision Trees (77.92% accurate), Random Forest (81.37% accurate), AdaBoost (83.21% accurate), and KNN (61.69% accurate) [2].

Published in 2024, a study by Ajayi et al. used network centrality measures to enhance flight delay predictions from machine learning models [3]. The researchers first constructed a network model in which airports served as nodes and flight routes as edges [3]. The calculated three network centrality measures at each node: degree centrality, betweenness centrality, and closeness centrality [3]. Once the scores were computed for all airports, the dataset was augmented by adding the network centrality measures for the departure and arrival airports, for a total of six additional features [3]. The researchers tested and trained three models: Random Forest (86.2% accurate), Gradient Boosting (85.8% accurate), and CatBoost (85.6% accurate) [3].

Pamplona et al. published research in 2018 predicting air traffic delays between Sao Paulo and Rio de Janeiro with a supervised neural network [4]. Aiming to predict delays only in January 2017, the authors built an artificial neural network with four hidden layers that had an average accuracy of 91.3% [4]. Their model suggested that the day of the week, block hour, and route have great influence on the flight delay and failed to consider meteorological factors [4].

In 2022, Shao et al. predicted flight delays with spatio-temporal trajectory convolutional network and airport situational maps [5]. This was a vision-based solution, intended to achieve a high forecasting accuracy for just one airport, Los Angles Internation Airport [5]. The authors showcase end-to-end deep learning architecture, known as TrajCNN, to capture spatial and temporal information from the situational awareness map, which shows statuses from airports across the country [5]. Though it only makes predictions for one airport, the proposed framework obtained good results (around 18 minute mean error) for predicting flight departure delay [5].

3 Methodology

A delayed flight is defined in this study as a flight for which the actual departure time is greater than 15 minutes beyond the scheduled departure time. This does not include flights that are canceled altogether. This cutoff threshold for flight delays was determined in order to isolate flights that were delayed most likely due to weather, as logistical or other routine delays may result in shorter delays. This reduces the size of the training dataset, but this tradeoff is necessary to balance long-term model accuracy.

3.1 Data Collection

Historical US domestic flight records from 31 airports were gathered using the DOT's On-Time Performance data from 2023 and enriched with the historical weather report for the given area at the time of departure floored to the previous hour, as more granular data is not available. The weather was gathered using Meteostat and joined using the latitude and longitude coordinates of the airport and timestamp. The dataset include the following 8 weather features:

- Elevation
- Temperature
- Dew Point
- Relative Humidity
- Chance of Precipitation
- Wind Direction
- Wind Speed

- Wind Gust Speed

Firstly, elevation was selected because it impacts temperature, pressure, and precipitation severity. High-elevation airports often experience unique challenges that might contribute to delays, such as reduced aircraft performance due to lower air density and greater snow accumulation [6]. Temperature influences atmospheric density and is closely tied to other meteorological factors such as wind patterns and type of precipitation [6].

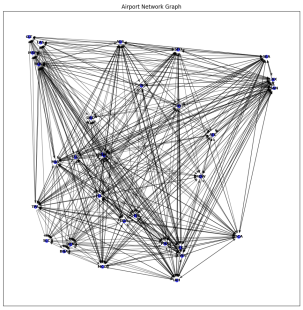
Next, dew point was selected because it indicates the level of atmospheric moisture which is a key indicator for fog formation, precipitation, and aircraft frame, propeller, or turbine icing [6]. Low dew points suggest dry conditions while high dew points are associated with fog and storms. Relative humidity is closely coupled with precipitation potential. It complements temperature and dew point and is used to backfill missing data as well as providing a complete picture of air moisture.

Precipitation chance predicts the likelihood of rain, snow, or sleet which disrupt flights directly [6]. Wind direction is important in relation to runway direction because crosswinds can deem a runway unsafe for takeoff. Wind speed and gust speed inform aircraft stability, turbulence, and the feasibility of takeoff and landing. Airports will often employ operational limits for wind speeds.

3.2 Data Exploration

After collecting historical flight and weather data, the data was explored and analyzed. First, the data was plotted as a network graph, with airports as nodes and flights as edges [Figure 1].

Figure 2: Network of US Domestic Flights 2023



Degree centrality measures the number of direct connections a node (e.g., airport) has in a network. For an unweighted graph, it is defined as:

$$C_D(v) = \frac{deg(v)}{N-1}$$

where $deg(v)$ is the degree of node v, and N is the total number of nodes in the network. Airports with high degree centrality act as regional hubs with many direct connections, which can significantly influence the spread of delays across the network.

Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It is defined as:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

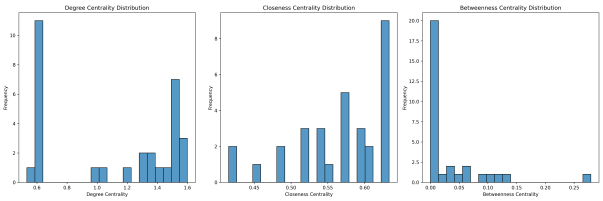
where σ_{st} is the total number of shortest paths between nodes s and t, and $\sigma_{st}(v)$ is the number of those paths that pass through node v. Nodes with high betweenness centrality are critical for maintaining flow within the network. Airports with high betweenness often serve as transit points, making them prone to cascading delays if disrupted.

Closeness centrality measures how close a node is to all other nodes in the network. It is given by:

$$C_C(v) = \frac{N-1}{\sum_{u \neq v} d(v,u)}$$

where $d(v,u)$ is the shortest distance between nodes v and u. Airports with high closeness centrality are well-positioned for efficient travel across the network. Such airports are likely to recover more quickly from disruptions or delays.

Figure 3: Network Centrality Measures of US Domestic Flights 2023



The distribution of each of these network centrality measures across all airports is seen in Figure 2. The first graph shows a distribution of degree centrality, with most nodes having either very few or many direct connections, indicating a network with a few hubs and many peripheral nodes. The second graph indicates that most nodes are relatively close to others in the network, suggesting a compact network structure. The third graph reveals that most nodes have low betweenness centrality, with only a few acting as significant bridges, highlighting the importance of certain nodes in maintaining network connectivity. The top ten airports and their values are listed in Table 1.

In addition to the aforementioned weather features, the three network centrality measures for each departure and arrival airport were added to the exploratory features for analysis. Other logistical features, such

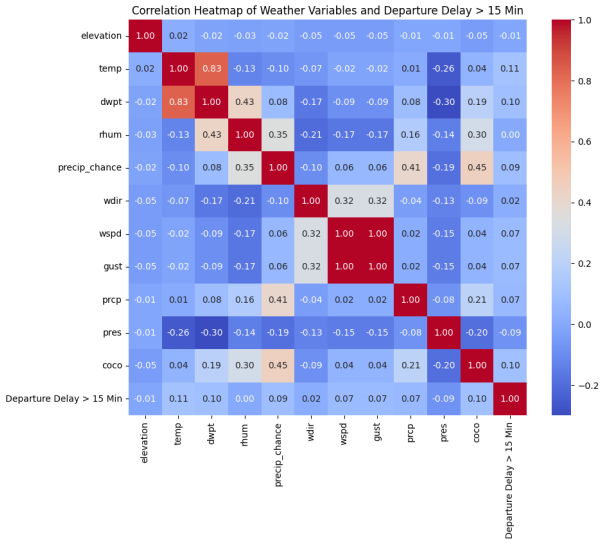
Table 1: Top 10 Airports by Betweenness Centrality

Airport	Degree Centrality	Closeness Centrality	Betweenness Centrality
HNL	0.9667	0.4154	0.2793
SAN	1.4667	0.5400	0.1264
IAD	1.4000	0.5143	0.1184
MIA	1.5000	0.5400	0.1000
PHL	1.3333	0.4909	0.0954
AUS	1.5333	0.5684	0.0621
LGA	1.0667	0.4154	0.0609
EWR	1.3667	0.5143	0.0552
DCA	1.3333	0.4909	0.0322
TPA	1.5000	0.5400	0.0310

as Quarter, Day of Week, Month, and Hour were also added to the exploratory features.

In our analysis, we plotted the distributions of each of these features across all flights. After normalizing and balancing our data, we then looked at each feature with respect to our target variable, Departure Delay ¿ 15 Min. Figure 3 shows a correlation heat map of each of our weather features. **Note:** this chart can/should be updated based on which data we are using (i.e. balanced or SMOTE data or both/neither; new features or weather features only)

Figure 4: Correlation Heat Map of Weather Variables and Departure Delay



TODO: Add plots from 2 or 3 most significant features

3.3 Feature Selection

3.4 Algorithm Analysis

After the data exploration process, many algorithms were put on the weighing scale to see which fits the problem the most.

3.5 Model Selection

The problem is framed as a binary classification task to predict whether a flight will be delay based on the weather features by the hour. To demonstrate a basic classification, Decision Tree were selected as the base model as it can capture the basic non-linear relationship. Logistic Regression was also added as a base model due to its ability to show the linearity relationship within the data. After comparing the statistics, the tree-based approach seem to have a more appropriate fitting to the dataset. Hence, leading to the decision to experiment with a bagging solution, Random Forest, to improve the accuracy of the model and possibly reduce the over fitting due to the imbalance dataset between on-time flights and delayed flights as Random Forest does average out all the predictions from multiple trees. At this stage, XGBoost was also added to the collection due to its computational efficiency. The downside to XGBoost was that it requires extensive hyper parameter tuning for the model to perform optimally. Due to the nature of the skewed dataset, one-class SVM model was also investigated as it tends to use in anomalies detection and in the specifically case of this problem, a delayed flight is an anomaly. Finally, Neural Network was also implemented with the ambition for it to model the hierarchical relationships between the features. Through the selections of algorithms described above, our main goal is to find a well balanced model that can interpret the complex relationships between the weather features and possible flight delay.

As the focus is on the classification tasks, performance metrics like accuracy, precision, recall, and F1-score were used to evaluate the models. While accuracy can give the overall view of how a model is performing, but due to the imbalanced dataset it is important to add precision and recall to use for analysis to ensure that there is a class balance in place. The goal is to have the values for recall, precision, and F1-score to be roughly the same to know that model is not over nor under fitting.

4 Experiments

4.1 Data

The experiment is based on three primary dataset. First is the historical flight On-Time Performance data from Airline Data Inc filtering only large hub airports. Secondly, the historical weather data of comes from Meteostat's Python package base on the coordinates and requested hours. Thirdly, the airports information was retrieved from the International Air Transport Association (IATA). Lastly, an API was connected to retrieve the live weather data for up to 100 hours ahead. To prepare the data for analysis, the On-Time Performance dataset was merged with the airport information to enrich the dataset with the coordination, elevation, and time zone. Based on each airport timezone and the flight scheduled departure time, it was

then converted to UTC in order to retrieve the historical and real-time weather data. At the end, all of the flights schedule was each merged with the corresponding hour bin of the weather features from the past weather dataset.

Data cleaning at this point included unit conversions, filling missing values with estimates based on other meteorological parameters, and bucketing wind direction into 8 categorical cardinal directions (e.g. North, South, etc.). This bucketing assisted with the one-hot encoding of the input vector. Flights were evaluated based on departure tardiness and flagged as delayed or on time. Canceled flights were scrubbed out of the dataset entirely because the cancellation could be unrelated to weather conditions. All rows without temperature data were dropped and other weather data points not available in the historical weather data were back-filled using industry standard estimation formulae based on the other weather features.

The data available via NOAA forecast contains different information than the historical weather reports, therefore some feature engineering was necessary. For example, precipitation is a key factor when determining the timeliness of flight departures, so measured historical volumetric precipitation was graded between 0-100 to match the format of forecast-ed precipitation chance.

After handling missing data, there were approximately 2.8 millions records for 2023, with 2.2 million flights running on time and 637K flights delayed. The extreme imbalance in the prediction class initially led to poor model performance, so all delayed flight rows were kept and an equally sized group of on-time flights were randomly sampled and shuffled into the data, resulting in a training dataset containing 1.27 million rows of flights, with an equal number of delayed and on-time flights.

4.2 Experimental Setup

TODO: I think this might be where we address hyperparameter tuning.

4.3 Results

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.66	0.67	0.63	0.65
Logistic Regression	0.57	0.57	0.57	0.57
Random Forest	0.66	0.65	0.66	0.66
XGBoost	0.60	0.61	0.57	0.59
Neural Network	0.56	0.56	0.56	0.56

Table 2: Model Performance

The data was scaled, encoded and randomly split into 80% training and 20% validation data. Several models were constructed to predict the presence of a flight delay: Decision Tree, Logistic Regression, Random Forest, XGBoost, One-Class SVM, and a Neural Network. The real-time weather forecast is transformed using the same scaler and fed as a scaled input vector to the chosen model to produce a prediction window of potential flight delays.

5 Discussion

Their performance is reported in Table 2. The Decision Tree and Random Forest models performed the best, achieving an F1 score of 0.65 and 0.66, respectively, while other models, including XGBoost and the Neural Network struggled due to the complexity of capturing nonlinear relationships in the data. The one-class SVM model, typically suited for anomaly detection, was fed the entire data set but was unable to produce meaningful results.

We attempted several techniques to improve model performance beyond under-sampling the on-time class such as using SMOTE, however, this method lead to overfitting, particularly with the decision tree-based models.

Further feature engineering was explored, such as creating interaction terms between weather variables, such as combining features to estimate accumulated precipitation and likelihood of snow. We also used airline seasonality data as input features, for example, flights that occur on or near Thanksgiving, Christmas, and the first and second weekends of August, which have seasonally high volume citation needed. However, these additional features did not improve model performance, likely due to the strong colinearity among existing and engineered weather features. Although there is much more travel during certain periods, airlines are also expected to be equipped and prepared to handle seasonality, and therefore this does not play a large role in determining delay likelihood.

To increase the robustness of the model, the hyperparameters of all models were tuned to maximize the F1 score. This yielded marginal improvements in precision and recall. Ultimately, the Random Forest model with tuned parameters showed the most consistent performance across metrics and was selected as the predictive model for weather forecasts.

An F1 score of 0.66 for the Random Forest model is exceptional performance given the even balance of precision and recall. It is also expected that not all delays are caused by inclement weather (what percentage of delays are weather related?) so an approximation in this case is acceptable. The model is applied to weather forecast data to provide an hour-by-hour prediction of delay likelihood at a given

airport.

There are no clear patterns in flights that were misclassified with higher frequency, therefore, model ensembling was not chosen to enrich predictions future flight delay predictions. The simple GUI included in the source code allows a user to input a forecast window, origin airport, and select between to two highest performing models. Data is retrieved for the specified window at the location of the specified airport. The weather data collected via this technique is transformed with the saved scalar used to transform the training data such that all of the input variables are the same. The model is then used to predict delays for future flights. Although the training data was an equal proportion of delayed and non-delayed flights, the model output properly classifies delays conservatively which is consistent with the relatively low volume of overall delayed flights.

6 Conclusion

7 Acknowledgements

We would like to thank you Dipasis Bhadra, a senior quantitative economist, from the Federal Aviation Administration for working with us to retrieve the dataset from Airline Data Inc and provide great internal insights that initiated us to the research topic.

8 Author Contributions

9 Data and code availability

The dataset and code is available through the link below, where it consists of of the data folder including the raw data and processed data outputs. It also contained all the preliminary and preprocessed code files went into the model. `https://drive.google.com/drive/folders/1XGWIiMgocHnJVsfm907OHzJJG8Rs_WI8?usp=sharing`

The cleaned up models and front end are available in this demo file: `https://colab.research.google.com/drive/1IBm1qJXOnbrqPRLAyyTK3MOMuhoPhD8C?usp=sharing`

References

[Com79] D. Comer. The ubiquitous b-tree. *Computing Surveys*, 11(2):121–137, June 1979.

[Knu73] D. E. Knuth. *The Art of Computer Programming – Volume 3 / Sorting and Searching.* Addison-Wesley, 1973.