Paper 175-2024

Flight Delay Analysis and Prediction

Eric Braga, Miguel Ángel Bravo Martínez del Valle, Riley Rutan, Fernanda Carrillo Escarcega, and Xinyu Du, San Diego State University

Abstract

Like most global industries, the COVID-19 pandemic caused fundamental shifts in the airline industry. As a result, the consumer is often left with the inconvenience of delayed and canceled flights for various reasons. In this project, we explored a large dataset of canceled and delayed flights from 2019-2023 from the US from the Department of Transportation along with weather data from NOAA, leveraged machine learning models to identify trends in delayed and canceled flights, and created a tool that consumers can use to predict the probability that their upcoming flight will be delayed or canceled. After using several machine learning models, we utilized an Extreme Gradient Boosting (XGBoost) model that achieved an accuracy of 0.68 and a weighted average f1-score of 0.70.

Introduction

In an age when the average American prefers to travel long distances by air, airline services must be reliable and cost-effective. The massive growth of this sector has led to crowded airports and stress on the American airline industry. Frequent flight delays and flight cancellations often cause the consumer's travel plans to be disrupted and are a financial burden on airline companies and the American economy. The Total Delay Impact Study [1] estimates that flight delays and cancellations in 2007 cost consumers $16.7 billion, cost airlines $8.3 billion, and had a direct negative impact on the US Gross Domestic Product by $4 billion. The inefficiencies of the airline industry are costly, and solutions to flight disruptions would be invaluable for the consumer and the US economy.

The period from 2019 to 2023 has been marked by an extraordinary wave of disruptions in the airline industry. The U.S. Department of Transportation's Air Travel Consumer Report [2] disclosed that only 1.29% of flights were canceled in 2023, a substantial decrease from the previous year's rate of 2.71%, underscoring a recovery from the peak pandemic years ("Air Travel Consumer Report: December 2023, Full Year 2023 Numbers"). Despite this improvement, challenges such as weather extremes, staffing shortages, and technical malfunctions persisted as significant impediments to reliable air travel.

Weather-induced disruptions, particularly thunderstorms, have been a critical factor in flight cancellations and delays. In a poignant example, a report on August 6, 2022, stated that the tracking service FlightAware reported that thunderstorms on August 5, 2022, led to the cancellation of about 1,400 US flights and delayed thousands more, underlining the vulnerability of the aviation network to natural forces ("US Flights Cancel Due to Thunderstorms").

Beyond meteorological issues, industry-wide staffing shortages exacerbated by the pandemic have strained operations. American Airlines, for instance, announced proactive measures to cut 1% of its July flights, a direct response to the extraordinary challenges posed by the post-pandemic resurgence in travel demand and workforce deficits ("The Future of Flying: More Delays, More Cancellations, More Chaos"). The shortfall in staffing is critical, with the aviation industry grappling with a deficit of approximately 32,000 commercial pilots, mechanics, and air traffic controllers. This shortfall led to a third of the nation's flights being delayed and 1 in 17 canceled during four days in June 2023, as reported by CBS News. Moreover, the scheduling practices of airlines have come under scrutiny for contributing to these disruptions. Unrealistic scheduling often overstated the carriers' capacity to service the tickets they sold, a practice that not only led to operational chaos but also eroded public trust in U.S. flights.

In this complex backdrop, our project harnesses a comprehensive dataset from 2019 to 2023, provided by the U.S. Department of Transportation, to analyze and understand the multifaceted nature of flight delays and cancellations. Through machine learning engineering, we aim to decipher the patterns that forecast these travel disruptions, striving to equip consumers with the foresight to plan their travels more effectively. This initiative stands as a response to the industry's turbulence, offering a data-driven tool to navigate the ever-present uncertainties of air travel.

About the Datasets

Flight Delay and Cancellation Dataset

Our primary dataset originates from the U.S. Department of Transportation, specifically sourced from the Bureau of Transportation Statistics, spanning the years 2019 to 2023. This comprehensive dataset, available via Kaggle, encompasses a vast array of flight-related information, comprising 32 attributes and a staggering 3 million rows. Among its contents are detailed flight routes denoting origin and destination points, event durations, as well as insights into the reasons behind flight delays and cancellations.

https://www.kaggle.com/datasets/patrickzel/flight-delay-and-cancellation-dataset-2019-2023/data

Weather Dataset

To enrich our analysis, we incorporated weather data sourced from the National Oceanic and Atmospheric Administration (NOAA). This dataset is called the Daily Global Historical Climatology Network (GHCN-Daily) and is available through their website. It provides a wealth of meteorological information essential for contextualizing flight operations within the United States. This data is a collection of many different daily measurements for stations across the globe, and many of these stations are located at or very close to airports. The five core elements of this data describe precipitation, snowfall, snow depth, maximum temperature, and minimum temperature. The additional measurement of average daily wind speed was included in this project because it is an essential consideration for pilots.

https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ncdc:C00861

https://noaa-ghcn-pds.s3.amazonaws.com/index.html

Airport Coordinates Dataset

To bridge the gap between the flight and weather datasets, we used the International Air Transport Association/International Civil Aviation Organization (IATA/ICAO) list data from http://www.ip2location.com. This data set includes the country code, region name, IATA code, and ICAO code. airport name, latitude, and longitude. The International Air Transport Association (IATA) code was used to merge the code on the flight data to assign geographic markers to each airport in our flight dataset.

https://github.com/ip2location/ip2location-iata-icao/blob/master/iata-icao.csv

Exploratory Data Analysis

	column_names	column_dtypes	column_missing
21	CANCELLATION_CODE	object	2920860
31	DELAY_DUE_LATE_AIRCRAFT	float64	2466137
30	DELAY_DUE_SECURITY	float64	2466137
29	DELAY_DUE_NAS	float64	2466137
28	DELAY_DUE_WEATHER	float64	2466137
27	DELAY_DUE_CARRIER	float64	2466137
19	ARR_DELAY	float64	86198
24	ELAPSED_TIME	float64	86198
25	AIR_TIME	float64	86198
15	WHEELS_ON	float64	79944
16	TAXI_IN	float64	79944
18	ARR_TIME	float64	79942
13	TAXI_OUT	float64	78806
14	WHEELS_OFF	float64	78806
12	DEP_DELAY	float64	77644
11	DEP_TIME	float64	77615
23	CRS_ELAPSED_TIME	float64	14
22	DIVERTED	float64	0

We began our analysis by loading all flight data into Python utilizing the Pandas library to extract essential information about the dataset. A method was created to return a summary of the metadata, including column names, data types, and the number of missing values. These values can be found in Table 1.

This first step provided valuable insights into the quality and completeness of the dataset.

After our initial exploration, we looked more closely at the categorical variables in our dataset to discover important insights about flight operations in the United States.

Our analysis showed that the top five airlines accounted for a significant portion, approximately 65.0% of the total flights in the U.S. Also, we found that around 27% of flights departed from the most frequent airports in the country.

Table 1. Flight Delay Data Summary

By examining the frequencies of flights by airline and origin city, we can later account for class imbalance in the data when creating a machine-learning model to predict flight delays. These frequencies can be found in Figure 1 and 2 below.

Figure 1. Flight Frequency by Airline

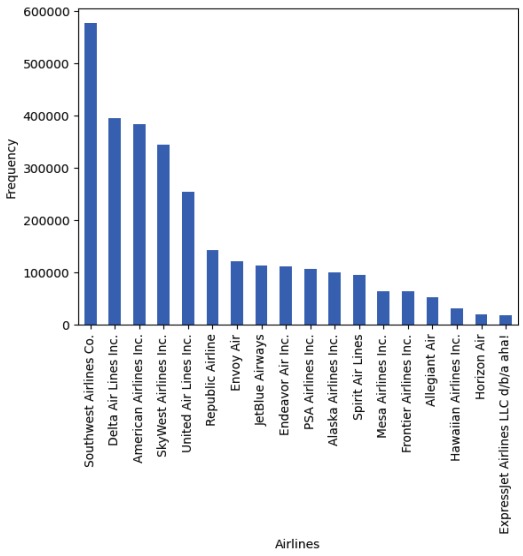
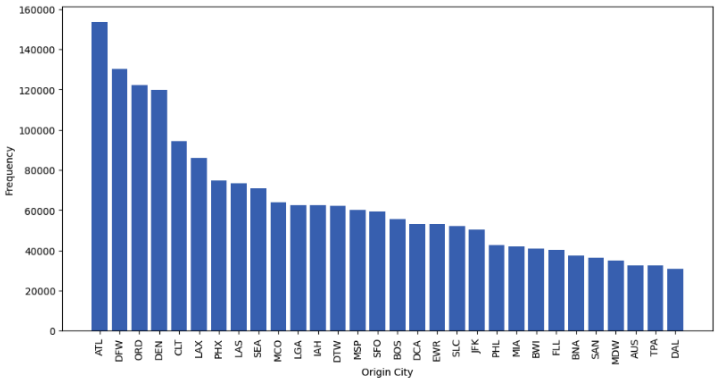


Figure 2. Flight Frequency by Origin City



More insights in the airline industry can be found when only examining flights that have been delayed. Among the different airlines, Southwest Airlines appears to be the carrier with the highest percentage of delayed flights, accounting for approximately 23.9% of the total delayed flights. Followed by American Airlines at 14.3% and Delta Airlines at 10.3%. These percentages can be found in Figure 3.

We also analyzed the total flights per airline and the percentage of those flights that have been delayed. Figure 4 displays variations in total flights and the amount of delays among airlines, offering insights into airline performance and reliability. Additionally, we analyzed the average delay time for flights departing from different cities in Figure 5. This allows us to identify cities with higher or lower average delay times, which can lead to researching some potential reasons for delays in specific regions.

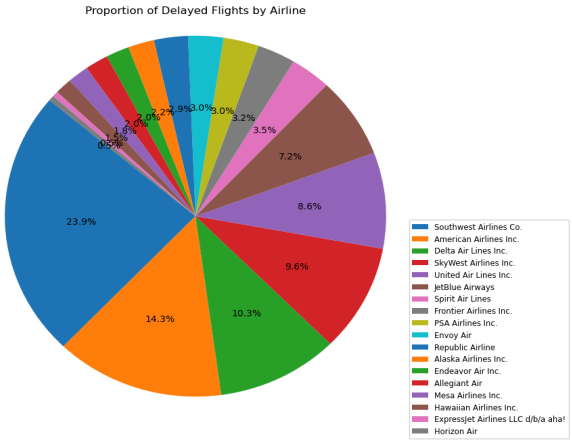


Figure 3. Delayed Flights by Airline

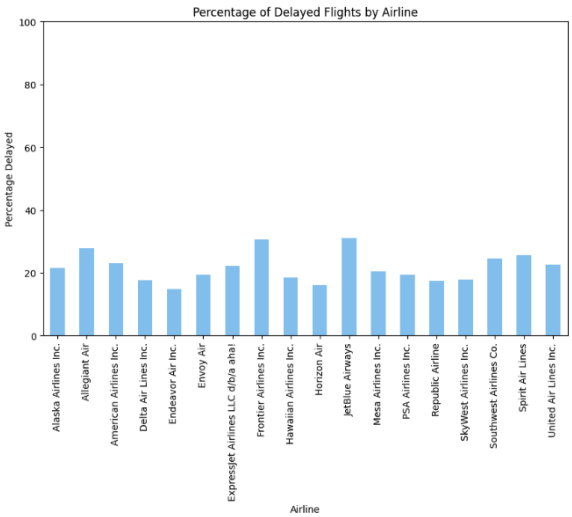


Figure 4. Rate of Flight Delays by Airline

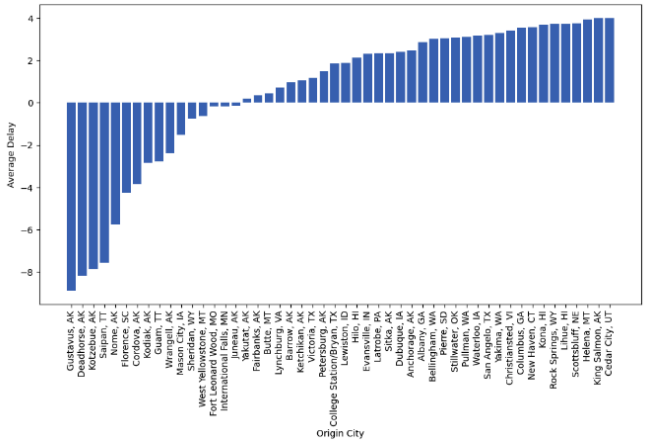


Figure 5. Average Flight Delay by Origin City

By looking at the average delay grouped by month in Figure 6, it is clear that the date on which the flight is scheduled is linked to the likelihood that it will be delayed. It is apparent that the average flight delay peaks in the summer and holiday months, while flights in the spring and fall are more often on time. For this reason, we decided to include the date variable in our predictive models.

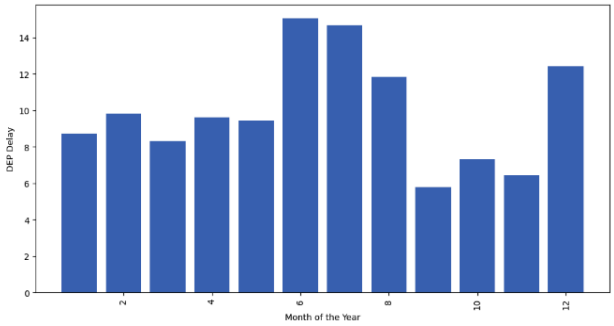


Figure 6. Average Flight Delay by Month

Methodology

Merging Flight and Weather Data

To merge weather data to the main flight data the flight data was merged with the IATA dataset to provide each origin and destination airport with latitude and longitude. Next, the ghcnd-stations.txt file from NOAA was turned into a data frame by using the read_fwf function in the pandas library. The fwf stands for fixed width format and was used because of the format of the .txt file. The station list and flight data were turned into geopandas data frames with the latitude and longitude being turned into a point geometry column. These tables are then merged with the geopandas functions sjoin_nearest. This function maps the closest weather station ID to the airport using the latitudes and longitudes. The origin airports were then used to fill out the destination airport with the appropriate station ID. By associating each airport with its nearest weather station, meteorological data such as temperature, precipitation, and wind conditions for origin and destination airports are incorporated with flight data.

NOAA has the weather organized with each measurement by year with the listed station daily measurement in parquet files. With all the files a process_parquet_file function was created that would read the parquet file format by renaming columns and dropping others. For efficiency, it only includes data from station IDs in our airport list. Then with an empty list, we create code that will iterate through every parquet file in a folder path and perform the process_parquet_file function on all of them. Then this list is concatenated to form a single data frame for all data for one weather variable. This is done for each variable and then merged for full weather data. The two now complete data frames of flights and weather are merged on location and date once on the origin location of the flight and a second time on the destination location of the flight.

Preprocessing Data

After our data is combined we address missing values and duplicates. We discard flights lacking weather information, resulting in a final dataset comprising 1,255,864 rows, significantly reduced from the original 3 million rows.

We created a binary target variable, called 15_DELAYED that marks flights as being delayed more than 15 minutes for arrival or departure, as this is the industry standard. We also feature-engineered the scheduled flight departure date and time, turning a DateTime object into several numerical features so our models could find patterns in the day of the week, day of the month, month, year, and hour of the day.

For feature selection, we eliminate redundant variables, retaining those pertinent to flight details, airline information, weather conditions, and indicators of delays. Intending to create a truly predictive tool, we aimed to only provide our models with information that would be available before a flight occurs. Finally, we split the dataset into training and testing sets to facilitate model development and evaluation.

We created pipelines for numerical and categorical features. Categorical features are OneHotEncoded, and categories are turned into separate features so the model reads numerical instead of string values. For numerical pipelines, all numerical features were normalized to prevent overfitting and underfitting.

Analysis (Models and Performance)

Logistic Classification Model

Logistic Regression models are widely accepted for binary classification because they model the odds of an event as a linear combination of independent variables. An initial logistic classification model from scikit-learn was created to establish baseline performance of predicting delayed flights from pre-flight information without weather data. As flights that are not delayed make up over 78% of the dataset, this class imbalance was adjusted for in the model. The results of this classification model are found below in Table 2, while the features with the highest importance in the model are found in Table 3.

Table 2. Logistic Model Results

	Precision	Recall	F1- Score	Support
On-Time	0.85	0.60	0.70	470,833
Delayed	0.56	0.60	0.39	129,167
Accuracy	0.60			600,000
Macro Avg	0.57	0.60	0.55	600,000
Weighted Avg	0.73	0.60	0.64	600,000

Table 3. Logistic Model Feature Importances

Feature	Importance
cat_DEST_CDB	0.844633
cat_DEST CITY Cold Bay, AK	0.844633
cat_ORIGIN_CDB	0.736002
cat_ORIGIN CITY_Cold Bay, AK	0.736002

This first model performed poorly with an F1-Score of .39 for delayed flights. Upon investigating the importance coefficients of each feature in the data frame below, it is apparent that the model placed high importance on flights from a few small airports and cities. There is a small likelihood that flights from these airports appear in the testing set, which led to poor performance. Moving forward, we should engineer these features to exclude origin and destination airports that are infrequent in the dataset. This model learned that flights coming in and out of small airports are a good predictor of whether a flight will be delayed, but does not perform well on unseen data because these cases are so rare in the dataset.

Moving forward, a function was created to turn all categories that appear in less than 1% of the data into ‘other’, to prevent overfitting on small airports and decrease the computational resources required to train the models.

Logistic Classification Model with Weather Data

We used a similar logistic regression classifier and included the weather data for the departing and arriving airport for each flight. The results are shown in the table below in Table 4.

	Precision	Recall	F1- Score	Support
0(not delayed)	0.76	0.99	0.86	142150
1 (delayed/canceled)	0.56	0.05	0.10	46230
Accuracy	0.76			188380
Macro Avg	0.66	0.52	0.48	188380

Weighted Avg	0.71	0.76	0.67	188380

Table 4. Logistic Model with Weather Data Results

This is a poor predictive performance from the model. Because on-time flights make up nearly 75% of the total flights, the model achieving an accuracy of 76% indicates that the model only slightly outperformed simply guessing that every flight is on time. This is also evident with the recall for not delayed being so high at 99% and the recall for delayed being only 5%. The next step would be to adjust this to take into account the class imbalance to see if this type of model has any viability in helping find a solution to the problem.

Moving forward, an argument of class_weight = 'balanced' was added to our models to address this class imbalance issue, and the argument n_jobs =-1 was added to allow parallel training to reduce training time.

Decision Tree Model

A decision tree model is one of the most powerful supervised learning methods for classification. It creates a flowchart-like tree structure of decisions and their possible consequences. The results of this model can be found below in Table 5.

	Precision	Recall	F1 Score	Support
0 (not delayed)	0.82	0.81	0.81	195889
1 (delayed)	0.34	0.35	0.35	55284
Accuracy	0.71			251173
Macro avg	0.58	0.58	0.58	251173
Weighted avg	0.71	0.71	0.71	251173

Table 5. Decision Tree Model Results

For the class "not delayed" (0), a precision of 0.82 means that out of all of the instances predicted as 0, 82% of them are not delayed. In this case, a recall score of 0.81 means that out of all the instances that are actually "not delayed", 81% of them were correctly identified as not delayed. These scores mean that the model has a relatively high precision and recall for predicting "not delayed" flights with an F1 score of 0.81, displaying a good balance between precision and recall.

Additionally, for the class "delayed" (1) a precision of 0.34 means that out of all the instances predicted as delayed by our model, only 34% were delayed. Similarly, a recall of 0.35 means that the model correctly predicted 35% of all delayed instances in the dataset. Thus, these scores suggest that our model's performance is relatively low. The precision tells us that when the model predicts a flight to be delayed it corrects about 34% of the time, and the recall captures only 35% of all delayed flights in the dataset.

Based on the scores obtained from this model it can be determined that even when the model scores well for predicting non-delay flights, there is room for improvement in the model's ability to correctly predict delayed flights.

Gradient Boosting Model

Gradient boosting is a machine learning approach that uses many decision trees that make few assumptions about the data. Regular gradient boosting algorithms are often short when handling large datasets. This is primarily because they cannot parallelize computation, which can lead to significantly longer training times. In contrast, XGBoosting is engineered to leverage the power of parallel processing, which enables it to perform computations much faster than traditional Gradient Boosting Machines (GBM). Furthermore, XGBoost incorporates a technique to calibrate machine learning models to minimize the adjusted loss function. This feature prevents overfitting or underfitting, ensuring that the model generalizes well to new, unseen data.

	Precision	Recall	F1 score	Support
0 (not delayed)	0.87	0.69	0.77	163691
1 (delayed)	0.37	0.65	0.47	46024
Accuracy	0.68			209715
Macro avg	0.62	0.67	0.62	209715
Weighted avg	0.76	0.68	0.70	209715

Table 6. Gradient Boosting Model Results

Initial results using default settings in a regular gradient boosting model showed a recall for the delayed class (class 1) of around 0.12, which was suboptimal, particularly for an imbalanced dataset like ours. Utilizing the scale_pos_weight parameter in XGBoost, helped to correct the model's bias towards the majority class by using the weights of different classes during the training process. For our case, a scale_pos_weight of 3.557 was used, calculated based on the support ratio of delayed to not delayed instances. This adjustment improved the recall for the delayed class significantly, highlighting the adaptability and robustness of XGBoost when dealing with imbalanced datasets:

For the not delayed (1) class, this model scored 0.87 for precision which means that out of all of the flights predicted as “not delayed” 87% of them are not delayed. This suggests that there is a high amount of correct predictions for flights that are not delayed. Additionally, the model scored 0.69 for recall, this indicated that out of all of the instances that are not delayed 69% were correctly identified as not delayed.

Secondly, for the delayed class (1) the precision score is 0.37 which indicates that out of all of the delayed instances, only 37% were delayed which suggests our model has a relatively low precision for delayed flights. In this case, the recall score was 0.65 meaning that out of all instances that are delayed 65% were correctly identified as delayed. This shows the model’s ability to identify delayed flights.

The model performs well in identifying flights that are not delayed (0) but it has a lower performance in identifying delayed flights. However, it captures a decent proportion of delayed flights with reasonable recall.

Following a comprehensive randomized grid search over various hyperparameters(n_estimators, max_depth, learning_rate, gamma, and colsample_bytree), the best model configuration found did not surpass the model's performance with scale_pos_weight=3.557.

In contrast, adjusting the scale_pos_weight parameter showed a more balanced treatment of the minority class, leading to a better recall. It effectively addressed the class imbalance by attributing a higher weight to the minority class during training, enhancing the classifier's sensitivity to delayed flights. As the recall for predicting delays is a critical performance measure in our scenario, we have decided to proceed with the scale_pos_weight adjustment in our final XGBoost model. This choice is aligned with our commitment to achieving the best possible performance in identifying delayed flights, which holds significant value in practical applications.

Neural Networks

The dataset we are utilizing for model creation comprises a total of 22 columns, with 1 serving as our 'Y' variable for prediction. Among these 22 columns, there are 12 variables linked to the meteorological conditions of the origin and destination airports. The remaining variables include airline, origin city, destination city, as well as pertinent data concerning air and airport congestion, such as time of day, day of the week, or month of the year.

We employed the 'tensorflow' library to develop this model, integrating different layers and optimizers. Before constructing and feeding our neural network, we need to finalize data preprocessing.

Specifically, our focus lies on the variables 'AIRLINE', 'ORIGIN', and 'DEST', corresponding to the airline name, origin city, and destination city, respectively. We intend to apply a function to this dataset, designed to aggregate categorical variables with a representation below 3.5% into a generic group labeled 'Others'. Thus, we transition from having 17 distinct airlines to 11, 221 origin cities to 6, and 220 destination cities to 6.

Our approach entails comparing outcomes derived from diverse models. Initially, we employ a sophisticated model encompassing all airports. Subsequently, we reevaluate all airports using a simpler model. Next, we focus solely on the four busiest airports, employing the simpler model. Finally, we scrutinize the busiest airport exclusively, once more utilizing the simpler model.

Complex Neural Network

We try to create and train a neural network consisting of 6 layers, a first input layer with as many inputs as attributes we have in our dataset (all but one, which will be the one we want to predict). All the layers are of type 'dense' and all the hidden layers are activated by the method 'relu'. The output layer is activated with 'sigmoid', since we are training a binary classification predictive model. On the other hand, we are going to use 'binary_crossentropy' as a loss function. We introduced 64, 150, 100 and 40 neurons respectively in each one of the hidden layers.

Using 10 epochs of 64 rows to see the performance:

Figure 7. 10 Epoch Complex Loss Function Graph

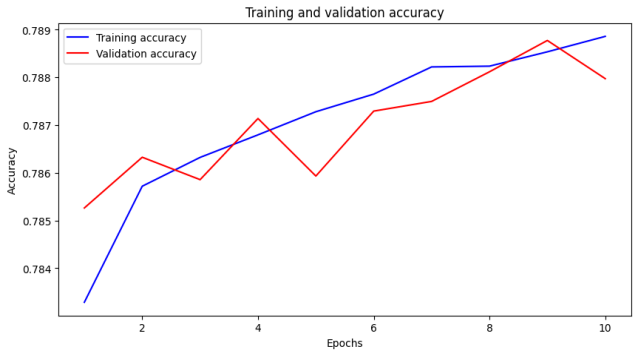


Figure 8. 10 Epoch Complex Accuracy Graph

Using 300 epochs:

Figure 9. 300 Epoch Complex Loss Function Graph

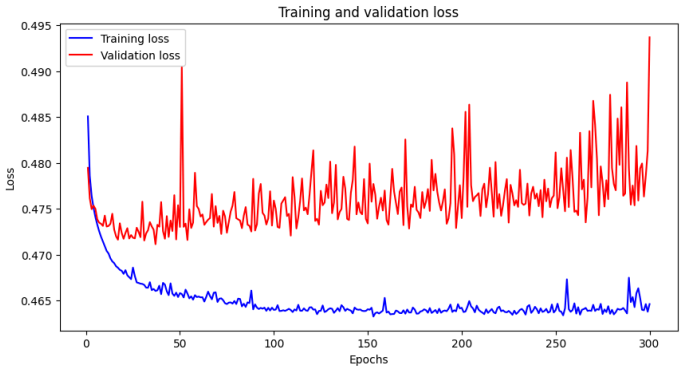
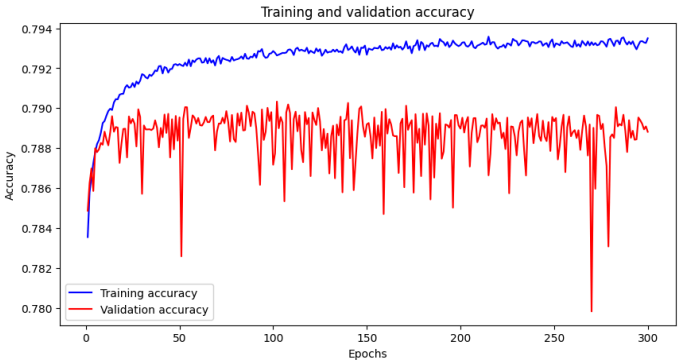


Figure 10. 300 Epoch Complex Accuracy Graph



Results Table:

	Precision	Recall	F1-Score
0 (Not Delayed)	0.81	0.96	0.88
1 (Delayed)	0.55	0.19	0.28
Accuracy	0.79		
Macro Avg	0.68	0.57	0.58
Weighted Avg	0.75	0.79	0.75

Table 7. Complex Neural Networks Results

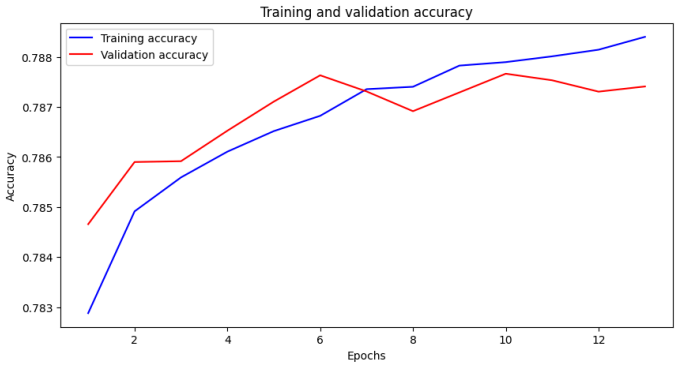
Simplified Neural Network

We decreased the number of layers and the number of neurons in each hidden layer, to see if reducing the complexity of the model could help the performance, we also added EarlyStopping method in order to stop training the model in case it does not improve anymore, so we don't have to wait for all the epochs.
We changed the number of layers, going from four hidden layers to three hidden layers with ReLu activation function, made from 30, 50 and 25 neurons, respectively. We also have the input and output layers, using 'sigmoid' as the activation function for this last layer, and the same loss function as the one we used for the previous model (''binary_crossentropy' ').

**All the Airports
Even though we tried to train the model for 30 epochs, it stopped after the 13th one, finding this results:

Figure 11. Simple Neural Network Loss Function Graph Graph

Figure 12. Simple Neural Network Accuracy



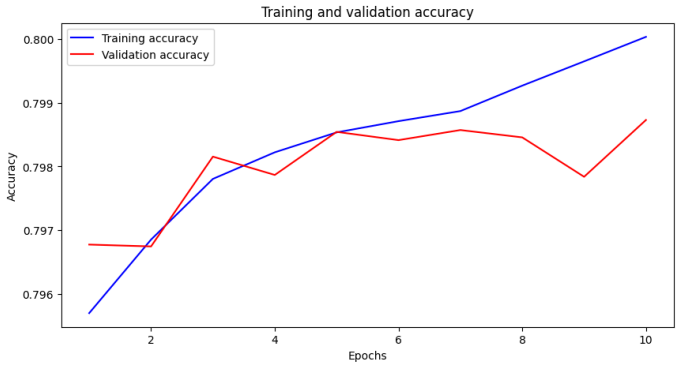
	Precision	Recall	F1-Score
0 (Not Delayed)	0.80	0.98	0.88
1 (Delayed)	0.58	0.12	0.19
Accuracy	0.79		
Macro Avg	0.69	0.55	0.54
Weighted Avg	0.75	0.79	0.73

Table 8. Simple Neural Networks Results

**The four busiest airports
Attempt to create a neural network that focuses only on the 4 airports with the most traffic.
First of all, we have to prepare the date, taking into account only the flights coming from the two airports with the most flights, as well as the two airports with the most flights arriving. These airports are.
With more departing flights we have ‘ATL’ (Atlanta, GA) and ‘DFW' (Dallas, TX) while with more arriving flights we also have ‘ATL’ and ‘ORD’ (Chicago, IL).
Even though we tried to train the model for 30 epochs, it stopped after the 10th one, finding:

Figure 13. Busy Airport Loss Function Graph

Figure 14. Busy Airport Accuracy Graph



	Precision	Recall	F1-Score
0 (Not Delayed)	0.81	0.98	0.89
1 (Delayed)	0.55	0.10	0.16
Accuracy	0.80		
Macro Avg	0.68	0.54	0.52
Weighted Avg	0.76	0.80	0.74

Table 9. Neural Networks Results for 4 Busiest Airports

**Atlanta Airport, the busiest one

We tried to run a model only taking into consideration Atlanta related data, to see how reducing data and attributes could help the neural network to find patterns and learn about the data.

Even though we tried to train the model for 30 epochs, it stopped after the 14th one, finding this results:

Figure 15. Atlanta Loss Function Graph

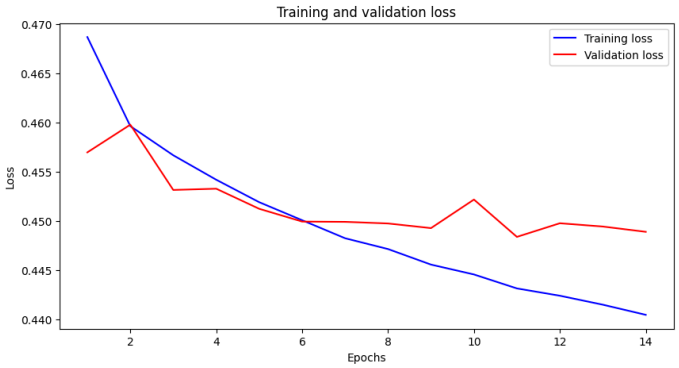
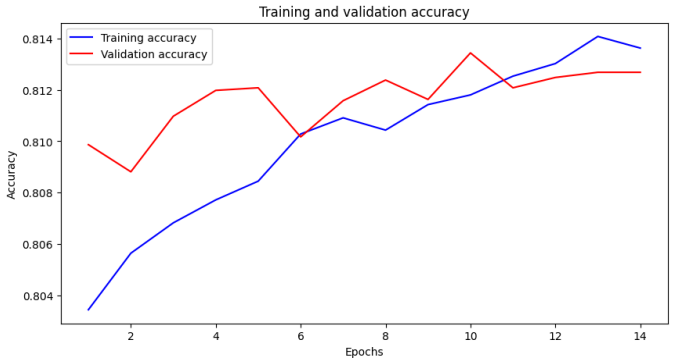


Figure 16. Atlanta Accuracy Graph



	Precision	Recall	F1-Score
0 (Not Delayed)	0.82	0.98	0.89
1 (Delayed)	0.58	0.12	0.20
Accuracy	0.81		
Macro Avg	0.70	0.55	0.55
Weighted Avg	0.77	0.81	0.76

Table 10. Neural Networks Results for Atlanta Airport

By comparing the results of all four neural network models below, it is apparent that the model that only utilized data from the Atlanta airport had the best results. This affirms that increasing the complexity of the database harms the model's performance. While simplified data increases performance, the consistently low recall score for delayed flights indicates that similar patterns are being learned by all models. This may indicate an inherent shortcoming of our dataset, that the features we input into our models are not great indicators of flight delay. Using all the airports we have a total of 1.2 million values, when we reduce to four airports we have a total of 350 thousand values, while when we use only those of the Atlanta airport, we have only 100 thousand values.

That is why we could assume that although the best model is the Atlanta model, the simple model in which we use all the data from our dataset would be the most complete, since it would only decrease by 2% in the accuracy of the negative class (not delayed), maintain the recall for this class, maintain the performance for the positive class and only decrease by 2% in the accuracy of the positive class. Using only a fraction of the total data in the Atlanta model, the performance improvement was only marginal. Therefore, a more robust model that utilizes the entire dataset was chosen to maintain the ability to predict flight status across the country.

Table 11. Simple Neural Networks Results

	Atlanta		Four Busiest Airports		All Airports (Simple Model)		All Airports (Complex Model)	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0 (Not Delayed)	0.82	0.98	0.81	0.98	0.80	0.98	0.81	0.96
1 (Delayed)	0.58	0.12	0.55	0.10	0.58	0.12	0.55	0.19
Accuracy	0.81		0.80		0.79		0.79	
Macro Avg	0.70	0.55	0.68	0.54	0.69	0.55	0.68	0.57
Weighted Avg	0.77	0.81	0.76	0.80	0.75	0.79	0.75	0.79

Random Forest

Random Forest is an ensemble learning method for classification and uses several decision trees while training. The output of this model is the class that was selected by the most trees. The results are found below.

Table 12. Random Forest Model Results

	Precision	Recall	F1-Score
0 (Not Delayed)	0.80	0.98	0.88
1 (Delayed)	0.61	0.13	0.22
Accuracy	0.79		
Macro Avg	0.71	0.55	0.55
Weighted Avg	0.76	0.79	0.74

Table 13. Random Forest Feature Importances

Feature	Importance
num_hour	0.092
num_DISTANCE	0.083
num__dest_AWND	0.073
num__ori_AWND	0.072
num__ori_TMAX	0.070

For our class of non-delay flights (0) the random forest model got a precision score of 0.80 that means that out of all of the flights that were predicted as “not delayed” were not delayed. Similarly, the recall score for this class is at 0.98 so out of all the instances that are not delayed, 98% were correctly identified as not delayed.

Additionally, for the delayed class (1) the precision score is at 0.61 suggesting that out of all of the flights predicted as delayed, 61% were delayed which indicates our model has a moderate ability to predict flights that are delayed. And the recall score is 0.13 which means that out of all of the flights that are delayed only 13% got correctly identified as delayed.

In summary, the model performs very well in predicting flights that are not delayed (0) with high precision and recall scores. However, the model performs low in delayed flight prediction with a very low recall score. This shows the models could use some improvement in predicting for class (1), delayed flights.

To create an accurate model, a decision tree was trained on only flights in and out of Atlanta (over 20 thousand flights). Similar model performance was observed on this subset of flights, and might indicate a shortcoming of the dataset, and not our model design.

KNN (K-Nearest Neighbor)

A k-nearest neighbor (KKN) model is a supervised learning classifier that uses a non-parametric function to plot datapoints, and classifies them based on their proximity to other datapoints. It is a very simple and widely used model. The results of this classification model are displayed in Table 14 below.

Table 14. KNN Model Results

	Precision	Recall	F1-Score
0 (Not Delayed)	0.81	0.88	0.85
1 (Delayed)	0.41	0.28	0.33
Accuracy	0.75		
Macro Avg	0.61	0.58	0.59
Weighted Avg	0.73	0.75	0.74

The KNN model got a precision score of 0.81 for the flights that are not delayed (0) which suggests that out of all of the non-delayed flights that were predicted by the model, 81% are actually not delayed. The recall score is at 0.89 indicating that out of all of the flights that were not delayed, 89% were properly identified as not delayed.

For the flight delayed class (1) the precision score is 0.41 this means that out of all of the data predicted as delayed, 41% were delayed. And recall score is 0.28 indicating that out of all the instances that are delayed, only 28% were properly identified as delayed.

The KNN performs better at predicting non-delayed flights (0) compared to delayed flights (1). While it has a high precision and recall score for non-delayed flights, it struggles to properly identify delayed flights.

Summary of All Model Results

The aggregation of the weighted average results of each machine learning model can be found below in Table 15.

The weighted average of precision, recall, and f1 score take the size of each predicted class into account. It is apparent that the Random Forest model has the highest scores and is highlighted in green. The scores of each model that give equal weight to the delayed and on time predicted class can be found in Table 16. The gradient boosting model performs best, with a f1-score of 0.62.

Table 15. Weighted Average Model Results Summary

Model Weighted Average	Precision	Recall	f1-score
Logistic Classification (without Weather)	0.73	0.60	0.64
Logistic Classification	0.71	0.76	0.67
Decision Tree	0.71	0.71	0.71
Decision Tree w/ RandomSearchCV	0.74	0.79	0.73
Random Forest	0.76	0.79	0.74
Gradient Boosting	0.76	0.68	0.70
Neural Network	0.75	0.79	0.75
K-Nearest Neighbors	0.73	0.75	0.74
Random Forest (ATL)	0.79	0.82	0.77
Neural Network (ATL)	0.77	0.81	0.76

Table 16. Macro Average Model Results Summary

Model Macro Average	Precision	Recall	f1-score
Logistic Classification (without Weather)	0.57	0.60	0.55
Logistic Classification	0.66	0.52	0.48
Decision Tree	0.58	0.58	0.58
Decision Tree w/ RandomSearchCV	0.68	0.55	0.54
Random Forest	0.71	0.55	0.55
Gradient Boosting	0.62	0.67	0.62
Neural Network	0.68	0.57	0.58
K-Nearest Neighbors	0.61	0.58	0.59
Random Forest (ATL)	0.71	0.56	0.56
Neural Network (ATL)	0.68	0.54	0.52

Web Application

We developed a web application to create a more informed and prepared consumer using streamlit, an open-source Python framework. This application, called “Flight Delay Predictor” allows the user to submit their flight data to receive a prediction from our model as to whether their flight will be delayed or on time.

We imported our XGBoost Model (because it performed the best out of our models with a macro average f1-score of .62) to power our application. We also imported our flight dataset for several uses. First, all airport codes from the dataset are used to create a dropdown for the user to select from. Because we do not expect the user to know the distance of their flight, we use the dataset to find the distance between the two airports the user has inputted. The user is then asked to submit their flight's date and time information. We also do not expect the user to know the weather data at the origin and destination of their flight, so the application finds historical weather data at these locations during the selected month and calculates the average weather conditions. Screenshots of this application being used to accurately predict the status of flights that our model has not seen before are shown below in Figure 17 and Figure 18:

Figure 17. Application Predicts Flight On Time

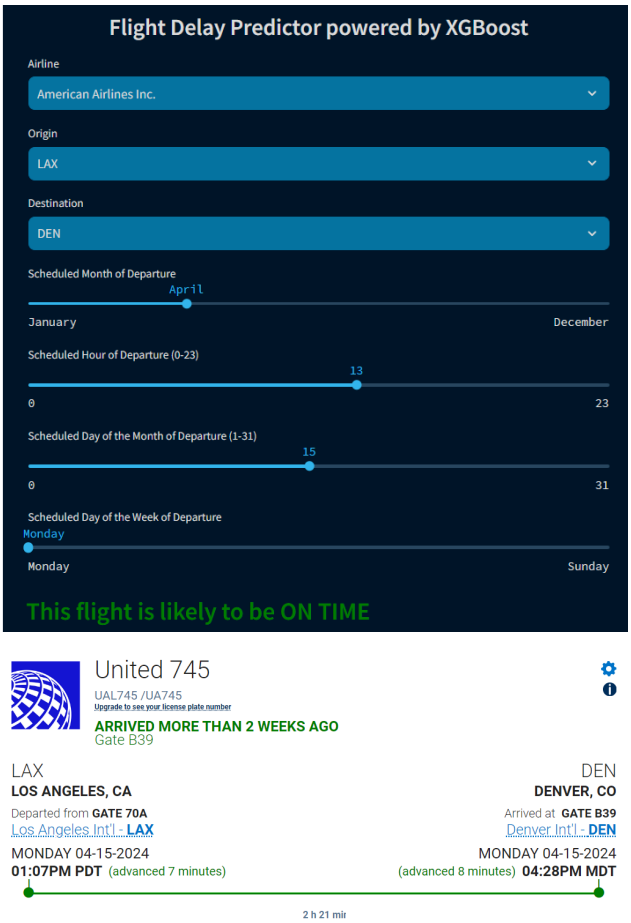


Figure 18. Application Predicts Flight Delay

Discussion

In our flight data, which has 1.2 million flights, there is a noticeable class imbalance concerning flight delays. Out of all of the instances in this dataset, only around 276,000 are identified as delayed while the majority that are considered on-time flights significantly outnumber them. This disparity in class distribution creates a challenge for predictive modeling tasks since models that are trained on imbalanced data often show biases toward the majority class while struggling to properly identify patterns and information from the minority class. In the context of our project, it was observed that while the models, including logistic classification, decision tree, random forest, gradient boosting, neural networks, and

k-nearest neighbors, perform well in predicting non-delayed flights, they face challenges in accurately predicting delayed flights. We believe one of the primary reasons contributing to this is the class imbalance present in our dataset.

Additionally, running complex models with a large dataset became challenging for our team due to long training times and lack of computational resources. Dealing with a smaller dataset would offer an easier environment for developing more complex models that could identify different patterns and relationships, thereby enhancing the accuracy of our predictions for delayed flights. This issue often limited our ability to run extensive grid searching techniques to optimize hyperparameters in our models.

While our selected features provided valuable insights into possible factors contributing to flight delays, they might not represent all the variables that can influence flight operations. The features that we selected were based on available data, but their predictive power for flight delays might be limited by the complexity of air travel dynamics. Moving forward, exploring alternative feature sets, adding more data sources, and employing advanced modeling techniques could offer a way to improve the accuracy and robustness of our predictive models for delayed flights. For example, we believe that having a unique plane identifier could identify which planes often have mechanical issues that lead to delayed flights.

When it comes to our application, we aim to enhance functionality by integrating weather data as an input the user can add. This addition allows the user to proactively assess potential flight delays caused by unforeseen weather events. This way our application becomes more adaptable, providing valuable insights into real-time conditions that could impact travel plans.

Conclusion

To conclude, while our models showed strong performance in predicting non-delayed fights, they encountered challenges in accurately predicting delayed flights. Despite the challenges posed by class imbalance, the need for more relevant features was identified as a significant factor influencing predictive accuracy. Tackling these issues by employing techniques such as oversampling, further feature engineering, and including more data sources like airport congestion levels or aircraft maintenance that could affect air travel. While our models have shown promising results in predicting non-delay flights, further work needs to be done to address the complexity of predicting delayed flights. Lastly, incorporating real-time data sources and more complex predictive analytics techniques could enhance the accuracy and reliability of the flight delay prediction model.

When examining all of our model's accuracy and weighted average f1-scores, they performed well with values in the high 70s and low 80s. However, when observing their precision and recall for the delayed class, there are apparent issues. This is more accurately captured in the average maco f-1 scores, as they provide equal weights to the precision and recall of both classes, regardless of their class balance. After using a wide variety of machine learning and deep learning techniques, we identified Extreme Gradient Boosting as the most effective method. While all of these models are effective in predicting on-time flights, XGBoost was the best model to predict delayed flights. We suspect that these shortcomings are mostly due to the data that we trained the model with, and not entirely the models themselves. Providing only features known before a flight takes off such as airline, origin, destination, date, and weather data is not enough information for a model to accurately capture a very complex national transportation system.

References

1) "Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the United States" , 2010

2) “Air Travel Consumer Report: December 2023, Full Year 2023 Numbers.” Air Travel Consumer Report: December 2023, Full Year 2023 Numbers | Bureau of Transportation Statistics,
https://www.bts.gov/newsroom/air-travel-consumer-report-december-2023-full-year-2023-numbers

Accessed 22 Apr. 2024.

3) "US Flights Cancel Due to Thunderstorms." The Economic Times, 6 Aug. 2022,
https://travel.economictimes.indiatimes.com/news/aviation/international/us-flights-cancel-due-to-thunderstorms/93390125.

Accessed 22 Apr. 2024.

4) “Flight Delays, Cancellations Could Continue for a Decade amid Airline Workforce Shortage.” CBS News, CBS Interactive,
https://www.cbsnews.com/news/the-future-of-flying-more-delays-more-cancellations-more-chaos/

Accessed 22 Apr. 2024.

5) FlightAware. "Flight history for flight UAL745 on April 15, 2024 from KLAX to KDEN." FlightAware,
https://www.flightaware.com/live/flight/UAL745/history/20240415/2024Z/KLAX/KDEN

6) FlightAware. "Flight history for flight NKS1777 on April 16, 2024 from KFLL to KEWR." FlightAware,
https://www.flightaware.com/live/flight/NKS1777/history/20240416/0017Z/KFLL/KEWR

Acknowledgments

We would like to extend our gratitude to Professor Ryan Lafler for their guidance and expertise in machine learning engineering. Their continued support, insightful feedback, and enthusiasm for teaching have been instrumental in shaping our research.

Contact Information

Riley Rutan is a student in the Masters of Science in Big Data Analytics program at San Diego State University, with a planned graduation date of Spring 2025. He received a Bachelor of Science in Physics from the University of California, Santa Cruz, and most recently worked as a business analyst for a healthcare technology company in British Columbia, Canada. Riley has interests in big climate data analytics, business analytics, healthcare data science, machine learning engineering, and data visualization.

email: riley.rutan@gmail.com

Eric Braga is a master’s student in the Masters of Science Big Data Analytics program at San Diego State University set to graduate in spring of 2025. He previously graduated with a bachelor's degree in Economics from California State University Northridge. He has worked for various businesses as a business analyst including a chocolate company in the Los Angeles area. He has an interest in the fields of urban planning, transportation, machine learning engineering, computer vision, and deep learning integration. He is working with The Smart Transportation Analytics Research (STAR) Lab.

email: ebraga890@gmail.com

Xinyu Du is a first-year master’s in the Big Data Analytics program at San Diego State University and plans to graduate in Spring 2025. He received a Bachelor of Science degree in Aerospace Engineering from the University of California, San Diego in 2022. Xinyu has interests in the fields of data visualization, business analytics, image processing, and machine learning.He is working with the Computer Vison Lab.

email: xinyu.xydu@gmail.com

Miguel Ángel Bravo Martínez Del Valle is a first-year student at San Diego State University, in the Big Data Analytics Master’s program, set to graduate in Spring of 2025. He took a bachelor's in Electronics, Robotics, and Mechatronics Engineering at the University of Málaga, back in Spain. Miguel has an

interest in Data Analytics, Machine Learning, and Artificial Intelligence. He is currently in two Research Groups, one in Data Visualization working with an Ireland company and he is also in the AI4Businnes team, starting his research in Spring 2024.

email: miguelangelbravo2000@gmail.com

Fernanda Carrillo is pursuing her graduate studies in Big Data Analytics at San Diego State University, with an expected graduation in Spring 2025. She holds a bachelor's degree in International Business, specializing in English and North America, with a focus on Management Information Systems, also from San Diego State University. With experience in both the technology and finance sectors, Fernanda has primarily worked in business operations roles. Currently, Fernanda is an active member of the Metabolism of Cities Living Lab, where she contributes to advancing synthesis science, enhancing data accessibility, and mentoring future scientists in aligning with the UN Sustainable Development Goals. Her passion lies in the intersection of machine learning engineering and leveraging data science for healthcare and social impact initiatives.

email: fernandacarrilloe@gmail.com