AeroForecast

Team #018

Ninaad L., Kartik N., Nathan W., Amish S., Jaegook K., Aryan M.

Our final website can be found at: Link to our website

INTRODUCTION

Flight delays are far from rare. An average of 30,000 delayed flights per day globally, and within the United States alone, that includes 7,000 to 9,000 flights per day. This impacts a significant amount of people's travel plans every day, and often these delays are last minute or unexpected so it's difficult for people to plan their personal or business-related circumstances around them. Therefore, there is value in predicting flight delay. A visualization tool to track and predict these delays is clearly a need in the status quo.

PROBLEM DEFINITION

People aren't aware of when their flight is going to be delayed or not, but by the time they reach the airport, it's often too late to make travel arrangements. By creating a solution, users of AeroForecast can prepare for a potential delay ahead of time.

LITERATURE SURVEY

Rao et al. offers a comprehensive exploration of big data systems, cloud services, and tools that cover data management and analysis (Rao et al., 2019). Their paper compares distributed file systems, examines ML tools, and surveys visualization and key technologies. We can incorporate a distributed file system with a cloud-based framework like AWS Spark on EMR or HDFS. Băicoianu et al. proposes efficient solutions for generating user-centric reports, emphasizing preprocessing and partitioning for faster, more relevant querying (Băicoianu et al., 2023). We can leverage these concepts in our project for faster reporting on specific flight detail queries. However, it's important to consider preprocessing overhead and other optimization techniques like caching.

Our team researched on visualizing maps and heatmaps. The paper by Neteck et al. focuses on the best performing libraries across marker clustering tasks (Neteck et al., 2019). Mapbox GL JS performed well and could handle large data, which is useful as it shows clustering libraries we could use for our heatmap/map. We can use Supercluster and Mapbox GL JS in our flight prediction. This paper is limited as it just focuses on clustering libraries. Klein et al. focused on problems with visualizing flights, including outlier spatio-temporal patterns and highlighting fine-scale detail (Klein et al., 2014). Their solutions use nested level-of-detail views. This paper was helpful in depicting visualizations for flight prediction. This paper did not detail predicting flights; there is no information about graphical flight representations. Severe weather events have an effect on flights. Glass, Davis, and Watkins-Lewis focus on the best way to visualize both time-specific flight as well as weather data, so air transport officials can better manage traffic (Glass et al., 2022). This paper was helpful in using airports as nodes and flights as edges, which could impact our own visualization process. We could use this approach to format our weather and flight data to yield informative visuals. However, this approach does not explore more severe delay-causing events. Also, they do not employ an airport heatmap format that we would like to use.

We also did an extensive review of current ML techniques within the paragraphs below. Yazdi et al. made a stack denoising autoencoder that they appended to a standard mode (Yazdi et al., 2020). We can adapt our approach to use a denoising autoencoder, but one shortcoming of this paper is that the model is heavy and training/inference time is long. Another novel method by Cai et al. is using a GOGCN model, which mixes Graph Convolutional Networks with Operational information (Cai et al., 2023). However, GCNs lack human-intelligible explanations so an addition of feature importance will help. Wang et al. combine simulation and real-time data mining to make an agent-based system (Wang et al., 2021). The benefit is the use of real-time data, but they do not incorporate weather patterns.

Shao et al. used aircraft trajectory and contextual data to predict departure delays via a convolutional neural network, achieving low error on an airport's data (Shao et al., 2022). We could apply similar spatio-temporal modeling but evaluate on multiple airports to improve generalizability. Similarly, Belcastro et al. implemented a Random Forest classifier on flight, weather and schedule data, showing good arrival delay prediction accuracy and scalability using a cloud platform (Belcastro et al., 2016). We can also use geographic information to capture flight delay relationships geographically. Ding proposed a multiple linear regression approach that provided decent arrival delay prediction accuracy using departure delay and route distance features (Ding, 2017). Our project can improve on this by incorporating weather data used by Belcastro et al. and incorporating binning outputs for higher accuracy.

Güvercin forecasts flight delays by clustering airports and fitting them into a common model, REG-ARIMA (Güvercin et. al, 2021). Truong uses causal ML on flight data to build a structural causal network for high-accuracy delay prediction and mitigation strategies (Truong, 2021). Aljubariy collects and integrates real-time IoT flight, weather, and air quality data to achieve 85.74% prediction accuracy, showing the value of live data (Aljubairy et al., 2020). We can adapt these methods to accurately predict flight delays. The downside of these models is they rely only on historical data and do not use live data.

Cai develops a spatio-temporal graph neural network to predict flight delays across airport networks. This is relevant for incorporating interdependencies between airports, but the implemented model may be too complex for our purposes (Cai et. al, 2022). Qu predicts flight delays by modeling aviation network topology and temporal correlations of relevant factors (Qu et. al, 2020). Their perspective on spatial and temporal factors is useful, but the work is not comprehensive as it focuses specifically on Chinese airports. Liu presents a generalized flight delay prediction method using ADS-B data and gradient boosting decision trees. This flexible data-driven approach could be extended for our purposes, but it lacks spatio-temporal aspects (Liu et. al, 2020).

The paper by Lundberg and Lee introduces a unified approach known as SHAP (SHapley Additive exPlanations) for interpreting predictions made by ML models (Lundberg, 2017). We can use SHAP to interpret our results and give insights on what the model predicts to users. On the other hand, this method has exponential complexity and heavy overhead.

PROPOSED METHODS

We break down our innovations to be the following: novel heatmap visuals using detailed APIs, data preprocessing and feature analysis, and ensembling predictive classification and regression models.

For the frontend of our application, we wanted user experience to be intuitive; travelers should waste no time in plugging in their information and getting amazing visual insights on potential delays and cancellations. We set up the foundation frontend stack, leveraging React, PrimeReact, and Bootstrap to develop a clean UI. We developed a screen where students can input their airline, flight number, and date using filtering and masks to make the form submission easier. We had discussions about customer information and came to a consensus that we need to choose data that is easy to find and super simple to input. This led us to using the airline, flight number, and date as our input data and felt it made the user journey simple.

Once the information is submitted, users are directed to a visualization grid that provides them all the necessary details about potential delays and cancellations. We have a simple pane that shows them our own prediction (whether the flight will be on-time, delayed, or canceled). The complexity of our model output also allows us to show them more statistics. In another pane, we have a donut chart showing our confidence percentage of each possibility occurring. Additionally, we also have a bar chart displaying the percentage chance a flight could be delayed for a set of ranges. A US map will be shown with their flight path from their departing to arriving airport. The map has the airports and their percentage of flight delays as a color scale, with white being lower likelihood of being delayed and red being a higher percentage. We do not have this in place, but we will add pertinent airport details (such as air traffic and weather surrounding the airport) as banners above the locations.

We plan to leverage data mainly from two sources: the Aerodata Box and CheckWX APIs. The CheckWX API is instrumental in providing us with comprehensive, real-time meteorological data tailored to aviation needs. This includes crucial information such as clouds, dewpoint, VFR, temperature, wind, and visibility, which directly impact flight schedules and safety. By leveraging the granularity and precision of this weather data, we can predict and visualize weather-induced delays with higher accuracy.

Simultaneously, the Aerodata Box API serves as the backbone for our flight data infrastructure. It offers us access to real-time updates on global flight statuses, including departure times, arrival times, in-flight progress, and other flight-related events. The power of Aerodata Box lies in its detailed and extensive dataset that encompasses an array of airlines and airports, ensuring that our application has global reach.

By synthesizing data from both APIs, we aim to create a robust dataset that not only provides a snapshot of live flight statuses but also contextualizes this data within the prevailing weather conditions, which is important to us as according to the FAA, weather accounts for over 70% of all recorded delays. This merged dataset will be the foundation for our predictive models and visualizations, enabling users to understand the connection between weather patterns and flight delays. The agility of our application hinges on the seamless integration of these two APIs, where the real-time nature of the data feeds into a dynamic and responsive user interface. Figure 1 in the appendix shows the diagram of our product.

We randomly sample 1000 data points each month from January 2022 to November 2023 in order to gather flight information data. This data is taken from the Bureau of Transportation Statistics, which includes columns such as date, airline, and origin/destination airports. We also appended columns from NOAA's historical weather dataset by airport code. For both origin and destination airports, we added average temperature, inches of snow, rain, wind, and other columns. Next is our preprocessing step and null value exploration. For each null entry, if the column is a delay column, we will fill it with 0. We parse column by column to determine what to fill the null placeholder with.

Now, we moved onto Exploratory Data Analysis. We want to visualize the data spread and determine the best bucket sizes for our departure delays. We also explore the official delay (15 minutes) percentage with regards to airline, date, and other features. We also run Pearson and Spearman correlation tests. We then explored which features are highly correlated to each other. Moving onto PCA, we bucketize on 30 minute intervals and run PCA with 2 and 5 principal components on our data with an output feature set to bucketized_delay. A scatter plot is created to visualize the separation between the 2 principal components. For the model, we conduct a two-stage analysis.

The first is the classification stage. A random sample of 10,000 rows is taken from the original dataset (df) for our current evaluation testing. Features (X) and labels (y) are defined based on this sampled data. We run a variety of classification algorithms (Random Forest, Logistic Regression, SVM, KNN, Decision Tree, XGBoost, AdaBoost), and each of these models is trained and evaluated using metrics like ROC AUC, Precision, and Recall. Receiver Operating Characteristic (ROC) curves are plotted for each classifier to visualize their performance.

The second stage first began as regression. Several regression algorithms (Linear Regression, Random Forest, XGBoost Regressor) are trained using the filtered data. Each regressor is trained and evaluated using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score. However, getting the data value right is very hard due to regression techniques, so we turned this into classification by using the buckets we had before. Simply, this turns into multiclass classification which ends up giving us much better results. Therefore we used the same variety of classification algorithms (Random Forest, Logistic Regression, XGBoost, AdaBoost) and metrics (ROC AUC, Precision, Recall).

EXPERIMENTS/EVALUATION

We employed multiple testbeds for the usability of our user interface, model performance, API reliability and data quality. These are the facets that our experiments are designed to answer. We employed a mixed methods approach for each component, including qualitative and quantitative tests. For our UI, we conducted structured interviews with a diverse set of participants, including frequent flyers, aviation enthusiasts, and casual travelers. Additionally, we utilized surveys to collect data from a broader audience. This approach allowed us to gather in-depth feedback on the UI, the intuitiveness of the navigation, and the overall satisfaction with the provided information. Participants were asked to complete specific tasks, such as searching for a flight, interpreting delay predictions, and navigating through different sections of the website. This helped us identify any navigational challenges or information gaps. Incorrect inputs were a common issue, so we mitigate this by incorporating a dropdown for airline selection (rather than typing) and Regex on the backend to ensure that the flight number follows the correct format. The core predictive model underwent a comprehensive performance evaluation to ensure its accuracy and reliability. This evaluation involved splitting the data into training and testing sets, employing K-fold cross-validation for robustness, and utilizing various metrics such as accuracy, precision, recall, and ROC-AUC score for a comprehensive assessment. Our metrics for our 2 stage model can be shown in Figure 2. For both stages we found that our results were similar to state-of-the-art for accuracy. However, due to the nature of the unbalanced data, our precision is well but recall is poor. Basically, we don't predict many delays, but when we do we are right. We can alter thresholds to deal with this, but stage 2 fixes this problem. Buckets allow the middle-ground of delay or no delay to be handled. To test our APIs, we made multiple test requests over 3 weeks (in the 100s) of increasing number and frequency at a variety of times throughout the day (to test load handling), and evaluated our responses to ensure that we can authenticate and request the proper data reliably. Finally, data cleaning and pre-processing was a crucial aspect of our project. Removing duplicates, handling missing values, and correcting inconsistencies were some of the steps we took. We also normalized data formats, especially for date and time fields, to ensure uniformity. Additionally, we transformed categorical data, such as airline and airport codes, using one-hot encoding to make these fields more suitable for our ML algorithms. Continuous variables like flight durations and weather conditions were segmented into more meaningful ranges to enhance the model's interpretability. We also made use of the FuzzyWuzzy package and Levenshtein distance as a metric to map weather names to Bureau of Transportation Statistics airport names due to the mismatched naming schemes. Our final dataset includes 21,319 rows and 36 learnable feature columns. We also experimented with generating views of the table for faster lookup on our website. For example, we make another table that contains airport codes, latitude, longitude, and delay counts for our map heat map visualizations.

First Model
Accuracy: 0.7852
Precision: 0.7655
Recall: 0.1118
AUC of ROC: 0.73

Second Model
Accuracy: 0.7872
Precision: 0.8236
Recall: 0.7872
AUC of ROC (per bucket):
(0.71, 0.67, 0.67, 0.69, 0.64, 0.67)

Figure 2

CONCLUSION/DISCUSSION

Flight delay predictions are inherently a difficult challenge as there are a multitude of factors that make up a successful on-time takeoff. Weather conditions, air traffic, mechanical issues, airport operations, crew issues, or technical failures, to list a few, can cause these delays. To accurately predict them, we would need access to a huge number of data streams, with some of these streams such as aircraft and engineering data being infeasible to access in real-time. In addition, the aforementioned attributes themselves contain large uncertainty around them and are difficult to predict. Publicly available data is limited as we are unable to access data on crew productivity, air traffic control, and other internal information that all play a part in potentially creating delays.

Our team worked with the data available publicly, and successfully collected and analyzed how date, airline, and weather can affect flight delays. However, our work has a few limitations:

1) Delays are inherently hard to predict, the massive number of factors makes it difficult to arrive at a reliably high rate of prediction.
2) Our APIs need to return data that contains the same feature names as the historical data, which is a challenge.
3) Despite our models, handling unforeseen or rare events such as snowstorms, accidents etc. poses a challenge in ensuring reliable accuracy.

Looking ahead, we are invigorated by the potential for growth and improvement. Recognizing the limitations in forecasting unforeseen events, we are committed to further refining our predictive algorithms and enhancing our data analysis capabilities. Our future endeavors will focus on integrating advanced meteorological data and leveraging machine learning techniques to better anticipate and manage the impacts of such unpredictable scenarios.

We see these challenges not as setbacks but as opportunities for innovation. The goal remains clear: to empower travelers with the most accurate information and predictive insights, making air travel more predictable and less stressful.

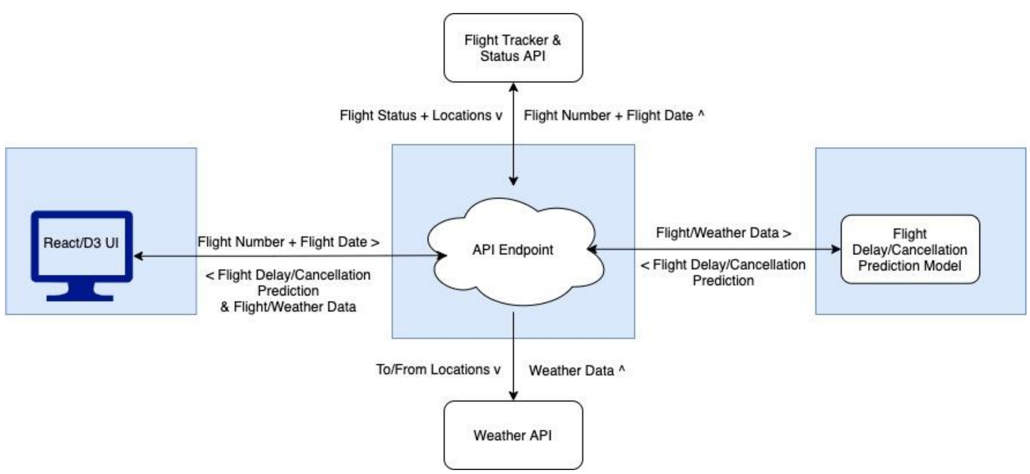
Our model training code can be found at this link here or in our Github.

All code can be accessed in the github repository: github.com/ninaadlakshman/AeroForecast and github.com/ninaadlakshman/AeroForecast-backend

**All team members have contributed a similar amount of effort.

APPENDIX

Figure 1



REFERENCES

Aljubairy, A., Zhang, W. E., Shemshadi, A., Mahmood, A., & Sheng, Q. Z. (2020). A system for effectively predicting flight delays based on IoT data. Computing, 102(9), 2025–2048. https://doi.org/10.1007/s00607-020-00794-w

Băicoianu, A., & Scheianu, I. V. (2023). Managing and Optimizing Big Data Workloads for On-Demand User Centric Reports. Big Data and Cognitive Computing, 7(2), 78–.https://doi.org/10.3390/bdcc7020078

Belcastro, L., Marozzo, F., Talia, D., & Trunfio, P. (2016). Using scalable data mining for predicting flight delays. ACM Transactions on Intelligent Systems and Technology, 8(1), 1–20. https://doi.org/10.1145/2888402

Cai, K., Li, Y., Fang, Y.-P., & Zhu, Y. (2022). A deep learning approach for flight delay prediction through time-evolving graphs. IEEE Transactions on Intelligent Trans portation Systems, 23(8), 11397–11407. https://doi.org/10.1109/tits.2021.3103502

Cai, K., Li, Y, Zhu, Y., Fang, Q., Yang, Y., &; DU, W. (2023). A geographical and operational deep graph convolutional approach for flight delay prediction. Chinese Journal of Aeronautics, 36(3), 357–367. https://doi.org/10.1016/j.cja.2022.10.004

Ding, Y. (2017). Predicting flight delay based on multiple linear regression. IOP Conference Series: Earth and Environmental Science, 81, 012198. https://doi.org/10.1088/1755-1315/81/1/012198

Glass, C., Davis, L., & Watkins-Lewis, K. (2022). A visualization and optimization of the impact of a severe weather disruption to an air transportation network. Computers & Industrial Engineering, 168, 107978.

Guvercin, M., Ferhatosmanoglu, N., & Gedik, B. (2021). Forecasting Flight Delays Using Clustered Models Based on Airport Networks. IEEE Transactions on Intelligent Transportation Systems, 22(5), 3179–3189. https://doi.org/10.1109/TITS.2020.2990960

Klein, T., Van Der Zwan, M., & Telea, A. (2014, January). Dynamic multiscale visualization of flight data. In 2014 International Conference on Computer Vision Theory and Applications (VISAPP) (Vol. 1, pp. 104-114). IEEE.

Liu, F., Sun, J., Liu, M., Yang, J., & Gui, G. (2020). Generalized flight delay prediction method using gradient boosting decision tree. 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring). https://doi.org/10.1109/vtc2020-spring48590.2020.9129110

Lundberg, S., Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. https://arxiv.org/abs/1705.07874

Netek, R., Brus, J., & Tomecka, O. (2019). Performance testing on marker clustering and heatmap visualization techniques: A comparative study on javascript mapping libraries. ISPRS International Journal of Geo-Information, 8(8), 348–. https://doi.org/10.3390/ijgi8080348

Qu, J., Zhao, T., Ye, M., Li, J., & Liu, C. (2020). Flight delay prediction using deep convolutional neural network based on fusion of Meteorological Data. Neural Processing Letters, 52(2), 1461–1484. https://doi.org/10.1007/s11063-020-10318-4

Rao, T. R., Mitra, P., Bhatt, R., & Goswami, A. (2019). The big data system, components, tools, and technologies: a survey. Knowledge and Information Systems, 60(3), 1165–1245. https://doi.org/10.1007/s10115-018-1248-0

Shao, W., Prabowo, A., Zhao, S., Koniusz, P., & Salim, F. D. (2022). Predicting flight delay with spatio-temporal trajectory convolutional network and Airport Situational Awareness Map. Neurocomputing, 472, 280–293. https://doi.org/10.1016/j.neucom.2021.04.136

Truong, D. (2021). Using causal machine learning for predicting the risk of flight delays in air transportation. Journal of Air Transport Management, 91, 101993–. https://doi.org/10.1016/j.jairtraman.2020.101993

Wang, C., Hu, M., Yang, L.,&; Zhao, Z. (2021). Prediction of air traffic delays: An agent-based model introducing refined parameter estimation methods. PLOS ONE, 16(4). https://doi.org/10.1371/journal.pone.0249754

Yazdi, M. F., Kamel, S. R., Chabok, S. J. M., & Kheirabadi, M. (2020). Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. Journal of Big Data, 7(1), 1–28. https://doi.org/10.1186/s40537-020-00380-z