Zander Edmond

Abigail Kubli

Final Project – Writeup

Our Data:

We found this dataset on Kaggle. After searching through different datasets, we found that Sports datasets were the most appealing. Then, after going through the different ones, we found the NFL Team Data 2003-2023 dataset. The beauty of this dataset is that it spans over 2 decades, giving me 20 years of data to pull from. Also has 35 different columns with different statistics that span over those decades. Things about this dataset that are interesting are the many measurable factors, such as wins, losses, yards, turnovers, and penalties, that reflect both team success and strategic choices. You can see how teams have evolved or devolved over the two decades with all the data that was recorded. You can connect data structures to a real-world dataset.

Data Cleaning:

We started our data cleaning by checking if there were any duplicates in our data set and removing them if necessary. Next, we saw missing values in the 'ties' and 'mov' columns. We decided to remove the ties question completely as it wasn't needed to answer either of our questions and it was missing the majority of its values. For the missing 'mov' values we divided the point difference by the number of games for each column and replaced that in the missing 'mov' values.

Research Question 1: How do turnovers, including fumbles and interceptions, impact a team's win percentage?

Research Question 2: How does rushing yards per attempt impact the win percentage compared to passing yards per attempt?

Data Structures:

The data structure we used to find the first answer was a list. We chose this for several reasons; a list can store different types of data in one place, so I could store integers, floats, and strings all in one place using a list. Using a list, I was easily able to slice my data set into eras to make it easier to read, and lists allow for that simple sorting. It was easy to build my list with only the columns that I needed. Lists are generally just the easier structure to work with, as they don't have many restrictions. Using a list also worked well with plotting my function into a scatter plot.

The data structure we used for the second question was an array. This allows us to create a 2D array. This array lets us use the columns and rows from our csv file to create the columns and rows in this array. This allows for data to easily be accessed using the two different indexes. Grabbing the data from an array to create scatter plots is very simple and easy. An array provides

easy access at a very fast speed. I've worked with arrays a lot in the past so implanting them seemed like it would work well and be easier with the data we have and the way we want to organize it.

Operations and Implementations:

For the first question, we wanted to select specific ranges of rows in order to create 5 different eras, to help avoid a cluttered scatter plot. To do this, I set team20xx (xx depends on which year I want), then set it to teamStats_list[x:xx], depending on which row block I wanted to select from. I then created cleaned lists with only the specific columns I needed; I did this by setting each list year to the columns I wanted to remove the unnecessary columns. I then extracted the win percentage and turnovers column for each era to place on the X and Y axis. I then converted them to numpy arrays to perform faster numerical operations. I then added a safety check to make sure there were no NaN, None, or infinite values, so the graph didn't break on me. I then added the best-fit line to the scatter plot, showing me the regression line. Lastly, I computed the correlation coefficient and R2.

To answer the second question, we went with the same set of ranges to create five scatter plots. To do this I created five different arrays with each array containing the win percentage, rushing yards, and passing yards from our data set for the range of years chosen. To add the specific columns, I wanted to the arrays from the data set I used the column_stack function from the numpy package. When creating the plots I would create two plots for each range of years, so one plot would show the win percentage vs. rushing yards and the other would show win percentage vs. passing yards. Next, I added a best-fit line to each of the scatter plots, as well as the correlation coefficient in the top corner of every graph, to show how strong the correlation was for every group of years both visually and numerically.

Performance Analysis:

For my implementation, I chose to use Python lists to store and process the turnovers and win percentage values. Lists are a simple and flexible concept, but their performance also depends on Python looping through each element one at a time. To measure how well the lists handled my analysis, I timed how long it took to extract turnovers and win percentages and compute the correlation for different input sizes. As my dataset increased, the list of operations became slower; the first 100 rows were fast, but as I added more eras and more rows, it became slower. Thus, showing that lists scale linearly (O(n)) and become less efficient as the dataset grows. Overall, lists worked well for storing mixed data types and slicing the dataset into eras, but they are not optimized for numerical computations. They are slower in mathematical performance in comparison to NumPy arrays, and the list implementation is best suited for small-to-medium-sized datasets or simple data extraction.

I chose to use python arrays using the numpy package to store the data I needed to create the scatter plots. This data consisted of the rushing and passing yards along with the win percentage values. To access an index in an array that is a constat scale (O(1)) which makes it quick to access any part of the array no matter how big it is. Arrays are mainly due to their fixed size but that isn't much of an issue for this data set because we already know the size the array will need to be to perform these operations. Overall, arrays made it simple to store the data types and insert things to the end of the array.