Draft

Oliver Ryder-Green

2022-12-03

Introduction

Flight delays are an inconvenience that almost all aviation passengers will experience at some point in their travels. Yet the burden of flight delays is not the same for all passengers. In particular, US passengers are not entitled to compensation for delays[1]. Yet, between 2013 and 2022, approximately one in every five flights from US airports was delayed by at least 15 minutes[2]. With more than 10 million scheduled passenger flights in the US each year[3], the cost to passengers of flight delays is substantial. Indeed, the *Federal Aviation Administration* estimates that flight delays in the US from 2016 to 2019 cost passengers US$62.6billion in total. Short of relying on airlines to inform them of expected delays, there is little that US passengers can do to reliably avoid flight delays. Therefore, I apply the classification methods discussed in class to determine which factors inform flight departure delays for domestic flights in the US.

Data from the *Bureau of Transportation Statistics* illustrates the prevalence of domestic flight delays. Among all US carriers, between 15–25% of departures were delayed from 2010 to 2022. Among the top US carriers[4], the proportion of delayed flights is persistently higher than average. Evidently, some US carriers exhibit fewer than average flight delays (e.g., Delta Airlines), but top US carriers tend to demonstrate more frequent flight delays than the industry as a whole.

For US passengers, the fact that top US carriers experience more frequent departure delays may be of interest in trying to avoid delays. That said, more frequent delays at top airlines do not necessarily imply more severe (i.e., costly) delays for passengers. The *Bureau of Transportation Statistics* data shows that, among all US carriers, mean departure delay lengths were between 7 and 17 minutes on average from 2010 to 2022. Unfortunately, top US carriers again appear to perform worse than the industry as a whole. Without exception, top US carriers exhibit longer-than-average delays at some point in the period.

The *Bureau of Transportation Statistics* data also highlights that the frequency of delays varies by origin airport. In line with the above, around one in every five flights from a US airport is delayed. There are clearly some airports that persistently experience more frequent delays, over 50% of all flights in some cases, and some airports that experience few or no delays.

The task of anticipating delays is extremely difficult for passengers. Many factor are hard to observe or nearly impossible to predict. That said, the data above suggests that some readily observable features may be useful for passengers trying to avoid delays. For instance, if passengers face a choice of carriers, they may be better able to avoid costly delays by choosing those that exhibit less frequent and shorter delays. The aim of this analysis is to identify such features that passengers might use to anticipate delays.

Data

To identify factors that inform whether a flight is delayed on departure, I use data from the *Bureau of Transportation Statistics' Airline On-Time Performance Data*[5] for January, March, September, and December in 2016, 2017, and 2018, respectively. The flight data contains 8,777 observations on US domestic flights and 21 features, such as the flight date, origin airport, carrier, destination, distance, and other flight level characteristics. I combine this data with weather data from *Weather Underground*[6]. The weather data contains weather observations from corresponding airport weather stations on flight departure dates.

[1]source:www.transportation.gov
[2]source:www.bts.gov
[3]source:www.faa.gov
[4]as measured by total number of flights serviced in 2010–2022.
[5]www.transtats.bts.gov/
[6]www.wunderground.com

Compiling and Cleaning

Flight Data

I manually download *Bureau of Transportation Statistics' Airline On-Time Performance Data* for January, March, September, and December in 2016, 2017, and 2018, respectively. I import the data and compile using Pandas in Python (see corresponding Jupyter NB). The resulting dataset has 5,851,068 observations and 21 features. To make the dataset manageable, I draw a random subset (fraction=0.0015) from each month-year sample. The resulting dataset contains 8,777 observations.

Weather Data

I use web-scraping methods in Python (see corresponding Jupyter NB) to acquire historical weather data from *Weather Underground.* I use airport codes corresponding to origin airports for departures in the flight data to scrape historical weather data from airport weather stations. I acquire observations on temperature, precipitation, sea level pressure, and max wind speed on the date of departure. The resulting dataset contains 6,190 observations.

Merged Data

I merge the flight and weather data on the date of departure and origin airport code. For the flight data, delays are identified as any flight departing more than 15 minutes late: `DepDel15=1` if delayed and `DepDel15=0` otherwise. Delays measured in minutes are given by `DepDelay`. Since the supervised learning methods I utilise rely on the assumption that **target variables** do not have missing values, I drop observations if both `DepDel15` and `DepDelay` are missing because such observations contain no useful information for the analysis.

Since I am interested in predicting delays and delay lengths using flight characteristics and weather observations, I consider the proportion of missing values for these predictor variables. I find that there are no missing observations in the flight data. However, around 70% of observations have missing values for `Day.Average.Temp`, `High.Temp`, `Low.Temp`, `Max.Wind.Speed`, and `Sea.Level.Pressure` (see Jupyter NB). Moreover, around 90% of observations have missing values for `Precipitation`. Dropping observations missing weather data is costly in terms of observations. Yet, the weather data is of interest in prediction and is likely independent of many flight characteristics. I choose to omit observations that have missing values for `Day.Average.Temp`, `High.Temp`, `Low.Temp`, `Max.Wind.Speed`, and `Sea.Level.Pressure`.

Further omitting observations that have missing values for precipitation may be discarding useful information: the correlation between `DepDel15` and `Precipitation` (0.059) and between `DepDelay` and `Precipitation` (0.026) are both non-negligible, and; mean precipitation is higher for delayed departures (0.446 inches) than for non-delayed departures (0.342 inches). Since `Precipitation` is correlated with other weather observations, I choose to impute missing values for `Precipitation` using k-Nearest Neighbours on weather data. I optimise parameter k by choosing $k \in (0, 100)$ to minimise the average MSE for out-of-sample prediction across 50 random sub-samples of complete weather data (see Jupyter NB). I impute missing values for `Precipitation` in the merged dataset using $k^* = 2$. The resulting dataset has 2,693 observations and 27 features.

Feature Engineering

There is structure in the data that may be useful to exploit. For instance, the data already splits flight dates into `Month`, `DayofMonth`, and `DayofWeek`, which may be relevant in predicting flight delays if, for example, weekend flights are more prone to delays. In a similar vein, I re-code `DepTimeBlk`, which gives the astronomical time interval in which a departure is scheduled, as a factor variable to be used in prediction. I likewise re-code `ArrTimeBlk`. The variables `CRSDepTime` and `CRSArrTime` give the scheduled departure and arrival times of a flight in astronomical time. I find the difference in minutes between scheduled departure and arrival times to categorise scheduled flight lengths by hour in `SchFlTm`. The variable `Distance` gives the flight distance in miles, I categorise flights by distance in `DistGr` in intervals of 500 miles. I assign `InSt=1` if a flight is within state and `InSt=0` otherwise using `OriginStateName` and `DestStateName`. Finally, I use a stricter definition of a departure delay than that of `DepDel15`. I assign `Delayed=1` if `DepDelay>0` and `Delayed=0` otherwise. The dataset used for the analysis contains 2,685 observations and 31 features[7].

Summary

The distribution of departure delay lengths is[8]:

I consider the features that may be useful in distinguishing between delayed and non-delayed flights. Delayed flights tend to have longer scheduled flight times:

Accordingly, delayed flights tend to have greater flight distances:

Methodology and Results

To identify important predictors of flight delays, I use supervised methods that are able to classify (i.e., predict) delayed and non-delayed flights. I compare Logit, Lasso, Ridge, Elastic Net, Random Forest, and Boosting methods. I consider the performance of these methods and the top 5 most important predictors of flight delays they each identify. I also consider the top 5 airlines, origins, destinations, and top flight and weather features the methods identify.

Since passengers likely wish to avoid costly delays, the performance of the methods is evaluated using the true positive and false negative rates of prediction. Accordingly, I cross-validate the fit of models using the AUC as the performance metric. Moreover, since passengers presumably wish to take their intended flight, I assess the methods using false positive rates. Finally, I consider the accuracy of prediction since a given passenger would wish to know whether a given flight is likely to be delayed.

To fit the models, I divide the merged dataset into training (n=) and test (n=) samples[9]. I fit the models on the training sample and cross-validate, where relevant, to tune model parameters. I then predict delays in the test sample using parametised models.

First, I fit a Logit model that yields the following:

The logit model performs relatively poorly at predicting delays; for instance, it assigns higher probabilities to delays over non-delays only marginally better than a random guess. Perhaps this is due to over-fitting on the training sample (i.e., the model delivers low bias in-sample but high variance out-of-sample). I therefore turn to shrinkage methods to strike a better bias-variance trade-off. In particular, I fit and cross-validate Lasso, Ridge, and Elastic Net models[10] that yield:

Evidently, prediction performance improves using shrinkage methods. In particular, Lasso, Ridge, and Elastic Net exhibit more desirable true positive and false positive rates; as a result, the respective AUCs are an improvement on logit. That said, false negative rates remain similar to logit. The possibility of 'creeping delays', caused by the interaction of features that increase the probability of a delay, suggest that non-linearities in the data may be an important aspect of prediction. I therefore turn to non-linear methods. I grow and prune a Random Forest[11] and subsequently consider the predictive power of a coalition of *weak*

[7]Note: I remove `Flights`, which gives the number of flights per flight journey, since it is equal to one for all observations and therefore contains no useful information.

[8]Early departures have negative delay lengths.

[9]Note: continuous variables are normalised.

[10]Note: I tune shrinkage, λ for Lasso (=), Ridge (=), and Elastic Net (=). I tune sparsity for Elastic Net (=).

[11]Note: I manually tune to optimise the maximum number of features (=) and number of trees (=).

learners using cross-validated Gradient-based Minimisation ("GBM")[12], which yield:

Non-linear models appear to offer no significant improvement on shrinkage methods. Random Forest, in particular, delivers the worst rate of prediction among the methods, despite having the best AUC. This is perhaps due to over-fitting on the data. That GBM outperforms Random Forest in prediction perhaps eludes to the fact that shrinkage methods are able to exploit sparsity in the training data to aid in prediction.

In summary, the fitted models yield:

Conclusion

The best performing method is Lasso. It delivers the highest true positive rate (), the second-highest true negative rate (), and the highest accuracy () among all methods. It identifies days of the month as being the most important predictors of flight delays. This is in contrast to Logit, Ridge, and Elastic Net, which select origins and destinations, and Random Forest and GBM, which select weather and flight features. That Lasso performs best in this context is unsurprising given the multiplicity of dummy variables generated by the features in the data. Many of the other models are likely over-fitting on the training data; Lasso mitigates this issue by exploiting sparsity[13]. Evidently, predictions are noisy: the top 5 most relevant predictors by category are relatively unstable between models. I use a rudimentary distance metric[14] to determine the stability of the predictions:

The most stable predictions across models are for which carriers are the most important in determining delays. This is perhaps unsurprising given the data presented above. A sensible choice for passengers might then be to avoid airlines with a history of delays. Happy flying!

[12]Note: I manually tune to optimise the learning rate (=), number of trees (=), and tree depth (=).

[13]Note: that cross-validated Elastic Net chooses α= speaks to this fact.

[14]For each top 5 category, I count the frequency of unique predictors given by the models. If all models agree, there would be 5 variables each occuring 5 times. I calculate the difference between the actual frequency of the predictors and 5. Then I sum the squared-difference to calculate the metric, which punishes top 5 categories twofold, for: (1) models that disagree with each other, and (2) the extent to which models disagree.