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 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

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

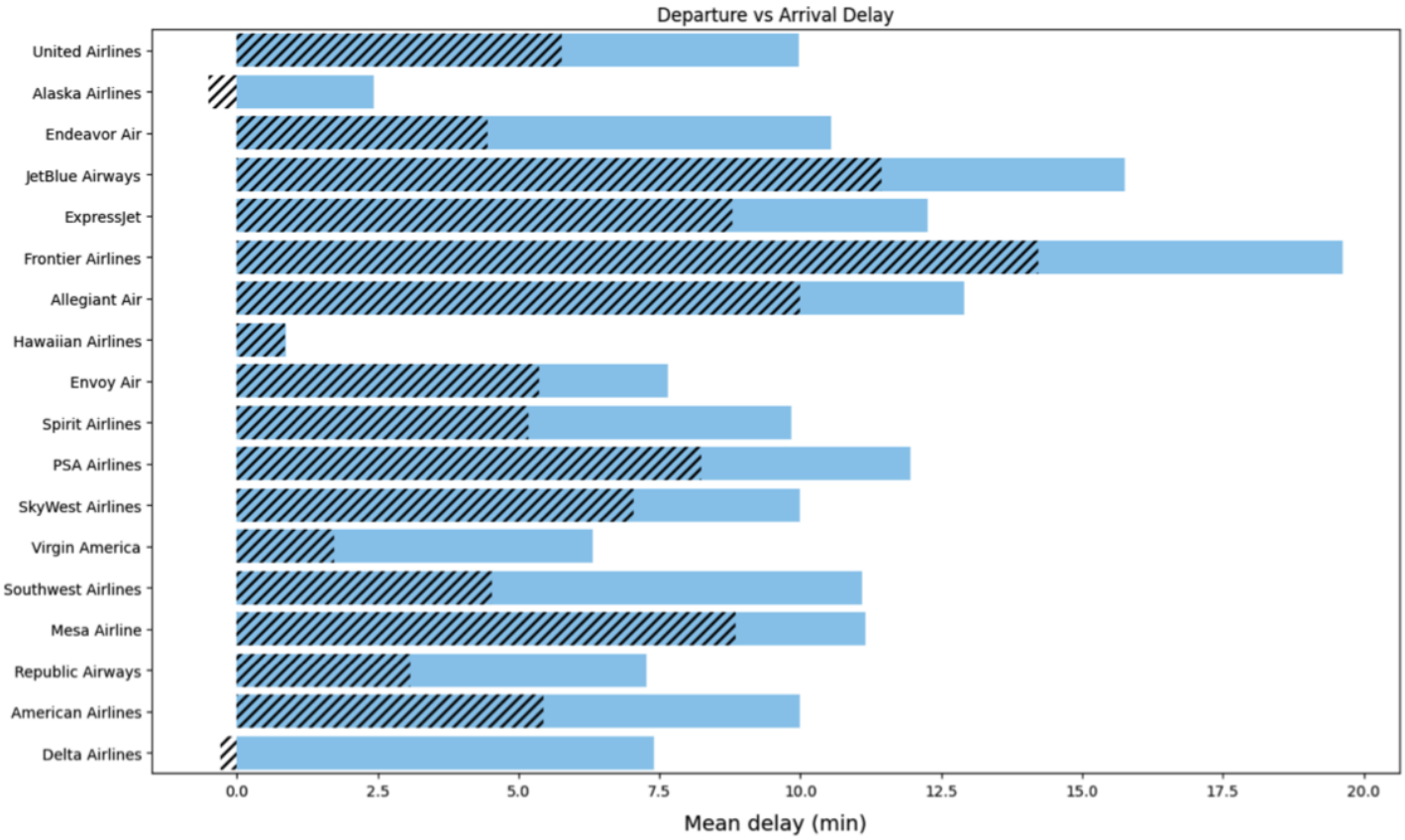

Figure 1: Sourced from https://medium.com/analytics-vidhya/using-machine-learning-to-predict-flight-delays-e8a50b0bb64c

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 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

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

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

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 numerical data in mind, our group discussed another algorithm: k-nearest neighbor. Unlike k-means, with k-nearest neighbor we would be able to return numerical predictions. Furthermore, since k-nearest neighbors is a supervised learning model, we would be able to leverage the extensive and well documented dataset we had found. One downside to this decision though, was that k-nearest neighbors is very intensive, and can take a while to run, especially with large datasets. Calculating distances between all our different data points would be very intensive, especially since we had over 100 million data points! Therefore, we cut our dataset down and settled on using KNN.

With a dataset and an algorithm in mind, the next step was choosing the programming language to implement this in. We decided on using Python, which is a very popular language for machine learning. 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.[4] 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

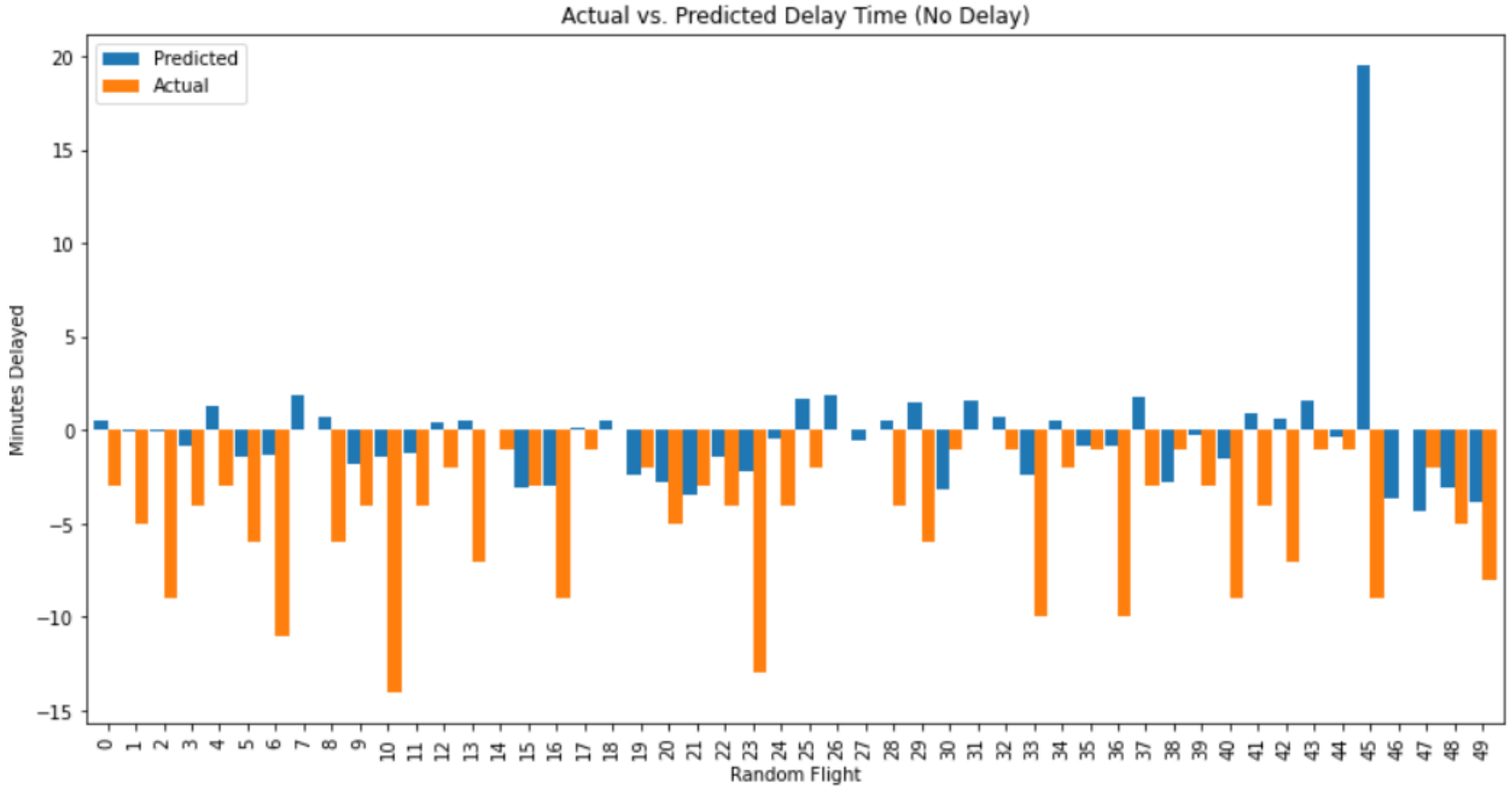

Figure 2: Flight delay predictions on flights with no delays

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

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 results with an accuracy of about 50%. We then compiled our results into three different visualizations while testing our accuracy: a set of data with no delayed flights, a set of data with only delayed flights, and a set of data with a random mix of delayed and non-delayed flights.

Take figure 2 for example. The accuracy for this chart can be denoted by how close in proximity the blue and orange bars are. Our model was not super accurate for these predictions. Because of this, we began to investigate further, and ran some predictions on flights that only had delays, to see if that would be more accurate.

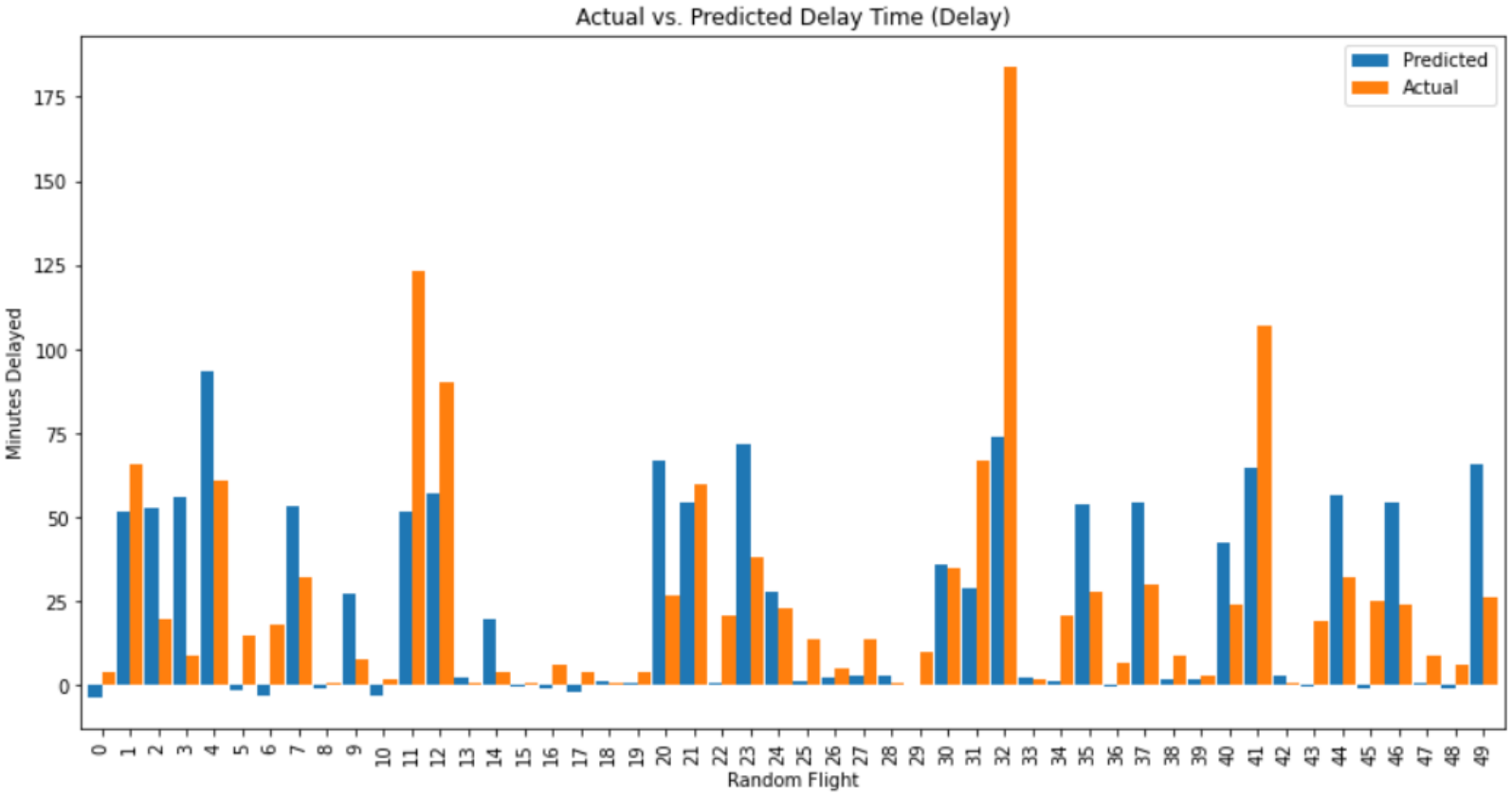

Figure 3: Flight delay predictions on flights with only delays

We created figure 3 to investigate if our model was better at making predictions for flights with delays. We found that in nearly 100% of cases, if a flight had a long delay time (25 minutes or more), our model would accurately predict that the flight would be delayed. Occasionally though, our models predictions would be off by a lot. The reason for this is because some aspects of a flight being delayed are chaotic and human-based (such as flight 32 in the chart), which our model has trouble predicting. Interestingly enough, our model proved to better at predicting how delayed a flight will be, rather than how early a flight will be. For comparison, take this chart versus the previous chart. There are a few reason for this, such as the amount of data. We found that there is more data for flights being delayed than for flights being early. Furthermore, there seems to be a human component to this dataset that our model has trouble picking up.

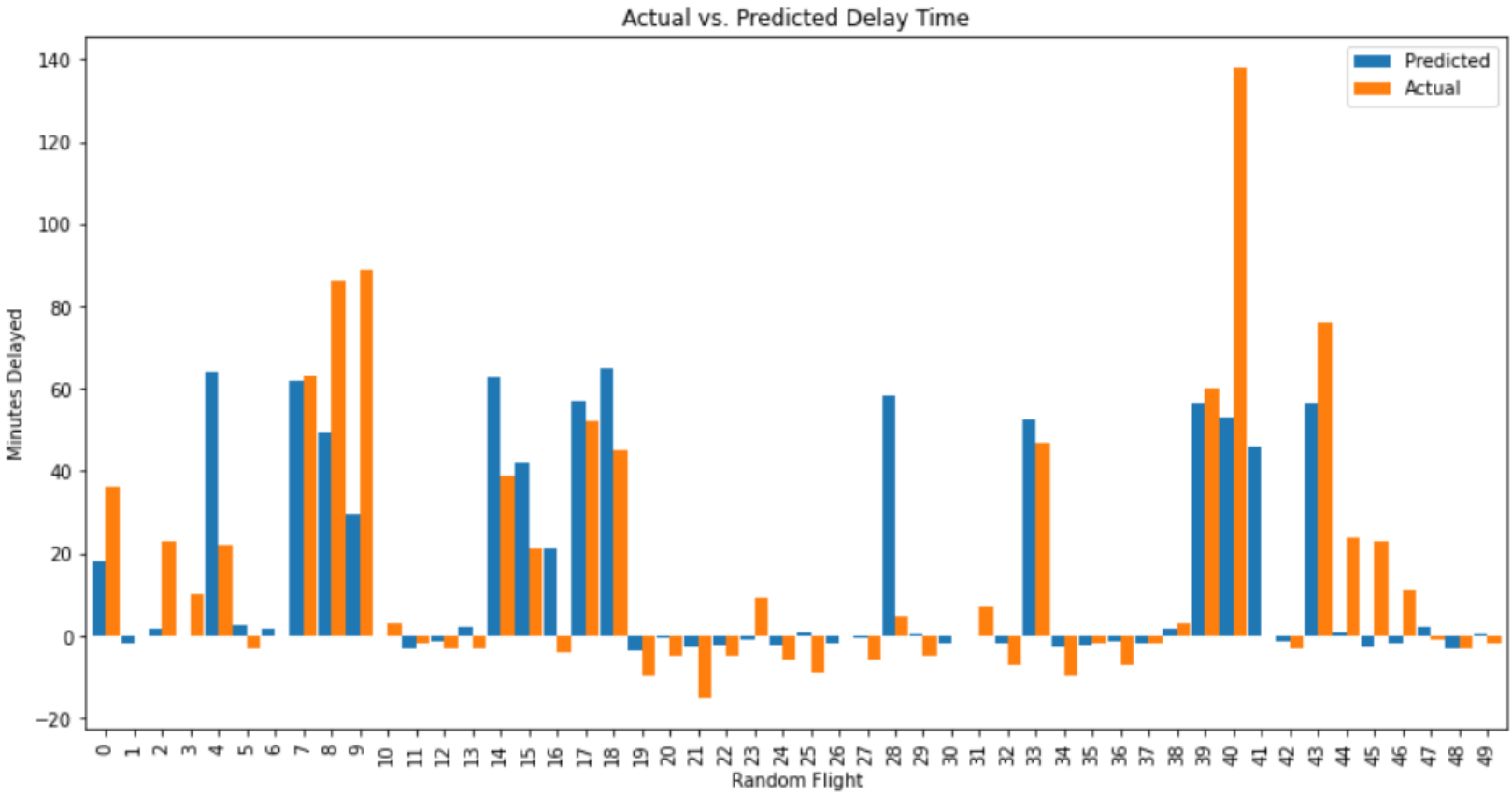

Figure 4: Flight delay predictions on 50 flights with differing amounts of delay

The final graph we generated was figure 4, which had both delayed and non-delayed flights. We were satisfied with our predictions, since they were fairly close to the actual results. This led us to the conclusion that our model is much more accurate at predicting delayed flights, than non-delayed flights. Furthermore, we found that delayed flights are usually much more late than non-delayed flights are early. Due to these reasons, we came to the conclusion that our model is more likely and better at predicting how late a flight will be rather than how early.

Overall, we succeeded in creating a prediction model for how delayed a flight will be. While our model is not great at predicting how early a flight will be, it is good at predicting how delayed a flight will be. It is also possible that there is some data not included in the datasets that would have made our predictions more accurate, especially because these datasets do not document human error. With more time, we could have further increased the accuracy of this model, although for the time we were given, we are satisfied with our results.

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