Jacob Wilkens

Professor Locklair

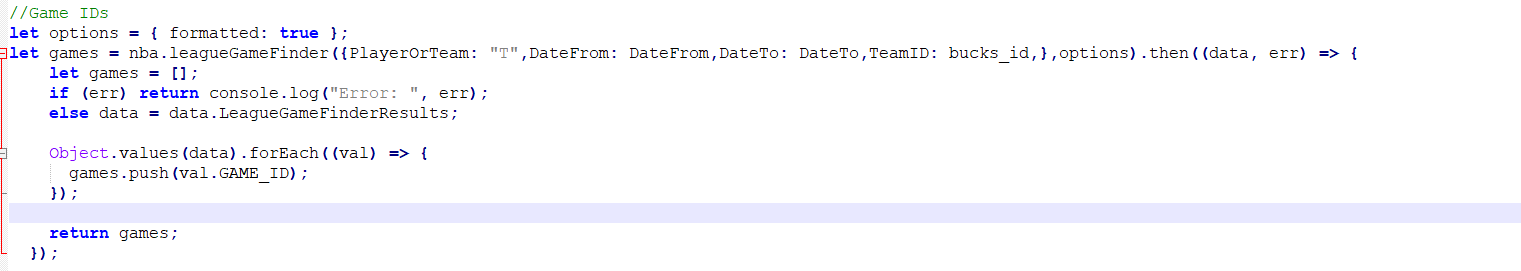
CSC 417

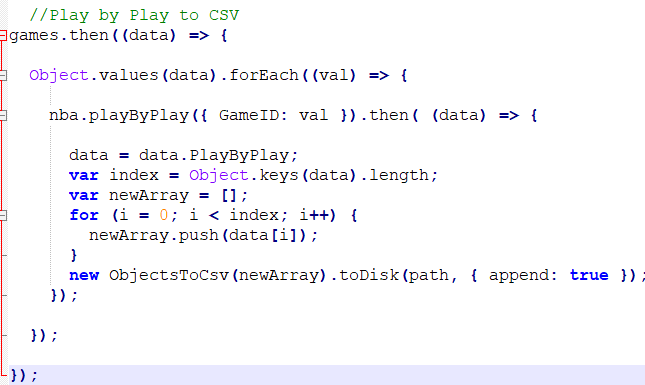
4 May 2021

Project Documentation

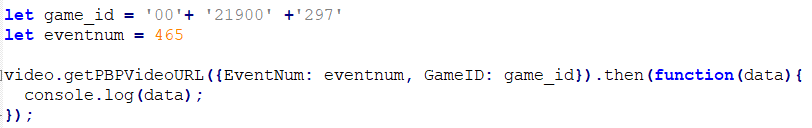
The goal of this project was to train a neural network to classify highlight plays based on play descriptions. Data preparation took way longer than expected, and I spent all my time on that. Since this happened, I ran out of time to implement the neural network. Now I will discuss the way I constructed and gathered my data. All files related to data collection and preparation will be included and referenced in the appropriate sections. Also screenshot of code, tables, and output will be provided as well.

The code for gathering the data can be found in “index.js”. This uses nodejs with 3 packages which were used in gathering the data. First package which is the basis of the data gathering is [**nba-api-clinet**](https://www.npmjs.com/package/nba-api-client). This first code snippet gathers the game ids for Bucks games in the 2019-20 season based on the data range provided.

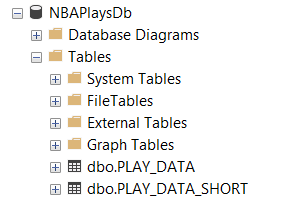


[**Objects-to-csv**](https://www.npmjs.com/package/objects-to-csv) is a package which saves given objects to csv. This is how I put together my CSVs and the data was gathered. This next code snippet saves a csv with play descriptions by passing in the game ids which the previous segment addressed. 

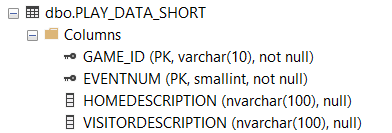
Last package to be used allowed for access to individual video of a certain play by inputting the given game id and event number. This package is [**nba-pbp-video**](https://github.com/mtthai/nba-pbp-video). Note that there was an issue with resolving the requests for both NBA packages, but this issue should be fixed in the node modules in my github project. This code snippet shows how to get the video for a certain play.

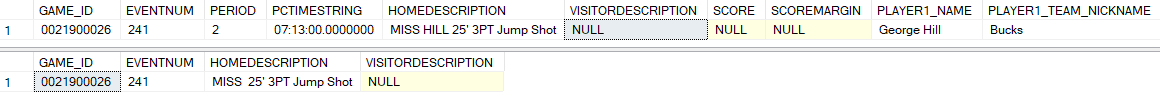


For my data, I planned to use a training and test set. My training set was “Data.csv” and test was “Test Data.csv”. These two csv files were inserted into my local SQL Server database. Both files had two separate tables for a total of four tables. I created a database called NBAPlaysDb. The table PLAY\_DATA held all the information in “Data.csv”.

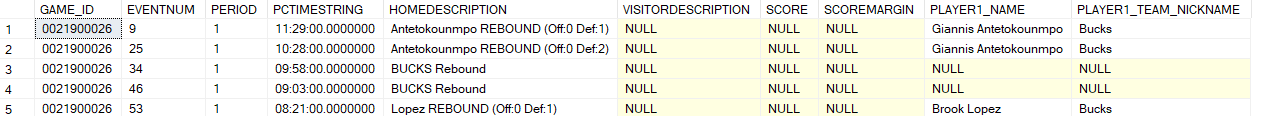


PLAY\_DATA\_SHORT held a subset of information which made the play descriptions as generic as possible. All records can be joined on GAME\_ID and EVENTNUM. This is an example of output. The goal was to eliminate any specific phrases from the play description. Some specific phrases could have included stat line, player names, and team names. Also some plays could have had two descriptions linked with an event. An example of this could be a block. One player gets blocked while the other got a block in their stat line. In the example below, the name Hill is removed.

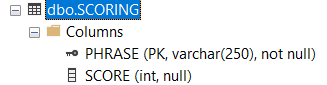


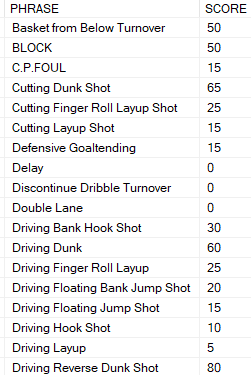


In my GitHub project, I will be uploading my SQL scripts which were written for updating, inserting, and deleting data from these tables. Most of my scripts are iterative by using multiple temporary tables to clean the data, and then finally the final generic plays were inserted into PLAY\_DATA\_SHORT. The same process was also done on TEST\_DATA and TEST\_DATA\_SHORT. These tables are exactly like the two tables shown above but with different names. The SQL scripts which were used for data preparation were “Final Rewrite Foul.sql”, “Final Rewrite.sql”, “Test Data Foul”, and “Test Data”.

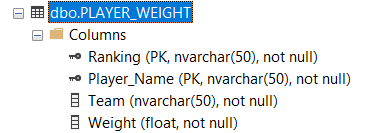
Records were eliminated from the data set. These records included certain fouls, rebounds, and substitutions. Records such as these did not have any significance regarding highlight plays. 

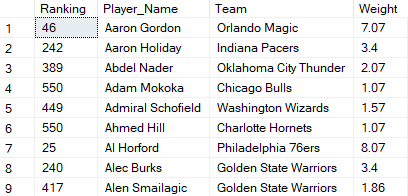
Calculating score was done by splitting each phrase on a ‘. The script that did this is “Insert into Scoring.sql”. A SCORING table was constructed to hold phrases and their scores. I scored each phrase individually. There was a total of 260 phrases. Some notes about the choices I made for scoring. Dunks were scored the highest. Numbers in the table represnted distance from the basket. For example a 26’ would be scored as a 26. If there was a miss, the score would always be 0.



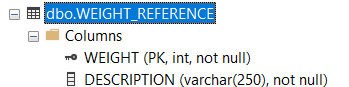


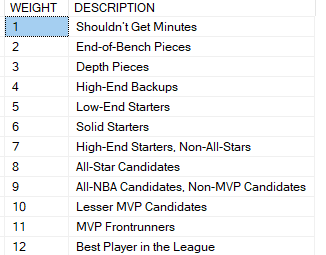
I also incorporated player scoring into the total as well. Players mentioned in the data set had a score which was used from [nbamath.com](https://nbamath.com/crystalbasketball-ranking-all-nba-players-for-2019-20/). This was a list based on player significance for the 2019-20 season. A table called PLAYER\_WEIGHT was created. This table held all players and their weight.



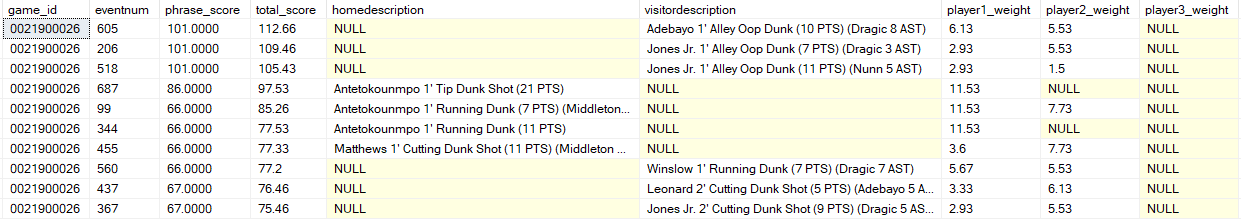


A reference table called WEIGHT\_REFERENCE was established to see what the scores meant.





The important part of the project was calculating the score for a play. “Calculate Score.sql” was the script which calculated the score. Here is some example output of the top 10 plays for a specific game. There is the phrase score and the player score.



Some final thoughts on my project. I spent a lot of time trying to figure out what I wanted to do with highlight plays. First, I wanted to do something with NLP, but I kept finding conflicting ideas. Then, I ran into some issues with finding data. From the beginning, I wanted to use nodejs, and I finally found a package to get the data I was looking for. There was an issue, however. Getting data took longer because I could only make a request for about 10 games every 15 minutes. The data prep took me way longer than I expected. There were some mistakes that happened, and I underestimated how long it would take. I did not get to writing the code for the neural network unfortunately. I learned that data preparation is a huge part of the process. For the neural network, I would have used KERAS.

I also learned that word descriptions can only provide so much information. This also was subjective data because I scored each of the phrases myself. Also, there can be more phrases, as I only had phrases which showed up in my data set of about 60 games from the perspective of the Bucks in the 2019-20 season. It was neat to have access to the video for most of the plays that were in my dataset. Overall, I did not have enough time to implement the neural network because all my time was put into data gathering and preparation. There is a new found appreciation for clean data.