# Prediction Model For Flight Cancellation Status

## Introduction

Air travel has become an essential part of modern life, connecting people all over the globe. However, the occurrence of flight cancellations can disrupt plans and inconvenience passengers, impacting both individuals and airlines. The ability to anticipate and mitigate such disruptions is crucial. Using machine learning, we created a predictive model that determines whether a flight will be canceled or not. We hope to aid airlines and travelers by providing information that can potentially make travel smoother and prevent issues regarding flight cancellations.

## 1 Dataset

### 1.1 Identify Dataset

The dataset we are using (Combined_Flights_2022.csv) is a collection of flights from 2022. Using this dataset, we hope to predict if flights will be canceled or not. The dataset was extracted from the Marketing Carrier On-Time Performance where we are just looking at the data from 2022 containing information on flight cancellations and delays with 61 columns although we will not be using all these columns. The main columns we will be using to conduct our analysis are 'Airline', 'Origin', 'Dest', 'Distance', 'Month', 'DayofMonth', 'DayOfWeek', 'DistanceGroup', 'CRSDepTime'. We noticed the most significant value in these columns when predicting flight cancellations.

### 1.2 EDA

**Basic Statistics**

The columns of this dataset represent data from flights from 2022 where each row is a separate flight. It includes information like date, departure delay, departure time, etc. There are missing values, where the data is Not Missing at Random because we could not get the data if the flight was canceled. To clean, we just dropped the missing values since whenever the flight was canceled, values in certain columns would not be populated.

We noticed that there is a major imbalance between canceled and not canceled flights. This makes sense because the majority of flights tend not to be canceled, and most flights end up departing. We still have to take this imbalance into consideration when constructing our model.

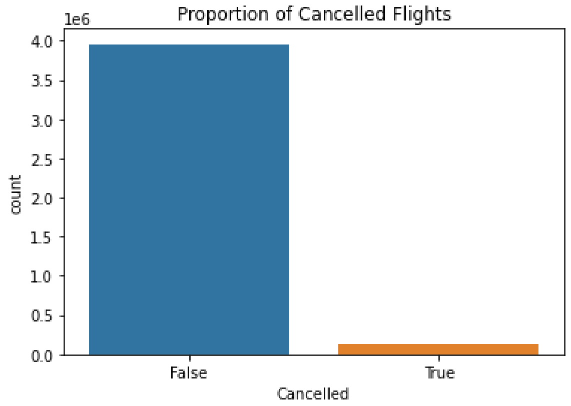


**Figure 1: Dataset Flight Cancellation Imbalance**

**Missingness**

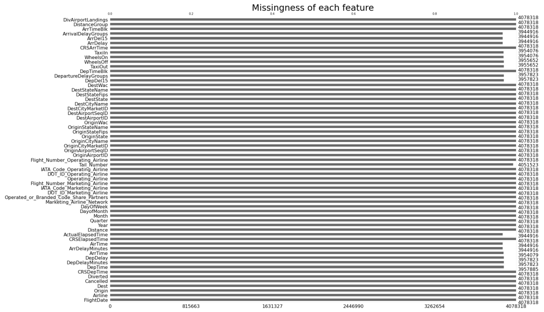


**Figure 2: Missingness of Columns**

We need to look at missing data to gain a better understanding of our dataset and how to build our model. We can see that the dataset is Not

Missing at Random because the missing values are not deliberately created as they would be in Missing by Design. The missingness in this situation is tied to flight cancellations rather than a deliberate part of the data collection method. The columns with the most missing data are ArrivalDelayGroups, ArrDel15, ArrDelay, WheelsOn, etc. This makes sense because these columns are relevant when the flight is not canceled meaning that a lot of this data is missing when the flight is canceled.

### Interesting Findings

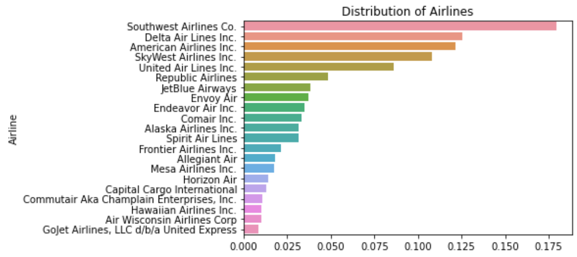


**Figure 3: Distribution of Airlines in Dataset**

In our flights dataset, we can view the distribution of airlines in the dataset to gain a better understanding of the prevalence of different carriers. Figure 3 shows the prevalence of 5 major carriers: Southwest, Delta, American Airlines, SkyWest, and United airlines. As these major carriers make up a substantial portion of the dataset, their impact on patterns and trends can be significant and it is important to keep that in mind as we predict cancellations and see if the carrier has anything to do with it.

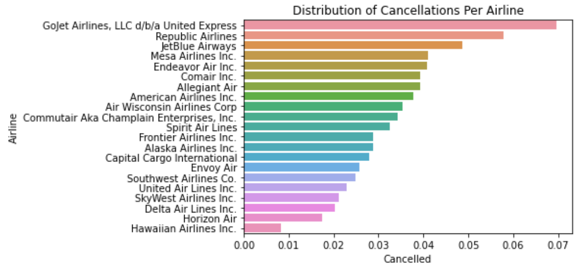


**Figure 4: Distribution of Cancellations By Airline**

Our examination of cancellations by airline shows that GoJet, Republic Airlines, and JetBlue stand out for a higher proportion of cancellations compared to other carriers. This could be due to potential operational challenges or other factors affecting their flight schedules. By incorporating a feature for airlines, we hope to capture the differences between carrieres to create a more accurate prediction model when predicting flight cancellations.

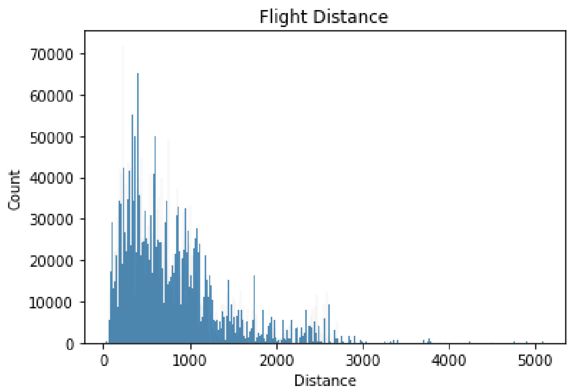


**Figure 5: Distribution of Flight Distance**

We decided to take a look at flight distance distribution to look at the variability of distances. Understanding the distribution of flight distances is important when creating a predictive model to predict flight cancellations. We can see that the distribution of flight distance is right skewed indicating that the data is skewed towards shorter flight distances. This is good to know when looking out for potential biases caused by the distribution's asymmetry. We would like to understand if flight distance has any impact on the flight being canceled.

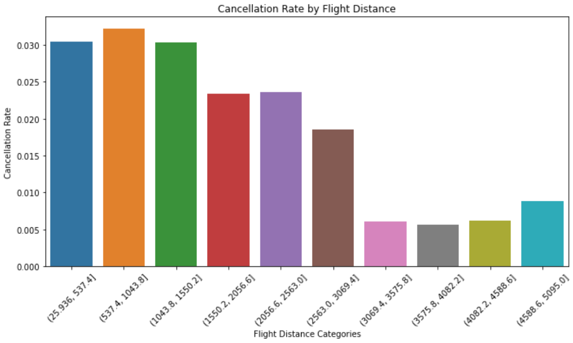


**Figure 6: Cancellation Rate by Flight Distance**

Our exploration into cancellation rates across different flight distances uncovers that the shorter flights tend to have higher cancellation rates compared to longer flights as shown in Figure 6. This could be because the distribution of flight distance is skewed right indicating that there are more shorter flights in the dataset. Factors such as weather or operational issues that are unique to short flights could play roles into why they have higher cancellation rates.

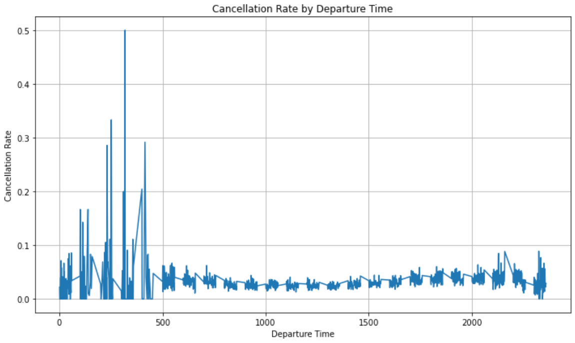


**Figure 7: Cancellation Rate by Departure Time**

According to Figure 7, we can see that earlier departure times have higher cancellation rates. This is an interesting pattern that we found. Integrating departure time in our model by understanding that association between specific departure times and cancellation probabilities allows our model to forecast cancellations more accurately. It could be that operational issues that occur earlier could be a reason why earlier flights tend to get canceled more.

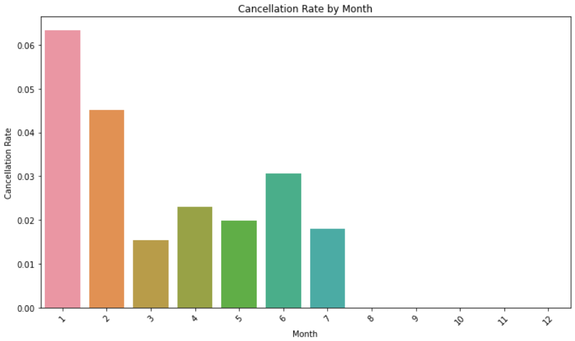


**Figure 8: Cancellation Rate by Month**

We can see that January has the highest cancellation rate out of all the months. Looking at Figure 8 overall, we can see that the first 7 months of the year have cancellations whereas the last 5 months do not, indicating that we do not have data for the last 5 months, and we need to keep this bias in mind. However, the different cancellation rate per month highlights the seasonal nature of flight cancellations. Factors such as adverse weather conditions prevalent during certain months, increased travel demand, or operational issues during certain months could play roles in influencing cancellation rates during this period. Understanding these differences will help us when creating our predictive model due to the distinct difference in flight cancellations during certain months. We hope to leverage this and incorporate month specific features into our model to make our model more accurate.

## 2 Predictive Task

### 2.1 Predictive Task

Given the available features, predict whether or not a flight will be canceled. This is a classic binary classification problem, and the target label in this dataset is 'Cancelled'.

### 2.2 Evaluation

As we found previously, there is a major imbalance in the canceled flights vs the non-canceled flights. Approximately 97 percent of flights are not canceled, so accuracy becomes a poor measure of our model's performance. Using accuracy as an evaluation method would largely bias our model because if the model just predicts that every flight will not be canceled, it would achieve an accuracy of the proportion of non-canceled flights (0.97).

**Equation 1:**

**F1 = 2 * (precision * recall) / (precision + recall)**

Therefore, we will use evaluation metrics such as precision, recall, and f1-score which is a combination of the precision and recall metrics (Equation 1). Precision is the ratio of instances that are true positives of the ones that the model predicts are positive, so a high precision in this case would mean that the model is accurate in not misidentifying non-cancellations as canceled. Recall is the ratio of true positive instances that the model correctly identifies as positive, in this case a high recall would indicate that the model correctly predicts most of the cancellations.

Recall is the most impactful measurement in our case since there is no real process to deal with cancellations once they have happened and is extremely inconvenient for both the passengers boarding a flight and the airline company losing money on that trip. Thus, maximizing recall could be more beneficial, but we will use f1-score as the overall metric to evaluate our models to get an equal balance of both precision and recall. However, we will still include both precision and recall separately.

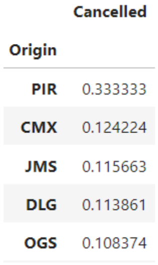
### Training and Test Data

For our baseline models, we decided to incorporate a 80-20 train/test split of our entire dataset where 80% of all our data would be used to train each model, and we would then evaluate our results on the testing set which was 20% of the dataset.

### 2.3 Baseline Models

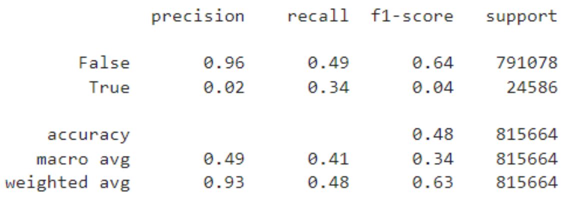
We will be trying to create a baseline model using 3 different classification models. All models are set on default hyperparameters except to prevent extreme overfitting or exceptionally long run times.

**Simple Baseline** is the baseline we used first to compare our improved models against. We wanted to keep it as simple as possible so we only used the [‘Origin’] feature which was the airport planned to depart from. We believed that the logistics of certain airports would make them more likely to cancel flights so we analyzed the proportion of canceled flights from each airport.



| | Cancelled |
|---|---|
| **Origin** | |
| **PIR** | 0.333333 |
| **CMX** | 0.124224 |
| **JMS** | 0.115663 |
| **DLG** | 0.113861 |
| **OGS** | 0.108374 |

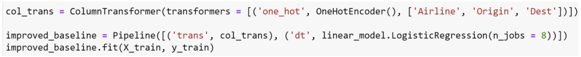
We also found that the proportion of canceled flights in the whole dataset was about 0.03 so our simple model predicts from this threshold. Airports that cancel more than 3% of their flights are predicted to always cancel and airports that cancel less than 3% are predicted to not cancel.



```
              precision    recall  f1-score   support

       False       0.96      0.49      0.64    791078
        True       0.02      0.34      0.04     24586

    accuracy                           0.48    815664
   macro avg       0.49      0.41      0.34    815664
weighted avg       0.93      0.48      0.63    815664
```

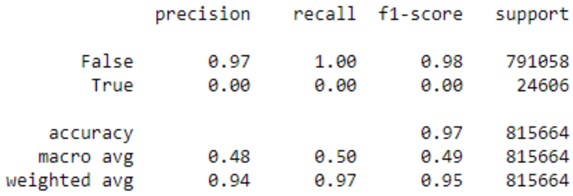
Our accuracy is 0.48 which means we predict the right label close to half of the time. Although the f1-score is higher at 0.64 for False labels, it is very low at 0.04 for True labels, showing that it is much better at predicting False than True. This makes sense since the data is very biased towards False values since most flights are not canceled. Our simple baseline falls into the ecological fallacy where we predict every flight in the airport is canceled based on the summary of the airport, which is not true. We hope to improve this simple baseline by adding the feature of [‘Airline’] and considering the interactions between the airports and airlines in a latent factor model.

**Logistic Regression** is a popular classification algorithm which uses a sigmoid function to model the probability that a given input belongs to the positive class. The logistic regression model is trained by finding the values of the coefficients that maximize the likelihood of the observed data. The process involves using optimization techniques to minimize the error between the predicted probabilities and the actual class labels in the training data. Once trained, the logistic regression model can be used to predict the probability of an instance belonging to the positive class, and a threshold can be set to classify it into one of the two classes.



```
col_trans = ColumnTransformer(transformers = [('one_hot', OneHotEncoder(), ['Airline', 'Origin', 'Dest'])])

improved_baseline = Pipeline([('trans', col_trans), ('dt', linear_model.LogisticRegression(n_jobs = 8))])
improved_baseline.fit(X_train, y_train)
```

For our Logistic Regression Baseline model, we will be using the basic categorical features [‘Airline’, ‘Origin’, ‘Dest’] and no quantitative features as inputs. We then one-hot encode each of these categorical variables to create our features and train a Logistic Regression model from Scikit-Learn set to the default parameters and then extrapolate our results.



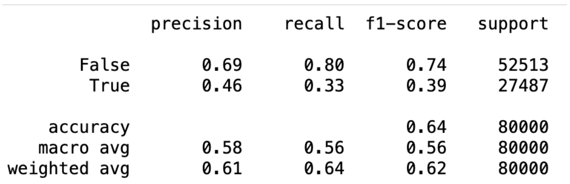
```
              precision    recall  f1-score   support

       False       0.97      1.00      0.98    791058
        True       0.00      0.00      0.00     24606

    accuracy                           0.97    815664
   macro avg       0.48      0.50      0.49    815664
weighted avg       0.94      0.97      0.95    815664
```

We find that although we have an extremely high accuracy of 0.97, our f1-score and recall for True values (canceled flights) is extremely low at 0. When we further explore the model, we find that it predicts only False values resulting in our .97 accuracy, and this makes sense because 97% of our dataset is non-cancellations. Some reasons why we believed that our baseline had such a low f1-score were that it did not have enough features, it was trained on an unbalanced (mostly False) dataset, and none of our hyperparameters were fine tuned. We hoped to make improvements on each of these attributes in our final Logistic Regression model.

**Decision Tree** is a classifier used to combat the issues faced by the Logistic Regression baseline model. This model makes decisions based on the rules it infers from the data features and can serve as a robust model for our prediction task. As we are working with an imbalance dataset containing very few positive labels, we made the use of sampling to create a random sample of 400,000 observations, where 25 percent of the observations are randomly selected canceled flights and 75 percent are non-canceled flights. Additional features such as ‘Month’, ‘DayOfMonth’, ‘DayOfWeek’, and ‘DistanceGroup’ were included in order to prevent our model from making overly biased predictions. Our created dataset was preprocessed by creating one-hot encodings for the categorical features ‘Airline', 'Origin', and 'Dest’, and then training our decision tree model using the sampled dataset. The decision tree hyperparameters were tuned to get the following:

| **Parameter** | **Value** |
|---|---|
| max_depth | 200 |
| max_features | 175 |
| max_leaf_nodes | 18 |
| class_weight | ‘balanced’ |

Using these hyperparameters, we were able to notice significant improvements compared to our previous model.



```
              precision    recall  f1-score   support

       False       0.69      0.80      0.74     52513
        True       0.46      0.33      0.39     27487

    accuracy                           0.64     80000
   macro avg       0.58      0.56      0.56     80000
weighted avg       0.61      0.64      0.62     80000
```

Our precision, recall, and f1-score values increased by 0.46, 0.33, and 0.39 respectively. Our results imply that our model is taking into consideration positive and negative labels in our dataset and is able to recognize differences in our imbalance dataset. The accuracy, on the other hand, decreased from 0.97 to 0.64, which is expected and implies that we are no longer overfitting our data. We will later analyze if we are able to improve our performance by using a Random Forest classifier.

**Baseline Models Summary**

Our initial baseline model addressed the major imbalance in our data, indicating that we cannot completely rely on the metrics presented by our dataset and make use of a trained classification model. Implementing a Logistic Regression Model illustrated the heavy bias towards non-cancellation of flights, which implied the use of sampling methods for effective results. By taking the two points into account, we were able to create a Decision Tree model trained on a randomly selected subset of data which overcame our overfitting and imbalance dataset problem.

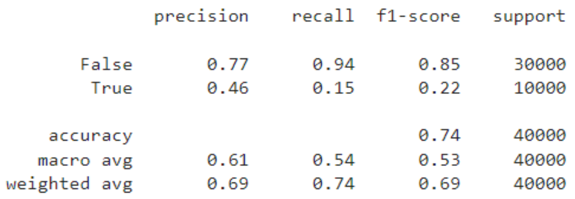
# 3 Model

## 3.1 Final Models

We will create our final models by improving on our 3 different baseline models using the same sampling method. All models now have tuned hyperparameters to improve our f1 scores.

**Latent Factor Model**

We wanted to improve on our simple baseline by looking at the interactions between airports and airlines to find if certain combinations are likely to be canceled. This model is similar to a recommender system in that it shows if a flight should be canceled based on the input pair of the departing airport and airline. We created a Latent Factor Model with 100 latent factors and a regularizer of 0.00001, then trained it for 100 iterations. The output for each prediction was a number from 0 to 1, with 1 being the most likely to be canceled. We found the best threshold to be 0.4 where airline and airport outputs greater than that are predicted to be canceled.

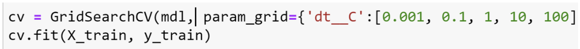


```
              precision    recall  f1-score   support

       False       0.77      0.94      0.85     30000
        True       0.46      0.15      0.22     10000

    accuracy                           0.74     40000
   macro avg       0.61      0.54      0.53     40000
weighted avg       0.69      0.74      0.69     40000
```

The latent factor model is a large improvement from the simple baseline with increases in f1 score and accuracy. By considering the interactions between airport and airline instead of just the airports, we reach an accuracy of 0.74. However our f1 scores are still unbalanced because we get 0.85 for False values and 0.22 for true values. The LFM is bad in the recall metric for True values which hurts the airline since they do not want to accidentally cancel flights that should be scheduled.

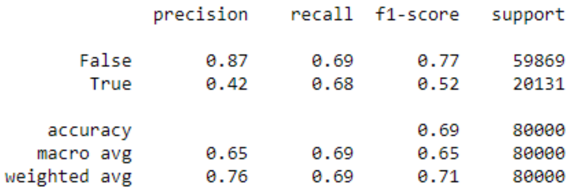
**Logistic Regression**

For our Logistic Regression Final model, we will be using the basic categorical features [‘Airline’, ‘Origin’, ‘Dest’, 'Month', 'DayofMonth', 'DayOfWeek', 'DistanceGroup'] and the quantitative features [‘Distance’, 'CRSDepTime'] as inputs. We also decided to create a balanced dataset while training the model and decided to go with a 1:3 cancellation to non-cancellation ratio in our subset of data used in the model, and we incorporated the same 80-20 train/test split of that data.



```
cv = GridSearchCV(mdl, param_grid={'dt__C':[0.001, 0.1, 1, 10, 100]
cv.fit(X_train, y_train)
```

Once we completed this step, we then one-hot encoded each of the categorical variables and trained a Logistic Regression model including all of our new features with Scikit-Learn. We also implemented a 5-fold cross validation and ran GridSearchCV on our model to find optimized hyperparameters and the best_estimator, C, that gives us the best results.



```
              precision    recall  f1-score   support

       False       0.87      0.69      0.77     59869
        True       0.42      0.68      0.52     20131

    accuracy                           0.69     80000
   macro avg       0.65      0.69      0.65     80000
weighted avg       0.76      0.69      0.71     80000
```

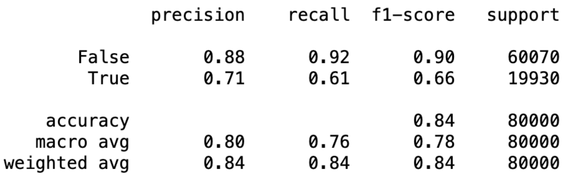
We find that our accuracy has taken a hit from our baseline model 0.97 to 0.69, but our f1-score of 0.52 and recall of 0.681 for True values (canceled flights) is significantly larger than the previous values of 0. We are now able to determine that our model really is predicting cancellation status rather than only predicting non-cancellations due to biased training data. However, the accuracy of our Logistic Regression Model has room for improvement, and although our f1 and recall is better, it is still not the best we can do. We have optimized our Logistic Regression Model, but our statistics did not meet our target goals (>0.80 accuracy, >0.60 f1-score), so we conclude that Logistic Regression may not be the most ideal model for this situation.

**Random Forest**

Lastly, we employed a Random Forest Classifier using categorical features [‘Airline’, ‘Origin’, ‘Dest’, ‘Month’, ‘DayOfMonth’, ‘DayOfWeek’, ‘DistanceGroup’] and numerical features [‘Distance’, ‘CRSDepTime’]. The Random Forest classifier makes use of multiple Decision Trees on subsamples of our dataset, which helps increase model performance and also prevent the issue of overfitting as seen in our baseline models.

Similar to previous models, we created our dataset using a 1:3 cancellation to non-cancellation ratio from our original dataset. Our data preprocessing included one-hot encodings of all of our categorical features which was used to train our RandomForestClassifier using the following tuned hyperparameters:

| **Parameter** | **Value** |
|---|---|
| n_estimators | 100 |
| max_depth | 1000 |
| n_jobs | 8 |
| max_features | ‘sqrt’ |
| class_weight | ‘balanced’ |



```
              precision    recall  f1-score   support

       False       0.88      0.92      0.90     60070
        True       0.71      0.61      0.66     19930

    accuracy                           0.84     80000
   macro avg       0.80      0.76      0.78     80000
weighted avg       0.84      0.84      0.84     80000
```

As a result, we were able to further improve our model performance, achieving an f1-score of 0.66 which is 0.14 higher than the one from our improved Logistic Regression model. We were also able to improve our precision by 0.29 from 0.42 to 0.71, with a tradeoff of a 0.07 decrease in recall. In addition, this model has achieved a higher accuracy than one of Logistic Regression, increasing by 0.15 from 0.69 to 0.84. We expected this increase based on the structure of a RandomForestClassifier, which makes use of multiple instances of our Decision Tree baseline model to become more informed on the data features and make more robust predictions while avoiding overfitting.

## 4 Literature

### 4.1 Existing Dataset

Our dataset was extracted from Marketing Carrier On-Time Performance, which is from the Bureau of Transportation Statistics. The Bureau of Transportation Statistics is a hub for statistics regarding aviation, freight activity, and overall transportation. It is accessible to the public and can be used to gain insights regarding transportation to enhance understanding on the topic. It is a credible source of information without any political bias, and it is used by Congress, researchers, and the public.

We found this dataset on Kaggle and it was already structured for us in a csv format. We proceeded to focus on the data from 2022, although there was data from 2018 to 2022. We chose 2022 because for our analysis it reflects the most recent trends, patterns or changes in carrier performance that is relevant as we create a predictive model to predict flight cancellations. Focusing on one year, we hoped to gain a more detailed understanding with focus without overwhelming by using too many years. We wanted to use this dataset to address the need within the aviation industry to anticipate flight cancellations and take necessary actions to proceed. It allows airlines to prepare and address operational issues to enhance customer satisfaction and operational efficiency.

### 4.2 Similar Projects

In the past, and in other similar projects, the following datasets and analysis have been used to either predict cancellations or flight delays as both are critical data that can be used to optimize current flying processes.

These projects[2],[3],[5],[6],[7] all include datasets with similar features to ours such as Month, Day, Date, Destination, Origin, Delays, etc. and use them in different models to predict cancellation statistics like ours.

Especially in the ScienceDirect article, written by Yu Yinyang, Hai Mao, and Li Haifeng, they used Logistic Regression, Support Vector Machine, Naive Bayes, and Decision Tree models/algorithms to predict cancellation. Across all our sources and data these models along with the ones we incorporated into our report seem to be the most common methods at approaching a classification problem such as this. While in the ScienceDirect study, the authors found SVM and Decision Tree to be most effective at predicting cancellations with an accuracy of about 90%, we found that a RandomForest Model performs even better than Decision when predicting cancellations based on our dataset. Although results may vary from dataset to dataset, tree-based algorithms seem to be the best solution when solving a problem like this.

Similar to ourselves, these projects all agreed that predicting cancellations was slightly difficult due to the larger amount of non-cancellations across all training data, and while other groups tried to increase tree depth and leaf nodes to account for this, we decided to balance our dataset while also playing with hyperparameters to reach the same conclusion and high accuracy, f1, and recall scores.

## 5 Results

### Final vs Baseline Model

Fortunately, our final model using a Random Forest was able to outperform our baseline models by a sizable margin, as well as the other improved model. The final Random Forest had a 31% improvement in accuracy compared to the baseline Decision Tree, and a 10% and 15% improvement over the latent factor model and logistic regression respectively. In addition, the Random Forest had much better f1 scores of 0.9 and .66 for True and False values, beating the other models by a good margin. By adding more features and tuning hyperparameters of models that were inherently more effective, we were able to end with a final model with a more optimal performance.

**Model Parameters**

The model parameters used in our model were a mix of categorical and numerical features:

| Feature | Type | Description |
|---|---|---|
| Airline | Categorical | Name of airline |
| Origin | Categorical | Flight location |
| Dest | Categorical | Flight destination |
| Distance | Numerical | Travel distance |
| Month | Categorical | Flight month |
| DayOfMonth | Categorical | Day of Month |
| DayOfWeek | Categorical | Weekday |
| DistanceGroup | Categorical | Distance group by destination location |
| CRSDepTime | Numerical | Departure time as recorded by Certification of Release To Service (CRS), allowing aircrafts to fly. |

Our final model contained multiple features unaccounted for in our previous models and randomly selected data which tackled the imbalance in our dataset. Additionally, our model made multiple uses of our Decision Tree baseline model, allowing it to reach optimal performance and have higher precision predictions, something our previous models failed to accomplish.

## References

[1] Bureau of Transportation Statistics. (2023, September 1). Bureau of Transportation Statistics. https://www.transtats.bts.gov/DL_SelectFields.aspx?gnoyr_VQ=FGK&QO_fu146_anzr=b0-gvzr

[2] Bureau of Transportation Statistics. (2023, September 1). *OST_R: BTS: Title from H2*. BTS. https://www.transtats.bts.gov/ot_delay/ot_delaycause1.asp

[3] *Connect to amadeus travel apis: Amadeus for developers*. Amadeus IT Group SA. (n.d.). https://developers.amadeus.com/self-service/category/flights/api-doc/flight-delay-prediction

[4] Mulla, R. (n.d.). *Flight status prediction*. Kaggle. https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022?select=Combined_Flights_2022.csv

[5] Shu, Z. (2022, March 17). *Analysis of flight delay and cancellation prediction … - IEEE xplore*. Analysis of Flight Delay and Cancellation Prediction Based on Machine Learning Models. https://ieeexplore.ieee.org/abstract/document/9731090

[6] Tang, Y. (2021, October 15). *Airline Flight Delay Prediction Using Machine Learning Models*. Airline flight delay prediction using machine learning models. https://dl.acm.org/doi/fullHtml/10.1145/3497701.3497725

[7] Yinyang, Y., Mo, H., & Haifeng, L. (2019, December 31). *A classification prediction analysis of flight cancellation based on Spark*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050919320241