Predicting Flight Cancellations

Project Overview

Flight cancellations have significant economic and logistical impacts, costing the aviation industry billions of dollars annually. Cancellations affect both airline operations and passenger satisfaction, often leading to cascading effects such as missed connections, rebookings, and logistical complications. This project aims to predict the likelihood of flight cancellations based on historical flight performance data and weather conditions. Accurately forecasting cancellations will enable airlines, airports, and passengers to take proactive measures, minimizing disruptions and enhancing decision-making for all involved. This project focused on the top 20 airports with the highest cancellation volumes in 2023. By narrowing the scope to these key airports, I obtained hourly weather data for each origin-destination pair to use for analysis and modeling.

High Level Overview of Results

For this binary, imbalanced classification problem focused on predicting flight cancellations, I evaluated several ensemble models, including Balanced Random Forest and Easy Ensemble, along with baseline models like Logistic Regression and Random Forest. Using 3-fold StratifiedKFold cross-validation and a custom grid search function, I optimized hyperparameters and tested different class imbalance handling techniques such as undersampling with RandomUnderSampler and NearMiss. The Balanced Random Forest Classifier was selected as the best model, achieving a near-perfect recall (0.9997) and a strong AUPRC (0.380), crucial for minimizing false negatives in cancellation detection. Although precision was lower (0.104), the model's ability to capture true positives justified this trade-off. Key features influencing the model included prior cancellation counts and time-related and weather variables, aligning with trends identified during exploratory analysis.

Data

Data for this project have been collected from several sources. The table below provides links to the sources.

- **Airline Performance:** Monthly airline performance data for 2023 comes from data published through the U.S. Department of Transportation (DOT), Bureau of Transportation Statistics. This data is available for download as csv files for each month and provides information such as flight schedules, on-time performance, cancellations, origin and destination details, and details about delays.
- **IATA Codes:** Data for the IATA codes, airline names, and DOT IDs was downloaded from the DOT. This table would be used to associate IATA codes with their respective airline providers.
- **Airport Details:** Data regarding the names and geographical information for all airports was downloaded from the DOT. Specifically the latitude and longitude data is what would be required in order to gather appropriate hourly weather data.
- **Weather:** Hourly weather data for 20 airports comes from an open source provider, Open Meteo. Data was downloaded using an API connection and specifying the locations and

variables of interest. Weather data includes information such as precipitation, wind direction and speed, snowfall, and weather codes.
- **Aircraft Registration:** Annual aircraft registration data was downloaded to provide details about aircraft types and registration.

Data Table	Source	Other Details
Marketing Carrier On-Time Performance (Beginning January 2018)	U.S. DOT, Bureau of Transportation Statistics	• Link to data definitions • Downloaded monthly performance data for all 2023 flights
L. Airline ID (IATA Codes, DOT ID and Airline name)	U.S. DOT, Bureau of Transportation Statistics	• Link to data source
Aviation Support Tables : Master Coordinate	U.S. DOT, Bureau of Transportation Statistics	• Link to data source
Historical Forecast Weather Data	Open Meteo	• Link to data source • Downloaded hourly weather data for 20 airports
Aircraft Registration	Federal Aviation Administration	• Link to data source

---

Data Cleaning and Wrangling

Notebook Links:
- Importing Data
- Weather Data

Airline Performance

The airline performance data for 2023 contained information regarding 7,278,739 flights over a 12 month period. The original data had 120 columns, many of which would become unnecessary for further analysis. Of these flights, 93,897 were canceled, representing 1.29% of all scheduled flights in 2023. Another 17,797 flights were diverted at least once, representing 0.24% of all flights.

Missing Data

Fifty-eight columns were missing more than 50% of their values. Many of these had to do with details for diverted airport landings and whether the diverted flights involved codeshare partners. I dropped the columns that were missing 100% of their values. Other columns were used to fill in details on diverted flights and then dropped later. Of the remaining columns with a high volume of missing data, I kept the following: `CancellationCode`, `LateAircraftDelay`, `SecurityDelay`, `NASDelay`, `WeatherDelay`, and `CarrierDelay`.

I checked what columns were missing 50% or less of their values. Seventeen columns were missing between 0.3% and 1.5% of their values. I suspected that most of these had to do with canceled flights since if a flight was canceled, actual performance data would not be available. For example, a canceled flight would not have actual arrival time data.

Of the diverted flights, 16,049 reached their final destination. For diverted flights that reached their scheduled destination, I replaced the `ActualElapsedTime` and `ArrDelay` with the values from the Diverted columns. I did this because of the note in the data definitions table: "Elapsed Time of Diverted Flight Reaching Scheduled Destination, in Minutes. The ActualElapsedTime column remains NULL for all diverted flights." Flights that were diverted and never reached their final destination were removed. This was approximately 1,600 flights.

Airline Operations

The `Operated_or_Branded_Code_Share_Partners` column identified whether the flight was operated by one of the airline's code share partners. A codeshare flight is an agreement between airlines to sell seats on each other's flights. This gives the appearance of airlines flying to more destinations. By doing so, the airlines typically share the revenue on that ticket. The `DOT_ID_Operating_Airline` column should be used to identify which airline carrier is operating the flight. I created a new column, `Code_Share_Flight`, denoting whether a flight was a codeshare flight to allow for segregating flights based on differing operations. Just over 500 flights had different Marketing Airline Flight Numbers to Operating Airline Flight Numbers. This number was minimal so I dropped the Operating Airline Flight Number column.

Geographic Information

I reviewed the `OriginCityName` and `OriginState` as well as `DestCityName` and `DestState` for any inconsistencies with the states. I only found one inconsistency and replaced that city name with the appropriate name. The airport is marketed as being in Cincinnati, OH but is actually in Hebron, KY. I also removed redundant geographic information from the dataset but preserved the information for full state name, FIPS and WAC in a dictionary for origin and destinations using the state code as a key and the rest as a tuple for the value.

Departure and Arrival Performance

The `CRSDepTime` and `CRSArrTime` columns represent the scheduled departure and arrival times for each flight in the timezone for origin and destination respectively. These two columns originally showed the scheduled times in integer format, and these would need to be converted to datetime values for analysis later on.

For the `DepDelay` vs `DepDelayMinutes` and `ArrDelay` vs `ArrDelayMinutes` columns, the only difference is that in the Minutes columns, early departures or arrivals are set to 0 instead of a negative number. This is redundant information so I removed these columns.

All the rows missing `DepTime` are canceled flights. Similarly all the rows missing `ArrTime` are canceled flights. There are 3,797 rows that have `DepTime` data and are missing `ArrTime` data, and all of these flights were canceled. It is likely that these flights were going to take off, left the gate, but did not actually get in the air and were subsequently canceled, therefore they would have departure data but not arrival data. I replaced the `DepTime` for these rows with np.nan values since they are all canceled flights.

I wrote a function `create_scheduled_actual_datetime` that would create new columns for scheduled departures and arrivals and actual departures and arrivals that were datetime values. First, I had to convert the 2400 values in the `CRSDepTime` and `CRSArrtime` columns (scheduled time columns)

to 0 so they would be read as midnight for the datetime conversion. Then, I had to concatenate the `FlightDate` with the schedule time columns and convert these to datetime objects, `scheduled_departure_datetime`, and `scheduled_arrival_datetime`. I had to adjust the `scheduled_arrival_datetime` for next day arrivals by identifying flights that had an arrival time before the departure time; these were adjusted by adding +1 day. Next, I replaced the actual departure times for canceled flights with null values. Finally, I calculated actual departure and arrival datetime objects. I filtered the dataframe for completed flights only, created the new columns, `actual_departure_datetime` and `actual_arrival_datetime`, based on the scheduled time + the `DepDelay` or `ArrDelay` and added these new columns to the original dataframe.

IATA Codes

IATA Codes data was for 1,737 airlines, more airlines than are represented in the flights performance data. I created a dictionary from the IATA dataset to map the `Airline_Mkt` and `Airline_Ops` columns from the flight performance data to the dictionary values, or the airline names.

Airport Details

The airports dataset had 19,197 rows of data representing airports from around the world. However, these were not all unique airports. This dataset also contained information about each airport's latest geographical information and previous information. I had to first filter for airports in the United States and then by `AIRPORT_IS_LATEST` == 1 and `AIRPORT_IS_CLOSED` == 0. This left me with 2804 airports. I dropped rows that were missing latitude and longitude data.

One of the things that would become essential later on was to merge flight performance data with hourly weather data. The hourly weather data was available in UTC time. All of my flight data's scheduled and actual departure and arrival times were in the local times for the origin and destination respectively. In order to compare flight performance data with weather data, all times would need to be converted to UTC and then rounded to the nearest hour.

First, I had to find the timezones for each airport. I created a function, `assign_timezone` that uses `TimezoneFinder()` from the `timezonefinder` library. Now the airports dataframe had time zones for each airport.

Next, I created a function, `add_timezone` that would add the `Origin_Timezone` and `Destination_Timezone` columns to the flights data by using left joins on the airport IDs and airport code columns.

Finally, I wrote a function `convert_flight_times_to_utc` that converts flight scheduled and actual departure/arrival times to UTC and rounds the scheduled times to the nearest hour for weather data joining. The function handles timezone conversions for both departure and arrival times based on the airport timezones by localizing the datetime columns first based on the `Origin_Timezone` and `Destination_Timezone` columns. Then it creates new datetime objects for the UTC times. It also addresses ambiguous and nonexistent times during daylight savings transitions. It returns 4 columns, `scheduled_departure_datetime` and `scheduled_arrival_datetime_utc` rounded to the nearest hour and `actual_departure_datetime_utc` and `actual_arrival_datetime_utc`.

Aircraft Registration

Aircraft registration data was downloaded as zip files by calendar year. Each folder had 7 txt files. I only needed the data on the aircraft registration, the aircraft, and the engine. The aircraft registration file contained data on 293,465 registered aircrafts. First, I filtered this by only looking for tail numbers that were in the airline performance dataset. Just over 19,000 flights were missing tail numbers, so they could not be merged with aircraft registration data.

Initially I only downloaded the data for 2023, however, I was missing registration data for 53 planes, so I also downloaded 2022 and 2021 in case they would have the missing information. This resolved some of the missing data but not all, I was still missing data on 22 planes.

I merged the aircraft registration data with the aircraft and engine data and saved this file as a csv. I have not yet used this for analysis due to time constraints.

Weather

Weather data was downloaded after conducting Exploratory Data Analysis on all 2023 flights. Due to the processing power and API call limitations, I could not download hourly weather data for all U.S. flights. I limited the further analysis, preprocessing and modeling to the top 15 airports for the highest volume of cancellations and the remaining 5 for the most delays that were not already included in the top 15 cancellations. More details about this selection process will be covered in the EDA section of the report.

Once airports were identified, I created a function, `get_hourly_weather_data`, to retrieve hourly data for each airport and then created separate dataframes for each airport's hourly weather data. First the function makes an API call to Open Meteo by specifying the start date, end date, features, and units for the data to be downloaded. The call was made by iterating through latitude and longitude pairs for each airport. The json file was then parsed into its metadata components and the hourly weather data as a new dataframe for each airport. Finally, it added the `airport` and `airport_id` to each dataframe so they could be merged with the flight performance data during the secondary phase of EDA.

Initial Feature Engineering

Notebook Links:

- Historical Performance

Calculating Historical Performance for Unique Flights

Each row in the flights dataset represents a scheduled flight, containing details such as flight date, carrier, origin, destination, scheduled departure time, arrival time, and performance metrics like delays and cancellations. To extract meaningful historical insights for each flight, I performed rolling window calculations to gather statistics based on recent flight history over a certain period of time. The idea for this aggregation approach to create new features was adapted from Ashish Jain.

Objective:

I computed historical performance statistics—such as delays, cancellations, and diversions—for each flight, using its recent history. For example, for a flight from Las Vegas (LAS) to Charlotte (CLT) on August 6th, 2023, at 11:59 PM, I wanted to know how many times a flight on the same route, operated by the same airline, and within the same time window, was delayed or canceled over the past 10 days, 20 days, 30 days and 90 days.

Key Variables Defining a Unique Flight

1. Carrier: The airline operating the flight.
2. Origin: The departure airport.
3. Destination: The arrival airport.
4. Departure Window: A specific time range during which the flight departs (e.g., morning, afternoon).

For each flight, I gathered historical data based on these four variables.

Process Overview:

1. **Grouping by Flight Attributes:**
   First step is to group flights by their carrier, route (origin-destination pair), and departure window.
   - Use the function `create_route_ids` to create a new dataframe containing unique routes made up of origin destination pairs and assign each route a `route_id`. This id was then added to each flight in the flights dataframe.
   - Use the function `create_time_windows` to create a new categorical column called `departure_window` with the categories and times below. These windows are based on the scheduled departure time in the local timezone.

Category	Time Range
Overnight	12 AM - 4 AM
Early morning	4 AM - 6 AM
Morning	6 AM - 11 AM
Midday	11 AM - 1 PM
Early afternoon	1 PM - 3 PM
Afternoon	3 PM - 5 PM
Evening	5 PM - 7 PM
Night	7 PM - 10 PM
Late night	10 PM - 12 AM

2. **Rolling Window Aggregations:**
   Next, for each flight, use the `calculate_flight_performance_aggregations` function calculate performance statistics (e.g., delay metrics, cancellations, and diversions) over rolling time windows (e.g., 10, 20, 30, and 90 days) that look back from the flight's scheduled departure date. The statistics include:
   a. Median and maximum departure/arrival delays
   b. Count of canceled flights
   c. Count of diversions
   d. Count of flights

3. **Handling Duplicate Data:**
   The aggregations are done for all specified time windows for each flight in the flights dataset. Some duplicates occurred when an airline had multiple flights on the same route and in the same time window. This caused the merge with the original flights dataset to result in more rows than expected, as one flight could be matched with multiple rows of aggregated data. After computing the rolling window statistics, I optimized the deduplication process by sorting the aggregated dataframe by key columns (`route_id`, `airline_mkt`, departure_window, `scheduled_departure_datetime`) and keeping the row with the highest number of n_flights. This ensures that for each combination of key columns, the row with the most complete flight data (i.e., the highest number of flights) is retained.

   This deduplication was applied with the function, `drop_agg_duplicates` before merging the rolling statistics back into the original flights dataframe.

4. **Time Windows:**
   Flights departing at similar times are grouped into "time windows" (e.g., early morning, afternoon), allowing for tracking consistent patterns in flight operations over time. This reduces the granularity of individual departure times and simplifies grouping.

5. **Imputed Missing Data:**
   There were 1,428 flights missing aggregate information for 10 day periods 1,213 for 30 day periods and 1,029 for 90 day periods . Of these 714 were flights in January. Since I only pulled data for January through December of 2023, there was not enough information before January 2023 to create aggregate information. I also identified 134 flights that were rare and canceled. Rare flights were defined by having fewer than 5 unique flights during the entire calendar year. I decided to impute values for flights missing performance data. I aggregated the rolling statistics by `airline_mkt` and `route_id` only (excluding `departure_window`). I then calculated the mean for each rolling statistic. Once these were calculated, I imputed the missing statistics for the flights missing historical performance data. After imputation, only 8 flights were still missing data. I drop these later during pre-processing.

Optimized Approach:

Rather than looping through each flight row by row, I leveraged vectorized operations with `pandas` to group flights by carrier, route, and departure window and applied rolling windows efficiently. This dramatically reduces computation time and allows the process to effectively scale large datasets with millions of rows. After calculating the aggregations, I merged the results back into the original flight dataset.

By combining these rolling window statistics with the original flight data, I enhanced the dataset with valuable historical features that can be used for further analysis and modeling.

Exploratory Data Analysis

Notebook Links:

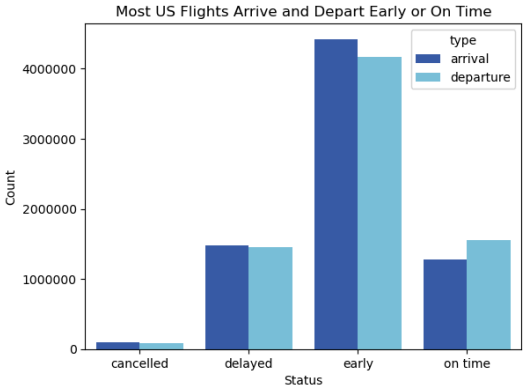
- Exploratory Data Analysis (all flights)
- Exploratory Data Analysis (top 20)

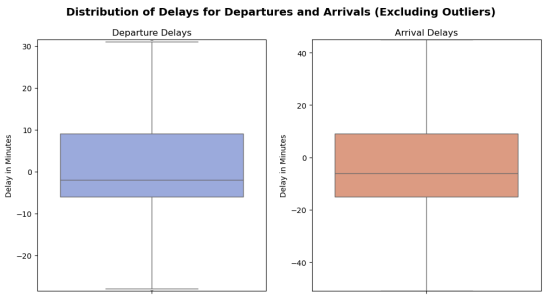
Exploratory Data Analysis was performed on all flights and then a subset of flights based on airports with the highest volume of cancellations and delays. This approach was taken for two reasons. First, I wanted to identify the features that affect cancellations across all flights. Second, I would not be able to get hourly weather data for all 359 locations due to API restrictions and the amount of data that would require.

Overall Data Attributes

There were 7,276,990 flight records and 84 features after performing initial cleaning of the 2023 flights performance dataset. Of these flights 7,183,093 were completed flights and 93,897 were canceled, representing a cancellation rate of 1.3%.

Most U.S. flights depart early or on time, with 57% departing early and 21% departing on time. Similarly, 61% of flights arrive early and 18% arrive on time. Approximately 20% of flights experience departure or arrival delays, representing just over 1.4M flights.





The Bureau of Transportation Statistics codes cancellations and delays using the following definitions[1]:

- Air **Carrier**: The cause of the cancellation or delay was due to circumstances within the airline's control (e.g. maintenance or crew problems, aircraft cleaning, baggage loading, fueling, etc.).
- Extreme **Weather**: Significant meteorological conditions (actual or forecasted) that, in the judgment of the carrier, delays or prevents the operation of a flight such as tornado, blizzard or hurricane.
- National Aviation System (**NAS**): Delays and cancellations attributable to the national aviation system that refer to a broad set of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume, and air traffic control.
- **Late-arriving** aircraft: A previous flight with same aircraft arrived late, causing the present flight to depart late.
- **Security**: Delays or cancellations caused by evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas.

[1] Understanding the Reporting of Causes of Flight Delays and Cancellations [source]

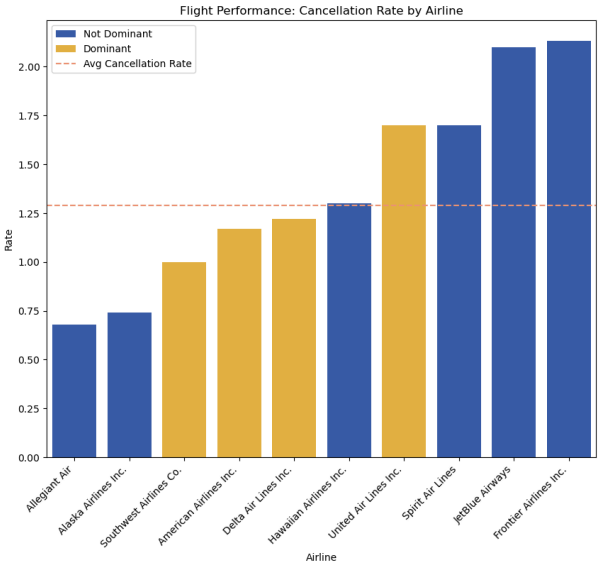
Of canceled flights, 55% of them were canceled due to extreme weather, 28% due to the carrier, 16% due to the National Air System and 0.2% due to security reasons.

The majority of flights depart between 6 minutes early and 9 minutes late and arrive between 15 minutes early and 9 minutes late. There are some extreme outliers for departure and arrival delays. Departure delays more than 31.5 minutes are considered outliers and arrival delays over 45 minutes are considered outliers, demonstrating that most delays are less than 1 hour long. Outliers range from 2 hours to 4 days. Approximately 895,000 flights experienced departures that are considered outliers and 662,000 experienced arrivals that were considered outliers.

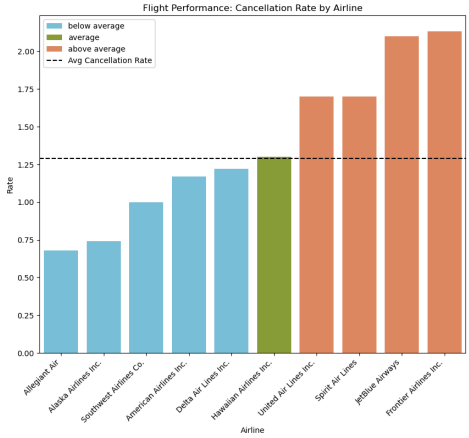
Airlines

There are 10 airlines that are marketed as operating flights in the United States. (Later I'll discuss the difference between a marketing airline and an operating airline.) Four of these airlines hold 82% of the market share of all completed flights, American Airlines (24%), Delta Air Lines (20%), Southwest Airlines (20%), and United Air Lines (18%). The other 6 airlines only hold 18% of the market share.

Three of the four dominant airlines have cancellation rates that are lower than the average cancellation rate of 1.29%. United Air Lines has a cancellation rate that was significantly higher than the average. Other airlines with higher than average cancellation rates included Frontier Airways, JetBlue Airways, and Spirit Air Lines. A one-way ANOVA test showed that the difference in cancellation rates among airlines was statistically significant with a p-value close to 0 and an F-statistic of 774.37. I then conducted a one-sample t-test for each airline comparing its cancellation rate to the average cancellation rate. I applied a Bonferroni correction by adjusting the p-value (0.05) by dividing this significance level (α) by the number of comparisons (number of airlines) to account for the number of tests I conducted. All but one airline had statistically significant differences in their cancellation rates compared to the overall cancellation rate. Hawaiian Airlines was the only airline with a cancellation rate that did not differ significantly from the average rate.



I conducted a Chi-Square test to compare airlines vs airline cancellation performance categories as predictors for cancellations. Although both are statistically significant, individual airlines are stronger predictors of cancellations than their airline cancellation performance grouping. The **difference in Chi-Square values (6962.65 vs. 5581.47)** highlights that individual airlines explain more variability in the cancellation rates than the grouped categories do.



Codeshare Flights

A codeshare flight is an agreement between airlines to sell seats on each other's flights. This gives the appearance of airlines flying to more destinations. By doing so, the airlines typically share the revenue on that ticket. I used the `airline_ops` column to identify which airline carrier is operating the flight. I created a new column, `code_share_flight` to distinguish whether a flight was a codeshare flight to allow for segregating flights based on differing operations.

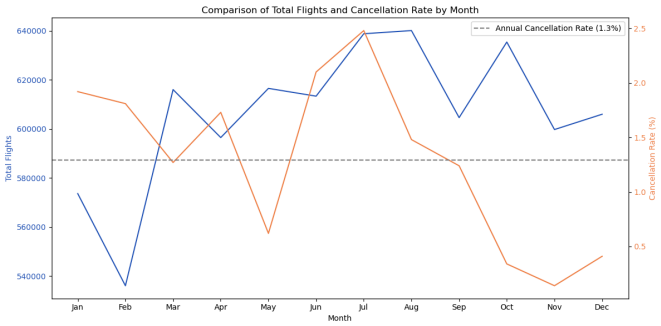
Four airlines have at least 1 codeshare partner: American (7), United (6), Delta (3), and Alaska (2). All other airlines marketed and operated their own flights in 2023. As a group, codeshare flights experienced more cancellations (1.5%) than non-codeshare flights (1.19%). A Chi-Square test demonstrated that this difference was statistically significant (**p-value < 0.0001**). Although statistically significant, the effect size, as measured by Cramér's V, was **040146**, indicating a weak association. This implies that while codeshare status is associated with an increased likelihood of cancellations, the practical impact may be limited.

Temporal Factors

Month

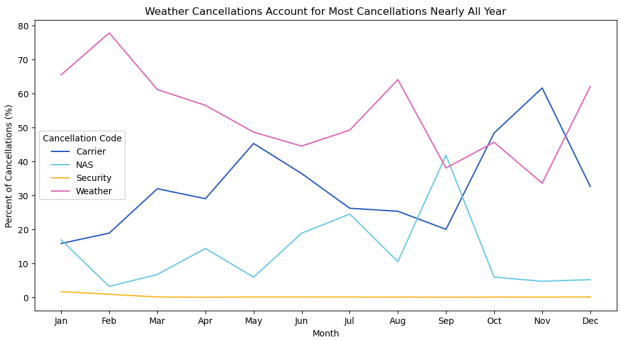
The chart below compares the total number of flights (blue line) and the cancellation rate (orange line) by month. The monthly cancellation rate peaks in the early part of the year (January to March) and then steadily declines, reaching its lowest point in November before slightly rising towards the end of the year. The total number of flights varies throughout the year, with noticeable dips in February and peaks during the summer months (June to August). The dashed line indicates the annual average cancellation rate of 1.3%. A Chi-Square test demonstrated that the relationship between month and cancellations is significant.

What is interesting is that the busiest travel months are not necessarily correlated with peak cancellations. There seems to be an inverse relationship between the total number of flights and the cancellation rate, particularly noticeable in the second half of the year, where higher flight volumes align with lower cancellation rates.



The chart below shows the monthly distribution of flight cancellations by reason: Carrier, National Air System (NAS), Security, and Weather. Weather is the leading cause of cancellations for most of the year, peaking at over 70% in January and remaining significant until late summer, before seeing a resurgence in December. Cancellations due to carrier issues fluctuate throughout the year, peaking in the fall (October and November) when they surpass weather as the leading cause.

NAS cancellations are consistently lower, with slight increases throughout the year until they drop off in the last quarter.

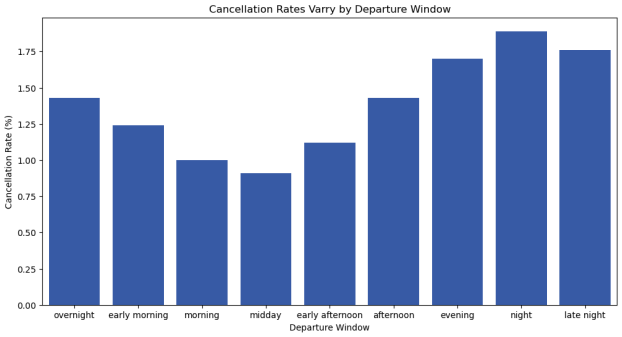


Security-related cancellations remain negligible across all months.

Weather has a strong impact on flight cancellations, especially in winter and late summer, while carrier issues become more prevalent during certain months.

Departure Time Window

The chart shows the percentage of flight cancellations segmented by different departure windows throughout the day. The night and late night time frames exhibit the highest cancellation rates both hovering around 1.75%. Morning, Midday and Early Afternoon departure windows have the lowest cancellation rates suggesting that flights scheduled during these times are less likely to be canceled.

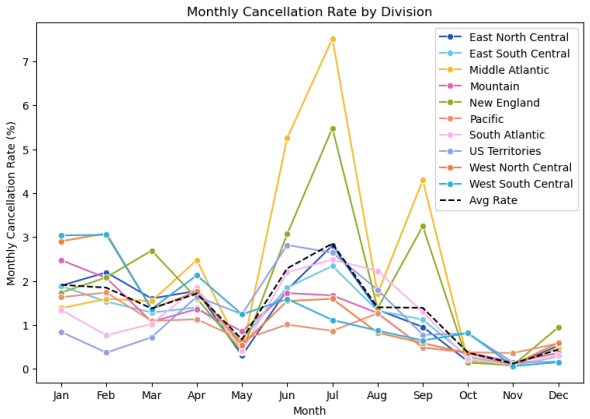


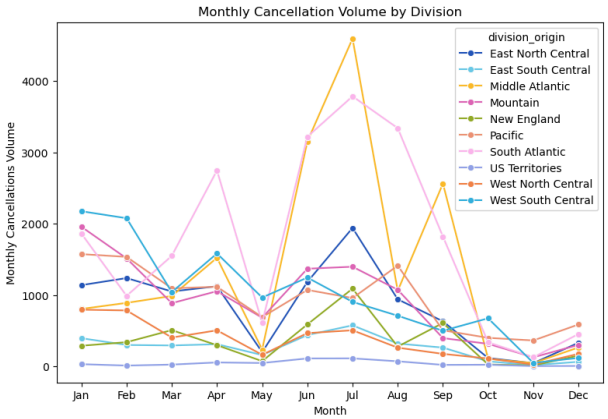
The higher cancellation rates during the night and late night hours could be due to factors such as limited crew availability, reduced operational capacity at airports, or carry-over effects from earlier disruptions in the day. Airlines may want to consider these trends when scheduling flights. They could plan more buffer time between flights during high-risk windows or allocate additional resources to minimize potential delays and cancellations.

Geographical and Temporal Factors

The Middle Atlantic and New England divisions, which make up the New England region, face the highest cancellation rates in June and July with rates double that of other divisions. West North and South Central divisions, which make up the Midwest region, have the highest cancellation rate during winter, which is consistent with that regions' weather patterns of colder temperatures, more snow, and higher speed winds.

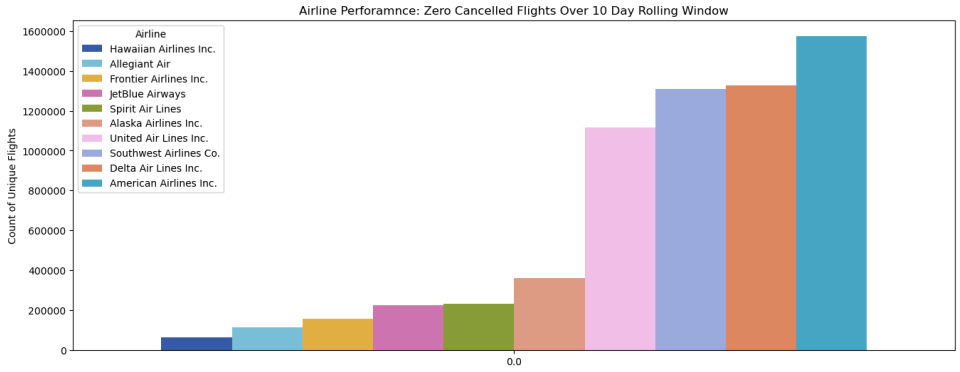
Although the South Atlantic region has the second highest cancellation volume, it has one of the more stable cancellation levels throughout the year. South Atlantic cancellations peak during the spring and the summer.





Historical Performance

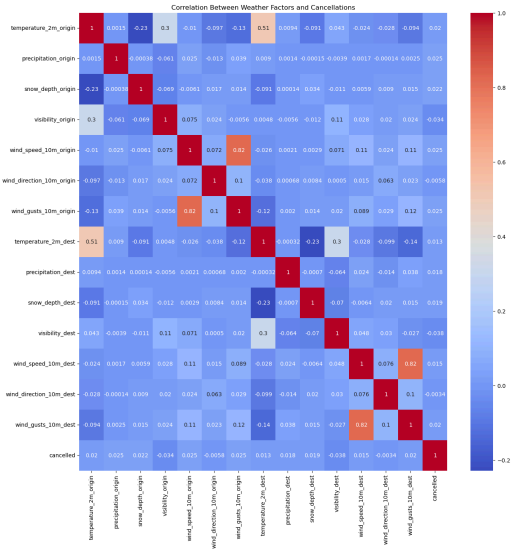
Unique fights that have cancellations within a 10 day window are considered outliers. The chart below shows that the majority of flights (5M+) do not experience cancellations over 10 day windows. Similar trends were shown with 30 and 90 day windows.



Unique flights that did experience at least one cancellation in the previous 10 days experienced much higher cancellation rates compared to unique flights that did not experience cancellations over a 10 day period.

Weather

Individual Origin and Destination Weather Variables and Cancellation Rates



This correlation matrix visualizes the relationships between various weather factors at both the origin and destination airports and the `cancelled` variable, which represents flight cancellations.

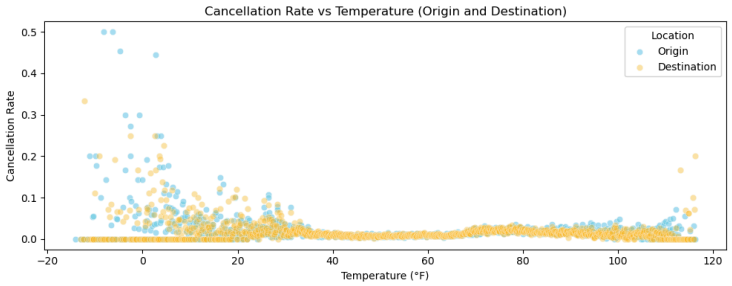
Most weather factors show weak correlations with cancellations, as indicated by values close to zero. While weather does have an effect, it might not be strong when considering individual variables alone.

To better understand the role of weather in flight cancellations, further analysis involving multivariate techniques or interaction effects may be needed to capture the combined influence of these weather variables.

This correlation matrix provides an overview of how weather conditions at the origin and destination airports relate to flight cancellations, emphasizing the need for more complex modeling to capture the interactions between factors.

Temperature

Flight cancellations tend to increase at temperature extremes, particularly in very cold weather (< 15 degrees F). This aligns with known impacts of extreme temperatures on airport operations and aircraft performance.

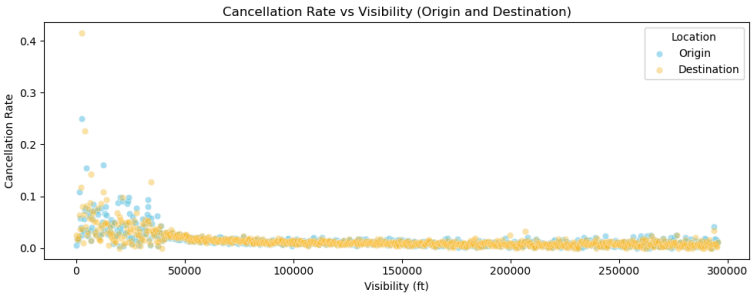


Precipitation

Flight cancellations increase with increased precipitation levels. Although the correlation is not very strong, it does impact operations at the origin and destination.

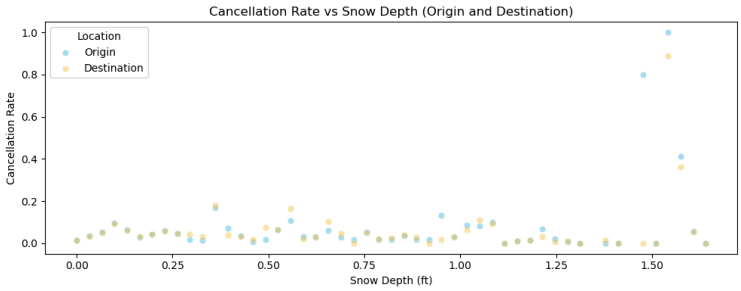
Visibility

Flight cancellations are sensitive to very low visibility conditions (< 50,000 ft.). This aligns with operational challenges faced during foggy or hazy conditions, which can disrupt flight schedules and safety. When visibility is within a normal range (> 50,000 feet), cancellation rates remain low and stable, indicating minimal disruption. Visibility at the origin and destination airports has a similar effect on cancellation rates, highlighting its consistent impact regardless of flight stage.



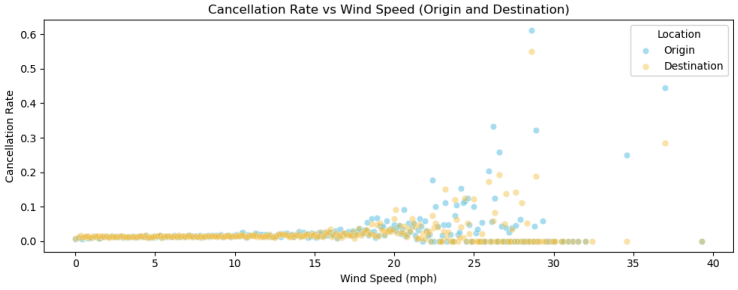
Snow Depth

The data shows that higher snow depth could be associated with a marginally increased cancellation rate. The oscillations at various depths could be related to different regions being able to handle varying snow depths.



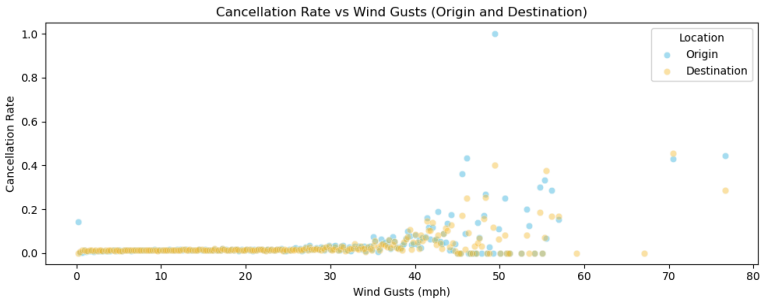
Wind Speed

Flight cancellation rates increase with higher wind speeds, especially above 20 mph, for both origin and destination locations. While most cancellations occur at lower wind speeds, there are noticeable spikes when wind speeds exceed 25 mph, indicating that strong winds contribute to higher cancellation rates.



Wind Gusts

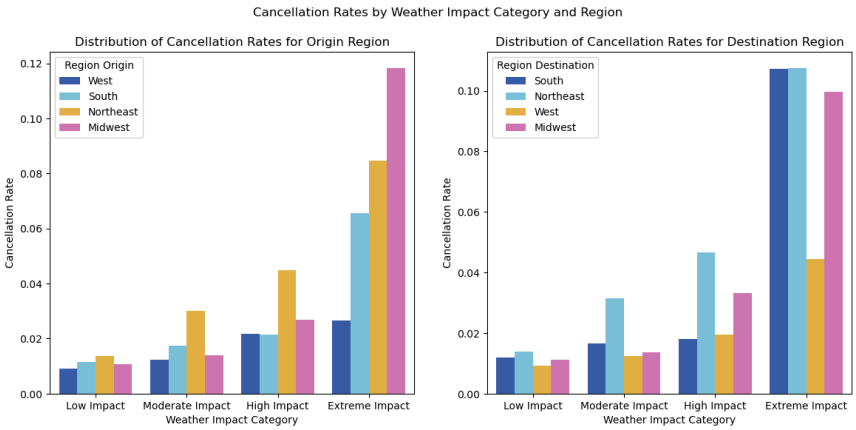
Flight cancellation rates increase significantly with higher wind gust speeds, particularly above 30 mph. Cancellations spike noticeably when wind gusts exceed 40 mph for both origin and destination locations, indicating that strong gusts are an impactful factor contributing to flight disruptions.



Weather Impact Categories

Each hourly weather datapoint has a feature for weather code. Weather codes represent different types of weather conditions[2] like rain, hail, haze, thunderstorms, freezing rain, and more. Initially I tried using the numerical representations themselves, however, more extreme weather does not necessarily mean a higher value. In order to address this, I created 4 weather code groups based on cancellation rates using a function called `weather_code_group`. The weather for each origin airport and destination airport was then assigned to each flight.

This visualization displays two bar plots comparing the cancellation rates of flights based on weather impact categories (e.g., Low, Moderate, High, Extreme) across different regions for both origin and destination airports. Cancellation rates increase as the weather impact category moves from Low to Extreme for both origin and destination regions. The Extreme Impact category shows the highest cancellation rates across all regions, indicating a significant effect of severe weather on flight operations.



Both origin and destination cancellation rates rise as weather impact severity increases. The Northeast and Midwest regions are more affected, particularly in extreme weather, while the West generally shows lower rates across categories. The South shows a significant increase in cancellations under Extreme Impact conditions, especially as a destination.

[2] Weather Code Table

Pre-Processing

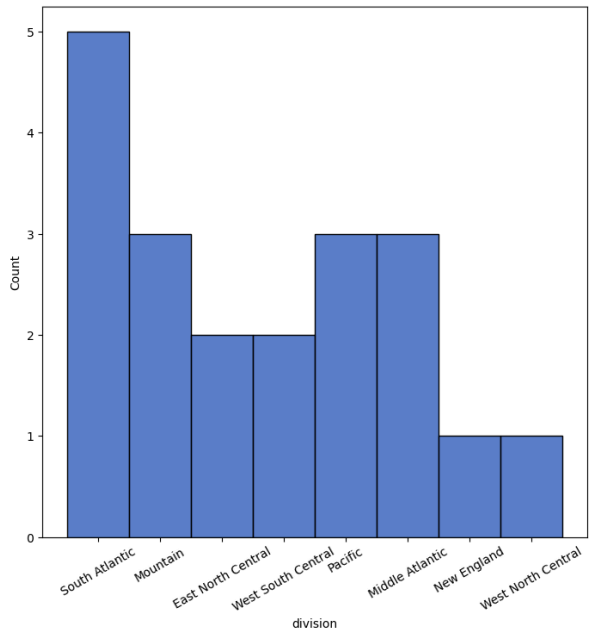
Notebook Link:

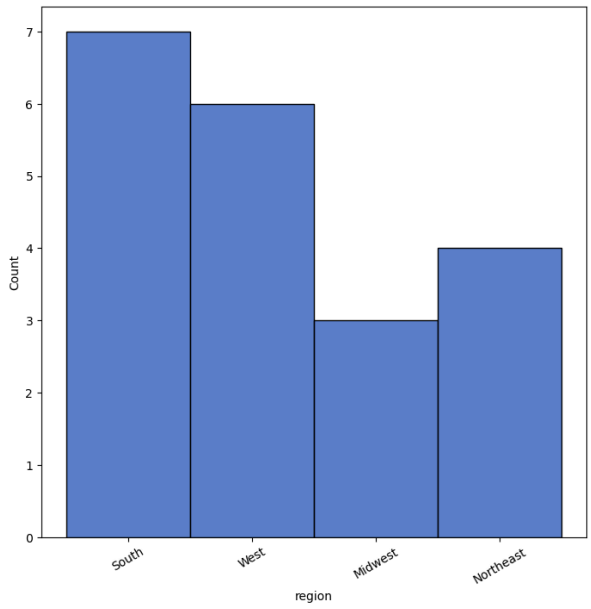
- Preprocessing

Top 20 Selection

I needed to select a subset of my entire dataset in order to combine flight performance with weather. There is a lot of overlap between the top airports in terms of total volume, total cancellations, and total delays. In order to select the airports I would use for the rest of the project, I decided to pick the top 15 airports based on total volume of flights. Then of the top 20 in terms of cancellations, I added the ones that were not already in the top 15 list. Then, I added the remaining airports based on top delays that were not already included in the top 15 for volume or top 20 for cancellations.

The charts below demonstrate how the top 20 airports are representative of the U.S. as far as including at least one airport for each division and region.





Feature Selection

I dropped columns that contained data that could only be acquired after a flight has reached its final destination. For example, columns with information on the amount of time a flight's departure or arrival was delayed, time it took to taxi in or out, cancellation codes, actual departure or arrival times, etc.

I only kept historical performance columns for a 10 day period instead of keeping them for 10 day, 30 day and 90 day rolling windows. Historical performance trends are similar across longer windows and therefore could introduce multicollinearity to the model.

Stratified Shuffle Split for Equal Class Imbalances

I used `StratifiedShuffleSplit` to ensure that I had equal class imbalances in my training data and testing data. Each split had the same cancellation rate, 1.44%. I concatenated the

Encoding

Ordinal Encoding
I used ordinal encoding for weather impact categories for origin and destination. During EDA, the data demonstrates how higher impact categories result in significantly higher cancellation rates compared to lower impact categories. Ordinal encoding ensures that the model understands the relationships between the categories and cancellations.

Target Encoding
I used target encoding for the following variables: `departure_window`, `month`, `origin`, `destination`, `region_origin`, `division_origin`, `region_dest`, `division_dest`, `airline_mkt`, and `airline_ops`.

In order to ensure there was no data leakage, I only used mean target encoding with the training data and then mapped the results to the appropriate columns in the testing data set.

I tested to see if region and division for origin and destination were highly correlated since divisions make up regions. The region and division had nearly perfect correlation with one another. I decided to remove origin and destination regions from the dataset and keep division since it was more granular.

Scaling

I applied StandardScaler to all numerical columns by fitting the scaler on the training data and then transforming the training and testing data. I then concatenated the categorical columns to the scaled numerical data to proceed with feature selection techniques.

Feature Selection

I wanted to check for any additional multicollinearity in the dataset so I used `variance_inflaction_factor` to calculate a VIF score for each feature. I removed features from the dataset with a VIF greater than 5.

I also used `SelectKBest` to limit the dataset to the top 10 features. I saved the reduced datasets as separate `X_train_reduced` and `X_test_reduced` to compare how less features affected model performance.

Train / Test Data Attributes

After reducing the number of features using VIF < 5, the final datasets had the following shapes:

- Train: 1,048,098 rows with 34 features
- Test: 449,185 rows with 34 features

The data with the reduced set of features using `SelectKBest` had the following shape:

- Train: 1,048,098 rows with 10 features
- Test: 449,185 rows with 10 features

Modeling and Model Evaluation

Notebook Link:

- Modeling

For this binary, imbalanced classification problem, I decided to try out several ensemble models that are well suited to handle class imbalances from the **Imblearn Ensemble** library, including Balanced Random Forest and Easy Ensemble. I also initially tested a base Logistic Regression Model, Random Forest Classifier, and Gradient Boosting Classifier.

Cross Validation Base Models

In order to understand the base level performance of these models, I used Cross Validation with 3 splits and measured the mean F1, recall and precision scores for each model. I also implemented StratifiedKFold splits to ensure each split had the same class imbalance ratio.

GridSearch

I created a function, `grid_search_model`, to evaluate several hyperparameter selections for the models that initially showed some promise during cross validation. The GridSearchCV was set up to score using recall, a cross validation that leveraged StratifiedKFold with 3 splits, and shuffled each split. The function takes the parameter grid along with a stratified subsample of the training data. I decided to use a subsample because the entire training dataset was taking several hours in addition to a lot of CPU to compute given the number of rows and features. The function returned the `best_estimator_` and `best_params_`. Once I identified the best hyperparameters, I trained the model on the full training dataset and evaluated it on the test set.

Different Class Imbalance Techniques

After training and making predictions using the models with their best hyper parameters, I also decided to try a few undersampling techniques from the **Imblearn Under Sampling** library including RandomUnderSampler, NearMiss (version 1), NearMiss (version 3), and TomekLinks. I proceeded with undersampling versus oversampling due to the size of the dataset (1M+ rows of data). Oversampling would have been inefficient given the volume of information on the negative class.

Evaluation

I created a function to evaluate each model and add results to a dataframe to compare different metrics across different models. The function captured the following metrics:

- Elapsed training time
- Precision
- Recall
- F1
- Accuracy
- Receiver Operating Characteristic Area Under the Curve (ROC AUC)
- Area Under the Precision Recall Curve (AUPRC)

Although I captured accuracy and ROC AUC as metrics, these are not very informative metrics when working with a highly imbalanced dataset. With the top 20 airports dataset, the cancellation rate is 1.4%. If the model were to guess not canceled 100% of the time, it would get an accuracy of 98.6%. Recall, F1 and AUPRC are the metrics I chose to prioritize for this project. More on that below.

The function also returned the predictions, prediction probabilities, and feature importances (if available). Below is a table of the model results after training models on the best hyperparameters.

Summary of Models & Performance

For this project, I made the assumption that correctly classifying as many true cancellations is the most cost-effective strategy for the target audience. I prioritized models based on high recall scores, capturing as many True Positives as possible. I also tried different probability thresholds to minimize the number of False Positives while not overly sacrificing recall.

Below is the results table ordered by recall and then F1 scores. I reviewed results for models that performed the best either based on recall score or F1 score for the positive class, cancellations.

Model	Accuracy	Precision	Recall	F1	ROC AUC	AUPRC	Train/Test Time
RF Random Undersample	0.876	0.104	1.000	0.188	0.971	0.362	3.5
Easy Ensemble	0.875	0.104	1.000	0.188	0.967	0.273	25.7
Balanced RF GridSearch	0.876	0.104	1.000	0.188	0.973	0.380	14.7
RF Tomek	0.878	0.105	0.999	0.191	0.971	0.371	55.0
Balanced RF Base	0.878	0.105	0.999	0.190	0.973	0.363	15.1
RF GridSearch	0.878	0.106	0.998	0.191	0.971	0.370	57.6
RF Undersample NearMiss 3	0.909	0.127	0.902	0.222	0.959	0.184	6.6
RF Undersample NearMiss 1	0.922	0.125	0.732	0.213	0.951	0.209	3.4
RF Base	0.988	0.858	0.183	0.301	0.977	0.528	74.8
GB Base	0.986	0.710	0.096	0.168	0.972	0.380	210.1

Model Selection

Optimizing for Recall and Precision: *Balanced Random Forest Classifier*

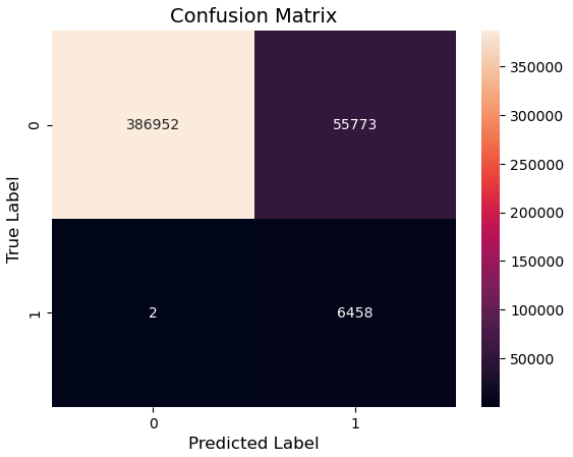
The model with a near perfect recall score of 0.9997 was the Balanced Random Forest Classifier Below is the classification report for this model.

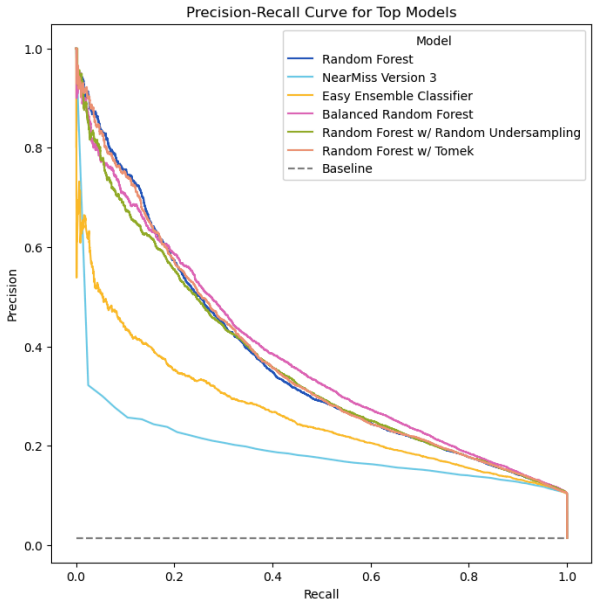
Classification Report: Balanced Random Forest Classifier				
Class	Precision	Recall	F1-Score	Support
0	1.00	0.87	0.93	442725
1	0.10	1.00	0.19	6460
Accuracy			0.88	449185
Macro Avg	0.55	0.94	0.56	449185
Weighted Avg	0.99	0.88	0.92	449185

Although recall is near perfect for the positive class, the confusion matrix demonstrates that of the 442,725 flights that were not canceled, the model predicted 55,773 of them to be canceled.

AUPRC (Area Under the Precision-Recall Curve) helps visualize the precision-recall trade-off across different probability thresholds. The Balanced Random Forest maintained the second highest AUPRC (0.379964), indicating it performs well when adjusting thresholds, allowing for better tuning based on operational needs (e.g., choosing a threshold that balances precision and recall more effectively).

The model with the highest AUPRC score of 0.528 achieves a reasonable recall of 0.95 when the probability threshold is adjusted to 0.65, resulting in 333 FN and 42,910 FP. The Random Forest model does not use any undersampling techniques, referred to as RF Base in the model evaluation table above.





The Balanced Random Forest model was selected for its superior ability to identify flight cancellations with high recall (0.9997) and the highest AUPRC (0.379964). This ensures that the model effectively detects cancellations, minimizing false negatives. While precision is lower, the trade-off is justified by the need to prioritize recall in this use case. The model's macro average recall score (0.94) and consistent F1-score further validate its suitability for the task, ensuring balanced performance across classes. Comparatively, this model outperforms others, making it the optimal choice for predicting flight cancellations.

Trade-Offs Between Precision and Recall

High Recall vs. Low Precision: Models like the Balanced Random Forest and Easy Ensemble Classifier prioritize recall, making them suitable for use cases where it's critical to detect all positive cases (cancellations), even if it means predicting more false positives. This trade-off is acceptable if the cost of missing a cancellation is higher than the inconvenience of a false positive.

Precision Impact: While precision is lower in high-recall models, it is important to understand the practical implications. For instance, a low precision means more resources may be used in verifying predicted cancellations that turn out to be false. However, in industries where preventing missed detections is more valuable, a high-recall, lower-precision model is justified.

Decision Analysis: Cost-benefit analysis could be used to find the optimal threshold that balances recall and precision for the most favorable outcome in practical scenarios. In order to proceed with this, I would need to understand the costs of cancellations posed to airlines and consumers. I'd want to investigate these two questions:

1. Does correctly predicting all true cancellations save airlines money?
2. What is the cost of incorrectly predicting a canceled flight?

Feature	Feature Importance
cancelled_sum_10D	62.82%
n_flights_10D	7.97%
month_encoded	5.95%
quarter	2.76%
visibility_dest	1.87%
visibility_origin	1.63%
temperature_2m_origin	1.59%
temperature_2m_dest	1.22%
day_of_month	1.07%
dep_delay_max_10D	0.95%
origin_encoded	0.89%
airline_ops_encoded	0.78%
distance	0.77%
wind_gusts_10m_origin	0.76%
wind_gusts_10m_dest	0.69%
wind_speed_10m_origin	0.68%
wind_speed_10m_dest	0.67%
hour_of_day	0.67%
route_id	0.66%
wind_direction_10m_dest	0.66%
wind_direction_10m_origin	0.66%
origin_weather_impact_category	0.58%
dest_weather_impact_category	0.50%
airline_mkt_encoded	0.50%
origin_division_encoded	0.46%
day_of_week	0.45%
precipitation_origin	0.45%
dep_window_encoded	0.45%
snow_depth_dest	0.33%
snow_depth_origin	0.28%
precipitation_dest	0.13%
code_share_flight	0.09%
div_airport_landings_sum_10D	0.03%
is_holiday	0.01%

Feature Importance

The best model's features and their importance can be seen to the left. The top predictive features align with the trends observed during exploratory data analysis. Flights that have had previous cancellations in the previous 10 days experienced significantly higher cancellation rates compared to flights that did not experience cancellations over specified window periods.

Months and quarters also showed that they were significant predictors during EDA and modeling. Similarly, several weather factors were also stronger predictors for the model.

Best Parameters

The best model's parameters were as follows:

- 'max_depth': None,
- 'max_features': 'sqrt',
- 'min_samples_leaf': 8,
- 'min_samples_split': 2,
- 'n_estimators': 100,
- 'sampling_strategy': 'auto'

Future Improvements

Feature Interactions: A potential future improvement for this project could be to understand and quantify the interaction between weather, time and historical performance features. For example, if visibility and temperature for origin and destination were multiplied, would it improve the model's ability to minimize false positives?

Predict Delays and Other Issues: Another potential future improvement for this project would be to extend the model's capability to predict not just cancellations but also significant delays or diversions, providing a comprehensive solution for airline operators.

Expand to All Airports: Data for modeling only included the top 20 airports for cancellations. Future expansion could include all airports to see how the model generalizes across other factors.

Credits & Thanks 🤗

I want to thank my mentor Jyant Mahara for coaching me throughout this process and making it feel less daunting.