Optimizing Airport Arrival Time

Iris Brook (irisb211@mit.edu)
Mackenzie Lees (mlees28@mit.edu)

December 2023

1 Introduction

In today's world, time is regarded as a critical resource, and the imperative for efficient travel planning, especially in air travel, has become paramount. A component of this is predicting and managing flight delays, an issue that impacts countless global travelers, airports, and airlines. This report delineates an advanced machine learning methodology to forecast flight delays and, more importantly, optimize airport arrival schedules. This dual-focused approach strives to achieve a balance between reducing waiting durations and mitigating the pressures associated with hurried airport arrivals.

The principal objectives of this project are two-fold. The initial aim involves the development of a predictive model utilizing Optimal Regression Trees. This model is designed to accurately categorize the amount by which a flight is delayed. The data used encompasses a variety of factors, such as meteorological conditions, airline-specific variables, and time-related factors. The secondary goal is to use Policy Trees to ascertain the best time for passengers to arrive at the airport. This decision-making model aims to offer recommendations that consider the forecasted flight delays and other factors, thereby improving the overall travel experience.

The innovative aspect of this project resides in the intersection of predictive analytics and decision-making frameworks. Although predicting flight delays using machine learning is not an unprecedented concept, using Policy Trees, specifically Optimal Policy Trees, for personalized airport arrival advice represents a relatively untapped area. This approach offers a more comprehensive solution to a frequent travel challenge, potentially transforming how passengers plan for airport arrivals.

The subsequent sections of this report will expand upon the methodologies implemented, encompassing aspects such as data collection, preprocessing, feature engineering, and the nuances of the employed modeling techniques. Following the methodological exposition, we will explain our findings, explore their broader implications, and offer practical, evidence-based recommendations from our analytical insights. The overarching goal of this project is to deliver a model that not only predicts with elevated precision but also converts these predictions into tangible, applicable guidance for people traveling.

2 Dataset Overview

The efficacy of our machine learning model in predicting flight delays and determining optimal airport arrival intervals hinges on the comprehensiveness and accuracy of our data sources. For this project, we meticulously curated a dataset from three primary sources, each contributing crucial aspects to our analysis. For our analysis, we focused on flights in March 2023 leaving from the Boston Logan International Airport.

2.1 Weather Data from the National Weather Service

Weather conditions are a fundamental factor in the intricate dynamics of flight scheduling and management. Our dataset, sourced from the National Weather Service's website [1], provides a range of weather metrics crucial for constructing a robust predictive model for flight delays.

This dataset has various weather variables, including temperature, precipitation, and visibility. Each of these elements plays a unique role in affecting flight schedules. For instance, temperature extremes can impact aircraft performance and runway conditions, while precipitation and visibility are directly linked to takeoff and landing safety protocols.

Incorporating this weather data into our predictive models offers a significant advantage. By analyzing historical weather patterns and their outcomes on flight schedules, we can train our models to recognize and anticipate similar patterns. This analysis not only enhances the accuracy of flight delay predictions but also provides valuable insights for airlines and airport authorities to proactively manage and mitigate the impact of adverse weather conditions on flight operations.

2.2 Flight Data from Massport Website

The flight-specific data set serves as a direct window into the dynamics of air travel, mainly focusing on the patterns and incidences of flight delays. Originating from Massachusetts Port Authority's official data [2], this dataset encompasses a

[1] `https://www.weather.gov/wrh/Climate?wfo=box`
[2] `https://www.massport.com/media/2oifruwd/0323-avstats-airport-traffic-summary.pdf`

range of flights departing from Boston Logan International Airport. As a significant hub, this airport provides a rich and diverse sample for analysis.

Key features within this dataset include flight numbers, which are unique identifiers for each flight and serve as a cornerstone for tracking and analyzing flight data. The scheduled and actual departure and arrival times are critical in quantifying delays and understanding their patterns. This temporal data helps not only identify the magnitude of delays but also discern patterns related to specific times of the day, days of the week, or seasonal variations.

Carrier information included in the dataset is also pivotal. Different airlines may have varying operational efficiencies and policies impacting their punctuality. By analyzing carrier-specific data, we can uncover trends and correlations between specific airlines and their propensity for delays, which can be a significant factor in predicting future delays.

Furthermore, the dataset includes details about flight destinations. This information is crucial as it allows us to consider the geographical and climatic variances that might affect flight schedules. For instance, flights to regions prone to severe weather might be more likely to be delayed.

2.3 Wait-Times from TSA Website

Airport security and baggage check-in wait times are essential in calculating optimal airport arrival intervals. These wait times significantly influence the time passengers must allocate for their airport journey, independent of the flight's punctuality. Utilizing data from the TSA [3], which provides detailed insights into average wait times per hour at Boston Logan International Airport's security checkpoints, enriches our model's capacity to predict the most efficient arrival times for passengers.

This dataset includes time-stamped information on security and baggage check-in checkpoint wait times, allowing for granular analysis of processing durations throughout the day and week. By correlating this data with flight schedules, we can identify patterns in busy and not busy periods at the airport.

Long-term analysis of this data can reveal trends and allow for predictive modeling of future wait times. This aspect is particularly pertinent in considering how changes in airport security protocols, staffing, and passenger volumes might impact wait times. For instance, increased security measures or reduced staffing might lead to longer wait times, necessitating adjustments in recommended airport arrival times.

2.4 Data Integration and Preprocessing

Integrating these diverse datasets, encompassing factors like weather conditions, wait times, and internal factors specific to each flight—provides a holistic view of the various elements influencing flight schedules and passenger experiences. This comprehensive approach is essential for developing a robust and accurate predictive model for optimizing airport arrival times.

Data Cleaning: This step involves identifying and rectifying errors or inconsistencies in the data, such as incorrect entries or outliers that might skew the analysis. Given the diverse nature of the data sources, this process was needed to maintain the integrity of the dataset.

Handling Missing Values: Missing data is common in large datasets, however, there were not many of these instances, so we decided to exclude incomplete records.

Encoding Categorical Variables: Many variables in these datasets are categorical (such as airline names, flight destinations, etc.). Encoding these into a numerical format was necessary for our machine-learning model to process them effectively.

Aligning Time Formats and Scales: We also had to ensure that the time-related data across different datasets aligned correctly. This involved standardizing time formats and scales (converting all times to a 24-hour format) to ensure consistency in our analysis.

In addition to these preprocessing steps, specific wait times were selected as benchmarks for assessing passenger experiences:

Baggage Check-In Wait Time: A time frame of 2 hours before the flight was selected for analyzing baggage check-in wait times. This choice was made because people typically consider checking their bags within this interval before their flight.

[3] `https://tsa.report/airport/bos/#wait-times`

Security Wait Time: For security wait times, a window of 1.5 hours before the scheduled flight time was chosen. This interval was selected to capture the most relevant period during which passengers typically go through security.

2.5 Feature Selection and Engineering

We selected and engineered features to comprehensively capture the dynamics influencing flight schedules and passenger experiences. The chosen features include variables such as the flight date, flight number, carrier code, and destination airport, which provide foundational context for each flight. Temporal variables like scheduled departure time and day of the week are incorporated to account for time-specific patterns in air travel. Weather-related features, including maximum and minimum temperatures, overall temperature, precipitation, and visibility, are necessary for assessing the impact of weather conditions on flight delays.

Additionally, we integrated operational aspects such as security wait time and baggage time, reflecting passengers' time in pre-flight processes. Delayed minutes, a direct measure of flight delays, offered insights into the flight's punctuality.

Regarding feature engineering, we introduced a weather severity column, which is defined as severe if precipitation exceeds two units or visibility drops below five. This helps in identifying weather conditions that significantly impact flight schedules. We also created treatment and outcome columns for the policy trees, where treatment refers to the proposed airport arrival time, and the outcome is a calculated measure considering the treatment, wait times, delayed minutes, and an additional 30-minute buffer for unforeseen factors.

We adjusted the delayed minutes variable to refine our approach, setting early departures to zero to ensure our model only accounts for actual delays. In cases of negative outcomes indicative of missed flights, we added a value of 90. This adjustment underscores that missing a flight is a more severe outcome than spending additional time at the airport. We also explored a range of treatment candidates, considering arrival times from 60 to 90 minutes before the flight in 5-minute increments to identify the most optimal arrival times.

3 Methodology and Models

This section discusses the methodologies and models we employed to achieve two key objectives: predicting flight delays and determining optimal airport arrival times. We experimented with and evaluated various modeling techniques, each chosen for their potential to unravel the complexities of air travel dynamics.

3.1 Predicting Delays

At first, we attempted to predict flight delays using gradient-boosted trees, Lasso regression, and Random Forest regressions. These models had significant discrepancies between predicted and actual delay times. Consequently, we tried out a binary classification approach, distinguishing flights as delayed or on time. Despite the improved performance in classification, the need for precise delay duration in minutes for optimal airport arrival time planning led us back to refining our regression models.

3.1.1 Baseline Model

First, we established a baseline model for predicting flight delays using a Decision Tree algorithm. We selected features pertinent to our analysis, including Precipitation, Visibility, Day of Week, Temperature, and Destination Airport, along with our target variable, Delayed Minutes. These features were deemed crucial in capturing the environmental, temporal, and operational factors influencing flight delays and, therefore, were used for the rest of the flight delay models. To avoid biases in model training and to ensure a randomized dataset, we shuffled the observations and randomly split the data into training and test sets in an 80-20 ratio.

To assess the model's predictive power and model's utility, we conducted a 3-fold cross-validation. These values were instrumental in evaluating the model's performance across different segments of the training data, offering a comprehensive understanding of its effectiveness. The baseline model's MSE, average from each fold, was 17.2, which is quite large.

Overall, this baseline model offered a preliminary gauge of predictive performance against which we could compare more advanced models explored in subsequent project sections. Its simplicity and interpretability were critical factors in its selection as the foundational model for our analysis. In contrast, the baseline model for policy trees simply assigned the same treatment for each flight, arriving at the airport 75 minutes before the flight.

3.1.2 Optimal Regression Trees

In the initial phase of predicting flight delays, we employed Optimal Regression Trees (ORTs), focusing on a subset of features specifically chosen for their relevance to flight delays.

To ensure robustness in our model, we divided our data into training and testing sets using a random split. For the modeling process, we initialized an Optimal Tree Regressor through Julia's Interpretable AI (IAI) framework. We chose this approach because it can generate interpretable models that efficiently and effectively handle regression tasks. To fine-tune our model and identify the best parameters, we utilized a grid search over various maximum depths ranging from 1 to 5. This range was selected to explore the trade-off between model complexity and overfitting, aiming to find an optimal depth that captures the underlying patterns in the data without being overly complex.

The utilization of OCTs in this context not only allowed us to predict delays accurately but also offered valuable insights into the relationships between different variables and flight delays.

3.1.3 Optimal Regression Trees with Hyperplanes

Following our initial exploration with Optimal Regression Trees (ORTs), we advanced our approach to predicting flight delays by incorporating ORTs with hyperplanes. This method evaluated whether including hyperplanes could enhance the model's predictive performance.

In our model, we set a parameter to consider all possible hyperplanes for each split. This allows the model to explore a more diverse set of splitting rules beyond the standard axis-aligned splits, thus potentially capturing more complex interactions between features. The grid search was again employed to fine-tune the model, this time exploring maximum depths ranging from 1 to 4. The reduced upper limit in maximum depth was chosen considering the increased complexity introduced by hyperplanes. By comparing the performance of this model against our initial ORT model, we aimed to determine the impact of hyperplanes on the model's accuracy and interpretability. This step was needed for our methodology, as it helped us understand whether the additional complexity introduced by hyperplanes translated into a significant enough improvement in predicting flight delays.

3.2 Predicting Optimal Airport Arrival Time

For our second model, we focused on using policy trees as a strategic approach for determining the most optimal times for passengers to arrive at the airport. The choice of policy trees for this task is based on their ability to make tailored recommendations based on various factors, effectively balancing the need for timely arrivals against minimizing unnecessary waiting, thus offering a nuanced solution to a common travel challenge.

3.2.1 Policy Trees

In our methodology for predicting optimal airport arrival times, we first developed a Policy Tree model using Python. Central to our model was the reward function within the Policy Tree, innovatively designed to calculate the reward for potential splits based on waiting time, the risk of missing flights, and weather severity, effectively balancing penalties against rewards for optimal arrival times.

In enhancing our methodology for the Policy Tree model, we further experimented with three sets of constants within our reward function, each representing a unique scenario in evaluating airport arrival times. Initially, we utilized a set of constants that offered a balanced approach to provide a well-rounded assessment of factors affecting airport arrivals, weighing penalties for missed flights and long waits against the rewards for arriving optimally. Next, we considered a scenario with a slightly higher penalty for missed flights, reflecting a more cautious approach. This adjustment was made to assess the impact of greater emphasis on avoiding missed flights, potentially at the expense of longer waiting times. Lastly, we explored a scenario highly sensitive to weather conditions, acknowledging their significant impact on flight schedules.

By varying these constants, we aimed to explore the model's performance under different priority settings, each reflecting a distinct perspective on balancing the risks of missed flights, the inconvenience of long waits, and the unpredictability of weather conditions. The Policy Tree was initialized with parameters for a maximum depth of 5 and minimum samples split of 2 to control its growth and complexity. We built the tree recursively based on the best splits identified.

Scenario	Missed Flight Penalty	Long Wait Penalty	Optimal Arrival Reward	Weather Severity Factor
Initial Constants	0.05	0.2	0.5	0.07
Balanced Approach	0.07	0.15	0.45	0.05
Weather Sensitive	0.05	0.1	0.4	0.1

Table 1: Constants for Different Scenarios in Policy Tree Model

3.2.2 Optimal Policy Trees

Next, we decided to implement Optimal Policy Trees (OPT), for which we utilized the IAI documentation on Optimal Policy Trees with Numeric Treatment[4].

Initially, we removed columns such as our treatment and outcome and other identifiers that do not contribute to prediction, like 'Date' and 'Flight Number.' The remaining features were our observations. We then randomly split the data 50:50 for training and test set.

Our treatments corresponded to arrival times before flights to the airport and outcomes related to the effectiveness of these interventions, adjusted for other factors(calculated as previously explained in 2.5). The range of potential treatment times is set between 60 to 90 minutes, incremented by 5 minutes, providing a set of candidate treatment times to evaluate.

We then used a reward estimator to quantify each treatment's benefit. This estimator uses a doubly robust method, combining propensity score estimation and outcome regression powered by random forest algorithms, to provide a balanced and unbiased estimate of the treatment effect. The propensity score helps balance the treatment groups to simulate a randomized trial scenario, while the outcome regression predicts the actual outcome of interest. The doubly robust method ensures that if the propensity or the outcome model is correctly specified, the estimator can still provide accurate reward predictions. Including minimum propensity values helped avoid overestimating rewards for treatments with low likelihood.

The reward estimator was then fitted to the training data, including finding the best fit for the given features and treatment times and predicting the reward for each instance in the training set. A grid search over policy trees with varying depths (2 to 7) is performed to identify the tree that best minimizes outcomes based on the calculated rewards.

Finally, with the policy tree trained, it can be used to prescribe optimal treatments on the training data. The prescribed treatments are then converted into numerical values for analysis. Predictions of outcomes and their rankings are also derived, providing insight into the expected effectiveness of each treatment. The model's predictive performance is further evaluated using the test data, predicting test rewards, and calculating the mean policy outcomes. The average difference between the policy and actual outcomes and the average test reward for a specific treatment time were calculated. These metrics provide a comprehensive view of the model's performance, highlighting its ability to prescribe effective treatment times and improve the adjusted outcomes.

4 Key Findings

4.1 Predicting Delays

Our initial models for predicting flight delays included Gradient-Boosted Trees, Lasso regression, and Random Forest regressions. These models were evaluated based on their Mean Squared Error (MSE) to assess the accuracy of the predicted delay times. However, they yielded unsatisfactory results with high MSE values, around 17, indicating that the predicted delay times were, on average, far from the actual delays.

4.1.1 Optimal Regression Trees

The Optimal Regression Tree (ORT) performed much better than the previous methods. Experimentation was done by incorporating and eliminating different combinations of features, and the best model resulted in an MSE of 0.08539 for the training set and 0.06596 for the test set.

[4] https://docs.interpretable.ai/stable/OptimalTrees/quickstart/policy_categorical/

As seen in 1a, the ORT model identified the Destination Airport as the primary factor influencing flight delays. Airports coded with an index below 8.5 generally experience shorter delays, whereas those with an index above this threshold are associated with longer delays. This suggests a systematic variance in delay times across different airports. Precipitation also emerged as a significant variable, with the model showing increased delays when precipitation exceeds 0.275, confirming the adverse effects of weather on flight punctuality. The Day of Week feature indicated that delays vary within the week, with a split at the threshold of 4.5, somewhat differentiating between weekdays and weekends. Temperature was another contributing factor, with the model indicating that extremes in temperature (both hot and cold) are likely to increase delay duration.

4.1.2 Optimal Regression Trees with Hyperplanes

We also experimented with Optimal Regression trees with HyperPlanes. As seen in 1b, the decision tree model uses the 'Destination Airport' as a main factor in predicting flight delays, with certain airports more likely to experience longer delays. The model intricately combines temperature and precipitation and their interactions with the destination airport. This suggests that these weather-related factors have a multiplicative effect on delays that vary by location. Additionally, Day of Week is integrated into the model with weather and airport variables, indicating that the flight timing within the week may interact with these conditions to influence delay times. This model sets thresholds for continuous variables such as temperature and precipitation, differentiating delay profiles for flights above or below these values. The following figure shows each variable and its corresponding coefficient.

Variable	**Coefficient**
Precipitation	0.1131
Temperature	-0.004598
Destination Airport	0.02044

Table 2: Optimal Regression Tree with Hyperplanes Variable Coefficients Matrix

The model also details each node's mean delays and sample sizes, providing a robust evidentiary basis for its predictions. However, including complex multi-factor splits, adds a layer of complexity to the model's interpretability.

This model took more time to compute and resulted in a very slightly better MSE of 0.06418. Due to this, in practice, we will use ORTs with parallel splits, which produce about the same result in significantly less time.

4.2 Predicting Optimal Arrival Time

In this project, applying policy trees was pivotal in optimizing airport arrival times, offering clear and interpretable results. This model allowed us to navigate the multifaceted aspects of airport operations, considering variables such as wait times, day of the week, and weather conditions. We were able to provide tailored arrival time recommendations, aligning with varied traveler scenarios and dynamically changing airport conditions. Furthermore, the structured nature of policy trees enabled us to identify and focus on critical factors affecting airport arrival times, ensuring alignment with our goal of reducing passengers' time at the airport. The clarity and data-driven approach of policy trees significantly contributed to making practical and trustworthy recommendations, ultimately enhancing the travel experience by offering optimized arrival times.

4.2.1 Policy Trees

In the results section of our policy tree analysis, we explored three distinct scenarios – initial, balanced with a high penalty for missing a flight, and weather-sensitive –each tailored with different reward constants to predict optimal airport arrival times. The analysis of the policy trees in each scenario, while anchored on the Carrier Code, also revealed the influence of other distinct features, with each tree exhibiting its unique splits and corresponding Mean Squared Error (MSE) values that measure the accuracy of the predictions.

In the initial scenario, the policy tree, 1c, which resulted in an MSE of 0.0319, created splits based not only on the Carrier Code but also on environmental factors like Visibility and Temperature. These features indicate the significance of current weather conditions on travel times, suggesting that adverse weather might lead to longer wait times at the airport or delayed flights.

The balanced scenario tree, 1d, with a slightly improved MSE of 0.0315, frequently utilized Carrier Code for its decisions. However, it also considered Destination Airport and Visibility. This reflects an understanding that the destination can affect departure times, potentially due to factors such as flight length or different airport procedures.

The policy tree's MSE increased to 0.318 in the weather-sensitive scenario, 2a, signaling a less accurate fit than the other models. Despite this, the model's heightened sensitivity to weather was evident, with Precipitation emerging as a critical feature alongside Carrier Code. This model's increased focus on weather highlights the crucial impact of weather conditions on flight delays and, consequently, airport arrival time recommendations.

Across all scenarios, the repeated selection of Carrier Code suggests that either the airline's operational characteristics dominate in determining arrival times or the reward function is not granular enough to capture the importance of all features fully. Yet, including weather and destination data points add to a complex interplay of factors that travelers and airlines must navigate. These results reinforce the need to consider a comprehensive range of variables to optimize travel plans effectively. The variance in MSE for each scenario further emphasizes the challenge of balancing these factors to improve the precision of the predictive model.

4.2.2 Optimal Policy Trees

The Optimal Policy Tree in 2b utilizes Security Wait Time and Baggage Check in Time as key features for making decisions, pinpointing Security Wait Time as a primary factor. The tree differentiates between instances where the security wait is less than 17.7 minutes and those where it is more. This split highlights the weight given to security wait times in predicting and addressing flight delays.

For instances where security wait times are under 17.7 minutes, the model further categorizes data based on the wait time and Day of Week. It prescribes different actions if the security wait is less than 15.52 minutes, with additional branching based on the days of the week. This suggests that the model considers the duration of wait times and weekly travel patterns, which can influence the prescribed actions.

Conversely, for higher security wait times exceeding 17.7 minutes, the model considers Baggage check in Time to determine its prescriptions. A key decision point is at the 11.25-minute mark for baggage check-in time, with the model offering varied recommendations for times below and above this threshold. Understandably, the model prescribes more drastic actions (higher prescription treatments) when both security and baggage check-in times are long.

The mean of the policy outcomes was 10.45 compared to a baseline of always prescribing the 75-minute treatment, which was 21.6156. This shows that the optimal policy tree performed better than the baseline as we tried to minimize outcomes. This tree can be used as a real-time decision-making tool for airport operations, designed to streamline processes and enhance passenger experience by proactively reducing time in the airport without missing a flight.

An example usage of this project could be the following. You have a flight on a Friday at 5:00 pm, there's 0 precipitation, 5.2 visibility, the temperature is 35 degrees, and you're flying to the Denver airport(index 30). Based on this information, your flight will be delayed by 3.065 minutes. Taking that into account, along with calculating security wait time(15.9) and baggage check in time(13.8) based on the departure time, we can figure out the best time to arrive at the airport. Based on our model, you should get to the airport 65 minutes before your flight to minimize wait time in the airport without missing the flight.

5 Impact

Implementing the Optimal Policy Tree model has demonstrated a notable improvement in predicting optimal airport arrival times, as evidenced by a mean outcome surpassing the baseline prescription of a 75-minute treatment. This enhancement demonstrates the model's efficacy and serves as a benchmark for operational efficiency, setting a new standard for airport time management strategies.

Capitalizing on this advanced predictive capability, airports should prioritize the management of security wait times. The model has highlighted Security Wait Time as a crucial determinant, suggesting that more efficient screening processes could significantly enhance passenger flow. By adapting to real-time data, airports can offer dynamic and precise arrival time recommendations, improving the overall passenger experience.

The variability captured by the model in wait times across different days of the week suggests that airports could benefit from day-specific operational plans. These plans could involve tailored staffing and resource allocation to manage the expected fluctuation in passenger numbers. Moreover, enhancing baggage handling procedures, particularly during extended security wait times, could further streamline the travel process, as indicated by the model's emphasis on Baggage Check-In Time.

The policy tree's actionable insights offer a foundation for proactive communication with passengers. By informing travelers of the best times to arrive at the airport based on up-to-the-minute data, airports can facilitate better travel planning and reduce the likelihood of missed flights. This level of communication can be achieved through direct engagement via mobile applications or airport display systems. Furthermore, the insights from the policy tree model should inform staff scheduling, ensuring that airport personnel are present when most needed. This optimized allocation of human resources can mitigate bottlenecks and enhance the efficiency of airport operations. Regular evaluation of the model's recommendations against actual outcomes is also essential, enabling airports to maintain the accuracy and effectiveness of the predictive tool.

Lastly, the policy tree model can assist airports in developing contingency plans for critical wait time thresholds, ensuring preparedness for peak travel periods. By acting on these recommendations, airports, and airlines can leverage the Optimal Policy Tree model to not only refine their operations but also to offer a predictably smooth and stress-free travel experience.

6 Possible Extensions

In considering future enhancements to our current model, several promising avenues for extension can further refine its utility and impact:

Development of Passenger-Facing Applications: A practical extension would be developing a user-friendly application or an interactive dashboard. This tool would enable passengers to input their flight details and receive personalized recommendations for their optimal airport arrival times. Integrating such an application would bridge the gap between the model's predictive power and passenger usage, making the insights accessible and actionable for the end-user.

Integration of Real-Time Data: To enhance the model's responsiveness, an extension could involve integrating real-time data feeds for weather and security wait times, potentially sourced from the Transportation Security Administration's website[5]. The ability to process live data would allow the model to adjust its recommendations based on current conditions, thus improving the accuracy and relevance of its predictions.

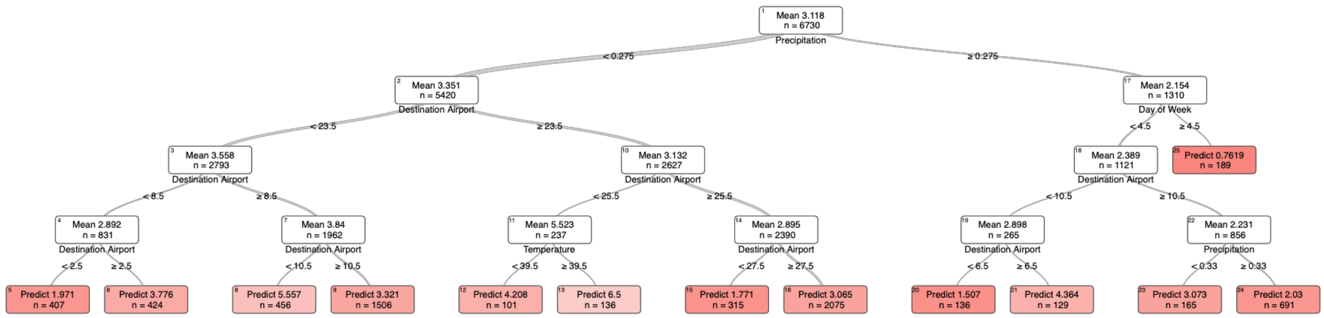
Conducting a Case Study: A qualitative extension could involve a case study where passenger stress levels are observed and measured in response to different arrival times. This human-centered approach would allow assessing the model's impact on passenger experience. Stress levels could be used as a direct outcome measure, potentially providing a more tangible indication of the model's effectiveness in improving passenger experience compared to calculated wait times.

By implementing these extensions, the project could significantly broaden the scope and utility of the model, ensuring that it not only remains technically robust but becomes more integrated into the passenger travel experience, ultimately leading to enhanced satisfaction and efficiency.

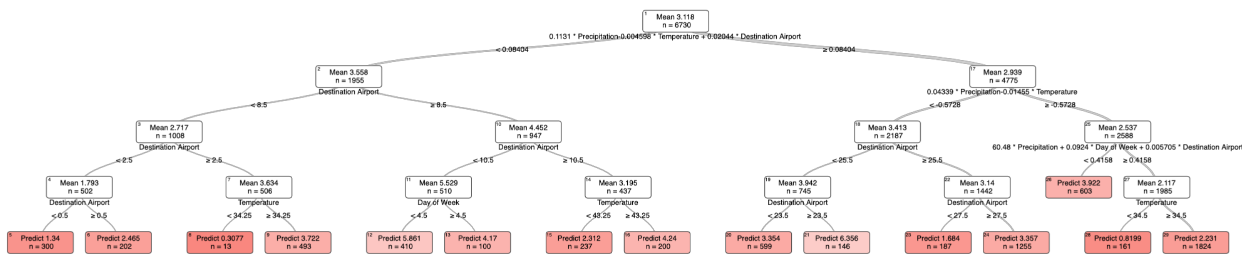
[5]`https://www.tsa.gov/mobile`

Appendix

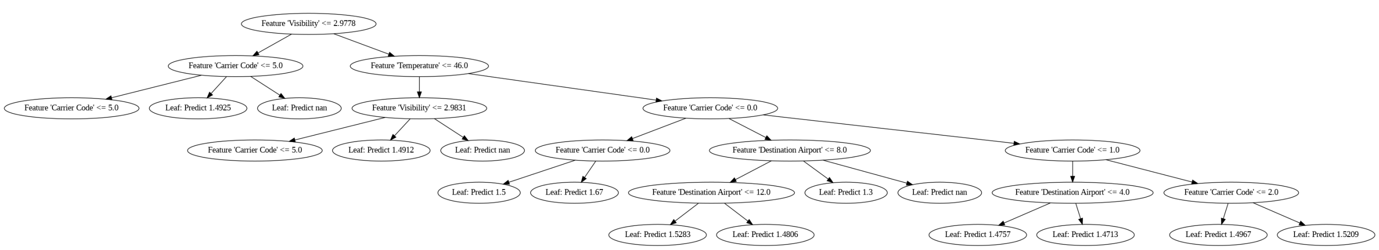
Figures



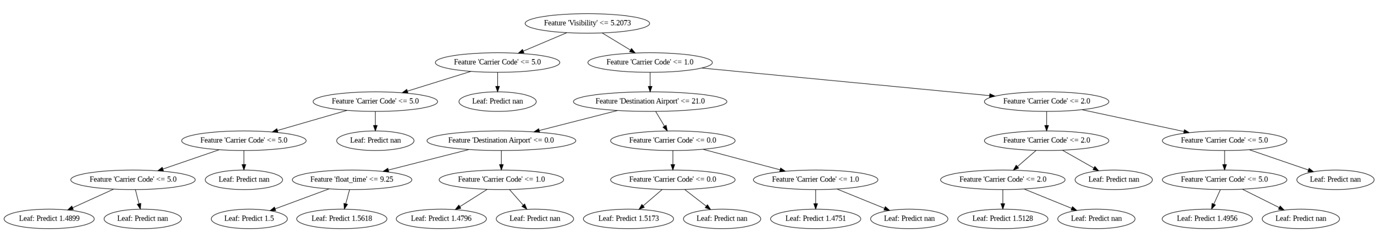
(a) Optimal Regression Tree



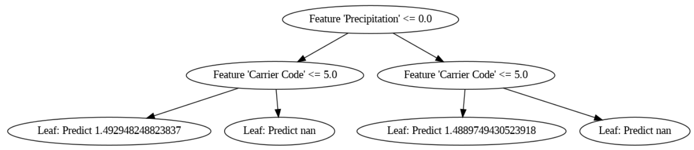
(b) Optimal Regression Tree with Hyperplanes



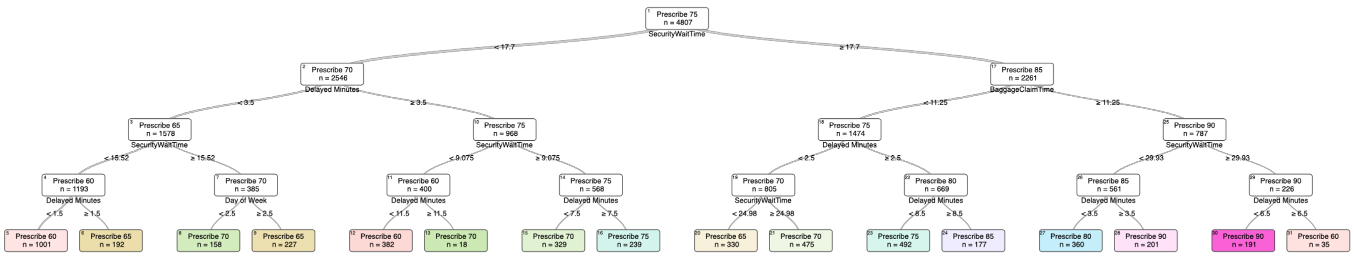
(c) Policy Tree with Initial Inputs



(d) Policy Tree with Balanced Inputs with Higher Penalty for Missed Flights



(a) Policy Tree Highly Sensitive to Weather Severity



(b) Optimal Policy Tree

Partner Contribution

Mackenzie focused on utilizing the tools provided by Interpretable AI (IAI), researching and understanding the topics in depth before implementing them. In addition, Mackenzie worked on data cleaning and feature engineering needed for analysis. She specifically worked on developing the Baseline decision tree, the Optimal Regression Tree, and the Optimal Policy Tree model. These approaches allowed us to model our data with a high degree of accuracy and interoperability.

Iris, on the other hand, focused on implementing policy trees in Python. She also contributed to data cleaning and making it suitable for her own analysis. Aside from our individual technical contributions, we both contributed equally to the writing of this paper and presentation.

Overall, our contributions to the project were equal.