Evan Primasing, Obadiah Sieg, Bricen Hicks

Math 4320

5/8/25

Dr. Mayo

Advanced Chatbot

The advanced chatbot is similar to our original chatbot, in that it outputs advanced stats for every single Division I college basketball team. However, the new and advanced chatbot now can sort through all of our original data and output it based on the user’s description. For example, users can input “Give me all college basketball teams with a three-point percentage > 35 and an assist-to-turnover rate < 2.” This will then sort through all of the data that we have in our csv file, and output any teams that match the query. This project was important because almost every traditional statistic is stuck behind complicated spreadsheets or websites that require a vast majority of your time filtering, scrolling, and looking through the data that you want to find. This chatbot eliminates that barrier and allows users to enter plain English text and get the statistics that they were looking for in a matter of seconds. Not only could it be used by coaches and fans, but also by broadcast teams who need to quickly find a statistic so that they can talk about it live on air. These broadcasters could have the chatbot pulled up to the side of them and during a timeout or media break they could easily access statistics in seconds that they otherwise would have needed minutes, which they don’t have, to find.

This project is extremely important because it allows users to quickly and efficiently find important stats about their favorite teams. This prevents users from having to constantly sort through data to find data that matches what they want to see. This could be used by college coaches, so they can have a greater understanding of what stats are truly important in determining the outcome of a game. This could save these coaches a lot of time, allowing them to spend more time game planning for the opponents and focusing on things that are more basketball-focused. Evan thought of the idea and proposed it to the rest of the group because he is very interested in sports data and how all of the data plays such a large factor into today’s sports.

We scraped all of the data from online. Most of our statistics came from <https://www.barttorvik.com/trankpre.php>, which has an extremely dense range of advanced and non-advanced stats. We also used <https://www.warrennolan.com/basketball/2025/index>, to get the rest of our advanced stats. There were still many more stats that we could have scraped from the internet, but we were strapped for time. Scraping the data took a lot of our time, as we had to constantly inspect pages and find the information that we were looking for in the elements tab. Our Python file for scraping all of our data was over one thousand lines of code, but most of it was copying and pasting similar functions that allowed us to scrape from these websites. We also allow users to sort the rankings by stats. Users can input “Show me the top 10 teams in total shooting percentage” or however many teams they want to show in whatever statistical category we have and it will quickly and efficiently sort the teams and display them. This is another feature that will allow users to easily get a greater understanding of the importance of statistics, and quickly find any stat that they are looking for.

The process that we used to create the chatbot was mostly using the code that you provided Dr. Mayo. We stripped the information that the user input, and used TFIDF vectorization to allow our model to get a greater understanding of what data the input matched. Upon stripping it to TFIDF, we fed the input to our CSV file that contained all of our data allowing it to grab the nearest match to the TFIDF score. Using the TFIDF scores also allowed us to clean up a lot of the fluff that a user enters, to ensure that it is down to bare bones. TFIDF Vectorization allowed us to emphasize meaningful words, allowing our bot to improve its accuracy in providing the data significantly. After cleaning and organizing the dataset, we mainly wanted to focus on developing the bot’s ability to parse natural language inputs. We began with a basic keyword search method but realized that it was extremely limited. Most of the queries contained comparators (“greater than”, “less than”), numbers, and logical conjunctions such as and or “or”. This made matching our keywords exactly ineffective. To fix this issue, we used regular expressions to extract the logical components and match them against the appropriate statistical categories in the CSV file.

We faced a lot of issues scraping the data. We had to completely learn how to do it, and many of the pages that we scraped from were extremely complicated. It took us a couple of days to figure out the best process for scraping the data, and then once we got the hang of it, we reused functions throughout to help us. Most of the data that we scraped was all in the exact same format, so we could just change the name of the page that we were scraping to match each individual stat and keep the rest of the scraping methods the exact same. Furthermore, allowing users to input multiple queries such as “show me teams with a three-point field goal percentage greater than 36 and a two-point field goal percentage greater than 50” was difficult because it was hard to get our bot to put the two queries together. It could easily show each input individually, but enforcing both queries to separate the teams was a difficult function that took us days to figure out. We also faced challenges when we allowed users to sort by conference as well. For some reason every time a user tried to sort by the B10 or B12 (Big 10 and Big 12 are both conferences), our bot was not able to bring all the teams up. We believe that since it is letters and numbers, B and then the 10 or 12, this throws an error in our bot. For example when a user enters “Can you show me teams in the B10” this is what our bot outputs “ Data bot: C:\Users\evanp\AppData\Roaming\Python\Python312\site-packages\sklearn\feature\_extraction\text.py:517: UserWarning: The parameter 'token\_pattern' will not be used since 'tokenizer' is not None' warnings.warn (The Citadel,296,362,94.5,116.3,46.90%,49.40%,27.20%,68.00%,46.60%,52.20%,8.30%,3.30%,65.7,0.452,0.328,63,40.30%,49.50%,29.20%,62.30%,11.6,0.9,76.9,47.10%,56.30%,34.70%,5.5,1.9,16.2,93.8356,117.6351,0.069436,-21.536, SC). We could not find any reason as to why this was causing this error, and The Citadel, the team it pulled up, is not in the Big 10 they are in the Southern Conference (SC).

If we had more time we would have loved to add stats from individual players, to allow coaches to get a greater understanding of exactly which players were the most important in their upcoming matchup as well as seeing how they fare from their stats. We would have loved to allow users to input queries such as “Show me the top five points guards in the SEC in assist percentage” This would have allowed an entirely new layer of analysis, that would have gone into even further depth. To do this we would have had to scrape even more data, ensuring that we had data from every single player in Division One college basketball. Unfortunately in the time that we had we were not able to do this, as scraping, processing, and cleaning the data is a long process that we would not have been able to do in time. Another idea we talked about adding was a user interface for our chatbot. We thought that it would be super cool if the user could interact with this on the web, rather than through their command line. This would have allowed users to visually see the data better, as well as giving them a better user experience rather than typing everything out through the terminal.

These are the websites that we used for referencing and scraping

<https://barttorvik.com/trankpre.php>

<https://www.warrennolan.com/basketball/2025/stats>