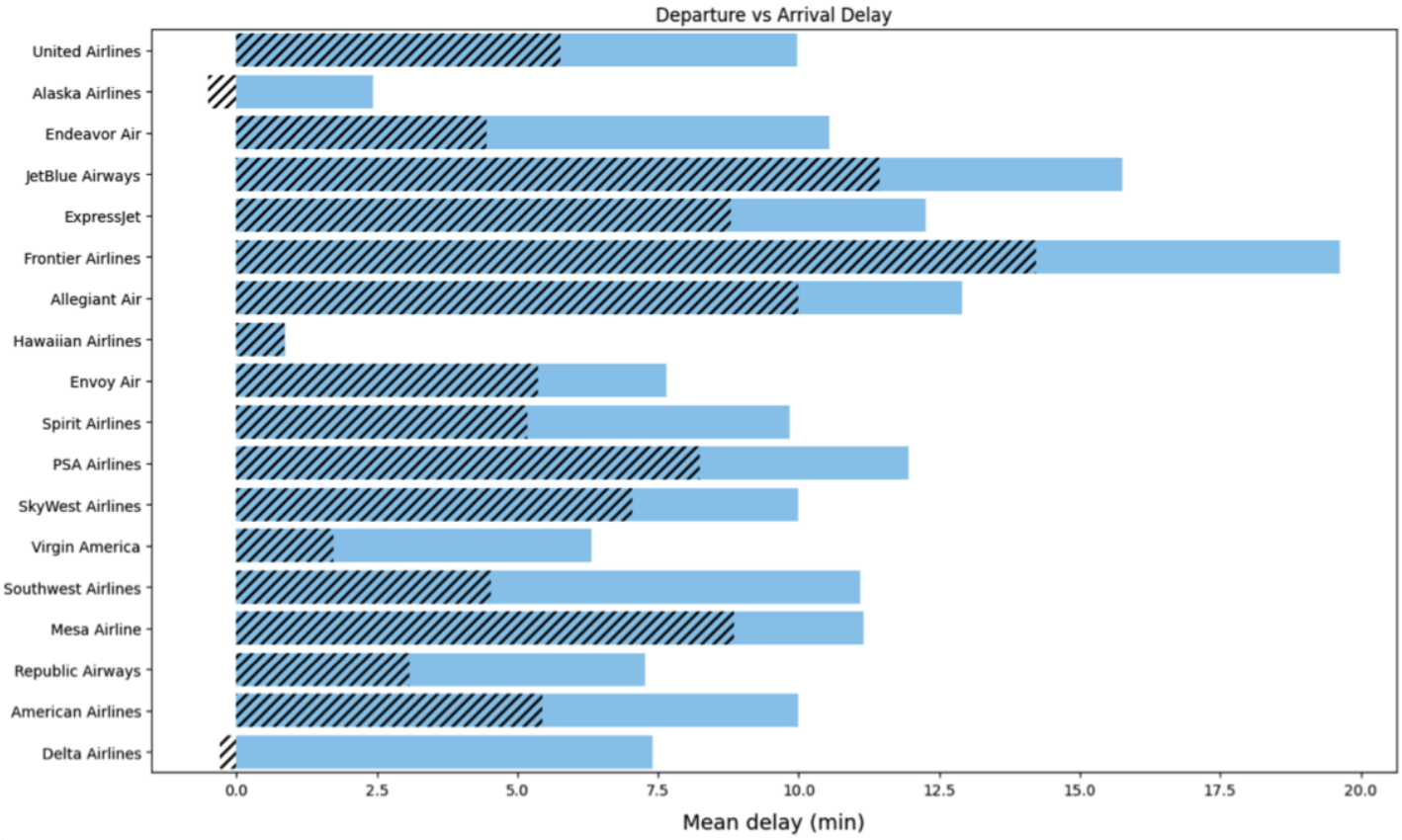
Michael Srouji
Max Sestero
Kishan Baliga
CSC 466 Project Report

Finding Airline Tardiness Using K-Nearest Neighbor

One of the most frustrating things in life is being all prepped and ready to ride on an airplane, whether for a business trip or a vacation, and the flight ends up being delayed by hours. Members of our group have personally been caught in this situation before, and we wanted to find a solution to this pesky problem. This issue is problematic and interesting, because it effects everyone, and is not a niche scenario. Furthermore, it's also a good problem because there is so much data available, due to how prevalent of an issue it is. Therefore, our group decided we wanted to create a solution for this issue, by creating a prediction system that would return whether or not the flight would be delayed, and by how much.

The first thing our group did was look for previous research done on this topic. We found a very interesting article written by Javier Herbas covering his approach on "Using Machine Learning to Predict Flight Delays".[1] Herbas used a very extensive amount of information to make this prediction, such as date, carrier number, flight number, origin, and destination in order to create an accurate model. What Herbas found was that flights with a departure delay would generally go faster during the trip, and have a smaller arrival delay. Furthermore, some airlines exhibited a much higher tendency to have delays than other airlines, such as Frontier Airlines which had the longest average departure delays. This article was a very interesting source

[1] https://medium.com/analytics-vidhya/using-machine-learning-to-predict-flight-delays-e8a50b0bb64c



Note: Figure is adapted from https://medium.com/analytics-vidhya/using-machine-learning-to-predict-flight-delays-e8a50b0bb64c

for our group, and we decided to use Herbas' research as the starting point for our own project.

We decided to search for a different dataset than Herbas, one that had even more categories of data. Our search for a dataset took us to the American Statistical Association, which has collected data on various topics for decades.[2] One of these topics was exactly what we were looking for, that being "Airline on time data." The American Statistical Association linked us to the Harvard Dataverse, with a very extensive amount of data on airline tardiness, ranging back all the way to 1987 all the way until 2008.[3] This data

[2] https://www.amstat.org

[3] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HG7NV7

was collected by this group by collecting flight records from each airport and compiling it. This was a great find, as it included any kind of data we might want to use, ranging from date and flight number, to weather conditions at the time. Now that we had found our data, we began to discuss what the best method to process all this data was.

We discussed among ourselves what the best method to process the data was. At first, our group considered decision trees, because there were many different factors as to why a flight may be delayed. By creating this decision tree we could easily see what factors are the most decisive as to whether a flight will be delayed or not, and we could trace the prediction given to us down the decision tree manually in order to validate the decision made. We quickly ran into an issue with this idea though: making decision trees is expensive, especially with such a large amount of data, and especially when you need to prune the tree. Making a prediction regarding how delayed a flight will be is not useful, if the prediction itself takes such a long time that you end up being late anyway.

With efficiency in mind, the next thing our group discussed was k-means. One of the advantageous things with k-means was that we could cluster a very large amount of data is a very short amount of time, by identifying the centroids of our data set and assigning each point to a particular cluster. This way, when our algorithm is given a prediction, it can assign that prediction to one of our existing clusters, and classify the delay based on the overall cluster. Although it would not be as easy to verify a decision as with decision trees, we could still trace the prediction to its cluster when manually verifying, which is not much more difficult. But, we quickly ran into an issue with k-means: we wanted to calculate a numerical flight delay, but k-means broadly groups data into a label, when what we wanted was accurate predictions. Therefore, although it was closer in scope to what we wanted, we decided against using k-means.

Now that we pinpointed the issues of efficiency and accuracy in mind, our group discussed our final method of execution: k-nearest neighbor. Unlike k-means, with k-nearest neighbor we would be able to return more accurate predictions, and unlike with decision trees, this process would be very efficient and not memory intensive. Specifically, we ended up deciding that we wanted to use a supervised learning model (which decision trees and k-nearest neighbor are) instead of an unsupervised learning model (which k-means is). Since we had a lot of training data, this would work in our favor, and we settled finally on using k-nearest neighbor.

With a dataset and a method of execution in mind, the next step was choosing the programming language to implement this in. There were many possibilities, each with their own advantages. For example, if we used Java, then we could be able to leverage our extensive knowledge of this language when designing our classes and data structures to process our dataset. Furthermore, if we used Java then we could be able to use GridDB, which is a library for Java that allows for very efficiency and scalable storing of data, which would be very useful when saving our model.[4] In the end, we decided against using Java. While it would have been possible to implement our algorithm in Java, we decided it would take too long, and that Java conventions had too many bells and whistles for our tastes.

Then, there is C++, which has the advantage of pointers and references. One of the issues with our dataset is that it was very large (giving data on all flights from 1987 to 2008), and storing this data as points in memory is not very feasible and would end up being very slow. Therefore, we would need a way to analyze our dataset, without loading the entire thing into main memory. C++ is very handy for this, due to the existence of pointers. With a pointer, we could create an iterator over our dataset, and

---

[4] https://griddb.net/en/

use a pointer in order to point to any single piece of data when assigning it. This would vastly cut down on the amount of data we store in main memory, which would make the process much faster. We decided against using C++ in the end though, because it is a difficult language to learn and to use, and not all of our group members had knowledge about how to code in C++.

The final language we discussed was Python, which is a very popular language for learning algorithms and is a very easy language to use. The biggest advantage with Python for our group was its accessibility, as all of us knew how to program in Python. Furthermore, Python has an extensive list of libraries at our disposal, such as Sklearn.[5] Sklearn provided to us all the tools we would need to model this issue and make a prediction, and we were all skilled in Python and with how to use libraries such as Sklearn, so we finalized our decision to use Python.

Finally, our group was ready to begin implementing. But, we still needed to make a plan for how to approach this problem. The first thing we did was define a set of categories we wanted to analyze. These categories were: year, month, day of month, day of week, unique carrier, origin, and destination. Year and month are relevant because flight tardiness has likely changed over time. day of month and day of week are interesting, because these categories illuminate if flights are more tardy on weekends or weekdays, and on certain holidays or not. Unique carrier is important because some planes and pilots may have a tendency to be late more than others. Finally, origin and destination are important, since a flight may be delayed due to weather or events at either point of travel, or may be delayed due to preparation times because of sheer distance.

Using these categories, we garnered some interesting results.

---

[5] https://scikit-learn.org/stable/index.html

Works Cited

[1] Herbas, J. (2020, November 8). *Using machine learning to predict flight delays*. Medium. Retrieved November 15, 2022, from https://medium.com/analytics-vidhya/using-machine-learning-to-predict-flight-delays-e8a50b0bb64c

[2] *American Statistical Association*. Default. (n.d.). Retrieved November 15, 2022, from https://www.amstat.org/

[3] Harvard Dataverse. (2008, October 6). *Data expo 2009: Airline on Time Data*. Harvard Dataverse. Retrieved November 15, 2022, from https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi%3A10.7910%2FDVN%2FHG7NV7

[4] *Open source time series database for IOT*. GridDB. (2019, August 8). Retrieved November 15, 2022, from https://griddb.net/en/

[5] *Learn*. scikit. (n.d.). Retrieved November 15, 2022, from https://scikit-learn.org/stable/index.html