Sandia National Laboratories

FLIGHT PREDICTION

Sean Lee, Atulya Kadur
Anthony Huang, Diana He, Christopher Lee, Samuel Duprey, Mukund Venkatesh,
Hari Ram Narayanan, Nicholas Chong, Siddharth Rao, Ben Eng, Ritwik Jayaraman,
Rahul Prabhu, Vinay Pattanashettar, Jonathan Walker, Adanma Adebayo-Kay

Introduction

Background:
When ground control loses contact with a flight, quickly determining its destination is crucial for public safety. Our model predicts the flight's destination using historical flight data, departure location, and partial flight trajectories.

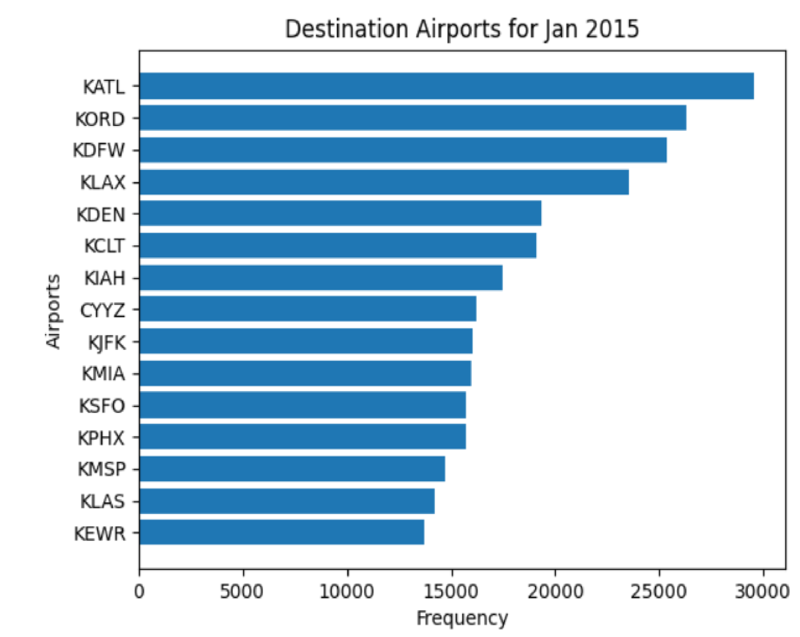
Motivation:
Predicting a flight's destination is an important challenge in motion analysis.

Tools & Resources Used:
Tracktable (Sandia Python library), Anvil (large Data Processing, Python, GitHub)

Problem

If a flight disappears, we want to know its destination. In this example, the origin airport (LAX) is known. When communication is lost, the model should predict the destination (JFK).

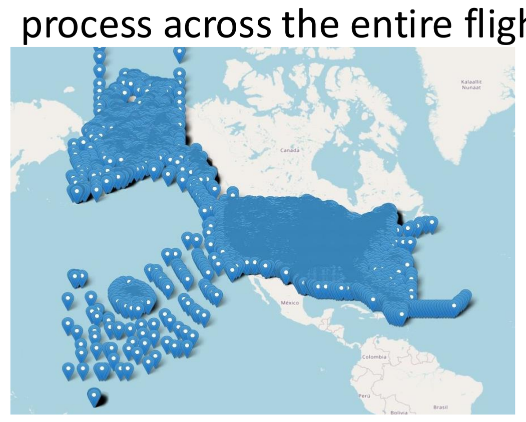
Data Exploration



- Flights are not evenly distributed with most flights arriving or departing from a major airport. This results in a "class imbalance problem" where flight prediction is skewed towards big airports.
- Teams must balance the high traffic in popular airports. One way to do it is to scale the traffic logarithmically

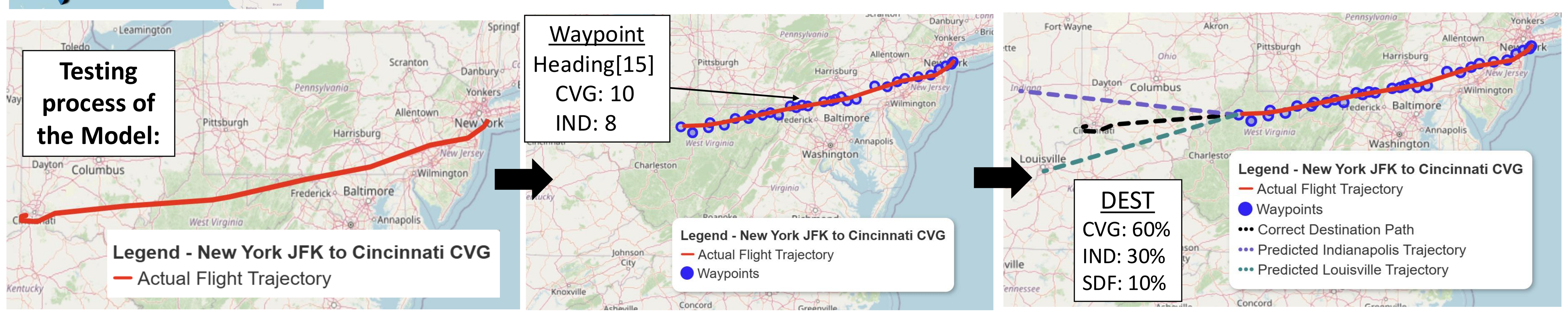
Waypoints Process

Description: The waypoints model uses historic data to predict the most likely destination of a flight segment. Each point along the flight path is matched to the nearest waypoint where the plane's heading direction is used to look up similar historic flight destination. By repeating this process across the entire flight path segment, the model creates a list of likely destination and the confidence levels of each prediction.



Waypoints: Used a publicly available list of over 60,000 waypoints from the FAA. Each waypoint is a set point across the US with latitude, longitude, and a list of destination: frequency maps for each 360 degrees of heading. An R-Tree is constructed with these waypoints for quick and efficient lookup of the nearest waypoint.

Training the Model: For all historic flights, we go through each flight point, find the nearest waypoint, index into the correct heading bucket, and update the frequency of the destination for that point.

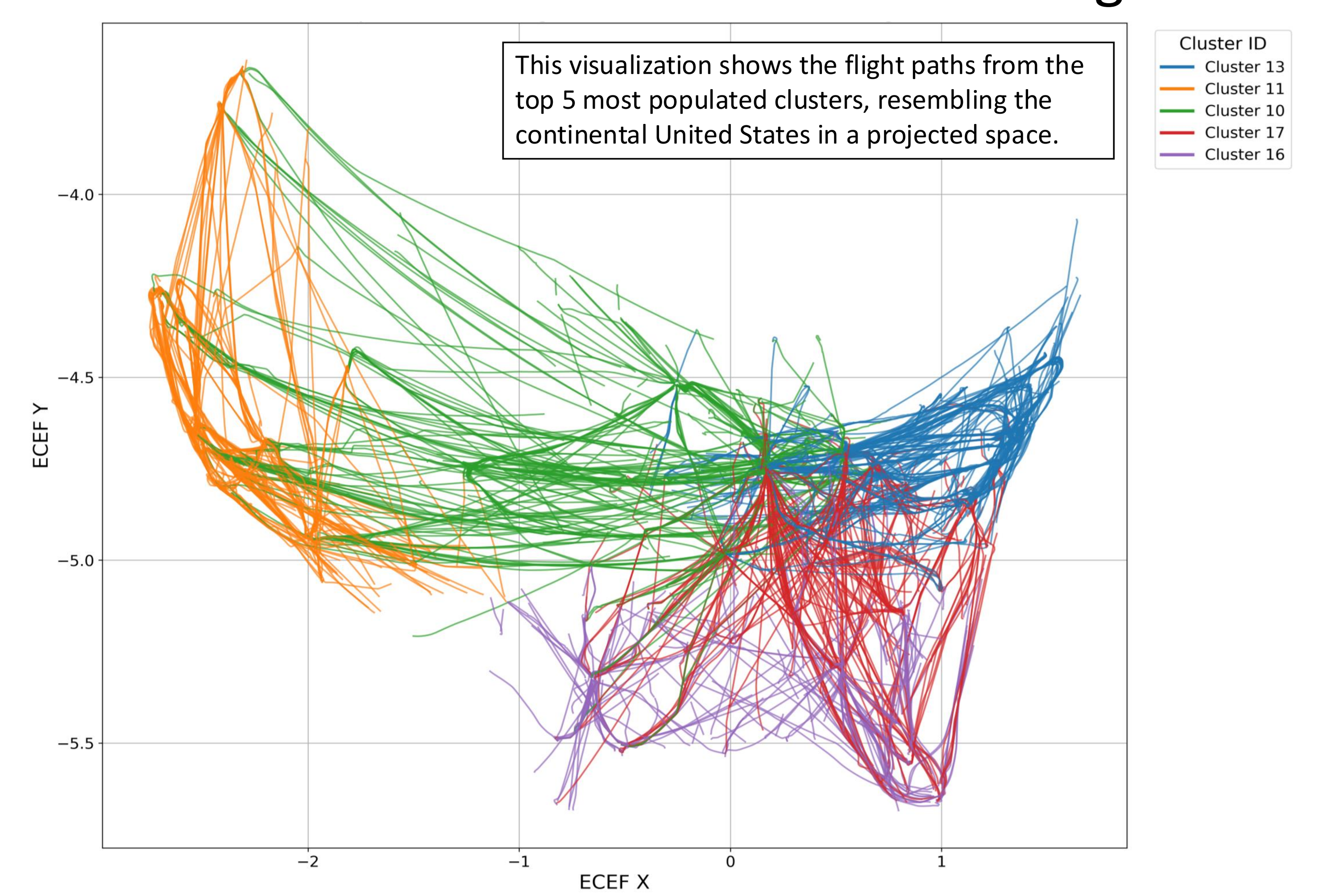


This is a full flight, testing input is any fragment of this, (0 – 60% for this example)

Heading is used to search the destination map in the bucket of the closest waypoint.

Values used to calculate a sorted list of (Dest, %) for a given number of possible destination.

Clustering Process



K-MEANS: We first assign each flight in our dataset to one of 20 clusters based on their overall displacement, heading, and start position.

⬇

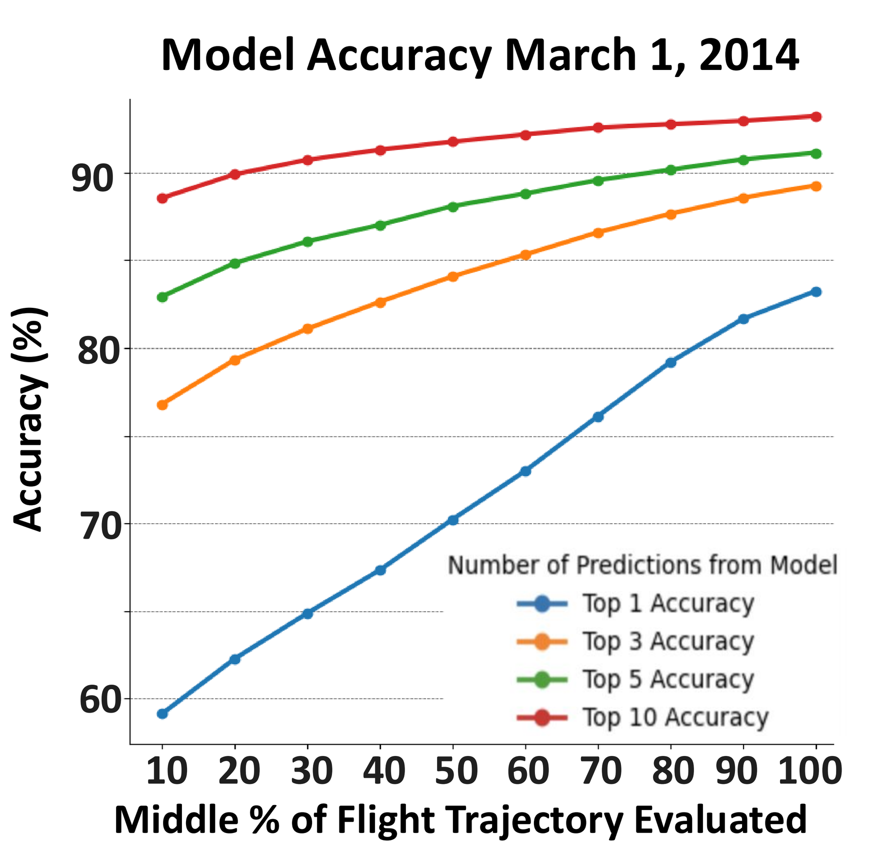
LSTM: Each cluster has an associated bi-directional LSTM trained on the cluster's flights. This LSTM predicts the continued trajectory of a given flight fragment, ending with the final destination as an (x, y) coordinate.

⬇

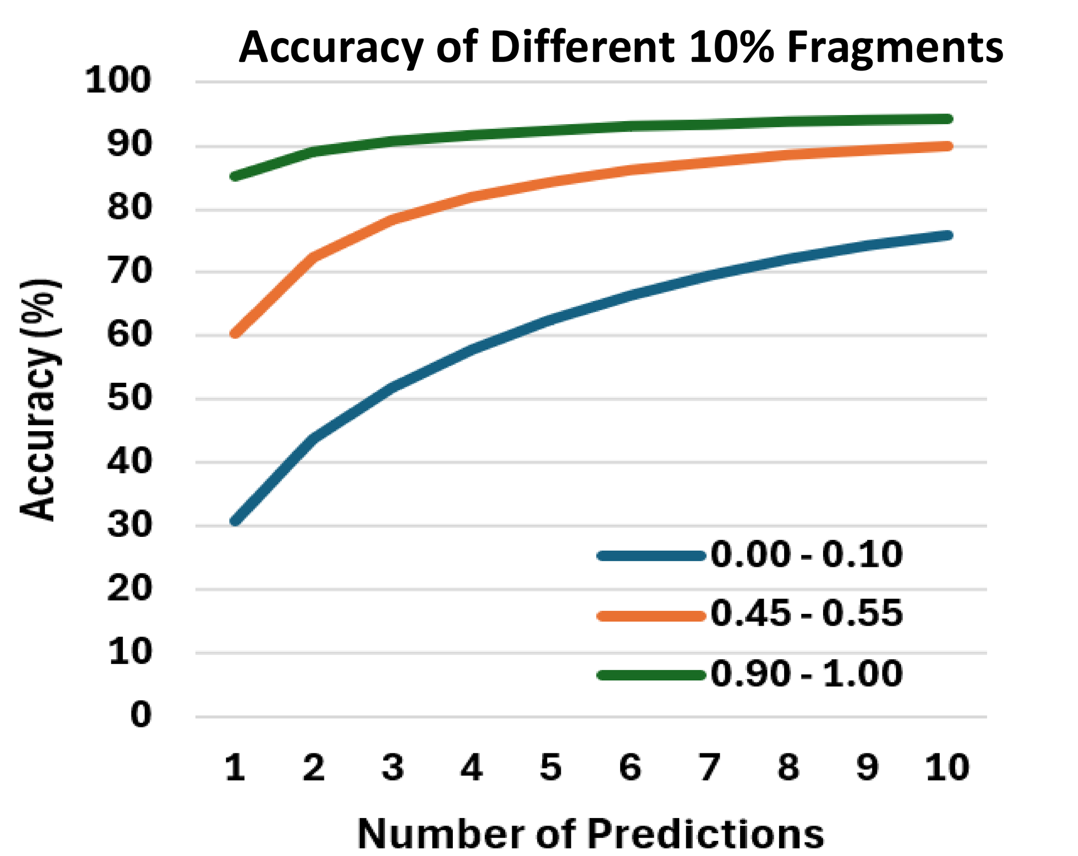
KNN: We use a K-Nearest Neighbors model to find a set of most likely destination airports to the final predicted (x, y) coordinate.

Description: Currently, we have completed data preprocessing (e.g., removing international flights), created our KMeans model, designed and trained our LSTM, and built a KNN-based representation of the various airports. We are now in the final stage of our pipeline: calculating the number of predictions required to correctly identify the airport at 60%, 80%, and 100% of the flight completion.

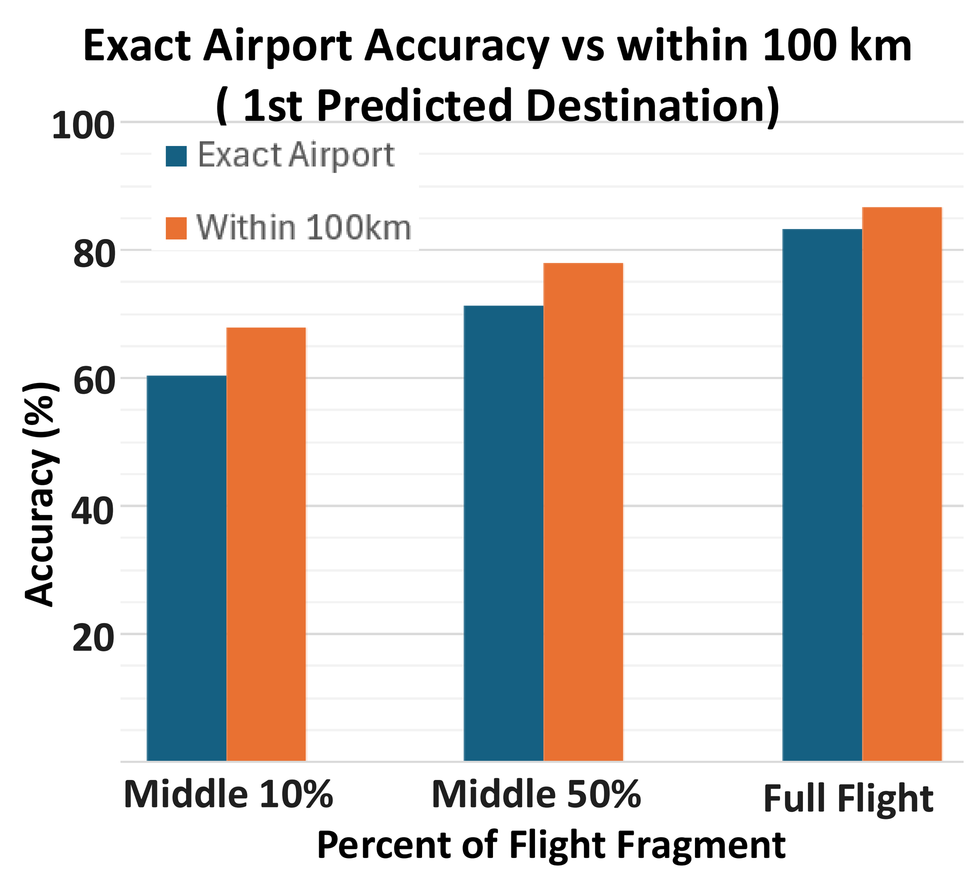
Waypoints Results



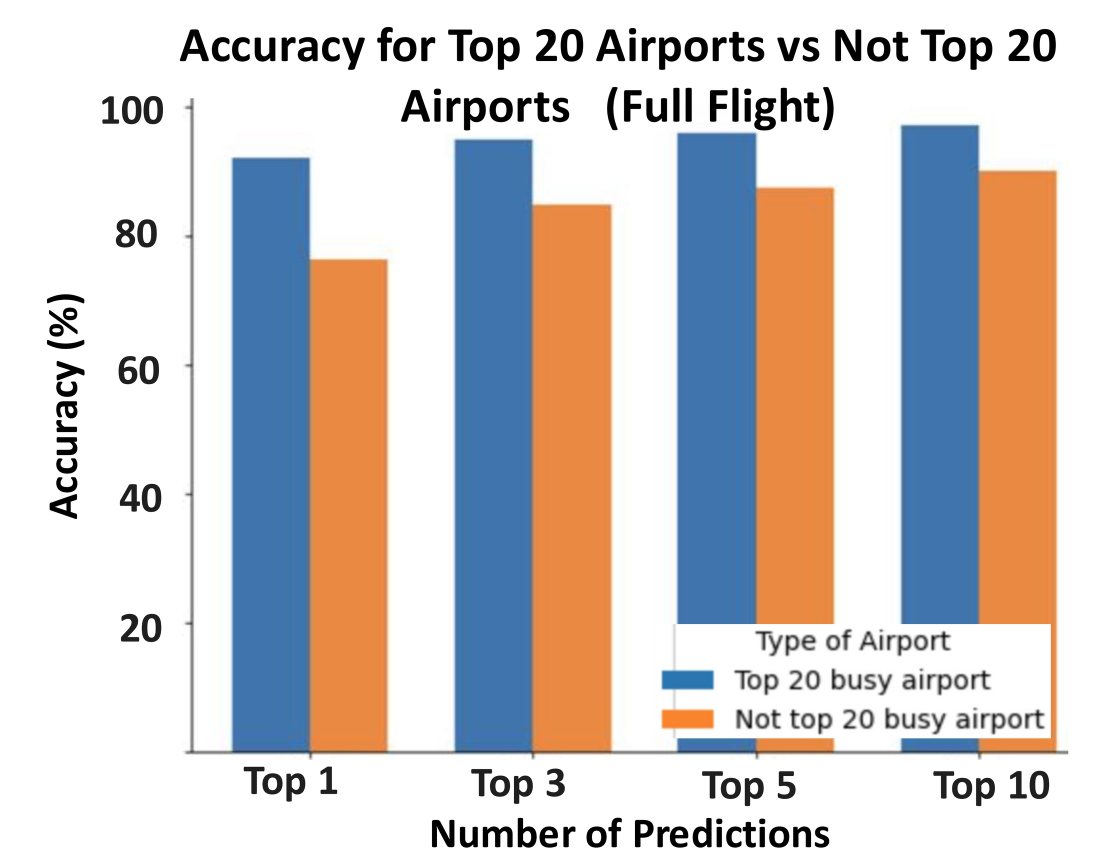
Each percent of the flight is taken from the middle of the overall trajectory. Fragments closer to the end have better results, even if they are shorter fragments sometimes.



Where the fragment is taken from the flight is very impactful for accuracy. The later a fragment is in the overall flight's trajectory, the noticeably more accurate the model is.



Our model predicts an airport as a flight's destination, For our first guess, our accuracy increases over 5% for most categories when also considering neighboring airports within 100km as correct.



Using the full flight, accuracy of flights from the top 20 busiest airports vs. all other airports varies slightly . Attempts to address the class imbalance by scaling traffic logarithmically has mitigated the differences between these categories.

Future Work:

Waypoints:
- Dynamically weight destinations at each waypoint based on a point's change in speed
- Create speed bucket layer in our model so each heading value will map to different speed ranges.

Clustering:
- Create final pipeline by connecting every element through python scripts.
- Evaluate coordinate to airport KNN Model and per cluster accuracy with tests and visualizations of different fragment sizes and number of predictions

References:

Tracktable and Python Documentation

Acknowledgments

Special Thanks to Dr. Andy Wilson, Ashley Arroyo, Pete Dragnev, Dr. Amelia Henriksen, and the Data Mine staff

PURDUE UNIVERSITY | The Data Mine | Sandia National Laboratories