Cam Osterholt

Dr. Biplav Srivastava

CSCE 581

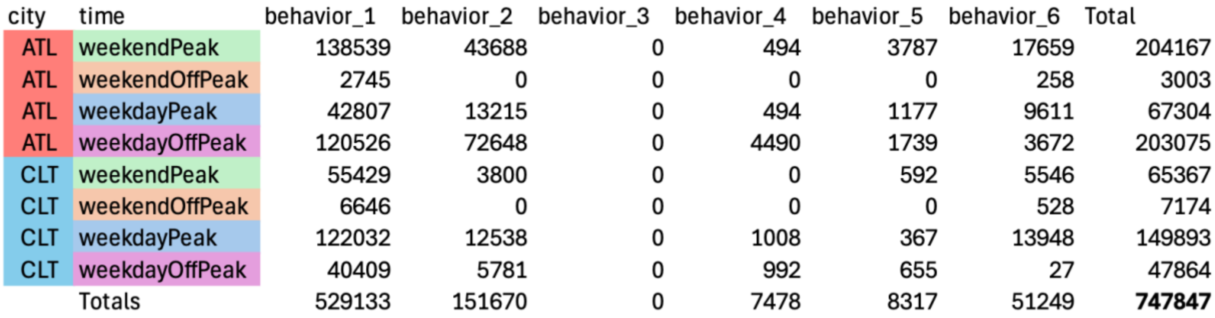
29 April 2025

Predictive Classification of Aircraft Approach Behaviors - Final Report

Key idea: Use flight summary statistics to predict the outcome of the flight based on the current location. The goal of this project was to enhance the tools available that can be used to understand the flight path using existing data from the FAA logs.

Who will care when done: Air Traffic Controllers, FAA, Flying Public

Data needed: Longitude and Latitude coordinates of tracking points for flights.



city	time	behavior_1	behavior_2	behavior_3	behavior_4	behavior_5	behavior_6	Total
ATL	weekendPeak	138539	43688	0	494	3787	17659	204167
ATL	weekendOffPeak	2745	0	0	0	0	258	3003
ATL	weekdayPeak	42807	13215	0	494	1177	9611	67304
ATL	weekdayOffPeak	120526	72648	0	4490	1739	3672	203075
CLT	weekendPeak	55429	3800	0	0	592	5546	65367
CLT	weekendOffPeak	6646	0	0	0	0	528	7174
CLT	weekdayPeak	122032	12538	0	1008	367	13948	149893
CLT	weekdayOffPeak	40409	5781	0	992	655	27	47864
	Totals	529133	151670	0	7478	8317	51249	**747847**

Methods: Classification using data summary (6 behaviors)

Evaluation: F1-Score & AUC ROC Curve

Users: Airport control systems, Pilots, Autonomous researchers, etc

Trust issue: Accuracy is one of the key metrics of many models, but in this case, it is of extreme importance in the aerospace industry. This points to the main trust issue of the predictions is safety. Any incorrect locations or predictions can result in aircraft crossing paths, which could be deadly. Those inaccuracies are especially common in older aircraft or those not as well maintained. They can also happen more frequently at smaller regional airports. The common factor in these reasons can be boiled down to economic status. Those who cannot afford the nicest flights or live in richer areas can be subject to less accurate data in their predictions. Moving forward, if this were to be implemented at scale, there would need to be a key focus on increasing the variety in airports picked and other ways to ensure data across all aircraft types, big or small.

Demonstration: To demonstrate its effectiveness, a flight was selected from Hartsfield-Jackson Atlanta International Airport (ATL) in Atlanta, Georgia, to track. ATL is one of the busiest airports in the world and serves as a hub for many Delta flights. This flight was taken during my weekday off-peak flight recording. I classified the data into six different categories:

1. No behavior (standing still)
2. Standard landing
3. Going around
4. Switching sides
5. Flyby
6. Unknown

The flight was classified as switching sides, meaning it comes from the opposite direction in which it lands. This can be tricky to detect because a normal flight would normally slow down on its approach to the airport, but switching sides keeps going at a level airspeed. I was able to see evidence of this in early versions of my model, which predicted classification 5 (flyby) until the very end. In this example, I use a Multilayer Perceptron classification model, as I will show later, this provides the best metrics out of the models I used. In this[1] video demo, we see the plane coming into the airspace from the east, flying past just south, then turning around to land east to west. The video demonstrates the flight at a rate of 1:20 (i.e., 1 second of video is 20 seconds of flight data). From 0:00-0:09 of the clip, the model is classifying the path as 6, meaning it has yet to give a prediction. At 0:10, we see a small window where it classifies it as a standard landing. The reason for this misclassification remains unclear; however, the benefit of this model is that it corrects itself very next second. From 0:11-0:37, it remains constant with its prediction switching sides until the plane completes its landing. This means it was able to determine the result 520 seconds, or almost **9 minutes**, before.

[1] https://youtu.be/hoQqRULs9rQ

Experimental results: After demonstrating the model's application on a single flight, I next evaluated its overall predictive performance. In my efficacy analysis, I compared my results to those generated by Gemini's latest 2.0 Flash. To not influence the baseline results, I provided it with basic questions. My first query was to ask, "I want you to take the data, split it, and give me the f1_score and auc-roc score to predict the behavior of this data," after providing my full data files. It trained models for each file, which I found to be too inaccurate due to their limited data size, so I reprompted to ask it to combine the data and then recalculate. Here is the response provided:

```
Metrics for Combined Dataset:
F1 Score: 0.4641
AUC-ROC Score: 0.4450
```

We can see this as above a fully random sample over the 6 classifications, but not better than a weighted average based on their occurrences. I found limitations in the model training, the model itself, so I then prompted the model to generate a refined script for evaluation, I could run that would provide a more accurate result. The code it generated is found in gemini.ipynb in my GitHub. Here are the metrics it provided.

```
Refined Metrics for Combined Dataset:
F1 Score: 0.4919
AUC-ROC Score: 0.7162
```

The AUC-ROC curve is much improved, but didn't do much in terms of F1 Score. My results are much improved over the LLM model. I started my project using a similar method to the code the LLM generated by analyzing each point individually. This kept giving me results way below the levels I expected. The results also had a large variance throughout the flight. To compensate for this, I instead relied on calculated metrics for the flight I derived from the dataset. This gave me greatly increased results. The metrics for my classifications are found here.

Related work: This is not the first time someone has attempted a project similar to mine. However, from my research, many attempts have been deprecated. The goals of many of these projects are based on trajectory planning rather than prediction. The difference is in the scope - money vs safety. Their question comes down to, how can we adapt our landing plans based on the variance in flight plans based on weather and other factors determining their arrival. Planning trajectories can result in increased bandwidth for an airport, My project attempts to prevent incidents that can arise while trying to land planes at an ever-increasing rate. NASA did put out a guideline in 2011 for projects like mine titled NASA's Survey and Method for Determination of Trajectory Predictor Requirements, yet it doesn't seem like many have taken action.

Discussion: Throughout this project, I learned a lot about the difficulty in data collection. The number of times I rewrote my training script to better group and organize the data was way more than I expected. I also appreciated the guidance to "Crawl, Walk, Run". It kept me within my scope and not attempt to go way overboard with the many improvement ideas. Speaking of, I had many ideas I kept note of throughout my research and training phases that could improve my models going forward. I can always continue to add more statistics in my flight summary, particularly more metrics that show the change in values rather than the total difference. While flying home over Easter weekend, I also noticed a possible next step for the project. While I was landing in Charlotte, there was a plane about a mile to our left landing on another runway at the same time. These parallel landings have become even more common in large airports in recent years, I would love to see the impact that other flight paths in the area have on the prediction model. Finally, my last plan going forward would be to increase the data sample. A vast majority of my data is in two classifications (1&2). Even though I weighted my model, the lack of representative samples in classifications 4, 5, and 3 prevents me from fully achieving the accuracy goals I wanted. The multiple airports were a big help, but 8 hours of data just wasn't enough. Though the real bottleneck was not the collection time but the time it took me to classify all hundreds of landings and flybys. The results of this project have not only been more accurate than I expected but also predicted some samples way earlier than I thought possible with that accuracy. Knowing what a plane was going to do nearly 9 minutes before landing feels beyond the intuition of the average person. The project highlighted the challenges of data processing and collection, the value of proper metric selection, and how incremental development could improve my development going forward. With further expansion and refinement, I do not doubt that this process could be scaled for real-world use.

Conclusion: According to the Federal Aviation Administration (FAA), the United States sees nearly 50,000 flights per day and 2.9 million passengers across 29 million square miles of airspace. Managing that volume of traffic leaves little room for error in the most dangerous part of the flight, landing. Emergencies can happen, and as the airline industry switches over to a new Voice over Internet Protocol (VoIP), there is an increased chance that an aircraft may lose communication. During that downtime, flight controllers must have an idea of what the aircraft is doing. Currently, their main way of doing this is using radar detection to see the current heading and location. The work has shown that prediction is possible and effective. I hope this project and the skills I learned can help increase aerospace safety for people across the globe.