Predicting Flight Cancellation and Delay using Flight Attributes and Weather Data

Sarah Ahmad, Edmond Chen and Jonathan Shiler
Contact: sarahahmad2019@u.northwestern.edu
EECS 349 | Northwestern University

Abstract

Our project aimed at solving two tasks: predict whether a flight would be cancelled and, if it were not cancelled, the departure delay in minutes. The first task is a binary classification problem and the second is a regression problem. We used two data sources: historical flight data from the American Statistical Association, and weather data from National Climate Data Center. Our best model for both tasks was a gradient boosting model, implemented using Microsoft's LightGBM library. With this model, we are able to predict cancellations at 99.9% accuracy, an improvement of ~5% from the ZeroR baseline. We are also able to predict departure delay approximately within 29 minutes of the actual delay, an improvement of 13 minutes from the ZeroR baseline.

1. Introduction

When we were presented with the opportunity to work on a machine learning task, our group immediately started brainstorming ideas and topics in which machine learning could serve a benefit in our lives. With winter weather already present in Evanston, we realized a lot of our peers' flight were being canceled and delayed. Additionally, one of our group member's flights was delayed numerous times and at risk of cancelation during the quarter on the way back to school. This got us thinking about if there was a way to predict delays and cancellations for individuals so as to allow them a better sense of control over their travel arrangements.

Our tasks are:

1. Predict whether flight is cancelled (binary classification)
2. Given that a flight is not cancelled, predict how long a flight is delayed (regression)

2. Data

We used historical flight data made available by the American Statistical Association [1]. This data set consisted of flight data from 1987 to 2008. We chose to work with the most recent year available in the data set,2008. Even after only selecting a subset of the years, we still had large amounts of data, with each year containing flight data for millions of flights. From the flight data, we decided to focus on flights out of O'Hare because not only was it the airport with the second most departing flights (the first being LAX), but also because we felt that O'Hare might have a greater variation of weather throughout each year than LAX. Additionally, we thought it would be more relevant for Northwestern students flying out of O'Hare whenever they travel home for breaks.

From the flight data set, we selected the following feature attributes: **departure time**, **carrier**, and **destination airport**. Departure time was given in hours and minutes so we decided to use Trifacta to preprocess the data set and break the departure time into two different categories, departure hour and departure minutes. We then used departure hour as the attribute.

We initially used only this data set for our machine learner, but after preliminary tests realized that we wanted to try to improve our learner's accuracy by including additional attributes. Specifically, we wanted to see how much of a role weather plays on flight delays, which is why we first tested our learner without weather data to have a baseline for comparison. In order to add weather data, we pulled from the National Climate Data Center's daily weather summaries [2]. The National Climate Data Center offers yearly weather reports for any location, so we were able to pull historical weather data for each day of the year 2008. After getting each date, we were able to join the flight data with the weather data on date using Trifacta. From the weather data, we used attributes such as **minimum temperature**, **maximum temperature**, and **average temperature**, **average precipitation**, and **type of precipitation**.

The target attributes were taken from the flight data set. For the first task, the target attribute was "**cancelled**" and the value is either 0 or 1. For the second task, the target attribute was **departure delay**, given in minutes.

3. Methodology

3.1 Preprocessing

We had three types of attributes: numerical, categorical (non-ordinal) , and categorical (cyclic ordinal) attributes. We did not need to preprocess numerical attributes since our machine learning models can handle the continuous values but we needed to process the two types of categorical variables in the following manner:

Non-ordinal categorical attributes are attributes such as **Airline Carrier** and **Destination Airport**. For these attributes, we applied one-hot encodings. This is because the values in the non-ordinal attributes are not necessarily linked with one another. For the sake of contradiction, suppose we label encode (rather than one-hot encode) American Airlines as 0, Turkish Airlines as 1, and Air China as 2. The average of American Airlines and Air China gives Turkish Airlines, which could be misleading and therefore should be avoided.

Cyclic ordinal attributes are attributes such as **Month**, **Day of Week**, and **Hour of Departure**. The ordinal property can be seen from the fact that the average of June and August can be reasonably regarded as July. The cyclic property can be seen when we consider that the average of February (month 2) and December (month 12) should reasonably be January (month 1), and not July (month 7). For these attributes, we applied sine and cosine preprocessing to account for these properties [3].

3.2. Data Partitioning

We split the training, validation, and test sets using scikit-learn's train_test_split function. The data was shuffled and we set a fixed random state, so that we could reproduce the same training, validation and test sets for each model to allow for fair comparison.

	Percentage of data (%)	**Number of rows**
Training	72	252273
Validation	18	63069
Test	10	35038

Total	100	350380

Table 1: Partitioning of data for Task 1. The data partitioning for Task 2 was very similar, the only difference being that we removed cancelled flights, which accounted for less than 5% of the data in Task 1.

3.3. Models

We tested four models for each of the two prediction tasks described in "Introduction" above. The four models were:

1. ZeroR: We implemented a ZeroR model to serve as a baseline. The ZeroR model outputs the majority class (for classification) and mean value (for regression).
2. Linear Models: We implemented Logistic Regression (for classification) and Linear Regression (for regression) using the Scikit-learn library.
3. Gradient Boosting: We used the popular framework LightGBM [4].
4. Neural Network: We implemented a Multilayer Perceptron (MLP) using the Keras library.

The architecture for the gradient boosting and neural network models are as follows:

Model	**Architecture**	
	Task I (Classification)	**Task 2 (Regression)**

LightGBM	objective = 'binary'
	metric = 'binary_logloss'
	learning_rate = 0.01
	boosting_type = 'gbdt'
	num_leaves = 100
	min_data = 50
	max_depth = -1
		objective = 'regression'
		metric = 'rmse'
		(The rest are the same as Task 1.)

MLP	Dense Layer (20, activation='relu') Dropout (0.3) Dense Layer (16, activation='relu') Dropout (0.3) Dense Layer (8, activation='relu') Dropout (0.3) Dense Layer (8, activation='relu') Dense Layer (1); Optimizer: Adam	Same as Task 1, except the last layer has an sigmoid activation function.

Table 2: Architecture of gradient boosting and neural network models

4. Results

4.1. Task 1: Predict whether a flight is cancelled (Binary Classification)

Model	**Accuracy (%)**		
	Training	**Validation**	**Test**
ZeroR	95.71	95.65	95.74
Logistic Regression	99.75	99.69	99.75
LightGBM	99.99~	99.96	**99.97**
MLP	99.80	99.78	99.76

Table 3: Model performance in terms of prediction accuracy. ~ indicates consecutive '9' values.

4.2. Task 2: Predict how long a flight is delayed given a non-cancelled flight (Regression)

Model	RMSE		
	Training	**Validation**	**Test**
ZeroR	41.35	41.05	42.52
Linear Regression	36.93	36.69	37.91
LightGBM	26.01	28.19	**28.85**
MLP	35.13	34.95	36.02

Table 4: Model performance in terms of root mean squared errors.

5. Analysis

5.1. Model Performance

For our first task of predicting cancellations, our results show that the gradient boosting model performs the best at 99.97% test accuracy. This is an improvement of about 5% from the baseline ZeroR model which has an accuracy of about 96%. We can attribute this to the imbalance in labels which is expected since most flights are not cancelled. Overall, all models were able to achieve above 99.7% test accuracy.

For our second task of predicting departure delay, our results show that the gradient boosting model significantly outperforms the linear regression and neural network model, scoring a RMSE of 28.85. This means that our model can predict departure delay approximately within 29 minutes of the actual delay, and this is an improvement of 13 minutes above the baseline ZeroR model which has an RMSE of 42.45.

While the gradient boosting model performed expectedly well on both tasks, the neural network model did not perform so well. This could be due to the hyperparameters. The neural network could be improved by fine tuning hyperparameters such as the number of hidden layers and the type of optimizer.

5.2. Feature Analysis

For the first task of predicting flight cancellations (binary classification), we discovered that the hour and minute of departure has a surprisingly high importance. These attributes have a higher importance than weather, which is interesting because we would normally expect weather to have a greater impact on flight cancellations.

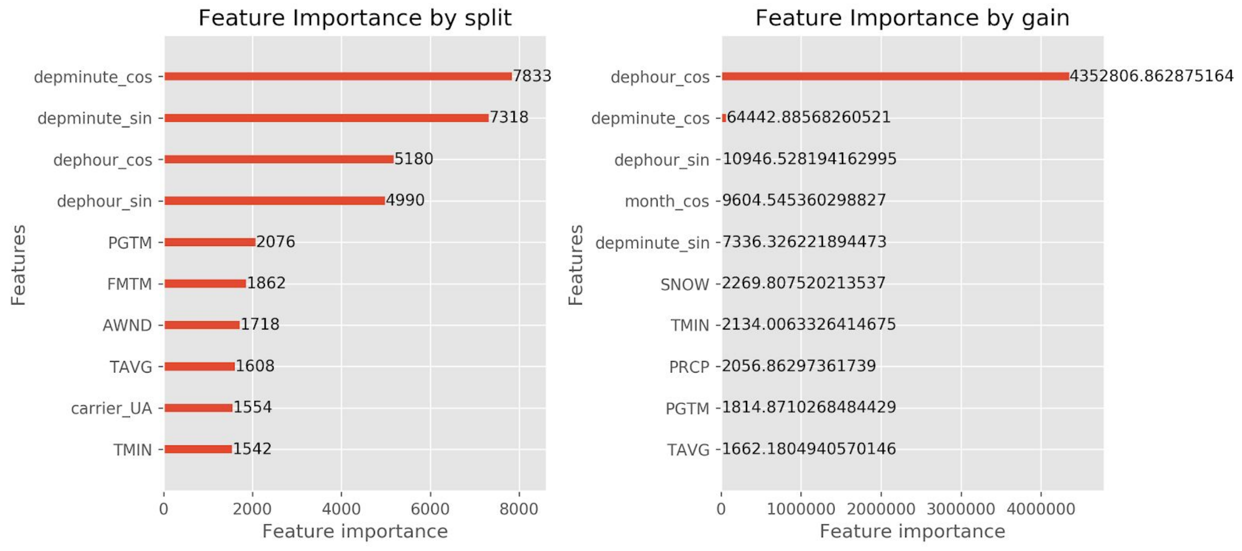

Figure 1: Feature importance by split (left) and gain (right) of LightGBM model in predicting cancellation (Task I)

For the second task of predicting departure delay (regression), the attributes of departing hour and amount of precipitation (PRCP) have the largest feature importances by split and by gain. Departing minute is one of the most important features "by split", but it was not as important when measured "by gain". The results from importance "by gain" makes more sense, because for example, the departing minute is not likely to affect the delay as much as weather.

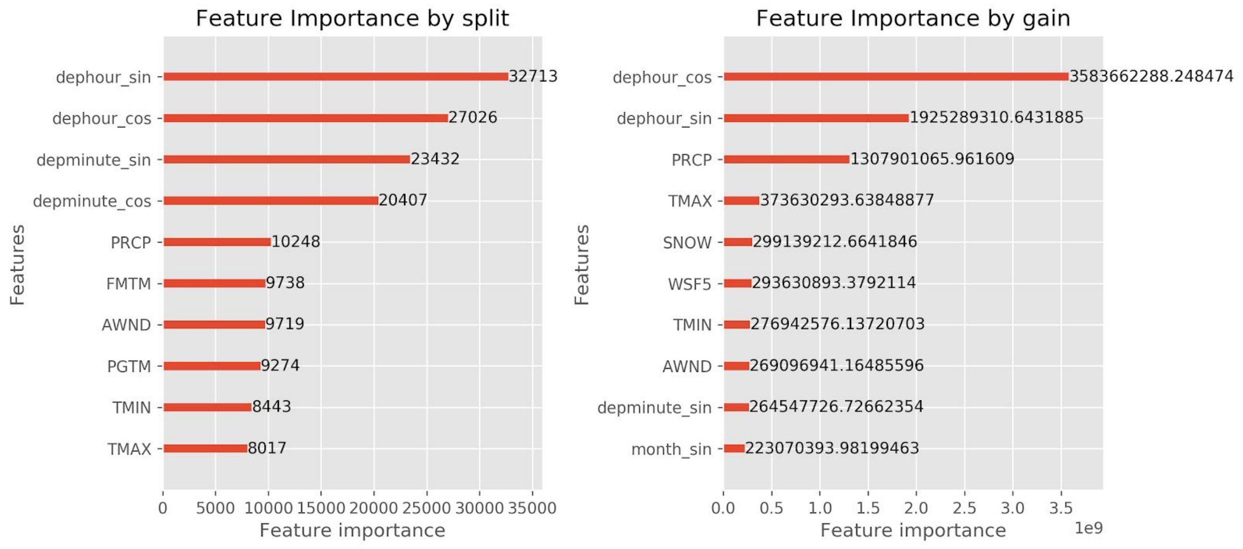

Figure 2: Feature importance by split (left) and gain (right) of our LightGBM model in predicting delay (Task II)

6. Conclusion & Future Work

In summary, our best model for both tasks was a gradient boosting model, implemented using Microsoft's LightGBM library. With this model, we are able to predict cancellations at 99.9% accuracy, an improvement of ~5% from the baseline ZeroR model. We are also able to predict departure delay approximately within 29 minutes of the actual delay, an improvement of 13 minutes from the baseline ZeroR model.

For future work, we think it would be beneficial to fine tune the hyperparameters in the neural network in order to see if a more accurate model can be uncovered. Additionally, while a year of flight and weather data meant hundreds of thousands of data entries to consider and preprocess, more years of flight and weather data could make our models more concrete and robust. Moreover, it would be nice to extend this analysis to all airports so that we can help individuals all over the world better make travel arrangements. This would then give us the opportunity to look at the interactions between airports and see which airports are the most problematic when it comes to delays and cancellations. Through all this, we hope to allow individuals to have a greater sense of planning confidence when making travel plans.

Roles

Edmond Chen	**Data extractions scripts, running machine learning algorithms, report drafting**
Sarah Ahmad	**Running machine learning algorithms, website design, report drafting**
Jonathan Shiler	**Running machine learning algorithms, report drafting, editing, and finalizing**

References

[1] https://stat-computing.org/dataexpo/2009/the-data.html
[2] https://www.ncdc.noaa.gov/
[3] https://ianlondon.github.io/blog/encoding-cyclical-features-24hour-time/
[4] https://github.com/Microsoft/LightGBM