Systems Analysis, Behavior and Optimization for Airport Operations

Yulong Quan (yq284), Ning Xie (nx49), Zihan Nie (zn39)
Jiaxin Zhang (jz2289), Jinglin Wang (jw2745)

May.10.2024
SYSEN 5200 Final Project

Cornell University

Abstract: This project delves into the complex dynamics of airport operations, focusing on systems analysis and optimization across four critical areas: reliability of aircraft landings, airport check-in and security processing, cargo operations, and aviation accident analysis. Each section utilizes distinct methodological frameworks to enhance operational efficiency and safety. Our findings reveal significant insights, such as the effectiveness of dynamic speed control in reducing landing accidents and strategies to manage check-in and security processes under varying customer arrival rates. Moreover, we developed an optimization model for cargo operations that minimizes transportation and idle costs, demonstrating potential savings of up to $19,825. Lastly, our analysis of aviation accidents employs a robust predictive model, achieving an accuracy of 83.3%, which identifies key factors influencing accident severity. These integrated approaches provide actionable recommendations for improving airport management and safety protocols, thereby enhancing overall operational efficacy and risk mitigation in airport environments.

Table of Contents

Q1: Reliability analysis of accident risk during aircraft landing

1. Introduction

Aircraft landings are critical phases of flight where the risk of accidents is significantly influenced by the spacing between consecutive aircraft, known as in-trail separation. Insufficient separation can lead to dangerous wake vortex encounters or simultaneous runway occupancy, both of which can cause severe accidents. The graph below briefly explains the factors that can lead to insufficient in-trail separation.

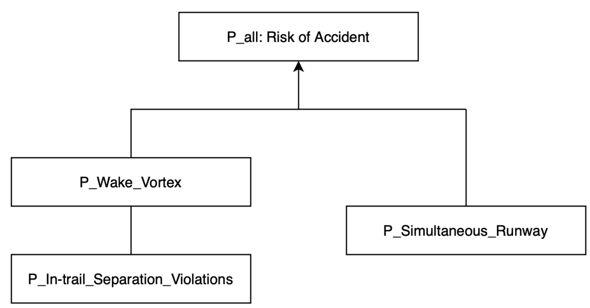

Figure 1: Factors lead to in-trail separation.

Our primary objective is to conduct a reliability analysis to estimate the risk of accident due to insufficient in-trail separation. We will calculate and compare the overall accident risk when the weather is good versus when the weather is bad.

Our secondary objective is to determine the tradeoff of applying a new dynamic speed control system. With different tradeoff levels, the risk of certain types of accident will increase or decrease accordingly. We will explore the level of new technology that should be utilized such that the overall accident risk due to insufficient in-trail separation is minimized. We will also investigate the influence of future weather conditions on the optimal tradeoff in the new technology.

2. Problem Modeling

From figure 1, we observe that the risk of insufficient in-trail separation can be calculated by:

$$P_{Insufficient\ in-trail\ separation} = P_{Wake\ vortex} * P_{in-trail\ separation\ violation} + P_{Simultaneous\ runway}$$

And we will investigate how to calculate the probability of the appearance of these three accidents.

Wake vortex encounters: We are given that the proportion of arriving aircraft by size is approximately 33% heavy, 46% large, and 21% small. And the probability of dangerous wake vortex with different size of lead and in-trail plane are shown in table 1. Denote that P_HH = P(Lead plane is heavy and In-Trail plane is heavy) = 0.53 and P_h = P(Plane is heavy) = 0.33, we can calculate P_Wake_Vortex by:

$$P_{Wake\,Vortex} = P_{HH} * P_h * P_h + P_{HL} * P_h * P_L + ... + P_{SL} * P_s * P_l + P_{SS} * P_s * P_s$$

In-trail Separation Violations: We are given the probabilities of rare events caused in-trail separation violation under different weather conditions, which contribute to wake vortex encounter. Let us denote P_1g = P(Misidentification of leading or trailing aircraft under good weather conditions) and P_9b = P(Air traffic control instruments will fail to detect separation violations under bad weather conditions). Then by description, we can calculate P_in-trail_separation_violations under good and bad weather by:

$$p\_g = ((p\_1g * p\_2g + p\_3g + p\_4g) + (p\_5g * p\_6g) + (p\_7g * p\_8g)) * p\_9g$$
$$p\_b = ((p\_1b * p\_2b + p\_3b + p\_4b) + (p\_5b * p\_6b) + (p\_7b * p\_8b)) * p\_9b$$

Simultaneous runway occupancy: We are given the probabilities of the rare events contributing to simultaneous runway occupancy under different weather conditions. Let's denote P_10g = P(Equipment failure prevents lead aircraft from leaving the runway under good weather). Then by description, we can calculate P_in-trail_separation_violations under good and bad weather by:

$$p\_simultaneous\_runway\_g = (p\_10g + p\_11g) * (p\_12g + p\_13g)$$
$$p\_simultaneous\_runway\_b = (p\_10b + p\_11b) * (p\_12b + p\_13b)$$

The relationship between different events that contribute to the wake vortex and simultaneous runway is shown in the graph below for better understanding.

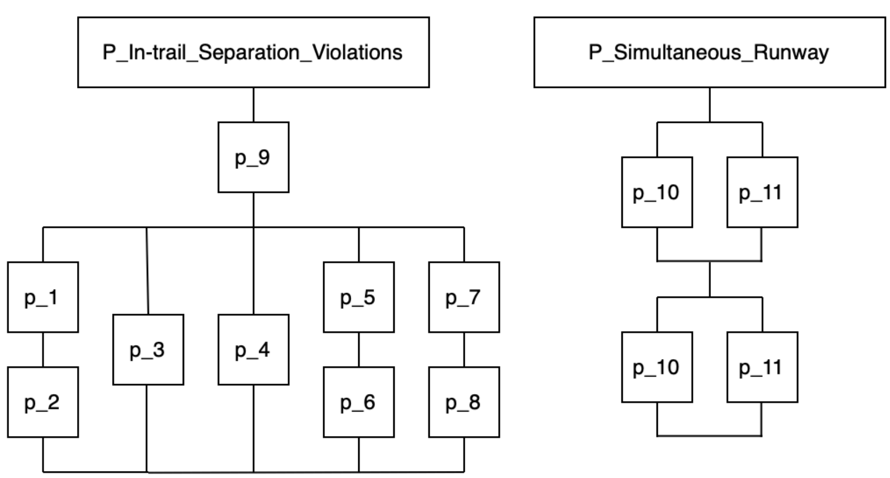

Figure 1: Relationship between different events for the wake vortex and simultaneous runway

Evaluating a new technology:

The tradeoff of applying this new technology includes two parts: Decrease in Probability that level pilot is unable to adjust and Increase in Probability that go-around cannot be initiated. The decreased responses to p_5 and the increased responses to p_12. Hence, we update the probability of p_5 and p_12 by:

$$P_5 new = p_5 - decrease\_p, \; P_{12} new = p_{12} + increase\_p$$

And then we recalculate the risk of Insufficient in-trail separation for each tradeoff level of the new technology under the condition that 80% of the time the weather is good and 20% of the time the weather is bad. As a result, we find the tradeoff level that minimizes the risk of Insufficient in-trail separation.

3. Result

The probability of each risk event under good and bad weather conditions is shown below.

Event	Probability
p_wake_vortex	0.576
p_in_trail_separation_distance_good_weather	1.596e-07
p_in_trail_separation_distance_bad_weather	2.155e-07
p_simultaneous_runway_good_weather	3.111e-07
p_simultaneous_runway_bad_weather	3.119e-06
p_insufficient_in-trail_separation_good_weather	4.031e-07
p_insufficient_in-trail_separation_bad_weather	3.244e-06

Table 2: Probabilities of Events

And the best trade level of the new technology to minimize the probability of insufficient in-trail separation is level 1 with probability of 9.6994e-07.

Q2: Simulation of airport check-in and security lines

1. Problem Overview

The problem involves exploring a queueing system at an airport with two main service stages: a check-in line and a security line. The primary focus is on understanding how the system behaves with variations in the arrival rate λ , particularly identifying periods of stability and potential instability where the system might become overwhelmed.

2. Problem Modeling

Service Process:

Check-in Process: Customers arrive and are serviced at one of the available check-in counters. The service time is lognormally distributed, which typically models time variables that cannot be negative and are skewed to the right.

Commute to Security: After check-in, customers commute to the security area. This time is modeled with a normal distribution, adjusted for non-negative times.

Security Screening: At security, customers are processed in an exponentially distributed time frame per available security station.

Statistical Metrics:

Average Number of People in System (L):

$$L = \frac{(number\_checkin + number\_commuting + number\_security)}{total\_simulation\_time}$$

Average Time in System (W):

$$W = \frac{(waiting\_time + service\_time + commute\_time)}{number\_completed}$$

Throughput (S):

$$S = \frac{number\_completed}{total\_simulation\_time}$$

Definition of Key Concepts and Variables:

Check-In Queue: Modeled as a multi-server queue where service times are log-normally distributed, reflecting the positive skewness typical of real-world service operations.

Security Screening Queue: A multi-server queue but with exponentially distributed service times, suitable for high-volume, memoryless service processes.

λ: Dynamically varying arrival rate of customers, influencing system input.

S_1: Log-normally distributed service time at check-in.

S_2: Normally distributed commute time between check-in and security, recalculated if negative to ensure physical realism.

S_3: Exponentially distributed service time at security, facilitating fast, memoryless processing.

n_1, n_3: Count of service stations at check-in and security, respectively.

L, W, S: Key performance metrics - average system occupancy, average time in system, and system throughput.

Simulation:

The simulation outputs detailed statistics on L, W, and S, which are analyzed to assess system performance, identify bottlenecks, and explore the impact of different levels of customer arrivals on system stability. The event-driven simulation follows these steps:

Initialization: Set simulation clock to zero, with no customers initially present.

Customer Arrival: Generate inter-arrival times based on λ, using an exponential distribution to model the time between arrivals.

Service Assignment: As customers arrive, assign them to available service stations or place them in queues: Check-in services use log-normal distribution for service times; Security uses exponential distribution.

Queue Management: Customers transition between services and are queued as necessary, based on station availability.

Completion and Departure: Track completions at each stage and log departures from the system.

State Updates and Metric Tracking: Continuously update system state and record data for performance metrics.

3. Result

Task 1.1:

To explore the relationship between the queuing system's performance measures and the arrival rate λ, we set up a discrete-event simulation using the SimPy package in Python. The system was modeled with a certain number of check-in (n1=25) and security (n3 = 10) stations, processing customers with lognormal and exponential service times respectively. The simulation was run for a period equivalent to 24 hours. To mitigate randomness and estimate these performance measures with higher accuracy, a Monte Carlo approach with 20 iterations for each arrival rate was employed.

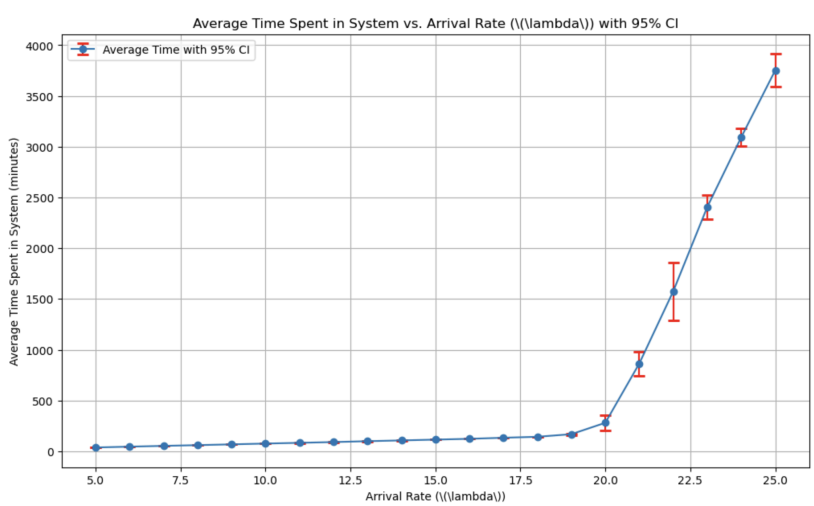

Figure 1 The average number of people in the system (L) as a function of the arrival rate

It is observed that the system remains stable with a gradual increase in L up to an arrival rate of about $\lambda \approx 20$. Beyond this threshold, there is a sharp increase, indicating the system's transition into an unstable state where the average number of people in the system escalates rapidly.
We validate this hypothesis by running the model longer for lambda = 20 and 21. We observe that L blows up for a longer time horizon when lambda = 21.

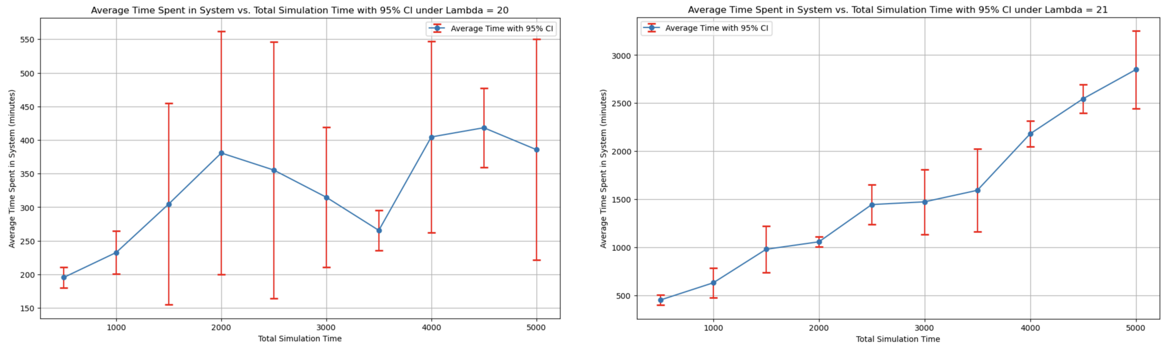

Figure 2: The average number of people in the system (L) for lambda = 20 and 21

Task 1.2:

We find a linear relationship between the average time a customer spends in the system L/Wand the arrival rate λ. This suggests that as more customers arrive per unit of time, the average time each customer spends in the system increases proportionally. This also align with the famous “Little’s Law” in queueing theory, that long term speaking, L = lambda * W.

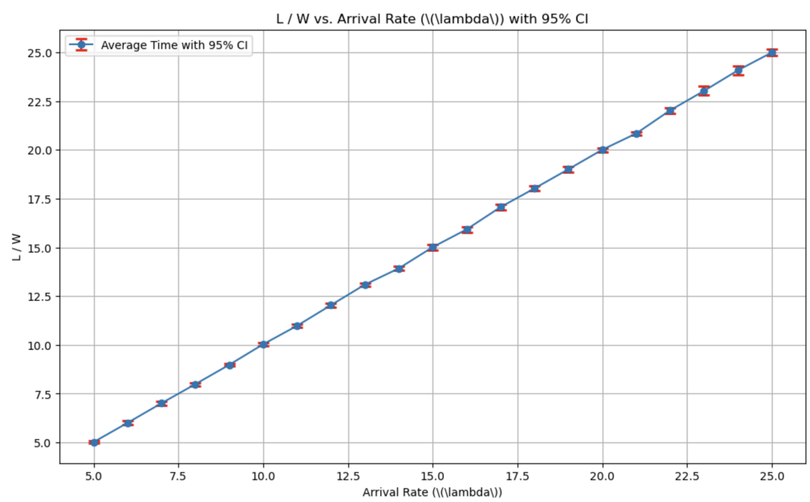

Figure 3: L/W vs. lambda

Task 2.1 & 2.2:

From figure 4, we observe that when n3 >= 6, the system becomes stable. The average S becomes stable when n3 >= 6. This observation is aligned with the conclusion we draw from figure 4.

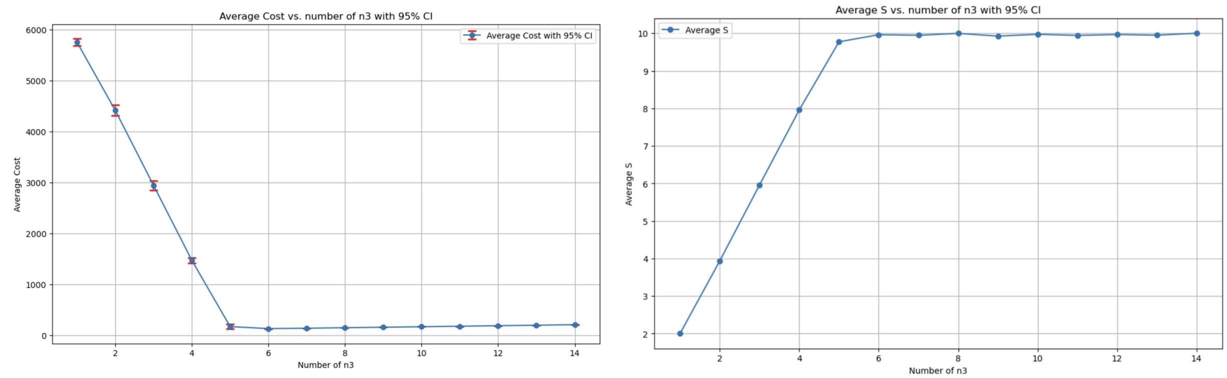

Figure 4 & 5: Number of screening station vs average cost and S

Task 2.3:

The average cost is minimized when n3 = 6.

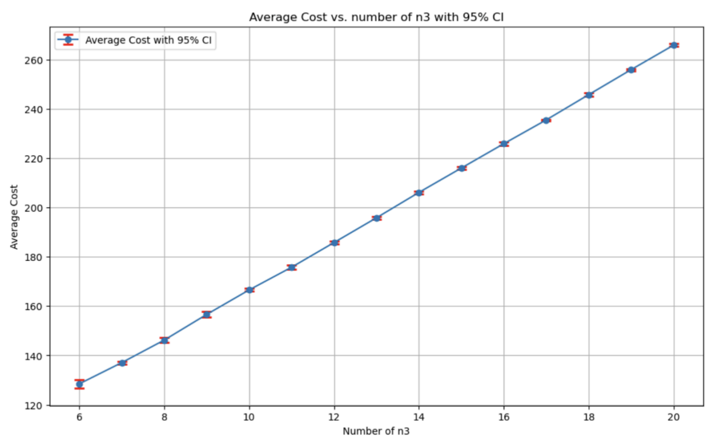

Figure 6: Number of screening station vs Average Cost

Q3: Optimization of cargo operations

1. Introduction

This section presents the formulation and solution of an optimization model designed to minimize the transportation and idle costs associated with airplane and goods transportation between three airports labeled as A, B, and C. The objective is to find an optimal schedule that meets operational constraints while minimizing the total cost. The objective function is designed to minimize the operational and idling costs associated with transporting goods:

$$Minimize: \sum_{day \in Days} \sum_{pair \in Pairs} \left(transport\_cost_{pair} \cdot y_{day,pair} + idle\_cost\_per\_goods \cdot z_{day,pair} \right)$$

This function aims to minimize the total transportation cost, which includes costs for using excess planes (y) and holding idle goods (z), across all days and airport pairs.

2. Problem Modeling

Total Airplane Constraint:

$$\sum_{airport \in Airports} i_{airport} = 1200$$

This constraint ensures that the total initial number of planes across all airports equals 1200.

Cumulative Goods Demand:

$$\sum_{day \in Days} x_{day,pair} \geq \sum_{day \in Days} d_{day,pair}$$

Each route's cumulative weekly cargo transport by planes must meet or exceed the total weekly demand specified for that route.

Daily calculation of excess planes and idle goods for each day and pair:

$$y_{day,pair} \geq x_{day,pair} - required\_daily\ z_{day,pair} \geq required\_daily - x_{day,pair}$$

These constraints account for the difference between the number of planes available and required for daily operations (excess planes) and the mismatch between supply and demand leading to idle goods.

Weekend Airplane Restoration:

$$init_{airport} + \sum_{day \in Days} \sum_{k \in Airports} (x_{day,(k,airport)} + y_{day,(k,airport)} - x_{day,(airport,k)} - y_{day,(airport,k)}) = init_{airport}$$

This constraint ensures that the number of planes at each airport is restored to its initial value by the end of the week, balancing arrivals and departures throughout the week.

3. Result

The optimization problem was solved with a minimum cost of $19,825. The daily scheduling of flights and the count of airplanes at each airport are presented in the tables 1 below.

Origin-Destination	Monday	Tuesday	Wednesday	Thursday	Friday
A-B	100	200	100	400	300
A-C	50	50	50	50	50
B-A	210	145	375	310	110
B-C	300	455	115	90	490
C-A	40	40	40	40	40
C-B	500	290	300	200	210

Table 1 Daily schedule of flights

Table 1 reveals specific patterns and variances in flight schedules that may be aligned with business needs, logistical requirements, or market demands. Understanding these trends is crucial for optimizing operations, improving service delivery, and potentially reducing costs through better resource management. The route B-C shows significantly high traffic on Tuesdays and Fridays, suggesting a peak demand for services between these locations at the start and end of the work week. Conversely, the B-A route exhibits a sharp peak on Wednesday, which might indicate mid-week supply or personnel shifts. The A-C router maintains a steady flow of 50 flights daily, indicating a consistent demand or a fixed contract for services throughout the week. Routes from A-B and C-B exhibit varied intensity, with the highest volumes towards the latter part of the week, particularly on Thursday and Friday for A-B. This pattern could reflect an accumulation of demand or strategic planning aligned with business or market activities that escalate towards the week's end. Furthermore, to analyze and verify the rationality of the scheme, the changes in the number of aircraft before and after optimal flight scheduling at each airport are plotted, as shown in Table 2.

Airport	Monday		Tuesday		Wednesday		Thursday		Friday		Next Monday	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
A	150	0	250	0	185	35	450	0	350	0	150	0
B	510	0	600	0	490	0	400	0	600	0	510	0
C	540	0	350	20	525	185	350	110	250	0	540	0

Table 2 Daily count of airplanes at each airport

From table 2, the complete utilization of planes at Airports A and B for most of the week could point to a need for careful management of plane maintenance and readiness to ensure sustainability. Meanwhile, the variability at Airport C requires robust flexibility in operations to adapt to the differing daily demands. Each airport starts with a certain number of airplanes every day but ends up with significantly fewer, or zero, indicating a full deployment of available resources.

Q4: Analysis of aviation accidents

1 Introduction

In the contemporary landscape of global aviation, maintaining and enhancing safety standards is not merely a regulatory requirement but a fundamental business imperative. Aviation accidents, although rare, can have catastrophic consequences including loss of life, significant financial liabilities, and damage to organizational reputation. The motivation for this study originates from a proactive approach to further mitigate these risks through sophisticated data analysis techniques. By leveraging historical accident data, this project aims to build predictive models that identify the severity of potential future accidents and key risk factors, thereby enabling more precise risk management strategies. The main objective of this analysis is to develop a robust predictive model that can accurately determine the severity of aviation accidents based on a comprehensive set of variables captured from past incidents. The specific objectives of the study include:

Determining Impact Factors: To identify and quantify the impact of various factors like aircraft model, weather conditions, and flight operations on the severity of accidents.

Actionable Insights for Risk Reduction: By identifying these key factors, we will provide actionable insights that can directly reduce the frequency and impact of future accidents.

2 Methodology

2.1 Data

Our dataset is collected by NTSB (National Transportation Safety Board). It was chosen due to its comprehensive coverage of significant factors that could influence the severity and outcomes of aviation accidents. It includes records of aviation accidents in the United States from 1982 to 2022. There are 31 columns and 68566 rows. The full description of columns can be found here.

2.2 Exploratory Data Analysis

2.2.1 Data Visualization

Prior to analysis, we visualize some key observations to better understand our dataset. Over the years, the number of total aviation accidents has been decreasing. And the board phase of flight with most accidents is the landing phase.

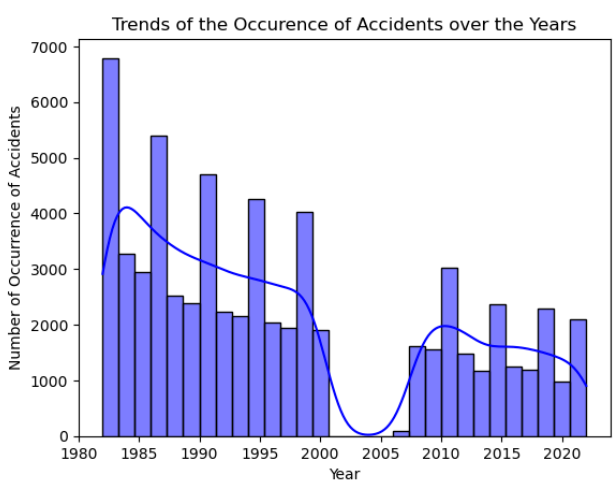

Figure 1: Trends of Aviation Accidents

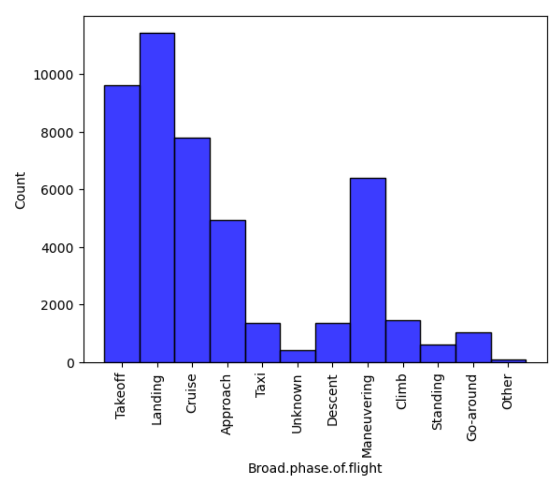

Figure 2: Accidents Distribution over Broad Phase

2.2.2 Drop columns with no predictive power

To begin with, our dataset contains columns that serve as a unique identifier or administrative details of the accident. We will drop these columns to simplify our model.

#	Column	Example	Description
1	Event.Id	20020909X01562	Unique Identifiers.
2	Accident.Number	SEA82DA022	Unique Identifiers.
3	Registration.Number	N2482N	Unique Identifiers.
4	Investigation.Type	Accident	Administrative details.
5	Report.Status	Probable Cause	Administrative details.
6	Publication.Date	01-01-1982	Administrative details.

2.2.3 Handling Missing Value

Prior to analysis, the dataset also underwent preprocessing to handle missing values and to ensure consistency in formatting, particularly with dates and categorical variables.

Approach 1 Feature Transformation: For the columns 'Airport.Code', 'Airport.Name', we think it would influence the severity of the accident. If this accident happened within 3 miles from the airport, it would have a better chance to cause more damage since the airport area tends to have a higher population density. Since Missing values in Airport.Code and Airport.Name suggest the accident did not occur near an airport within 3 miles, we decide to convert them into a single binary column called 'Near_Airport' indicating whether an accident occurred within 3 miles of an airport.

Approach 2 Filling with data transforming from other columns: Our columns Year, Month, Day, and Weekend contain lots of missing values. While the column “Event.Date”, with the format MM/DD/YYYY indicating the day that accident happened, has no missing values. We can convert 'Event.Date' to datetime to extract year, month, day, and check for weekends. So that we can fill in the missing value of these time columns. Similarly, we also extract the state information of the column “Location” to fill in the missing value of the column “State”.

Approach 3 Filling with Mode of the column: The values of “Weather.Condition”, “Aircraft.Damage”, “Engine.Type” are dominated by one specific value. Hence, we can fill the missing value with the mode of the value of the column.

Approach 4 Filling with Model-Based Imputation: We find that values from "Number.of.Engines" and "Purpose.of.flight" are correlated with the column "Model". Our hypothesis is that if two planes are in the same model, then they should have the same number of engines. When we test our idea, we find that out of 9564 distinct models, only 175 of them contain different numbers of engines. Similarly, we find that certain models can be more preferred to fulfill certain purposes of flight. We group the columns "Number.of.Engines" and "Purpose.of.flight" by 'Model' and calculate the most common value for each model. And then we fill in the missing value of these two columns with the most popular value.

Approach 5 Directly drop: After all methods we try, we still have a few missing and "Unknown" values. We decided to directly drop them since they are not a lot. By the end of handling missing and "Unknown" values, the total rows of our dataset change from 68566 to 66972, which is acceptable.

2.2.4 Target Features

Our dataset contains a few columns like Total.Fatal.Injuries to measure the severity of an accident. Since they are all numerical features, we plan to utilize them to generate a new feature called "Survival.Rate" to better measure the severity of accidents. When deciding the feature to measure the severity of the accident, we think all columns "Survival_Rate", "Injury.Severity", and "Aircraft.damage" should all be included to measure the severity. We decided to make a new target feature "composite_severity_score" as the average of three features, as there is no information given prior about the weight of these features.

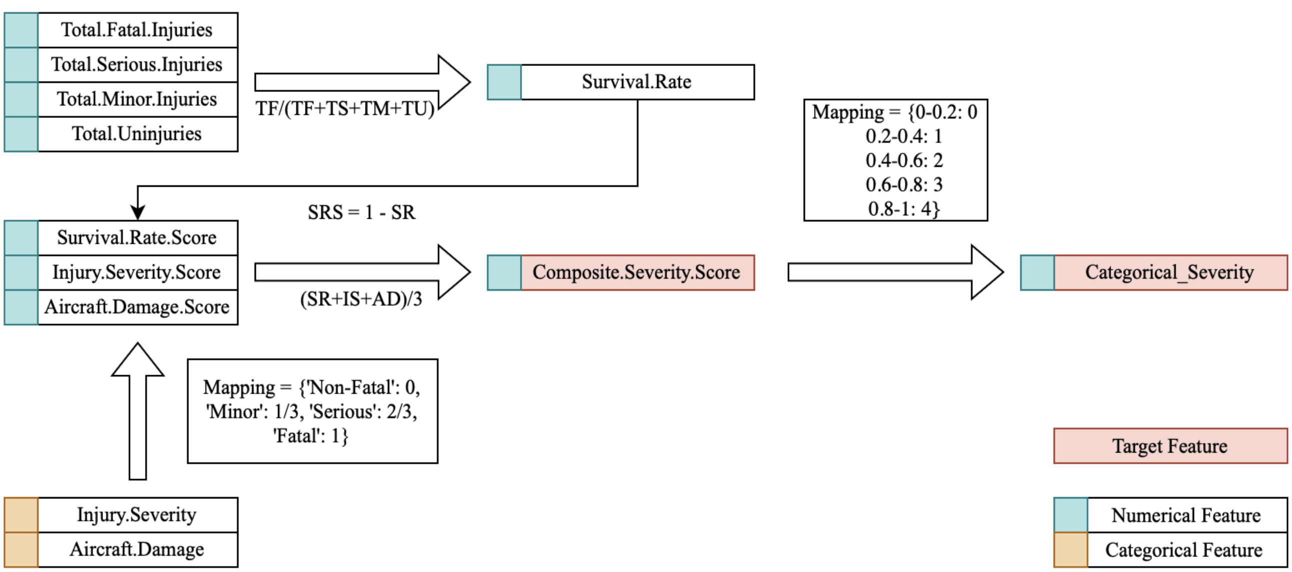

Figure 3: Target Feature Generation Process

2.2.5 Encoding Categorical Feature

After dropping and merging all the features, we are left with 21 features – 13 of them are predictive features, 7 of them are binary indicators, and the last 1 is the target features. Among these 13 features, "Model" and "Make" columns have thousands of unique values, which make them unsuitable for label-encoding. Hence, we implement frequency encoding for features with high-cardinality and label encoding for the rest of categorical features.

2.3 Model Selection

We start with the linear regression model. However, after we plot the distribution of the target feature, we observe that the distribution of points does not align well with the assumption of linearity, which is fundamental for linear regression models. The distinct vertical bands of data points at specific actual severity scores suggest that the severity scores might be categorical or discrete in nature, rather than continuous.

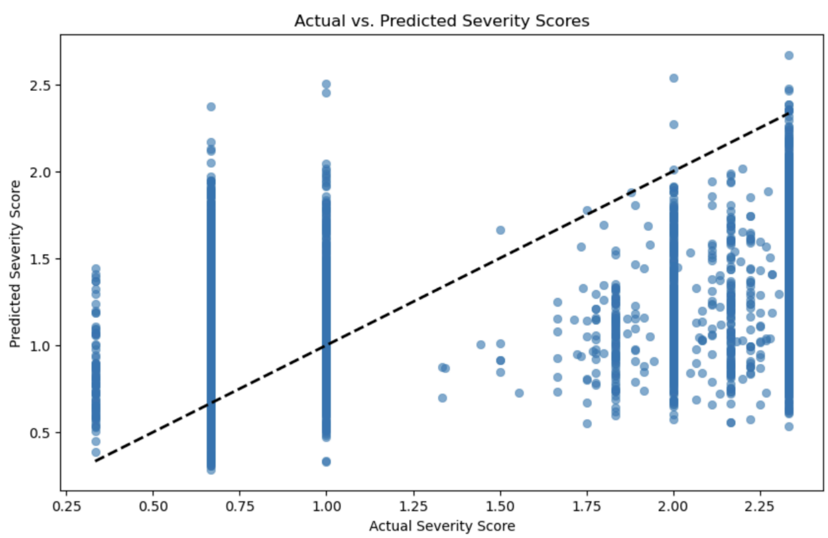

Figure 4: Target Feature Value Distribution

We switch to the random forest model for the non-linear patterns observed in the scatter plot. It can also automatically handle interactions between features such as "Model" and "Number.of.Engine". Lastly, it provides a package to measures of variable importance easily.

2.4 Model Development

We plan to use two metrics to measure the performance of the random forest model.

Accuracy: Representing the proportion of correctly predicted instances out of all predictions.
AUC-ROC: How much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s.
We plan to try different numbers of trees in the forest to optimize the model. Besides, we will implement GridSearchCV to find the optimized parameter of max depth, min samples split, min sample leaf and max features. We will use Gini Impurity to calculate feature importance to show which shows the significance of different factors that cause aviation accidents.

3 Result

We test the performance of the random forest model under the number of trees in the forest of 50, 100 and 200. The graph shows a marked improvement in both accuracy and AUC-ROC as the number of estimators increases from 50 to 100, indicating that more trees help the model to generalize better over the training data. Beyond 100 trees, the increase in accuracy begins to plateau, suggesting diminishing returns with further increases in the number of trees, although the AUC-ROC continues to increase slightly. This trend suggests that while increasing n_estimators beyond a certain point may continue to provide minor improvements in model discrimination ability (as measured by AUC-ROC), the overall accuracy does not significantly benefit while the run-time doubled. Considering the trade-off between computational cost and performance gain, we think n_estimators = 100 is the best choice. The result of the 5-fold cross validation is very close to each other, ranging from about 0.825 to 0.827. This tight range suggests that the model is stable and performs uniformly across different partitions.

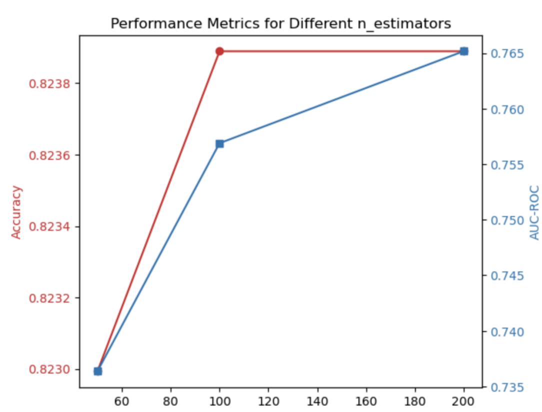

Figure 5: Performance of Model under Different Estimator

The most significant feature influencing the model's decisions is Model_Freq, which represents the frequency encoding of the aircraft model. This suggests that certain models of aircraft are more strongly associated with the severity of the aviation accident. The next most important features are State_Location and Year, followed by Month_Abbr and Make_Freq, indicating that the location and time-related attributes play crucial roles in the model's performance. It's noteworthy that features related to the flight's specifics, such as Broad.phase.of.flight, also show considerable importance, highlighting how specific conditions of operations influence the outcomes. The lower half of the chart shows features with missing or unknown data, which significantly contribute less, underscoring the importance of complete and accurate data for model accuracy.

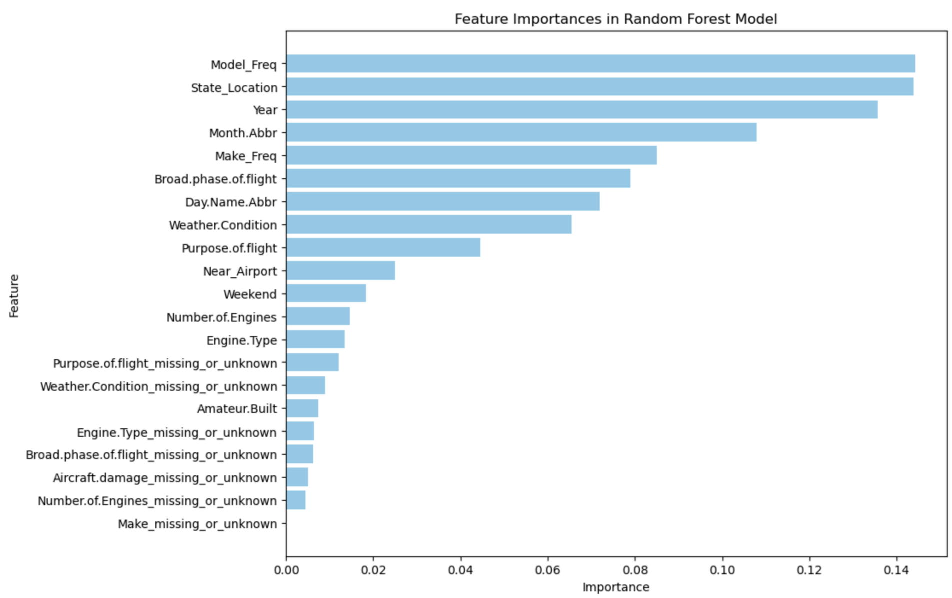

Figure 6: Feature Importance in Random Forest Model

We also utilize the method of GridSearchCV to optimize the rest of parameters in the random forest model. We find that with the setting {max_depth: 30, max_feature: sqrt, min_samples_leaf: 1, min_samples_split: 10}, we can improve the accuracy from 0.830 to 0.833.

4 Conclusion

Our predictive model, utilizing a Random Forest approach, successfully identified several critical factors influencing the severity of aviation accidents. The model performed well with a selected configuration of

100 trees, achieving an accuracy of approximately 83.3% and a satisfactory AUC-ROC score, which signifies its capability to distinguish between different levels of severity. Notably, the 'Model_Freq' feature, representing the frequency encoding of the aircraft model, emerged as the most significant predictor, underscoring the impact of specific aircraft models on accident severity. Other important factors included location-related attributes and temporal aspects such as the year and month of the accident.

We are reasonably confident in our results, given the model's robust performance metrics and the comprehensive cross-validation process employed during development. The use of GridSearchCV for hyperparameter tuning further enhanced our model's reliability by optimizing its settings for better accuracy.

Given the model’s strong performance and its insightful findings, I would advocate for its application within our airline company. Here are some of the actionable operational adjustments:

Enhanced training for flight operations for specific plane models. Allocate more source and attention for flight operations for a specific month.

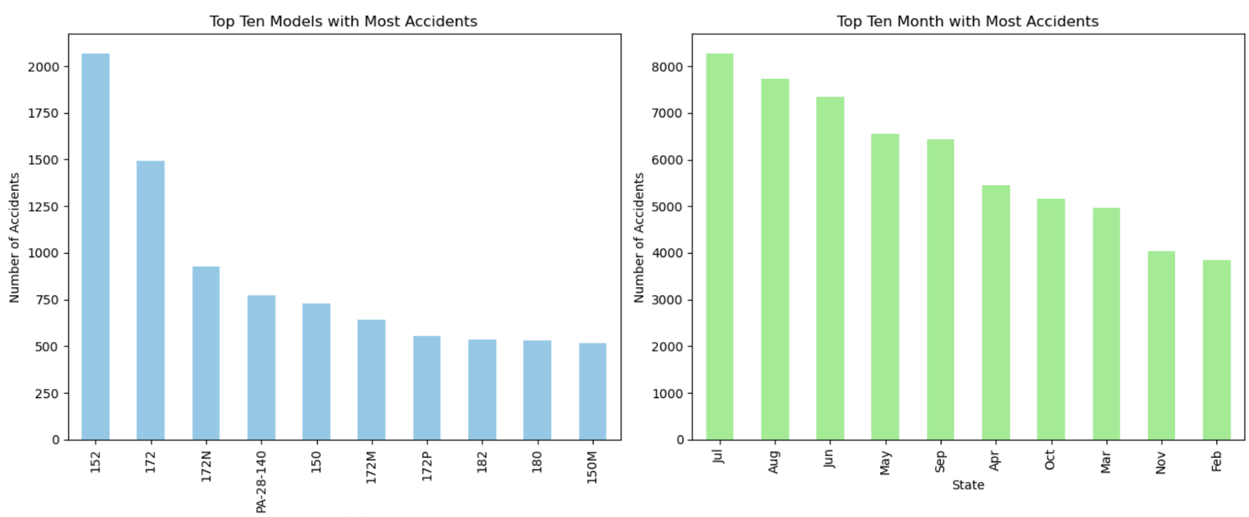

Figure 7: Top Ten Models and Month with Most Accidents

By leveraging these insights, we can enhance our risk management strategies, refine our safety protocols, and ultimately improve decision-making processes. This could lead to better resource allocation, more effective accident prevention measures, and possibly a reduction in related costs, thereby supporting both operational efficiency and safety enhancements.

Reference:

[1] Minca, A. (2024). Project_2024. SYSEN 5200. Ithaca, NY, 14850: Cornell University.

[2] National Transportation Safety Board. Aviation accident database & synopses. Retrieved from https://www.ntsb.gov/_layouts/15/ntsb.aviation/AviationDownloadDataDictionary.aspx

Appendix:

The code for all of the four questions can be found at:
https://github.com/yq284/SYSEN-5200