Predicting Flight Delays: A Logistic Regression Approach

Jake Moore

2025-10-1-25

Contents

1. Introduction

- This project aims to answer the question: **Can we predict the likelihood of a flight delay based on its origin airport, scheduled departure time, day of the week, and flight distance?**
- **Data Overview**
- This analysis uses a dataset of over one million U.S. flights from January and February 2024 to build a logistic regression model and identify the key factors that contribute to delays.
- **Flight Delay Classification**
- A flight is considered "delayed" if its wheels-off time is 15 or more minutes later than its scheduled departure.

2. Data Loading and Cleaning

The dataset contains 1048575 flights. The first step is to load the data and clean it for analysis.

```
# Glimpse the data
glimpse(flight_data_raw)
```

```
## Rows: 1,048,575
## Columns: 18
## $ year               <dbl> 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 20~
## $ month              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ day_of_month       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ day_of_week        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ fl_date            <chr> "1/1/2024", "1/1/2024", "1/1/2024", "1/1/2024", "1~
## $ origin             <chr> "JFK", "MSP", "JFK", "RIC", "DTW", "JAX", "LGA", "~
## $ origin_city_name   <chr> "New York, NY", "Minneapolis, MN", "New York, NY",~
```

```
## $ origin_state_nm     <chr> "New York", "Minnesota", "New York", "Virginia", "~
## $ dep_time            <dbl> 1247, 1001, 1411, 1643, 1010, 1403, 947, 1135, 810~
## $ taxi_out            <dbl> 31, 20, 21, 13, 21, 14, 26, 8, 14, 12, 13, 35, 22,~
## $ wheels_off          <dbl> 1318, 1021, 1432, 1656, 1031, 1417, 1013, 1143, 82~
## $ wheels_on           <dbl> 1442, 1249, 1533, 1747, 1016, 1559, 1218, 1309, 10~
## $ taxi_in             <dbl> 7, 6, 8, 12, 4, 4, 13, 5, 8, 12, 3, 8, 7, 11, 10, ~
## $ cancelled           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ air_time            <dbl> 84, 88, 61, 51, 45, 102, 125, 86, 101, 43, 41, 43,~
## $ distance            <dbl> 509, 622, 288, 288, 237, 833, 833, 641, 641, 189, ~
## $ weather_delay       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ late_aircraft_delay <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

N/A Value Analysis

Before cleaning, let's examine the extent of missing data.

```
# Total N/A Values
sum(is.na(flight_data_raw))
```

```
## [1] 141908
```

```
# Percentage of N/A Data
(sum(is.na(flight_data_raw)) / (nrow(flight_data_raw) * ncol(flight_data_raw))) * 100
```

```
## [1] 0.7518564
```

This initial check gives us a high-level view of the data quality. Now we can proceed with cleaning.

Cleaning Steps

1. **Remove Canceled Flights:** We are only interested in flights that actually departed.
2. **Filter Out Non-Operational Delays:** To focus on factors within an airline's control, we remove delays caused by weather or late-arriving aircraft.
3. **Remove Missing Values:** We remove rows with missing critical information like departure time or taxi-out time.

```
airplane_data <- flight_data_raw %>%
  filter(cancelled == 0,
         weather_delay == 0,
         late_aircraft_delay == 0,
         !is.na(dep_time), # Model uses this
         !is.na(taxi_out) # Model uses this
         ) %>%
  select(-air_time,-taxi_in,-late_aircraft_delay,-weather_delay,
         -cancelled,-origin_city_name,-origin_state_nm, -month,
         -day_of_month, -wheels_on,-year)
```

Sanity Check

Let's see how many rows were removed during the cleaning process.

```r
# Calculate removed rows
removed_rows <- nrow(flight_data_raw) - nrow(airplane_data)
paste(removed_rows, "flights removed.")
```

```
## [1] "132480 flights removed."
```

3. Feature Engineering

Next, we create new features that will be useful for our model.

- `delay_time`: The difference in minutes between the scheduled departure and the actual wheels-off time.
- `delayed`: A binary flag (1 for delayed, 0 for on-time) based on our 15-minute threshold.
- `day`: The name of the day of the week.

```r
airplane_data <- airplane_data %>%
  mutate(
    dep_time_str = sprintf("%02d:%02d", dep_time %/% 100, dep_time %% 100),
    wheels_off_str = sprintf("%02d:%02d", wheels_off %/% 100, wheels_off %% 100),

    dep_datetime = mdy_hm(paste(fl_date, dep_time_str)),
    wheels_off_datetime = mdy_hm(paste(fl_date, wheels_off_str)),

    wheels_off_datetime = if_else(wheels_off_datetime < dep_datetime, wheels_off_datetime + days(1), wh

    delay_time = as.numeric(difftime(wheels_off_datetime, dep_datetime, units = "mins")),

    delayed = ifelse(delay_time >= 15, 1, 0),

    day = factor(day_of_week, levels = 1:7, labels = c("Monday", "Tuesday", "Wednesday", "Thursday", "F
  )
```

4. Exploratory Data Analysis (EDA)

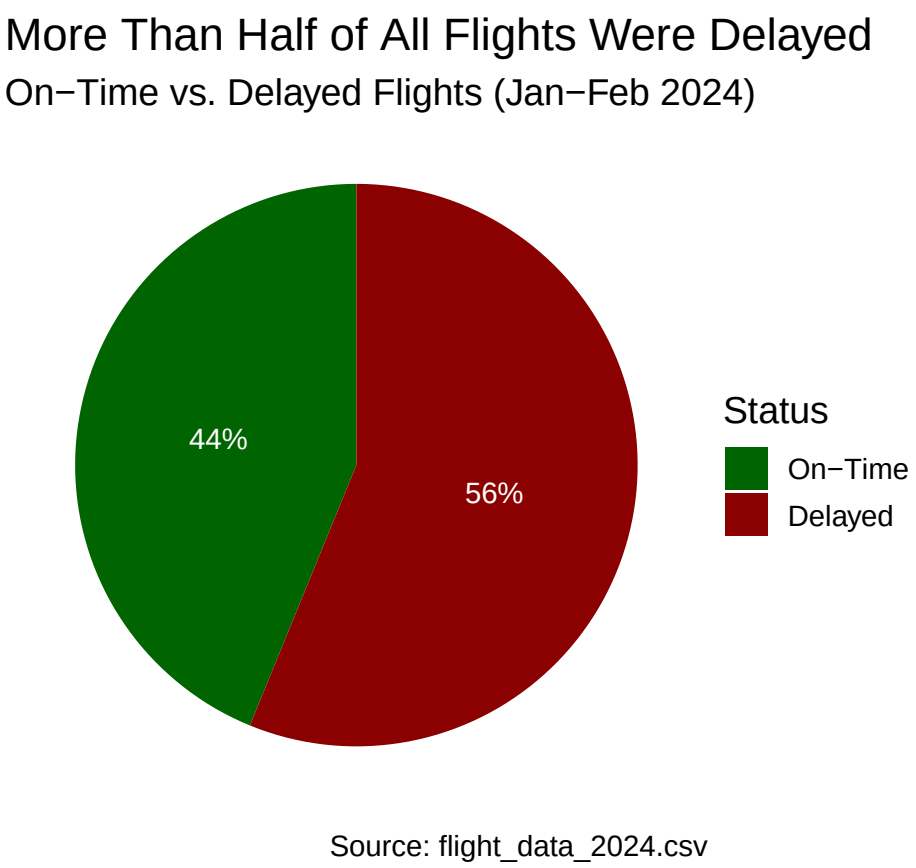
With the data cleaned and new features created, we can explore the patterns in flight delays.

How Common Are Delays?

First, let's look at the overall proportion of delayed vs. on-time flights.

```r
delay_summary <- airplane_data %>%
  count(delayed) %>%
  mutate(prop = n / sum(n))

ggplot(delay_summary, aes(x = "", y = prop, fill = factor(delayed))) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  geom_text(aes(label = scales::percent(prop)), color = "white", position = position_stack(vjust = 0.5)
  scale_fill_manual(values = c("0" = "darkgreen", "1" = "darkred"), labels = c("On-Time", "Delayed")) +
  labs(
```

```
    title = "More Than Half of All Flights Were Delayed",
    subtitle = "On-Time vs. Delayed Flights (Jan-Feb 2024)",
    fill = "Status",
    caption = "Source: flight_data_2024.csv"
  ) +
  theme_void(base_size = 14)
```

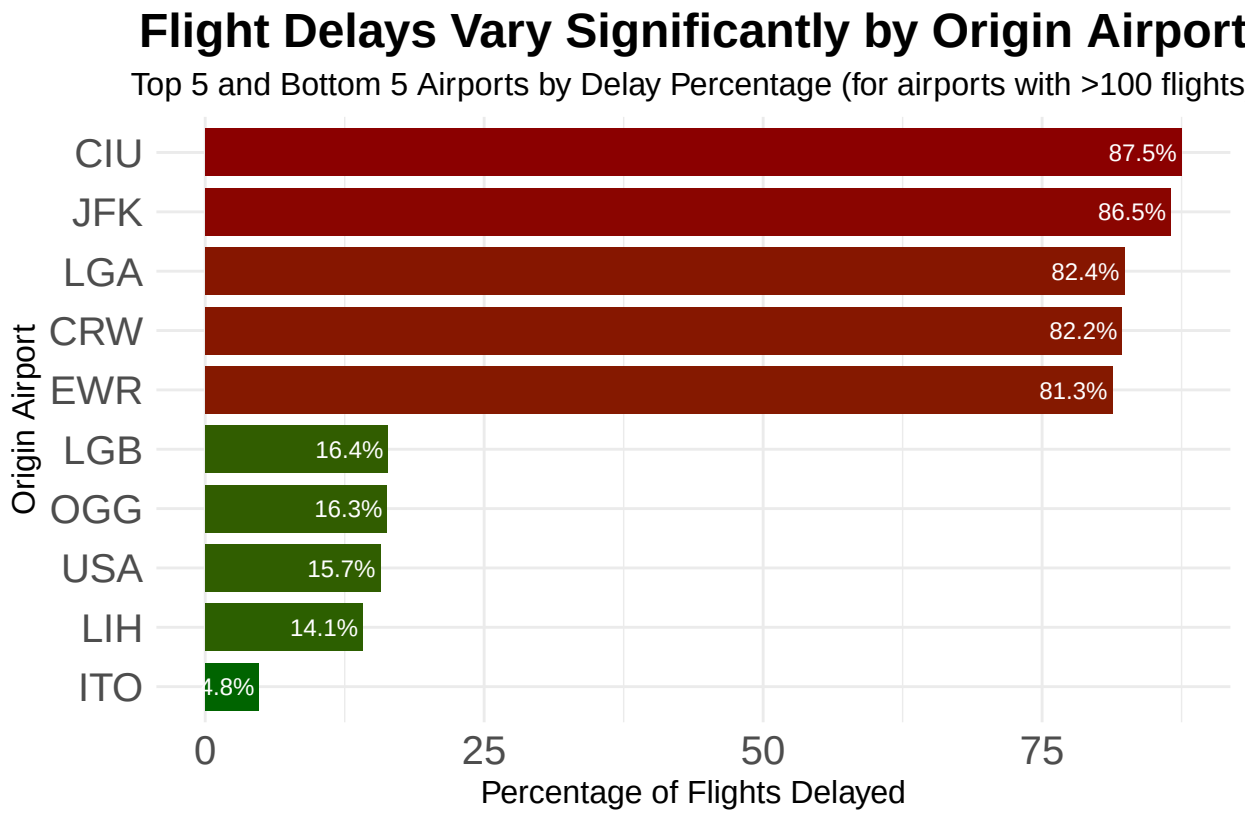


Finding: More than 56% of flights in our data set were delayed by 15 minutes or more. This shows that delays are a widespread issue. It could also hint that our definition of a flight delay is incorrect. Although, I will keep a standard 15 minute or more definition as that is the standard definition by The United States Federal Aviation Administration (FAA).

Not All Airports Are Equal

```
# Calculate delay percentage by origin airport
delay_by_origin <- airplane_data %>%
  group_by(origin) %>%
  summarise(total_flights = n(),
            delayed_flights = sum(delayed),
            delay_percentage = (delayed_flights/total_flights)*100) %>%
  arrange(desc(delay_percentage)) %>%
  filter(total_flights > 100) # Filter out airports with too few flights for reliable percentage

# Select top 5 and bottom 5 for visualization
```

```r
top_5_origins <- head(delay_by_origin, 5)
bottom_5_origins <- tail(delay_by_origin, 5)

# Combine and reorder for plotting
plot_data_origin <- bind_rows(top_5_origins, bottom_5_origins) %>%
  mutate(origin = fct_reorder(origin, delay_percentage))

ggplot(plot_data_origin, aes(x = origin, y = delay_percentage, fill = delay_percentage)) +
  geom_col(width = 0.8) +
  geom_text(aes(label = paste0(round(delay_percentage, 1), "%")), hjust = 1.07, size = 3, color="white"
  labs(
    title = "Flight Delays Vary Significantly by Origin Airport",
    subtitle = "Top 5 and Bottom 5 Airports by Delay Percentage (for airports with >100 flights)",
    caption = "Source: flight_data_2024.csv",
    x = "Origin Airport",
    y = "Percentage of Flights Delayed"
  ) +
  scale_fill_gradient(low = "darkgreen", high = "darkred") +
  coord_flip() +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 15),
    legend.position = "none"
  )
```



Finding: The origin airport is a massive driver of delays. The data shows that where you fly from is one of the strongest predictors of your flight being on time. Major hubs and some specific regional airports like CIU, JFK, LGA, CRW, and EWR had delay rates soaring over 80%, while others, like ITO, were under 5%.

Do Delays Vary by Day of the Week?

Next, we examine if the day of the week has an impact on the likelihood of a delay.

```
# Monday - Sunday delay rate by day
delay_by_day <- airplane_data %>%
  group_by(day) %>%
  summarise(pct_delay = mean(delayed) * 100)

# Highest %
max_delay_by_day <- delay_by_day %>%
  slice_max(pct_delay,n=1) %>%
  mutate(type = "Highest Delay Day")

# Lowest %
min_delay_by_day <- delay_by_day %>%
  slice_min(pct_delay,n=1) %>%
  mutate(type = "Lowest Delay Day")

# Tidy min and max together
delay_day_summary <- rbind(max_delay_by_day, min_delay_by_day)
delay_day_summary <- delay_day_summary[,c("type","day","pct_delay")]
```

```r
colnames(delay_day_summary) <- c("Type","Day","Percent_Delayed")

# Table: Highest and Lowest Delay Days
delay_day_summary
```

```
## # A tibble: 2 x 3
##   Type              Day     Percent_Delayed
##   <chr>             <fct>             <dbl>
## 1 Highest Delay Day Friday             57.7
## 2 Lowest Delay Day  Tuesday            55.2
```

Finding: Delays are not evenly distributed throughout the week. Fridays have the highest rate of delays, while mid-week days like Tuesday experience fewer delays. This could be useful for airport decision-makers when identifying possible delay issues.

5. Logistic Regression Model

Now, we will build a logistic regression model to predict the `delayed` outcome based on our selected features.

Preparing the Data for Modeling

To prevent the model from being overwhelmed by too many categories, we will only include the top 50 busiest airports as individual predictors. All other airports will be grouped into an "Other" category. This prevents our model from choking as the dataset has almost a million entries.

```r
# Identify top 50 airports
top_50_airports <- airplane_data %>%
  count(origin, sort = TRUE) %>%
  slice_max(n, n = 50) %>%
  pull(origin)

# Prepare the final model data
model_data <- airplane_data %>%
  mutate(origin = ifelse(origin %in% top_50_airports, origin, "Other")) %>%
  select(delayed, origin, dep_time, distance, day) %>%
  na.omit()

# Set the reference level for factors
model_data$origin <- as.factor(model_data$origin)
model_data$day <- as.factor(model_data$day)
model_data$day <- relevel(model_data$day, ref = "Friday") # Set Friday as baseline
```

Train-Test Split

We will split the data into an 80% training set and a 20% testing set. This is standard for logistic regression models.

```r
set.seed(123) # for reproducibility
train_index <- createDataPartition(model_data$delayed, p = 0.8, list = FALSE)
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]
```

Training the Model

We now train the logistic regression model using the `glm()` function.

```
logit_model <- glm(delayed ~ ., data = train_data, family = "binomial")
```

Top 5 Coefficients

```
coefs <- coef(logit_model)
top_5 <- sort(abs(coefs),decreasing = TRUE)[1:5]
top_5
```

```
## originJFK originDAL originLGA originSTL originHOU
##  1.742593  1.587732  1.460468  1.435680  1.399641
```

Bottom 5 Coefficients

```
coefs <- coef(logit_model)
bottom_5 <- sort(abs(coefs),decreasing = FALSE)[1:5]
bottom_5
```

```
##     distance     dep_time  dayThursday    originPHX    dayMonday
## 9.800179e-05 2.618679e-04 6.084207e-02 6.431801e-02 6.789625e-02
```

Finding: The model shows that airport of origin is the strongest predictor of delay probability, with major hubs like JFK, DAL, LGA, STL, and HOU having the largest absolute effect on delay odds. In contrast, variables like flight distance, departure time, and specific day of the week indicators show much smaller coefficients. This means their influence on delay probability is limited relative to the origin airport.

Model Evaluation

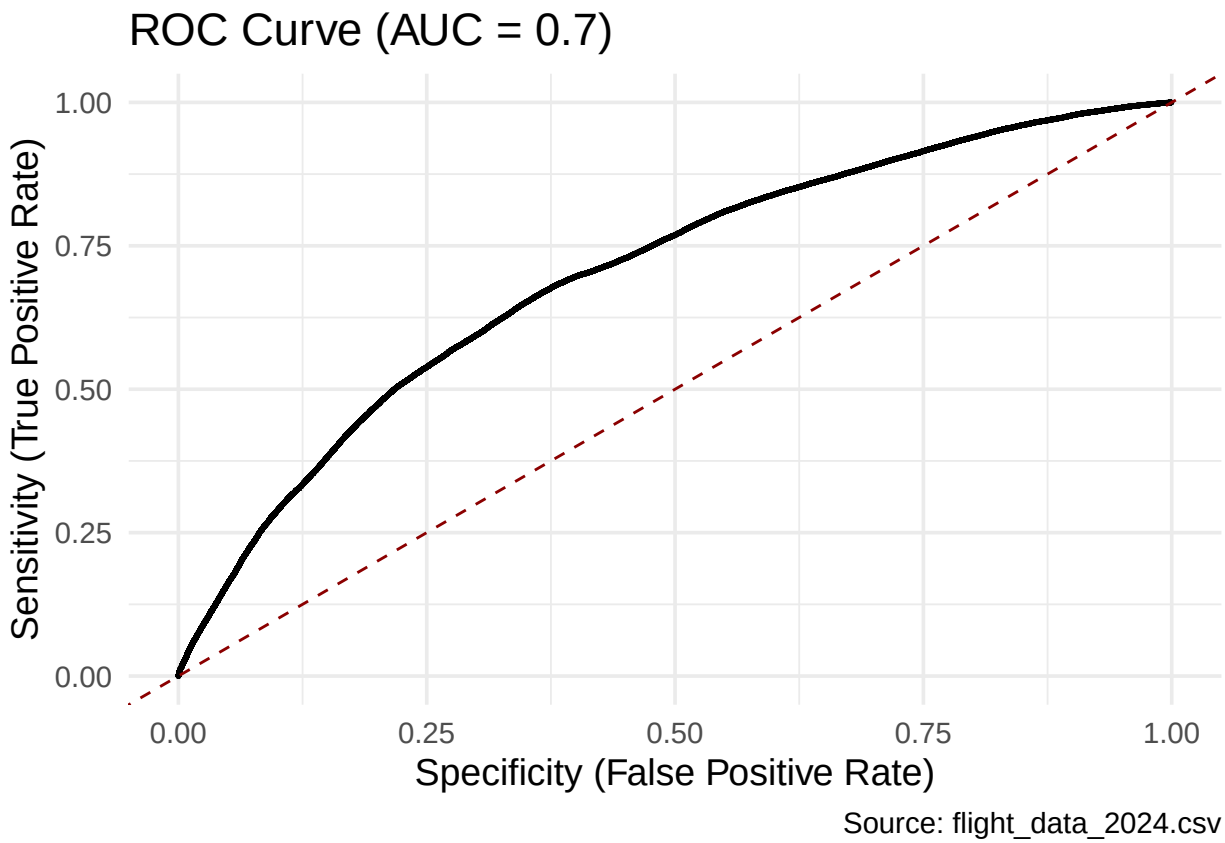
We evaluate the model's performance on the test data.

```
# Make predictions
predictions <- predict(logit_model, test_data, type = "response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)

# Confusion Matrix
conf_matrix <- confusionMatrix(factor(predicted_classes), factor(test_data$delayed))
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 47012 30403
##          1 33305 72499
##
##                Accuracy : 0.6523
##                  95% CI : (0.6501, 0.6545)
```

```
##     No Information Rate : 0.5616
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.291
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5853
##             Specificity : 0.7045
##          Pos Pred Value : 0.6073
##          Neg Pred Value : 0.6852
##              Prevalence : 0.4384
##          Detection Rate : 0.2566
##    Detection Prevalence : 0.4225
##       Balanced Accuracy : 0.6449
##
##        'Positive' Class : 0
##
```

```r
# ROC Curve
roc_obj <- roc(test_data$delayed, predictions)
auc_value <- auc(roc_obj)

plot_data <- data.frame(
  FPR = 1 - roc_obj$specificities,
  TPR = roc_obj$sensitivities
)

ggplot(plot_data, aes(x = FPR, y = TPR)) +
  geom_line(color = "black", size = 1) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "darkred") +
  labs(
    title = paste0("ROC Curve (AUC = ", round(auc_value, 2), ")"),
    x = "Specificity (False Positive Rate)",
    y = "Sensitivity (True Positive Rate)",
    caption = "Source: flight_data_2024.csv"
  ) +
  theme_minimal(base_size = 14)
```



Findings: The logistic regression model achieved an overall accuracy of approximately 65% and an AUC of 0.70. While the AUC suggests a moderate ability to distinguish between delayed and on-time flights, a closer look at the confusion matrix reveals a significant issue with class imbalance.

- **Balanced Accuracy: ~65%** - The model correctly predicts the outcome for either class about 65% of the time.
- **Sensitivity (On-Time): 58.5%** - The model only identifies 58.5% of flights that were actually on-time.
- **Specificity (Delays): 70.5%** - The model successfully identifies 70.5% of all flights that were actually delayed.

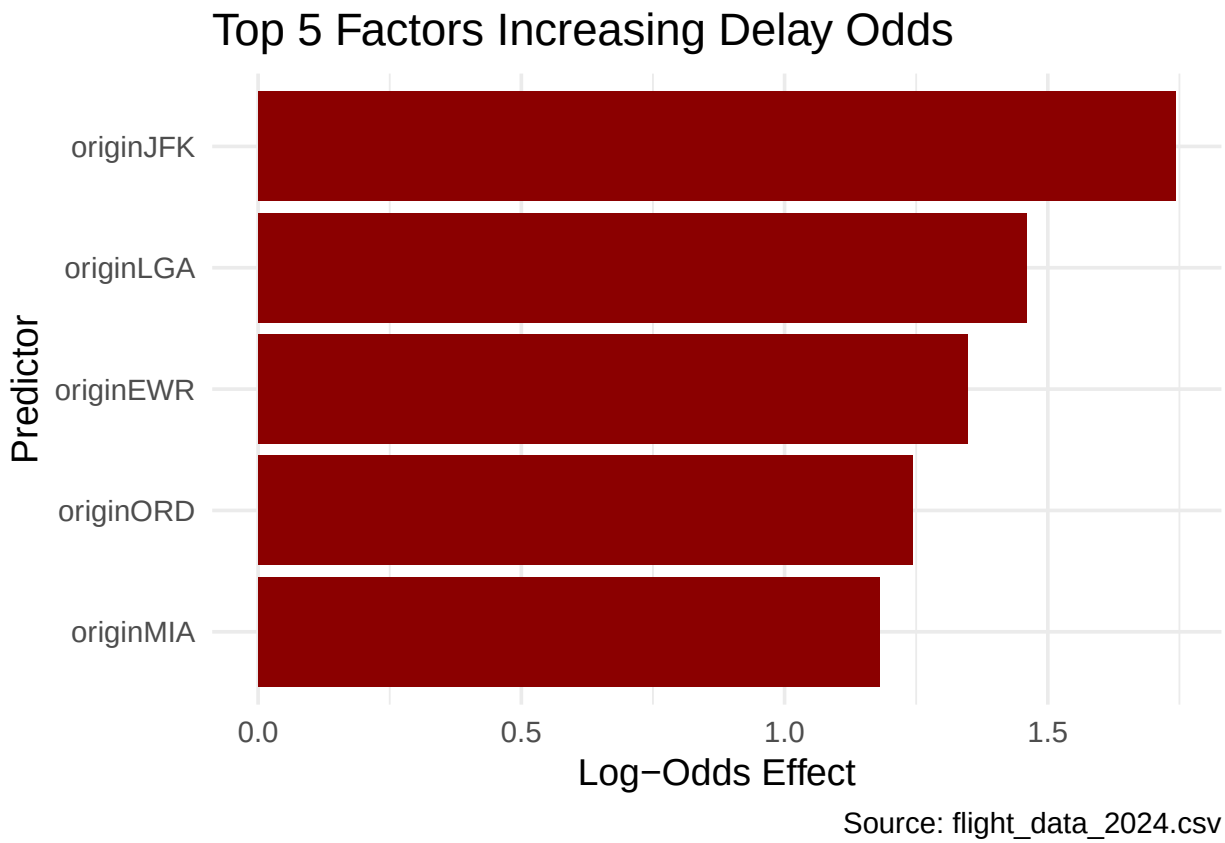
The key takeaway is that the model performs unevenly across classes: it's pretty good at catching delays, but it does so at the cost of raising many "false alarms" (predicting a delay for a flight that ends up being on-time). This tendency to over-predict delays reflects a bias that could be addressed by adjusting the probability cutoff threshold.

Interpreting the Model: What Factors Matter Most?

Finally, we examine the model's coefficients to see which factors have the biggest impact on delay odds.

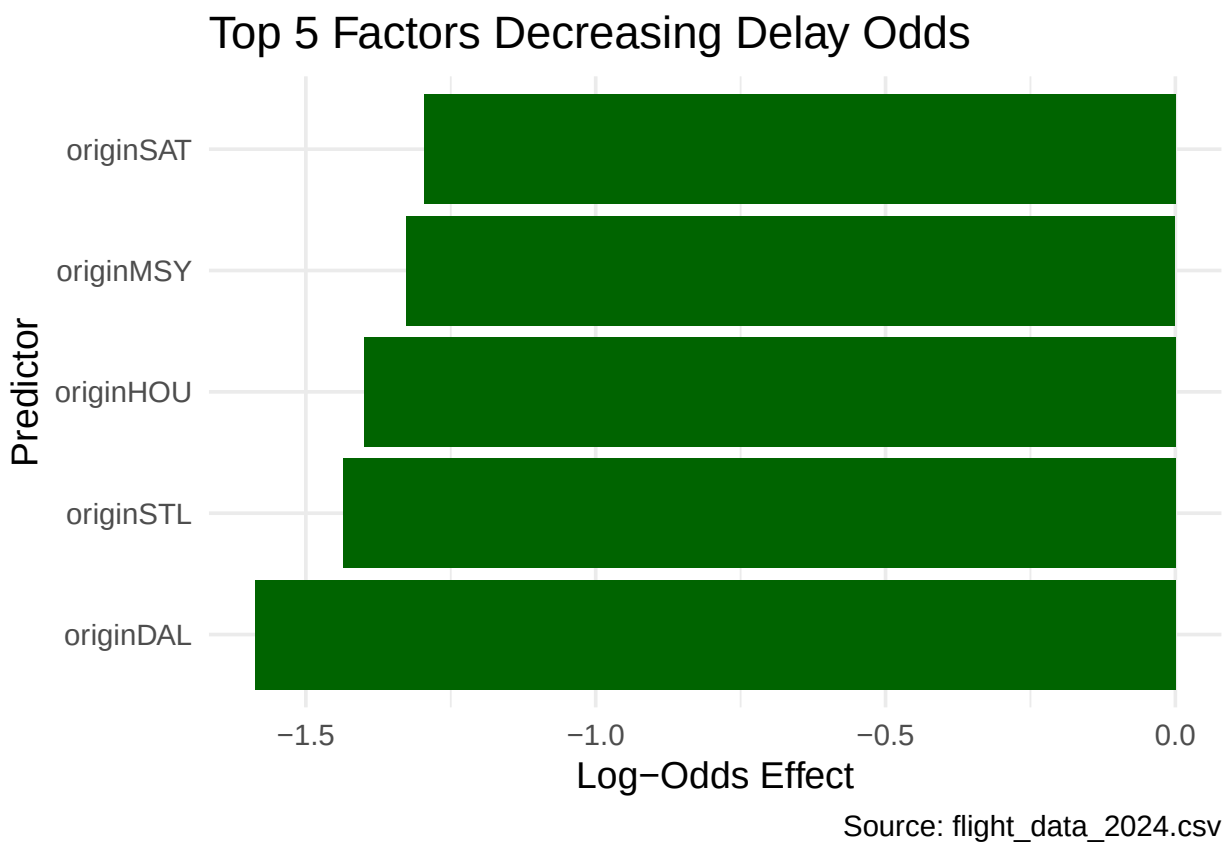
```
# Extract and tidy coefficients
model_summary <- tidy(logit_model) %>%
  filter(term != "(Intercept)") %>%
  mutate(
    odds_ratio = exp(estimate),
```

```
    variable_group = case_when(
      grepl("origin", term) ~ "Airport (Origin)",
      grepl("day", term) ~ "Day of Week",
      TRUE ~ "Numeric"
    )
  )

# Top 5 factors that INCREASE delay odds
top_pos <- model_summary %>%
  arrange(desc(estimate)) %>%
  slice(1:5)

ggplot(top_pos, aes(x = reorder(term, estimate), y = estimate)) +
  geom_col(fill = "darkred") +
  coord_flip() +
  labs(
    title = "Top 5 Factors Increasing Delay Odds",
    x = "Predictor",
    y = "Log-Odds Effect",
    caption = "Source: flight_data_2024.csv"
  ) +
  theme_minimal(base_size = 14)
```




```
# Top 5 factors that DECREASE delay odds
top_neg <- model_summary %>%
```

```
  arrange(estimate) %>%
  slice(1:5)

ggplot(top_neg, aes(x = reorder(term, estimate), y = estimate)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(
    title = "Top 5 Factors Decreasing Delay Odds",
    x = "Predictor",
    y = "Log-Odds Effect",
    caption = "Source: flight_data_2024.csv"
  ) +
  theme_minimal(base_size = 14)
```



Finding: The origin airport is by far the most influential predictor. Major hubs like JFK, EWR, and ORD significantly increase the odds of a delay, while smaller airports like DAL and HOU are associated with on-time performance.

6. Conclusion & Reflection

This analysis successfully built a logistic regression model that predicts flight delays with ~65% balanced accuracy, confirming that **where you fly from and on what day have a significant impact.** Large hubs and Fridays are key risk factors. Even with limited variables, the model provides a solid foundation for operational insights.

Limitations:

- **Time Scope & Seasonality:** The data only covers Jan-Feb 2024. Northern airports dominated the high-delay list, likely due to winter weather. These results may not generalize to summer, when different weather patterns and travel patterns happen.
- **Model Bias:** The model is better at identifying delayed flights than on-time ones, leading it to over-predict delays and create false positives. Although this helps ensure that almost all delays are flagged, it sacrificed some accuracy for on-time flights. Further adjustment might be needed in our model.
- **Definition of "Delay":** Our standard 15-minute threshold is a binary simplification. It treats a 16-minute delay and a 3-hour delay the same. The actual severity of the delay is lost and would be very different in a real-world context.

Next Steps & Implications

- **Incorporate More Features:** The model's accuracy could be significantly boosted by adding more relevant variables like **weather data, airline information, and the destination airport,** which I imagine would account for a lot more of the delay time.
- **Deeper Analysis:** Analyzing a full year of data could help us capture seasonal trends.
- **Predicting actual length of delay:** From the perspective of a passenger or an airline, a 24-hour delay or a 15-minute delay are the same, and I think that could cause a lot of issues in our data. Predicting the actual length of a delay and then using that information with the information we have here could also help us have more actionable insights.
- **Operational Use:** Even with its current limitations, the model provides a good amount of information to decision makers at airports. Airlines and airports can use these insights to focus mitigation efforts on the most vulnerable times and locations, such as adding buffer time to schedules at congested hubs on Fridays to reduce disruptions.