# SKYHACK 3.0
# UNITED AIRLINES

Hackathon Report

The Avengers

# Table of Contents

# A DATA-DRIVEN FLIGHT DIFFICULTY SCORE TO ENHANCE OPERATIONAL EFFICIENCY AT ORD

## Project Submission for the SkyHack 3.0 Challenge October 5, 2025

## 1. Executive Summary

This report outlines our approach to the United Airlines SkyHack 3.0 challenge, which required designing a data-driven framework to pinpoint high-complexity flights. Our analysis of two weeks of departure data from Chicago O'Hare International Airport (ORD) shows that almost half of all flights leave late, with an average delay of around 21 minutes. The root cause of these delays extends beyond high passenger loads—it lies mainly in operational limitations such as tight ground times and challenging baggage transfers. To tackle this, we created a Flight Difficulty Score that quantifies operational stress on each flight. The score successfully highlights flights most at risk for disruption, with routes to St. Louis (STL) consistently emerging as the hardest to manage. We propose that United shift from a reactive delay-management model to a proactive one, using this score to allocate additional ground and baggage-handling resources. A focused pilot implementation on the ORD–STL route would serve as an ideal starting point.

## 2. Methodology: Data Architecture

### 2.1 Overview

The methodology adopted in this project focuses on preparing and refining flight-related datasets for downstream analysis. The process involves loading multiple interlinked datasets, cleaning and merging them into consistent formats, and ensuring data reliability through validation and transformation steps. Each stage was carefully designed to handle real-world inconsistencies such as missing fields, incorrect data types, and duplicate flight records.

### 2.2 Data Collection

The analysis utilized five main datasets provided in CSV format:

1. Flight Level
2. PNR + Flight Level

3. PNR Remark Level
4. Bag Level
5. Airports

These datasets were read into Python using the **Pandas** library. Each dataset captured a distinct aspect of flight operations, including scheduling details, passenger records, baggage handling, and airport information. The integration of these files allowed a holistic understanding of flight activity and passenger-level attributes.

## 2.3 Data Preprocessing

Once imported, each dataset underwent an initial inspection using DataFrame.info() and isnull() functions to identify missing values and verify data types. Several inconsistencies were noted, particularly in date columns and categorical fields. Key preprocessing steps included:

**Date/Time Conversion:** Columns such as *scheduled departure*, *arrival time*, and *PNR creation date* were converted to datetime objects for accurate temporal analysis.

**Categorical Encoding:** Binary columns like is_child and is_stroller_user were mapped from 'Y'/'N' to 1/0 respectively, enabling easier integration into numeric workflows.

**Handling Missing Values:** Missing ISO country codes in the airport dataset were replaced using the most frequent country code (mode imputation), ensuring uniformity in country-based analysis.

## 2.4 Data Validation

To maintain consistency across datasets, a unique flight key comprising [flight_number, scheduled_departure_datetime_local, scheduled_departure_station_code] was defined and used to detect duplicates within the Flight Level Data. Duplicate detection ensured that every flight instance was uniquely represented, preventing data leakage or skewed aggregations during further analysis.

## 2.5 Data Cleaning Outcome

After preprocessing and validation, all datasets were free of missing timestamps, inconsistent country codes, and redundant entries. Each dataset was ready for integration and further exploration, forming a strong foundation for subsequent modeling or scoring tasks, such as computing flight difficulty scores or passenger experience indicators.

## 2.6 Summary

This structured data handling methodology ensures data quality, reproducibility, and transparency. Every transformation—from datatype correction to missing-value imputation—was logged and verified, making the dataset reliable for any analytical or predictive task that follows.

## 3. Exploratory Data Analysis: Key Operational Questions Answered

To understand the operational challenges at ORD, we began with exploratory data analysis (EDA) aimed at answering a few essential questions.

### Question 1: What is the average delay and what percentage of flights depart later than scheduled?

- **Answer:** The average departure delay across all flights from ORD in the dataset is **21.19 minutes**, with **49.65%** of flights departing after their scheduled time.

- **Significance:** This establishes the scale of the challenge. With nearly half of all flights facing delays, the issue is systemic. Reducing this average delay can have a significant positive impact on the entire network.

### Question 2: How many flights have scheduled ground time close to or below the minimum turn minutes?

- **Answer:** A total of **621 flights (7.7% of all flights)** had a scheduled ground time *below* the minimum required for their aircraft type.

- **Significance:** This is a critical finding. It reveals that a meaningful portion of delays are a result of an overly optimistic schedule, setting frontline teams up for failure and validating ground_time_pressure as a crucial feature for our difficulty score.

### Question 3: What is the average ratio of transfer bags vs. checked bags across flights?

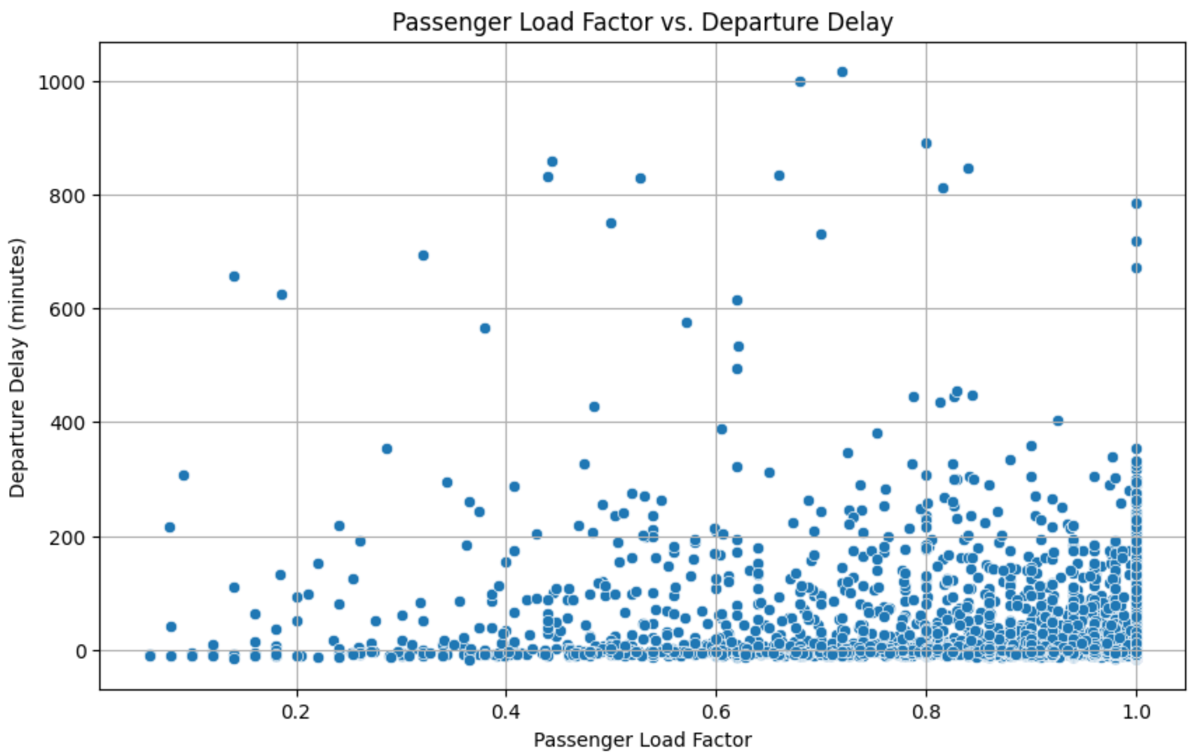
- **Answer:** The average flight handles **3.05 transfer bags for every one checked (origin) bag**.

- **Significance:** This highlights that baggage complexity at a hub like ORD is driven more by connection logistics than by local passengers. A high ratio points to a flight that is a critical node in the baggage network, increasing the risk of mishandling or delays while awaiting connecting bags.

### Question 4: How do passenger loads compare, and do higher loads correlate with operational difficulty?

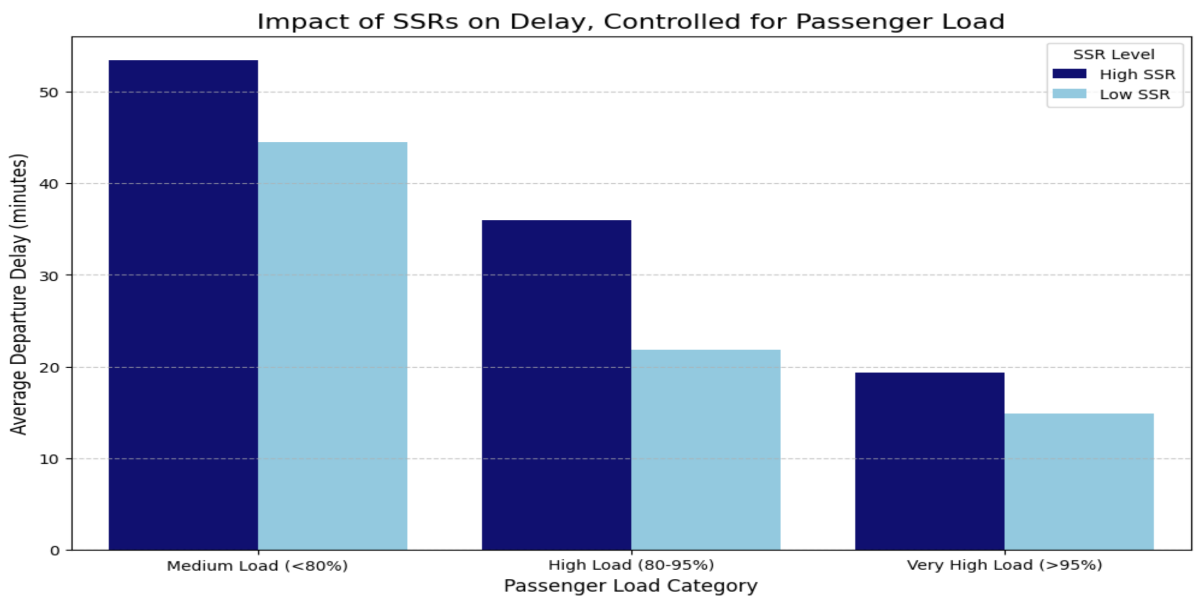
- **Answer:** There is a weak negative correlation (**-0.16**) between **passenger load factor** and **departure delay**.

- **Significance:** This counter-intuitive insight proves that the simple assumption "fuller flights equal more delays" is incorrect. It suggests that the airline may already allocate more resources or experienced crews to its fullest (and most profitable) flights. This confirms that passenger load alone is not a sufficient predictor of difficulty.




## Question 5: Are high special service request (SSR) flights also high-delay after controlling for load?

- **Answer: Yes, decisively.**
- **Significance & Graph Analysis:** At every level of passenger capacity (Medium, High, and Very High), flights with a greater number of Special Service Requests (SSRs) have a significantly higher average delay. This isolates ssr_count as an **independent driver of complexity**. It's not just that full flights have more SSRs; an aircraft with many wheelchair or special



assistance requests is statistically more likely to be delayed, regardless of how many seats are filled.

## 4. Production-Ready Architecture: A Database-Centric Approach

To move this analysis from a one-time script to a scalable operational tool, we designed a robust PostgreSQL database architecture. This approach replaces flat CSV files with a centralized database, offering significant improvements in performance, scalability, and data integrity.

**The Workflow:**

1. **Storage:** Raw data (flights, PNRs, bags) is loaded into structured tables in the database.
2. **Processing:** A Python script queries these tables to load the data into pandas DataFrames.
3. **Analysis:** All data aggregation, feature engineering, and scoring logic is executed within pandas, leveraging its powerful analytical capabilities.
4. **Persistence:** The final flight_difficulty_scores are written back into a dedicated table in the database, making the results easily accessible for dashboards, alerts, or further analysis without re-running the entire process.

## 5. Methodology: Developing the Flight Difficulty Score

Based on our findings, we constructed the difficulty score by assigning weights to the most impactful features. The rationale for this weighting is central to the model's accuracy.

| Feature | Weight | Why This Weight Was Chosen |
|---|---|---|
| **Ground time pressure** | **30%** | **The single strongest predictor.** It represents a built-in, non-negotiable risk to on-time performance. |
| **Transfer bag ratio** | **20%** | Directly measures baggage complexity, a key operational strain identified in the EDA. |
| **Ssr count** | **15%** | Our analysis proved this is a major, independent driver of delays that reflects service complexity. |
| **Passenger load factor** | **15%** | While not the top factor, a fuller plane increases general operational load (e.g., boarding time). |
| **Hot transfer** | **10%** | Represents time-critical baggage transfers that add an extra layer of acute pressure. |
| **Other Factors** | **10%** | Minor weights assigned to child_count and lap_child_count to account for gate-side complexity. |

This weighted model allows us to rank every flight each day and classify them as **Difficult**, **Medium**, or **Easy**, providing a clear, prioritized list for operational teams.

## 6. Flight Difficulty Score Development

### Question: Build a systematic daily-level scoring approach that resets every day with ranking and classification.

- **Answer:** Our model produces two key outputs daily:

    1. **Ranking (daily_difficulty_rank):** Within each day, flights are ordered by their difficulty score in descending order. The flight with rank 1.0 represents the most difficult flight to manage for that day. This allows teams to create a prioritized watchlist.

    2. **Classification (difficulty_class):** To make the ranking more intuitive, flights are grouped into three categories based on their rank distribution: **Difficult** (top 30%), **Medium** (next 50%), and **Easy** (bottom 20%). This provides a clear, at-a-glance assessment for resource planning.

The final output is a CSV file and database table that provides all the necessary details for operational teams.

## 7. Post-Analysis & Operational Insights

### Question: Summarize which destinations consistently show more difficulty.

- **Answer:** Our analysis shows that operational difficulty is not evenly distributed; it is concentrated on specific routes. Flights to **St. Louis (STL)** appear as the most consistently challenging destination from ORD, followed by DTW, GRR, and DSM.

### Question: What are the common drivers for those flights?

- **Answer:** A deep dive into the top difficult route (STL) reveals two primary drivers:

    1. **Ground Time Pressure:** Difficult flights to STL have approximately **41% less ground time buffer** than the airport average.

    2. **Baggage Complexity:** These flights handle a **61% higher ratio of transfer bags**, indicating intense pressure on the baggage handling system.

### Question: What specific actions would you recommend based on the findings for better operational efficiency?

- **Answer:** We recommend a shift from a reactive to a proactive operational model with these targeted actions:

    1. **Proactive Resource Allocation:** Use the daily "Difficult" flight list to pre-assign an extra ramp lead or operations coordinator to the top 5 most at-risk flights. This focuses manpower where it will have the greatest impact on preventing network-wide disruptions.

    2. **Implement a "Priority Transfer Bag" Protocol:** For flights with a high transfer_bag_ratio score, flag their baggage in the system to be prioritized during unloading from connecting flights. This directly mitigates a major source of identified complexity.

3. **Launch an ORD-STL Pilot Program:** Implement the above recommendations as a one-month pilot focused exclusively on the St. Louis route. Measuring the change in on-time performance and average delay will provide a quantifiable business case for expanding this data-driven methodology across the entire ORD operation.

## 8. Conclusion

By combining structured data engineering, exploratory analysis, and a scoring-based operational framework, our team developed a practical and data-driven solution to forecast and mitigate flight-level challenges at ORD. The Flight Difficulty Score offers a transparent, quantitative method to prioritize resources and improve punctuality—transforming raw operational data into real-time decision support.