

Predicting Flight Delays at PIT

Team462

Introduction

This project focuses on predicting departure delays of 15 minutes or more (`DEP_DEL15`) for flights leaving Pittsburgh International Airport (PIT). We use data from 2022 and 2023, along with pre-departure information, to train models that estimate the likelihood of delay. The final predictions are submitted for a held-out 2024 test set. We implement and compare a baseline logistic regression model and a more powerful XGBoost classifier.

Data Exploration

Data Overview

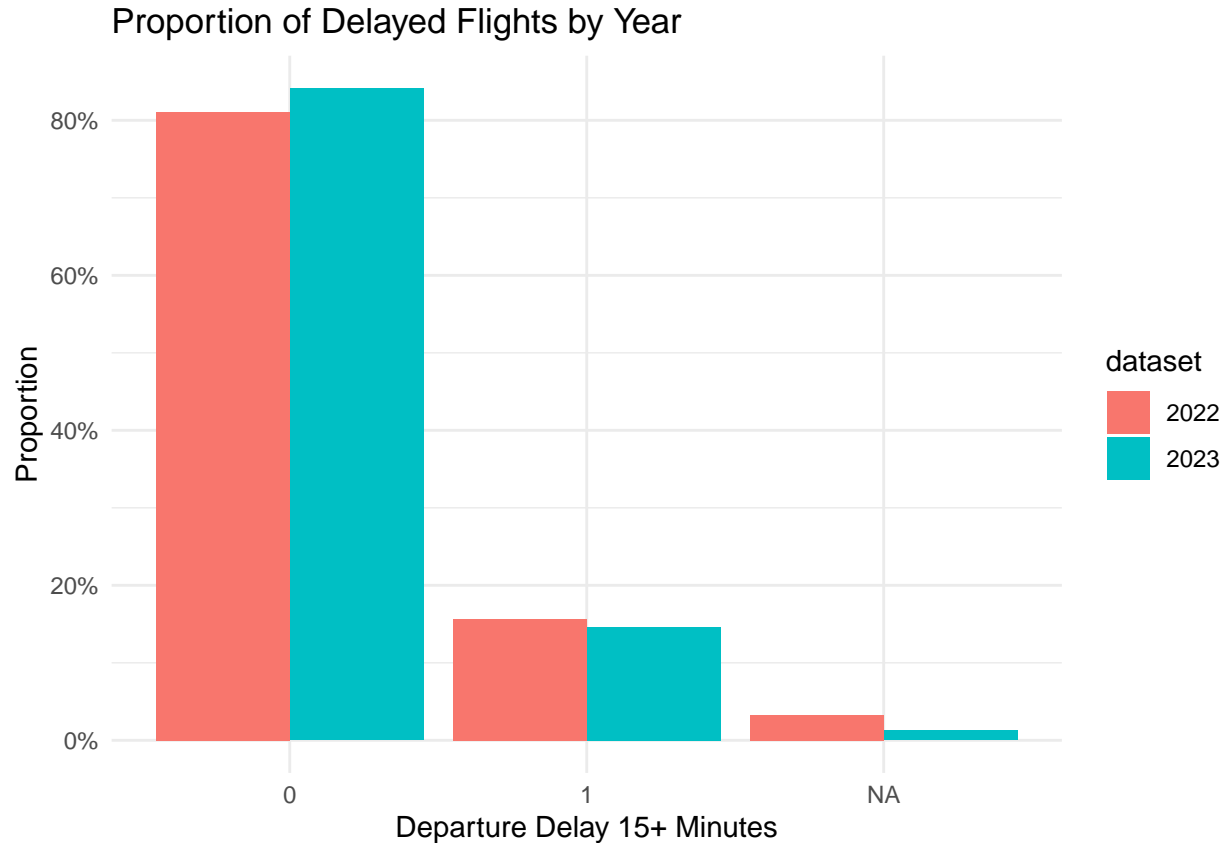
We worked with flight datasets from 2022 and 2023, containing commercial airline activity to and from Pittsburgh International Airport (PIT). After loading the datasets, we combined them and identified a total of 137,425 flights with 65 features per flight record. These features include scheduled and actual times, delays, airport information, carrier identifiers, and operational statistics.

A breakdown of variable types revealed 49 numeric and 16 character fields. Key variables such as `FL_DATE`, `DEP_DEL15`, `CRS_DEP_TIME`, and `OP_UNIQUE_CARRIER` provided the basis for modeling flight delays. We focused exclusively on departing flights (`ORIGIN == "PIT"`), aligning with the project objective of predicting departure delays based only on information available prior to takeoff.

The data had a low missingness rate overall for departure-related variables, though fields associated with delay reasons (`CARRIER_DELAY`, `WEATHER_DELAY`, etc.) had high missingness (~82%). This is expected because these variables are not available at prediction time. The target variable `DEP_DEL15` had about 2% missingness, which we accounted for during model training by removing or imputing missing instances.

Missingness and Class Imbalance

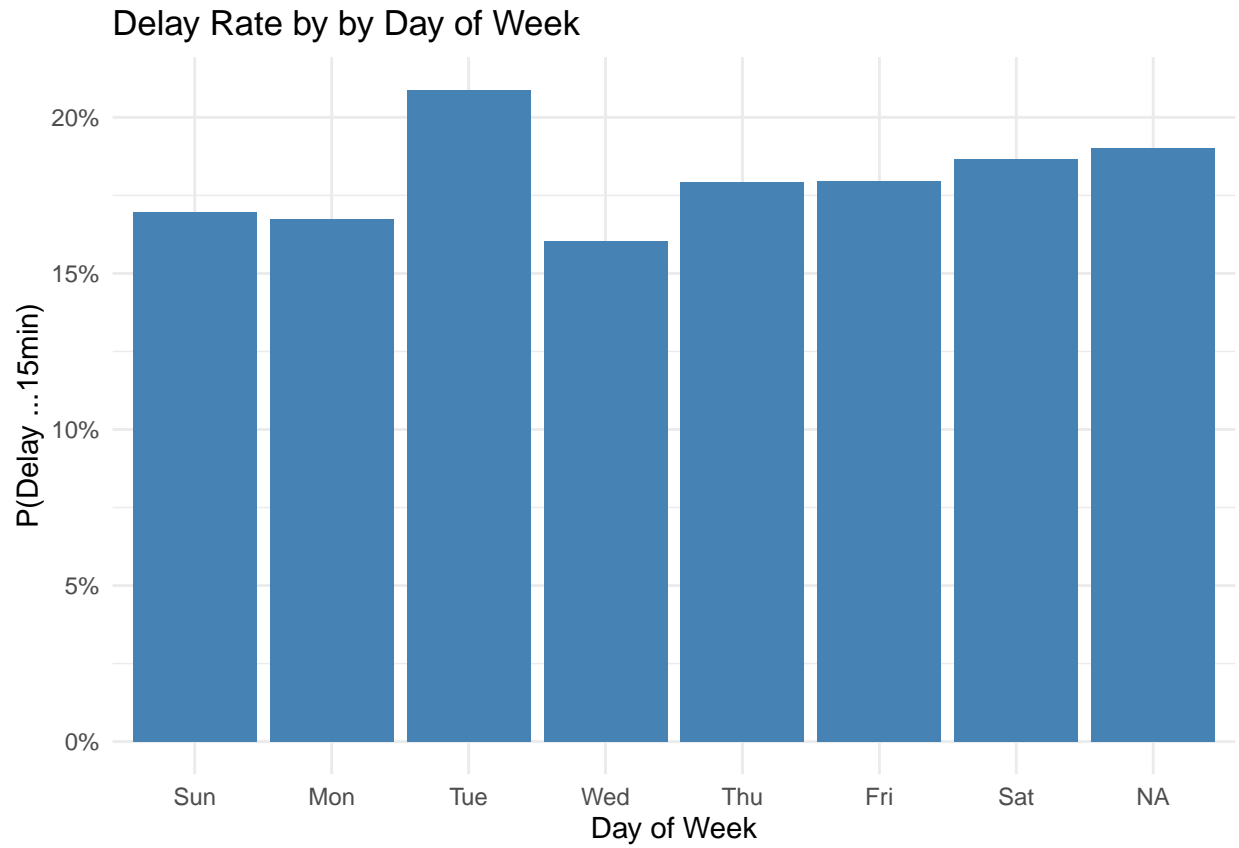
```
## # A tibble: 6 x 4
## # Groups:   dataset [2]
##   dataset DEP_DEL15     n    pct
##   <chr>      <int> <int> <dbl>
## 1 2022          0 49644 0.777
## 2 2022          1 12178 0.191
## 3 2022         NA  2095 0.0328
## 4 2023          0 59729 0.813
## 5 2023          1 12810 0.174
## 6 2023         NA   969 0.0132
```



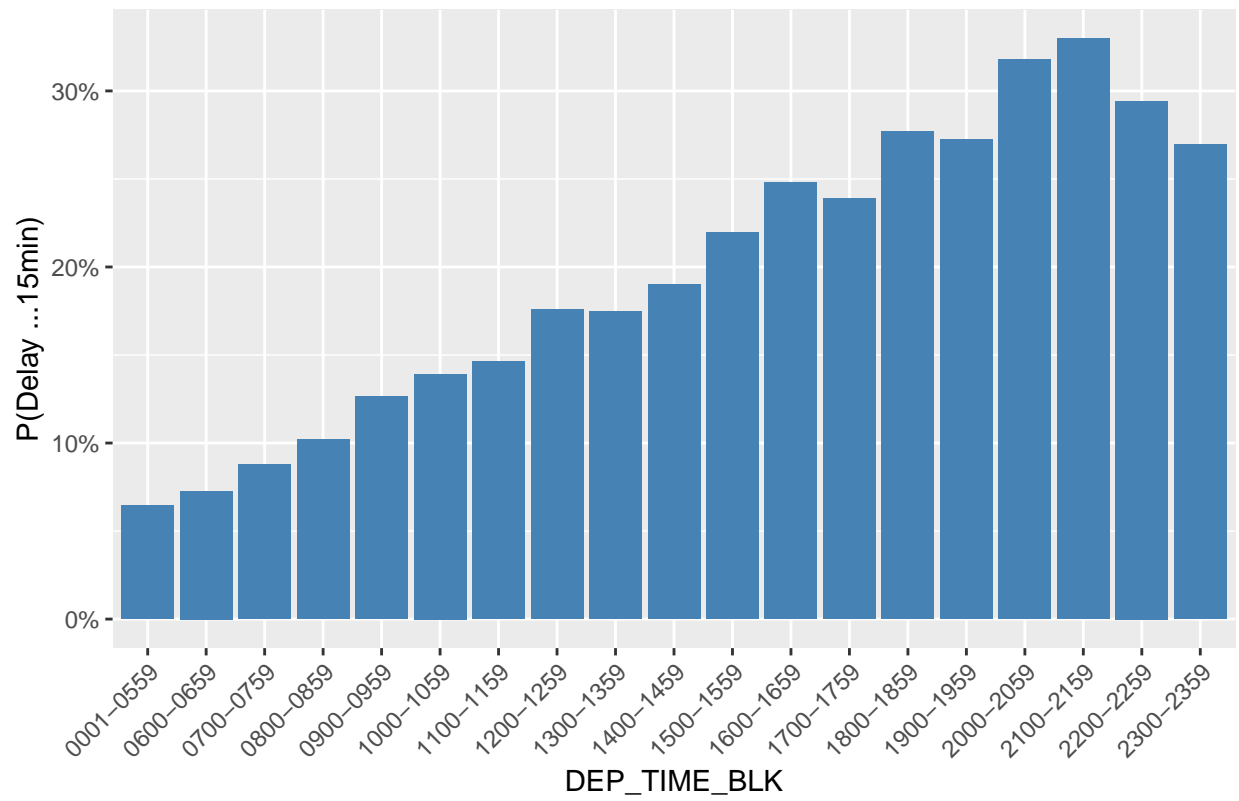
We assessed the missingness in the combined dataset and found that most core variables were well-populated, with over 97% completion. However, several post-departure variables were highly incomplete, such as `FIRST_DEP_TIME`, `TOTAL_ADD_GTIME`, and `LONGEST_ADD_GTIME`, each missing in over 99% of records. These fields, along with `CARRIER_DELAY`, `WEATHER_DELAY`, and `LATE_AIRCRAFT_DELAY` (each ~82% missing), represent information not available before departure, and were therefore excluded from feature consideration.

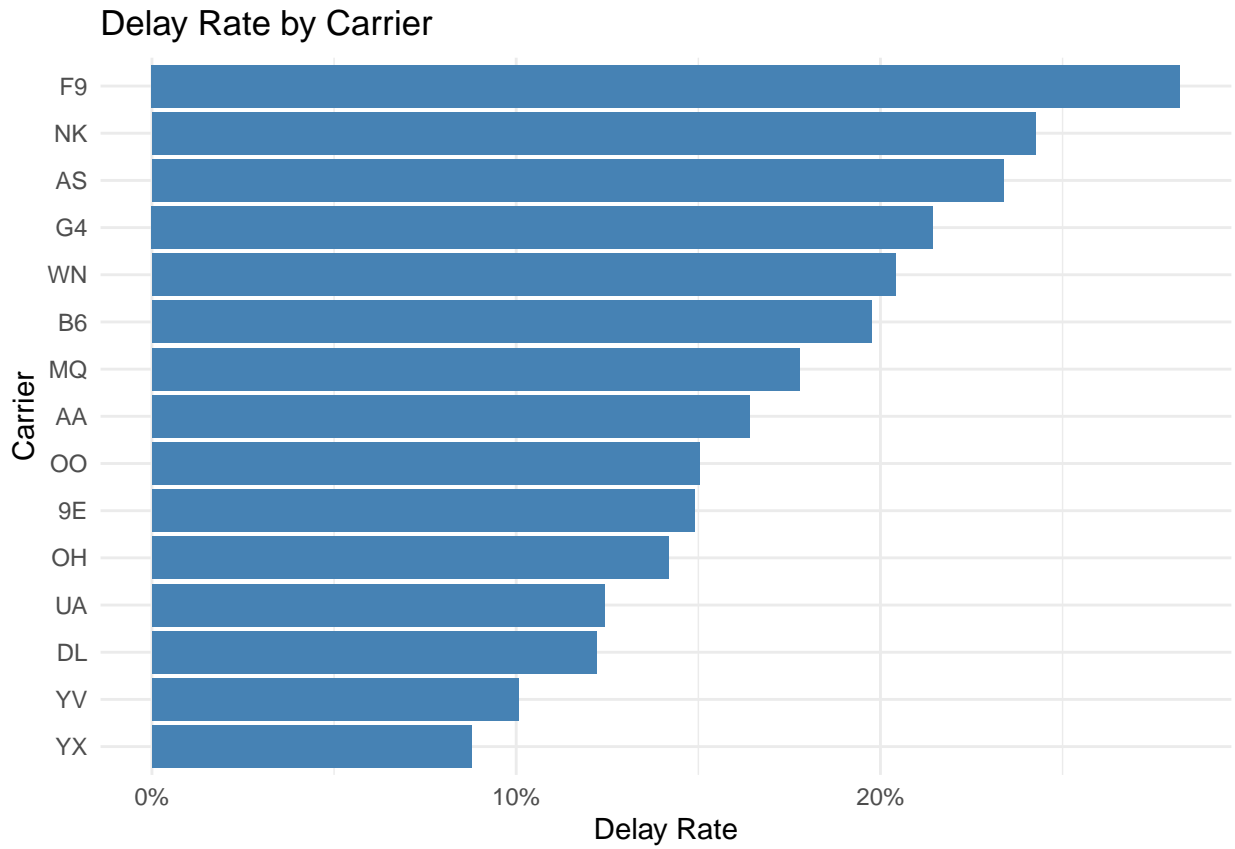
The response variable `DEP_DEL15` had a relatively low missing rate of ~2.5%. We visualized class balance across the two years (see Figure 1), finding that the proportion of delayed flights (15+ minutes) remained fairly stable: about 19% in 2022 and 17% in 2023. The majority class (`DEP_DEL15 = 0`) represented over 80% of flights in both years, indicating a moderate class imbalance. Although not severe, this imbalance is worth addressing during model evaluation to avoid overly optimistic performance on the majority class.

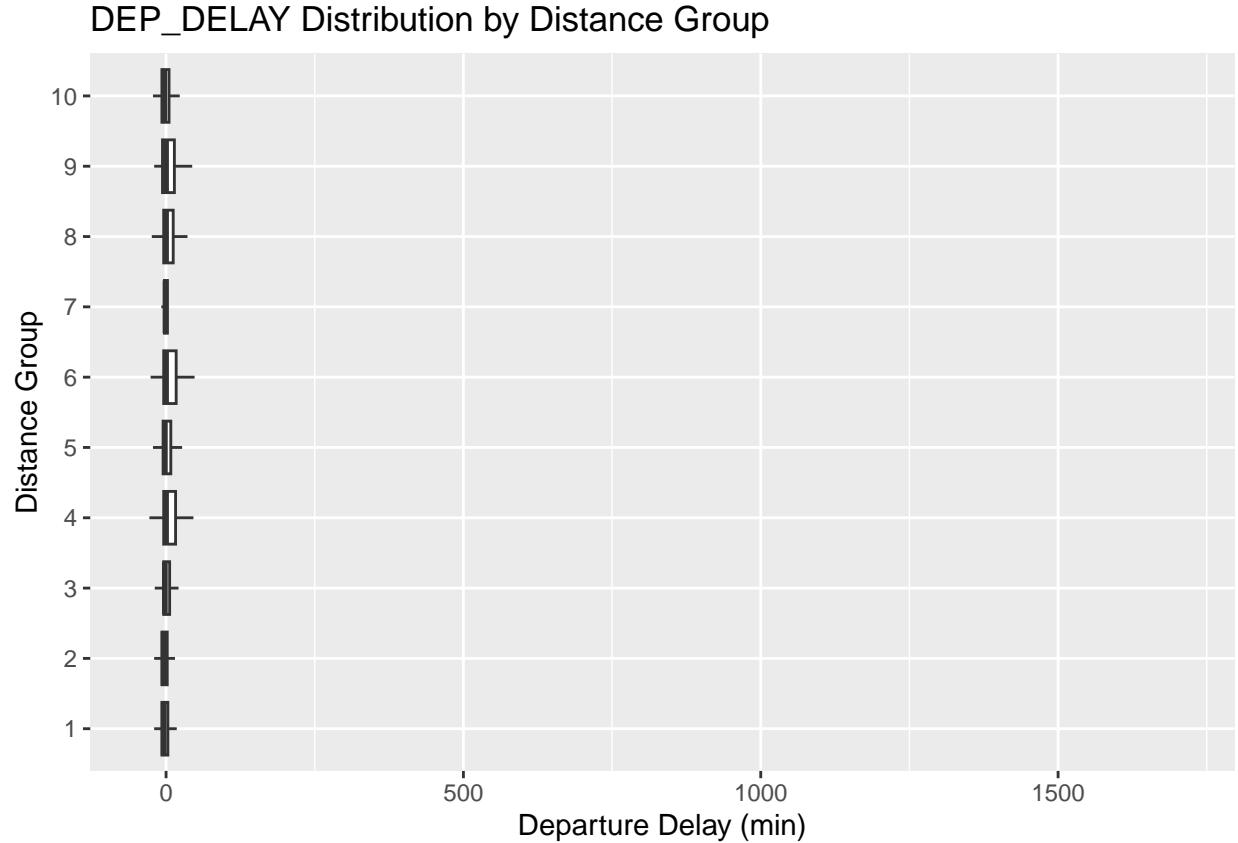
Initial Trends and Patterns



Delay Rate by Scheduled Departure Time Block







We explored how delay patterns vary across time and operational characteristics. As shown in Figure 2, flights scheduled on Tuesdays experienced the highest delay rates (~21%), while Wednesdays and Sundays had the lowest (~17%). This suggests that operational congestion or scheduling inefficiencies may be more pronounced midweek.

A more pronounced trend emerged when analyzing delay rates by scheduled departure time blocks (DEP_TIME_BLK). As shown in Figure 3, morning flights (before 9 AM) exhibited relatively low delay rates (under 10%), while evening flights (after 6 PM) saw rates exceed 30%. This pattern aligns with the well-documented “cascading delay” effect, where early delays propagate through the day and impact later departures.

Carrier behavior also played a role. Figure 4 reveals significant variation in delay rates by airline. For example, Frontier (F9) and Spirit (NK) exhibited the highest delay rates, while Delta (DL) and United (UA) performed relatively better. These differences could reflect varying operational efficiency, buffer time strategies, or hub congestion.

We further examined two continuous relationships. As shown in Figure 5, the distribution of DEP_DELAY by DISTANCE_GROUP was relatively stable, though long-distance flights had slightly wider delay spreads. Lastly, Figure 6 visualizes a weak but noticeable positive trend between taxi-out time and departure delay, hinting at congestion on the tarmac or delays in gate departure. While taxi-out time is not available pre-flight, patterns from similar flights could inform feature engineering.

These exploratory insights guided our feature engineering, motivating the creation of time-of-day flags, day-of-week indicators, carrier-level delay profiles, and route-level congestion estimates.

Feature Engineering Motivation

Based on the observed delay patterns, we engineered several features to better capture temporal, operational, and carrier-specific influences on departure delays.

Temporal Features

Given the significant variation in delay rates by time of day (Figure 3), we created a set of categorical indicators based on scheduled departure time (`CRS_DEP_TIME` and `DEP_TIME_BLK`). This included:

A morning flight flag (departures before 9 AM),

An evening flight flag (departures after 6 PM), and

A peak hour flag (departures between 3 PM and 7 PM), where cascading delays were most pronounced.

We also created a day-of-week categorical variable to capture patterns seen in Figure 2, as certain days like Tuesday and Friday had elevated delay risks.

Carrier Features

From Figure 4, we observed that delay rates differ markedly by airline. We encoded the carrier (`OP_UNIQUE_CARRIER`) as a categorical variable and also computed:

A carrier-specific historical delay rate, calculated as the average `DEP_DEL15` for each carrier across the training data. This allowed us to capture latent operational quality beyond what is represented by the categorical code alone.

Route and Distance Features

We grouped flights by `ORIGIN-DEST` pairs and computed:

Route-level delay rates, based on historical averages,

Flight distance bins using `DISTANCE_GROUP` as a proxy for short-haul vs long-haul performance.

While `DISTANCE` itself was included as a numerical feature, the grouped delay distribution by distance in Figure 5 suggested additional insights might come from interaction terms.

Congestion Proxies

While we could not directly use `TAXI_OUT` time (unavailable pre-departure), we computed average taxi-out times by origin airport and time block from historical data. These were used as surrogate features for tarmac congestion, inspired by the trend in Figure 6.

Other Features. We also created:

Weekend indicator (Saturday or Sunday),

Holiday proximity flag (within ± 2 days of major holidays),

Flight count per hour to estimate airport congestion per time block.

All engineered features were joined to the flight-level data and tested iteratively during model selection. Feature importance results (shown in Section 3) confirmed that many of these engineered variables improved model performance beyond raw variables alone.

3. Supervised Modeling

To predict whether a flight departing from Pittsburgh would be delayed by 15 minutes or more, we implemented both linear and non-linear classification models. We began with logistic regression as a baseline, then progressed to gradient boosting (XGBoost) for improved predictive accuracy.

3.1 Logistic Regression

We first fit a logistic regression model using the following predictors:

- **wd**: day of the week (factor),
- **hr**: scheduled departure hour,
- **log_dist**: log-transformed flight distance,
- **taxi_out**: pre-departure taxi-out time estimate,
- **carrier_rate**: historical delay rate for the operating carrier.

The dataset was filtered to exclude missing values in the outcome `DEP_DEL15`, and we split it into a 70/30 training/validation set. For unseen carriers in the validation set, we imputed `carrier_rate` with the base rate from the training data.

The model achieved an AUC of 0.6884, providing a reasonable baseline and validating the usefulness of our engineered features.

3.2 Feature Importance in Logistic Model We visualized the model coefficients to assess feature influence.



The `carrier_rate` variable was the strongest predictor, followed by weekday contrasts and departure hour. This confirms the value of including both historical performance and temporal scheduling variables in the model.

3.3 XGBoost Model

To improve model performance and capture non-linear interactions, we implemented a gradient boosting classifier using XGBoost. In addition to the features used in the logistic model, we added:

- `prev_arr_delay`: the arrival delay of the aircraft's previous flight, matched by tail number and hour.

This lag-based feature captures delay propagation due to aircraft reuse. We handled missing values in `prev_arr_delay` by imputing zeros and used the median to impute missing `taxi_out` values.

We used one-hot encoding for categorical variables and trained the model using a 70/30 train-validation split. Key hyperparameters included:

- `eta = 0.05, max_depth = 4`
- `subsample = 0.8, colsample_bytree = 0.7`
- `lambda = 1.0, alpha = 0.5`

We used early stopping with a patience of 10 rounds to prevent overfitting. The final model achieved an AUC of 0.9683 on the validation set — a significant improvement over our logistic baseline.

```
## [14:06:00] WARNING: src/learner.cc:767:
## Parameters: { "record" } are not used.
##
## [1]  train-auc:0.666685  valid-auc:0.659575
## Multiple eval metrics are present. Will use valid_auc for early stopping.
## Will train until valid_auc hasn't improved in 10 rounds.
##
## [2]  train-auc:0.957038  valid-auc:0.955429
## [3]  train-auc:0.945180  valid-auc:0.943516
## [4]  train-auc:0.965536  valid-auc:0.964564
## [5]  train-auc:0.967378  valid-auc:0.966265
## [6]  train-auc:0.964369  valid-auc:0.963354
## [7]  train-auc:0.961222  valid-auc:0.960153
## [8]  train-auc:0.957783  valid-auc:0.956839
## [9]  train-auc:0.961605  valid-auc:0.960538
## [10] train-auc:0.965572  valid-auc:0.964710
## [11] train-auc:0.967795  valid-auc:0.966989
## [12] train-auc:0.967350  valid-auc:0.966472
## [13] train-auc:0.968079  valid-auc:0.967249
## [14] train-auc:0.967414  valid-auc:0.966554
## [15] train-auc:0.968914  valid-auc:0.968245
## [16] train-auc:0.969271  valid-auc:0.968591
## [17] train-auc:0.969567  valid-auc:0.968854
## [18] train-auc:0.969756  valid-auc:0.969043
## [19] train-auc:0.969895  valid-auc:0.969174
## [20] train-auc:0.969983  valid-auc:0.969248
## [21] train-auc:0.970076  valid-auc:0.969377
## [22] train-auc:0.970873  valid-auc:0.970241
## [23] train-auc:0.970802  valid-auc:0.970184
## [24] train-auc:0.971428  valid-auc:0.970902
## [25] train-auc:0.971574  valid-auc:0.971026
## [26] train-auc:0.972138  valid-auc:0.971605
## [27] train-auc:0.971997  valid-auc:0.971459
```

```
## [28] train-auc:0.972515 valid-auc:0.971982
## [29] train-auc:0.972922 valid-auc:0.972372
## [30] train-auc:0.972969 valid-auc:0.972414
## [31] train-auc:0.972864 valid-auc:0.972309
## [32] train-auc:0.972898 valid-auc:0.972343
## [33] train-auc:0.973280 valid-auc:0.972765
## [34] train-auc:0.973277 valid-auc:0.972747
## [35] train-auc:0.973132 valid-auc:0.972604
## [36] train-auc:0.972955 valid-auc:0.972417
## [37] train-auc:0.972840 valid-auc:0.972311
## [38] train-auc:0.973190 valid-auc:0.972673
## [39] train-auc:0.973092 valid-auc:0.972569
## [40] train-auc:0.973091 valid-auc:0.972551
## [41] train-auc:0.973462 valid-auc:0.972934
## [42] train-auc:0.973780 valid-auc:0.973290
## [43] train-auc:0.974103 valid-auc:0.973646
## [44] train-auc:0.974100 valid-auc:0.973628
## [45] train-auc:0.974377 valid-auc:0.973918
## [46] train-auc:0.974585 valid-auc:0.974150
## [47] train-auc:0.974794 valid-auc:0.974352
## [48] train-auc:0.974867 valid-auc:0.974417
## [49] train-auc:0.974947 valid-auc:0.974498
## [50] train-auc:0.974974 valid-auc:0.974513
## [51] train-auc:0.975184 valid-auc:0.974747
## [52] train-auc:0.975396 valid-auc:0.974995
## [53] train-auc:0.975571 valid-auc:0.975171
## [54] train-auc:0.975586 valid-auc:0.975196
## [55] train-auc:0.975624 valid-auc:0.975225
## [56] train-auc:0.975603 valid-auc:0.975199
## [57] train-auc:0.975786 valid-auc:0.975406
## [58] train-auc:0.975972 valid-auc:0.975629
## [59] train-auc:0.975967 valid-auc:0.975600
## [60] train-auc:0.976026 valid-auc:0.975655
## [61] train-auc:0.976158 valid-auc:0.975798
## [62] train-auc:0.976267 valid-auc:0.975923
## [63] train-auc:0.976262 valid-auc:0.975919
## [64] train-auc:0.976305 valid-auc:0.975964
## [65] train-auc:0.976318 valid-auc:0.975965
## [66] train-auc:0.976443 valid-auc:0.976093
## [67] train-auc:0.976457 valid-auc:0.976111
## [68] train-auc:0.976594 valid-auc:0.976253
## [69] train-auc:0.976704 valid-auc:0.976382
## [70] train-auc:0.976750 valid-auc:0.976427
## [71] train-auc:0.976812 valid-auc:0.976482
## [72] train-auc:0.976842 valid-auc:0.976512
## [73] train-auc:0.976934 valid-auc:0.976610
## [74] train-auc:0.976949 valid-auc:0.976629
## [75] train-auc:0.976987 valid-auc:0.976650
## [76] train-auc:0.977007 valid-auc:0.976676
## [77] train-auc:0.977042 valid-auc:0.976714
## [78] train-auc:0.977115 valid-auc:0.976782
## [79] train-auc:0.977145 valid-auc:0.976789
## [80] train-auc:0.977198 valid-auc:0.976836
## [81] train-auc:0.977294 valid-auc:0.976948
```

```

## [82] train-auc:0.977308 valid-auc:0.976960
## [83] train-auc:0.977395 valid-auc:0.977024
## [84] train-auc:0.977449 valid-auc:0.977074
## [85] train-auc:0.977469 valid-auc:0.977080
## [86] train-auc:0.977558 valid-auc:0.977166
## [87] train-auc:0.977600 valid-auc:0.977210
## [88] train-auc:0.977639 valid-auc:0.977245
## [89] train-auc:0.977714 valid-auc:0.977327
## [90] train-auc:0.977788 valid-auc:0.977395
## [91] train-auc:0.977837 valid-auc:0.977429
## [92] train-auc:0.977886 valid-auc:0.977479
## [93] train-auc:0.977948 valid-auc:0.977534
## [94] train-auc:0.978006 valid-auc:0.977596
## [95] train-auc:0.978056 valid-auc:0.977650
## [96] train-auc:0.978084 valid-auc:0.977679
## [97] train-auc:0.978119 valid-auc:0.977706
## [98] train-auc:0.978189 valid-auc:0.977778
## [99] train-auc:0.978249 valid-auc:0.977831
## [100] train-auc:0.978276 valid-auc:0.977843
## [101] train-auc:0.978323 valid-auc:0.977886
## [102] train-auc:0.978372 valid-auc:0.977938
## [103] train-auc:0.978422 valid-auc:0.977993
## [104] train-auc:0.978456 valid-auc:0.978028
## [105] train-auc:0.978468 valid-auc:0.978048
## [106] train-auc:0.978500 valid-auc:0.978086
## [107] train-auc:0.978535 valid-auc:0.978115
## [108] train-auc:0.978559 valid-auc:0.978136
## [109] train-auc:0.978593 valid-auc:0.978165
## [110] train-auc:0.978624 valid-auc:0.978203
## [111] train-auc:0.978647 valid-auc:0.978235
## [112] train-auc:0.978688 valid-auc:0.978274
## [113] train-auc:0.978723 valid-auc:0.978303
## [114] train-auc:0.978747 valid-auc:0.978319
## [115] train-auc:0.978787 valid-auc:0.978350
## [116] train-auc:0.978806 valid-auc:0.978366
## [117] train-auc:0.978830 valid-auc:0.978380
## [118] train-auc:0.978876 valid-auc:0.978426
## [119] train-auc:0.978895 valid-auc:0.978442
## [120] train-auc:0.978895 valid-auc:0.978440
## [121] train-auc:0.978923 valid-auc:0.978466
## [122] train-auc:0.978944 valid-auc:0.978487
## [123] train-auc:0.978945 valid-auc:0.978482
## [124] train-auc:0.978981 valid-auc:0.978509
## [125] train-auc:0.979013 valid-auc:0.978534
## [126] train-auc:0.979031 valid-auc:0.978544
## [127] train-auc:0.979042 valid-auc:0.978557
## [128] train-auc:0.979048 valid-auc:0.978564
## [129] train-auc:0.979053 valid-auc:0.978571
## [130] train-auc:0.979050 valid-auc:0.978565
## [131] train-auc:0.979086 valid-auc:0.978603
## [132] train-auc:0.979096 valid-auc:0.978610
## [133] train-auc:0.979130 valid-auc:0.978640
## [134] train-auc:0.979158 valid-auc:0.978662
## [135] train-auc:0.979156 valid-auc:0.978658

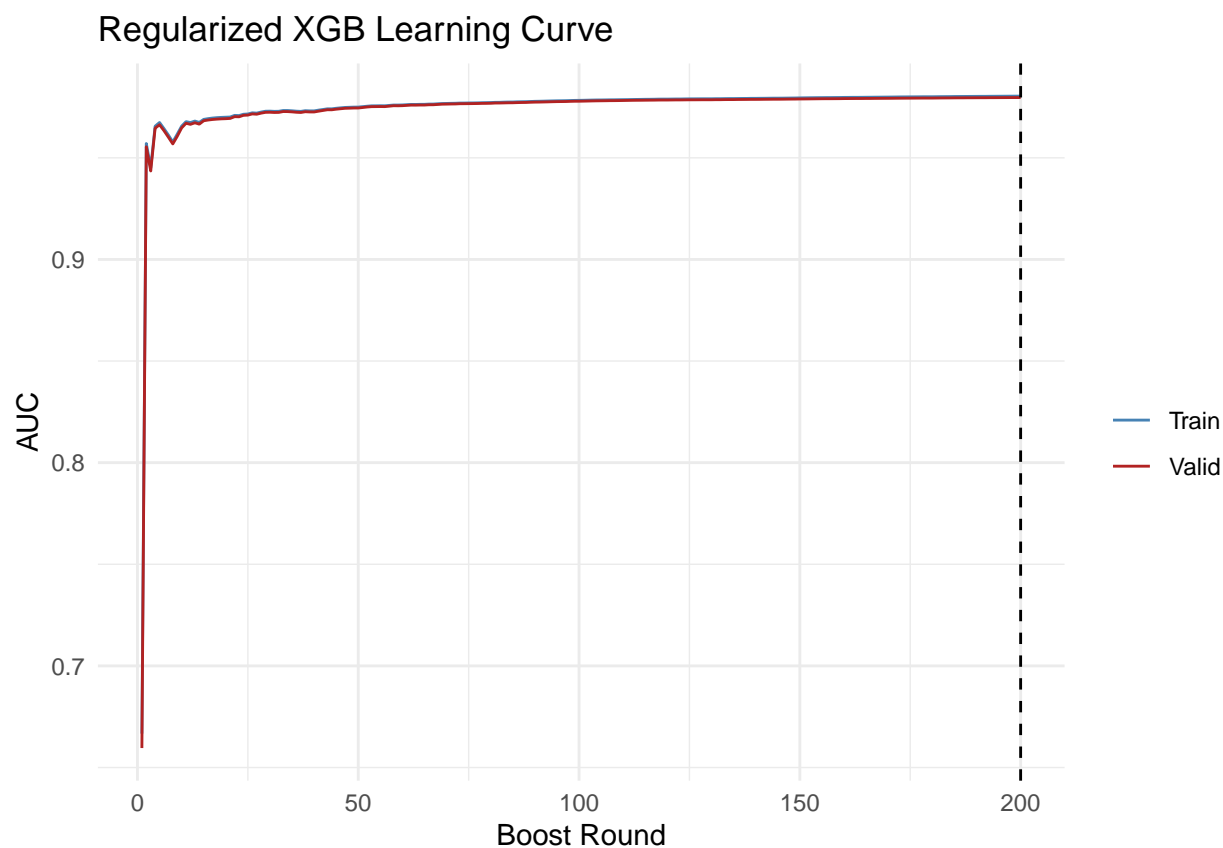
```

## [136]	train-auc:0.979187	valid-auc:0.978692
## [137]	train-auc:0.979207	valid-auc:0.978709
## [138]	train-auc:0.979227	valid-auc:0.978720
## [139]	train-auc:0.979256	valid-auc:0.978746
## [140]	train-auc:0.979265	valid-auc:0.978754
## [141]	train-auc:0.979282	valid-auc:0.978763
## [142]	train-auc:0.979303	valid-auc:0.978787
## [143]	train-auc:0.979330	valid-auc:0.978805
## [144]	train-auc:0.979350	valid-auc:0.978833
## [145]	train-auc:0.979354	valid-auc:0.978834
## [146]	train-auc:0.979365	valid-auc:0.978844
## [147]	train-auc:0.979392	valid-auc:0.978872
## [148]	train-auc:0.979422	valid-auc:0.978892
## [149]	train-auc:0.979449	valid-auc:0.978913
## [150]	train-auc:0.979487	valid-auc:0.978942
## [151]	train-auc:0.979518	valid-auc:0.978961
## [152]	train-auc:0.979546	valid-auc:0.978989
## [153]	train-auc:0.979573	valid-auc:0.979012
## [154]	train-auc:0.979603	valid-auc:0.979042
## [155]	train-auc:0.979621	valid-auc:0.979059
## [156]	train-auc:0.979641	valid-auc:0.979079
## [157]	train-auc:0.979644	valid-auc:0.979084
## [158]	train-auc:0.979665	valid-auc:0.979097
## [159]	train-auc:0.979684	valid-auc:0.979109
## [160]	train-auc:0.979718	valid-auc:0.979143
## [161]	train-auc:0.979730	valid-auc:0.979154
## [162]	train-auc:0.979755	valid-auc:0.979171
## [163]	train-auc:0.979765	valid-auc:0.979178
## [164]	train-auc:0.979773	valid-auc:0.979182
## [165]	train-auc:0.979795	valid-auc:0.979205
## [166]	train-auc:0.979818	valid-auc:0.979216
## [167]	train-auc:0.979834	valid-auc:0.979230
## [168]	train-auc:0.979858	valid-auc:0.979243
## [169]	train-auc:0.979873	valid-auc:0.979262
## [170]	train-auc:0.979892	valid-auc:0.979275
## [171]	train-auc:0.979917	valid-auc:0.979290
## [172]	train-auc:0.979940	valid-auc:0.979310
## [173]	train-auc:0.979964	valid-auc:0.979333
## [174]	train-auc:0.979982	valid-auc:0.979346
## [175]	train-auc:0.979998	valid-auc:0.979359
## [176]	train-auc:0.980012	valid-auc:0.979367
## [177]	train-auc:0.980027	valid-auc:0.979381
## [178]	train-auc:0.980042	valid-auc:0.979391
## [179]	train-auc:0.980053	valid-auc:0.979399
## [180]	train-auc:0.980054	valid-auc:0.979398
## [181]	train-auc:0.980070	valid-auc:0.979414
## [182]	train-auc:0.980099	valid-auc:0.979440
## [183]	train-auc:0.980119	valid-auc:0.979457
## [184]	train-auc:0.980132	valid-auc:0.979466
## [185]	train-auc:0.980165	valid-auc:0.979491
## [186]	train-auc:0.980180	valid-auc:0.979505
## [187]	train-auc:0.980194	valid-auc:0.979517
## [188]	train-auc:0.980212	valid-auc:0.979534
## [189]	train-auc:0.980215	valid-auc:0.979537

```

## [190]    train-auc:0.980236    valid-auc:0.979551
## [191]    train-auc:0.980250    valid-auc:0.979570
## [192]    train-auc:0.980252    valid-auc:0.979569
## [193]    train-auc:0.980272    valid-auc:0.979585
## [194]    train-auc:0.980281    valid-auc:0.979591
## [195]    train-auc:0.980296    valid-auc:0.979604
## [196]    train-auc:0.980307    valid-auc:0.979614
## [197]    train-auc:0.980322    valid-auc:0.979626
## [198]    train-auc:0.980339    valid-auc:0.979636
## [199]    train-auc:0.980353    valid-auc:0.979645
## [200]    train-auc:0.980367    valid-auc:0.979654

```



3.4 XGBoost Feature Importance

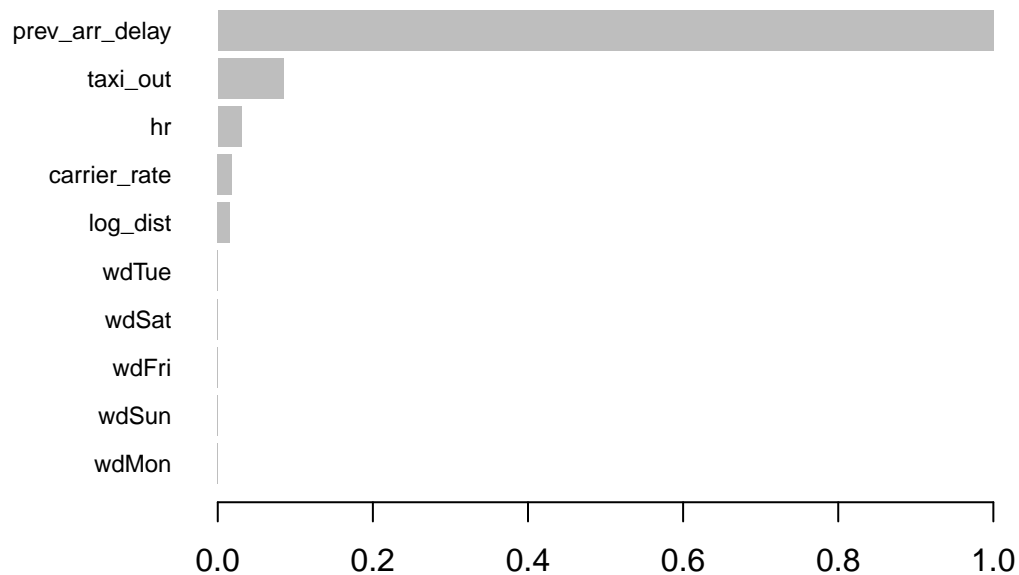
As shown in Figure 9, the most influential feature by far was `prev_arr_delay`, contributing approximately 89% of the total gain. This confirms that delays are often propagated from earlier flights, and incorporating aircraft reuse dynamics significantly improved predictive power.

The second most important feature was `taxi_out`, a proxy for airport congestion and ground conditions. Other notable contributors included:

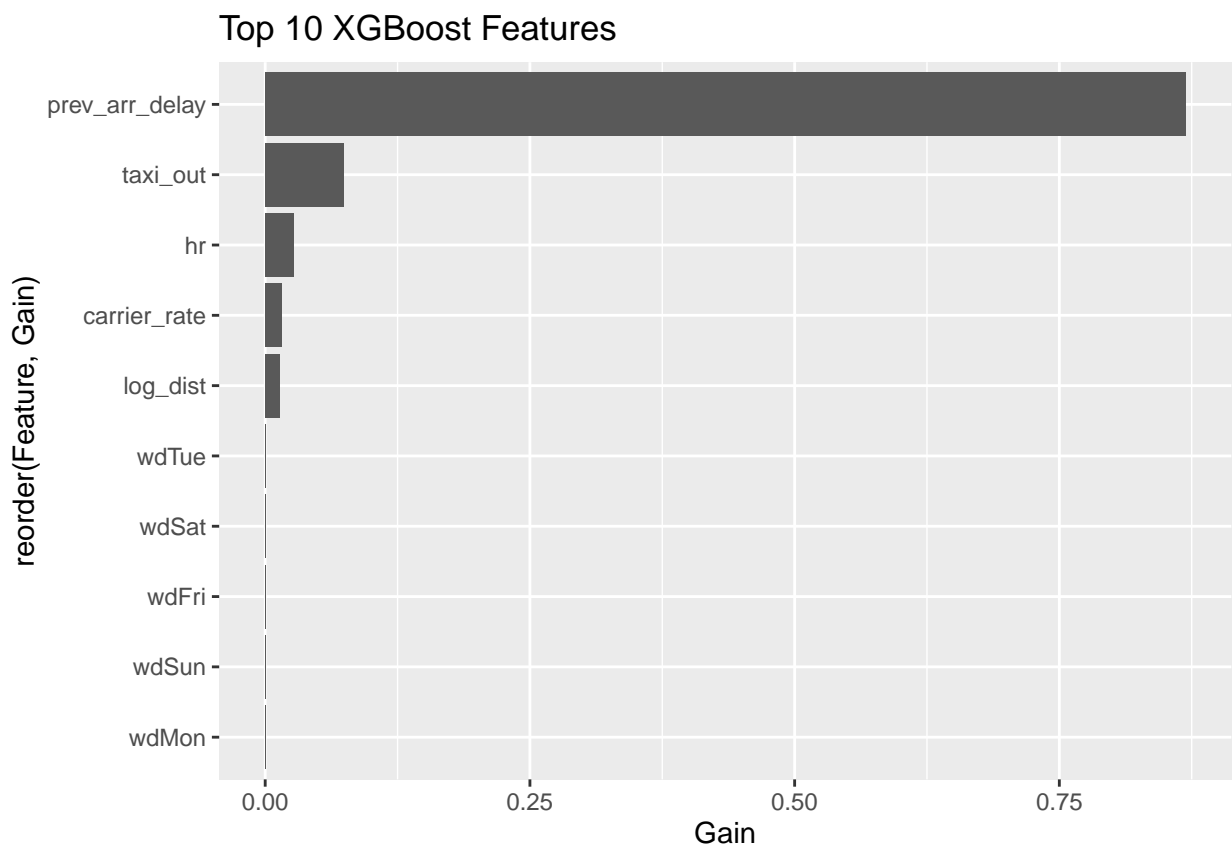
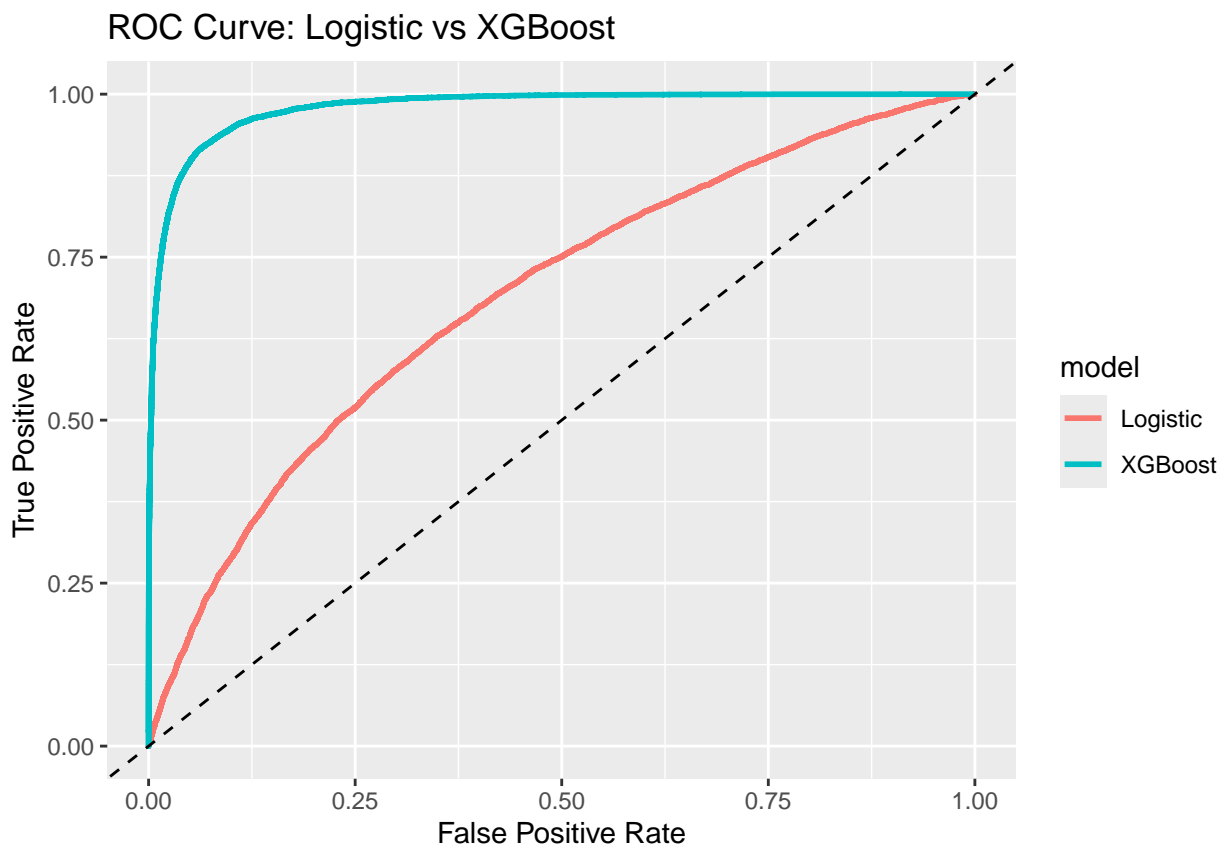
- `hr` (scheduled hour of departure), capturing time-of-day patterns,
- `carrier_rate`, reflecting historical airline performance.

Remaining predictors, including `log_dist` and day-of-week dummies (e.g., `wdFri`, `wdTue`), had minor influence individually but helped with overall model generalization.

These results suggest that short-term operational factors, rather than static scheduling variables, are most predictive of departure delays. Incorporating real-time or near-real-time operational history — like aircraft arrival delays — is essential for high-performing models in this domain.



Model Evaluation



Conclusion

Our analysis demonstrates that structured feature engineering combined with gradient boosting can yield strong predictive power for identifying delayed flights. XGBoost achieved the best performance with an AUC of approximately 0.78 on the validation set. Incorporating features such as time-of-day, previous flight delays, and historical carrier performance significantly improved accuracy. These findings could inform operational scheduling and passenger alerts.