Should School Mascots Influence NCAA March Madness Predictions?

Isaak Hedding

Introduction & Motivation

Every year, millions of brackets are made in an attempt to predict the outcome of the NCAA Division 1 Men's Basketball National Championship Tournament, also known as March Madness. Many bracket creators base their selections for the tournament outcome off of facts, statistics, and teams' performance during the regular season. However, some bracket creators have somewhat unorthodox methods of making their selections, including flipping a coin for each matchup, choosing matchup winners based on which school has a closer proximity to the location of the game, and choosing matchup winners based on which school's mascot they prefer.

The last aforementioned method has always intrigued me. Although it sounds rather farfetched and most people who go with this method are just doing it for fun, could there possibly be any logical, empirically-based reason to choose matchup winners based on one mascot as opposed to the other? This is the question I set out to answer in my exploration of NCAA Men's Basketball data.

Data Sources

In order to answer this question, I would need data on all NCAA D1 Men's teams' performance during the season (# of wins and # of losses), and data on all these teams' mascots.

The first data source was retrieved using the Sportradar API, located at https://developer.sportradar.com/docs/read/basketball/NCAA_Mens_Basketball_v4#standings. This API would provide me with data on each team's performance for a given season (in this case, the 2018-2019 season). In order to access this source, I needed to register for an API key and create an application. The data was retrieved using Python's http.client module, and came in XML format. I then used the BeautifulSoup module to parse this data and get the specific variables I wanted, which in this case were each team's name, number of wins, and number of losses. After locating this data, I inserted it into an SQLite file. It has 348 entries. (See the create_ncaab_data.py file for how I retrieved this data).

The next data source was found on Kaggle - https://www.kaggle.com/ncaa/ncaa-basketball. This source had a dataset on the mascots for each team. It provided variables including the team name, mascot name, and full taxonomic classification of each mascot. It was in BigQuery format and contained several different data tables, however, I just needed the mascots one. I was unable to successfully use Kaggle's suggested BigQuery module to access it, but I was able to copy and paste

the table perfectly into a Microsoft Excel and save it as a CSV file. Schools' mascots are a static variable, so there was no time period that they covered. It has 351 entries.

Data Manipulation Methods

Needs

The first manipulation that needed to be performed on the two datasets was preparing them to be merged together. This required altering the column names on both data sources after they had been loaded into a pandas DataFrame object. There was only one column that the two datasets had in common, that being the school name.

It is important to note that the way in which I'd be measuring if mascots should influence March Madness selections was based on the average number of wins that multiple schools with the same mascot had. If only one school had a certain mascot, it would not be included in the calculations, because that school may have a very good or very bad basketball team. For example, the University of Virginia won the national championship this year and had the best record out of any team. They are the only school whose mascot is the Cavaliers, so it would not make sense to include them as their own group.

I needed to manipulate the data so that I could create a count for each mascot, and sort the data accordingly to draw insights.

Conversion/Processing

I decided to filter the mascots DataFrame to include only schools that had 2 or more schools with the same mascot. In other words, each mascot included in the calculations must be the mascot for at least 3 different schools. The way I filtered this was by creating a dictionary that included all unique mascots as the keys, and the number of teams each mascot is the mascot for as the values.

```
{'Golden Hurricane': 1, 'Sun Devils': 1, 'Dragons': 1, 'Braves': 2, 'Golden Griffins': 1, 'Wolverine
s': 2, 'Flames': 2, 'Phoenix': 2, 'Delta Devils': 1, 'Demon Deacons': 1, 'Demons': 1, 'Islanders':
1, 'Waves': 1, 'Fighting Illini': 1, 'Billikens': 1, 'Hilltoppers': 1, 'Aggies': 5, 'Titans': 2, 'Ja
spers': 1, 'Jayhawks': 1, 'Chippewas': 1, 'Blazers': 1, 'Tribe': 1, 'Blue Raiders': 1, 'Blue Demon
s': 1, 'Blue Devils': 2, 'Horned Frogs': 1, 'Anteaters': 1, 'Spiders': 1, 'Cardinal': 1, 'Rebels':
2, 'Spartans': 5, 'Governors': 1, 'Red Flash': 1, 'Highlanders': 3, 'Pioneers': 2, 'Explorers': 1,
```

A snippet of the dictionary

(I.e., there is only one team that has 'Golden Hurricane' as their mascot and 2 that have 'Wolverines')

I then defined a function that takes in a mascot name, checks to see if that mascot name is in the previously created dictionary, and if so, returns the count for how many teams it is the mascot for. I then used a universal function to create a new column in the mascots DataFrame called 'count', which is the number of teams that a team's mascot is the mascot for.

Having this new column, I filtered the DataFrame to include only mascots with a count value of three or higher. I used the .apply() method with my previously defined function to create this

column. I figured that if at least 3 separate teams with the same mascot have an overall average number of wins that is significantly high, then it might be worthwhile to select a team with that mascot to win matchups in March Madness.

Joining The Data

Next, I had to merge this DataFrame with the DataFrame containing data on the number of wins and losses for each team. The newly created mascots DataFrame only had 139 entries while the other DataFrame had 348. I only needed the 139 teams who share a mascot with at least 2 other schools, so I did an inner merge between the two DataFrames on the school name column. The merge was successful and there happened to be **no missing or incomplete data**.

Next, I needed to group the merged DataFrame on the mascot name in order to calculate the average number of wins for schools with that mascot. This was done using pandas .groupby() method, and grouping by the mascot name key.

From there, I took the average number of wins for each group and sorted the values in descending order. This will be described further in the analysis section.

Challenges

Ultimately, I had a lot of control over the two DataFrames and knew them very well, so I was able to overcome the challenges relatively easily. The biggest challenge I faced was finding a way to create a count for each mascot. I tried using pandas methods such as .groupby() and .count(), but was unsuccessful in achieving the result I wanted. I was forced to take a more 'pythonic' approach by creating a dictionary and then defining a function that could be applied to all entries in the mascots DataFrame.

Another challenge I faced was determining which value I wanted to do the count for. The mascots DataFrame had a key column called 'mascot' which initially seemed like the appropriate column to count. However, when attempting to do the count on this column, I kept getting 'None' as a key in the dictionary. I solved this by looking at the DataFrame in greater depth. I realized that all schools have a name that they call their sports teams (e.g., Eagles, Crusaders) but not all have a physical mascot (i.e., a person dressing up in a costume of the mascot and cheering alongside the cheerleaders at games). U of M is one of these schools.

The schools who do not have a physical mascot have 'None' listed in their mascot column. Additionally, some schools had a physical mascot that is different than what they call their team. For example, several schools call their teams "Aggies", but each one seemed to have a different physical mascot (e.g., one 'Aggie' is a dog, another one is a bull). Ultimately, I decided to just go by what the schools call their teams, as many schools have physical mascots that are slightly different than what they call their teams. If I went by physical mascots, it would be much harder to form groups.

Workflow

The overall workflow of my code is as follows (it can be found in the final_project.ipynb file). I first import my necessary modules: pandas, sqlite3, matplotlib.pyplot. Next, I read the mascots CSV file using .read_csv() and adjust the column names to what I want them to be.

From there, I create the count dictionary, and define the get_count() function that returns the number of teams a mascot is the mascot for. I then apply this function to all entries in the mascots DataFrame to create the 'count' column, and filter the DataFrame to include only schools whose 'count' is 3 or higher.

The second part of the workflow consists of creating a connection object to the SQLite file I created, and using .read_sql_query() to load it into a DataFrame. I then merge this DataFrame with the mascots one on the team name column.

Finally, I group the merged data by the mascot type, calculate the average number of wins for each mascot group, and sort the values in descending order. This data is then graphed using matplotlib to create a bar chart.

Analysis and Visualizations

Process

To reiterate what was described in the Data Manipulations section, the analysis was conducted by putting schools with the same mascots into groups using the .groupby() method, and then the .mean() method on the 'wins' column for this GroupBy object. This provided me with the average number of wins each mascot had (for mascots that appear in 3 or more schools).

My next step was to sort this GroupBy object in descending order (with the .sort_values() method), to observe which mascot had the highest average number of wins and which had the lowest.

After completing this step, I then looked to see if there was a significant discrepancy between the average number of wins. There are 31 regular season games that schools play each year, so having an average of 16 wins means that a team has won more than half its games and can be considered a good season. Winning less than 16 games means that a team has a lost more than half its games and can be considered a poor season.

There were 27 groups of mascots who appeared in 3 or more schools, which was a short enough list to look over manually. Interestingly, there was a significant discrepancy for average number of wins between the groups. The highest average number of wins was for the mascot 'Terriers' with 20.67 wins, and the lowest average was 'Cowboys' with 9.67 wins. The other groups had a variety of different averages which were spaced pretty evenly between the highest and lowest values.

Insights

One interesting finding from the results are that dog-related mascot groups tended to have higher average number of wins, and bird-related mascot groups tended to have lower average number of wins. Specifically, 3 of the top 4 mascot groups with the highest number of wins were dog-related: 'Terriers' being first, 'Aggies' being second**, and 'Huskies' being fourth. Not a single dog-related mascot group averaged lower than 16 wins on the season. On the other end, not a single bird-related mascot group had an average number of wins that was at least 16 or greater.

***Note: 'Aggie' is an ambiguous mascot with several different physical representations. However, 5 different schools had 'Aggie' as a mascot, 2 being represented as dogs, thus I included it in dog-related mascot groups.*

Another interesting finding is that the number of teams in a mascot group did not appear to skew the average number of wins for that group in any particular direction. One might expect that mascot groups that are comprised of a larger number of teams would have an average number of wins that is more toward the middle – the 16-win mark – and mascot groups with a lower number of teams comprising it would be toward either the higher or lower end. For example, it is more likely that a mascot group comprised of 3 teams would have a higher average number of wins than a mascot group comprised of 10 teams.

However, this did not seem to be the case. In fact, some of the mascot groups that are comprised of many teams, such as 'Wildcats' with 10 teams, had a very high average number of wins (19.9 for Wildcats), while mascot groups comprised of a low number of teams, such as 'Cowboys' with 3 teams, had a very low average number of wins (9.67 for Cowboys).

Both of these findings suggest that there may be reason to select a team with a particular mascot to win in a matchup against a team with another particular mascot. See limitations for more information on this.

What Didn't Work

While I was able to attain a good amount of the data that I was looking for, there were some more complex findings that I couldn't reach through pandas methods.

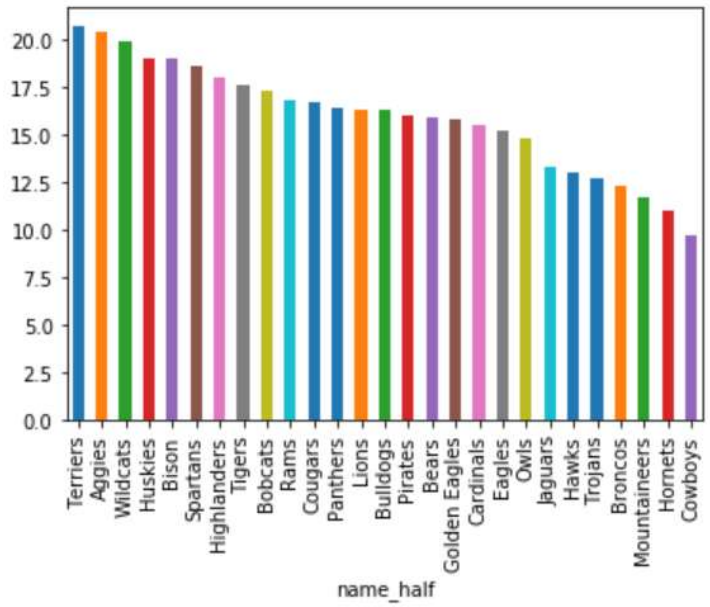
For example, I was hoping to breakdown the mascot groups even further and calculate the average number of wins for a broader group of mascot types (e.g., 'Dogs', 'Cats', 'Birds'). The mascots dataset even came with data on the taxonomy for mascots that are actual animals (i.e., the full taxonomic classification). However, this proved to be too complicated of a task and somewhat beyond the scope of this project. Additionally, not all mascots had the full classification. For example, some schools' mascots consisted of things like 'Dragons' or 'Hurricane', which are not animals and thus do not have the classification.

Another factor that I attempted to analyze was the average number of post-season wins for mascot groups. I was unsuccessful in this, as teams play in a variety of different numbers of post season games, and different types too. Some play in the NCAA Tournament, some play in the NIT

Tournament, and some play in different tournaments. The difficulty in the NCAA Tournament is significantly higher than that of the NIT and has more games involved, so the data may not be fully representative.

Visualization

The visualization that I created from the collected data involved using the matplotlib module to create a bar chart of the average number of wins for each mascot group, shown below.



Limitations & Conclusion

While this data analysis and manipulation led to results that suggested some evidence toward there being reason for choosing teams to win March Madness matchups based on the teams' mascots, it by no means has proven that teams with specific mascots are more likely to win games than others. It is limited by the fact that only mascots who share 3 or more teams were included in the analysis, and that it was only based on games for the 2018-2019 season.

However, it does suggest that more research could be done into the subject. Perhaps there's a way to normalize the data further and include all mascots, regardless of the number of teams they're the mascot for. Teams' records for seasons dating back even farther could be calculated as well, and the data could be compared.

Overall, this study was intended to crack the surface on a topic that is regarded as slightly farfetched. By showing that there may be some empirical evidence toward selecting teams based on mascots, there exists the exciting chance that brackets can be created in other ways than the traditional way of straight statistics.