**CSCI 185: Final Project**

**Project Title:** Basketball by the Numbers: A Deep Dive into NBA Statistics

**Group Members:** Parneet Kaur and Lydia Myla

**Dataset:**

For our project, we have chosen to scrape data from the following website: [NBA 2023-24 Season Stats](https://www.basketball-reference.com/leagues/NBA_2024.html#all_per_game_team-opponent). This website provided us with the stats of every player and gave us information on their overall performance during the course of the season, such as their averages of points per game, field goal percentage, three-point percentage, rebounds, and assists.

**Project Idea and Objective**

For this project, our group has decided to analyze data received from NBA stats. From this dataset, we can extract meaningful insights from the NBA 2023-24 season, which can help us identify players’ strengths and weaknesses. Given the data we will scrape, it will include players’ statistics that will allow us to analyze and decide what we want to work with. For example, a high 3P% is an indicator that a player is good at making three-point shots and this could indicate that the player has a points per game average (PPG). In Data Preprocessing, we will be converting raw player statistics into a categorical format. For example, when dealing with a continuous feature (e.g. Type of Scorer), then the threshold condition can be that PPG must be greater than 15 to qualify as a “high scorer” and lower than 10 to qualify as a “low scorer”. We are using several algorithms to uncover hidden relationships between basketball player performance metrics. First, we will apply the Apriori algorithm to generate simple association rules by converting continuous metrics (e.g., ppg) into categories. While Apriori is useful for interpreting basic correlations, it can be slow with larger datasets. To explore more complex relationships, we will implement FP-Growth, which generates multi-level association rules and handles larger datasets efficiently. Then, we’ll use CHARM to identify crucial rules by removing redundant ones, ensuring that we focus on the most informative patterns. Finally, we will also use a content-based filtering algorithm (cosine similarity) to recommend similar players, such as finding players similar to Stephen Curry.

**First Steps**

* **Fetching the HTML Content**

To find data to work with, we decided to scrape stats from a website called Basketball Reference. Here, we chose the NBA Season that we wanted to work with, the 2023-24 season, and then copied the URL of that webpage. At the beginning of our code, we set the url to the website link, fetched the data using “User-Agent: Mozilla/5.0”, and used BeautifulSoup to parse through the HTML content to find the stats.

* **Importing Library**

In this initial step of our data analysis process, we lay the groundwork by importing essential libraries and loading our dataset. First, we imported critical libraries and modules tailored to our needs, including requests, BeautifulSoup, pandas, and multiple libraries and their functions used to display/visualize our results. These libraries provide comprehensive support for numerical computation, data manipulation, and a diverse range of plotting options.

* **Preprocessing**

After finding the stats, we extracted the headers, column names, and the rows, players’ data. Following this step, by using pandas’ DataFrame function, we created a dataframe to place our extracted data. After this, we cleaned our data by dropping irrelevant columns and rows with missing values. Once our data was pre-processed, we saved it to a csv file to be accessed throughout our project code.

**Algorithms and Visual Analysis**

* **Apriori Algorithm and Association Rules**

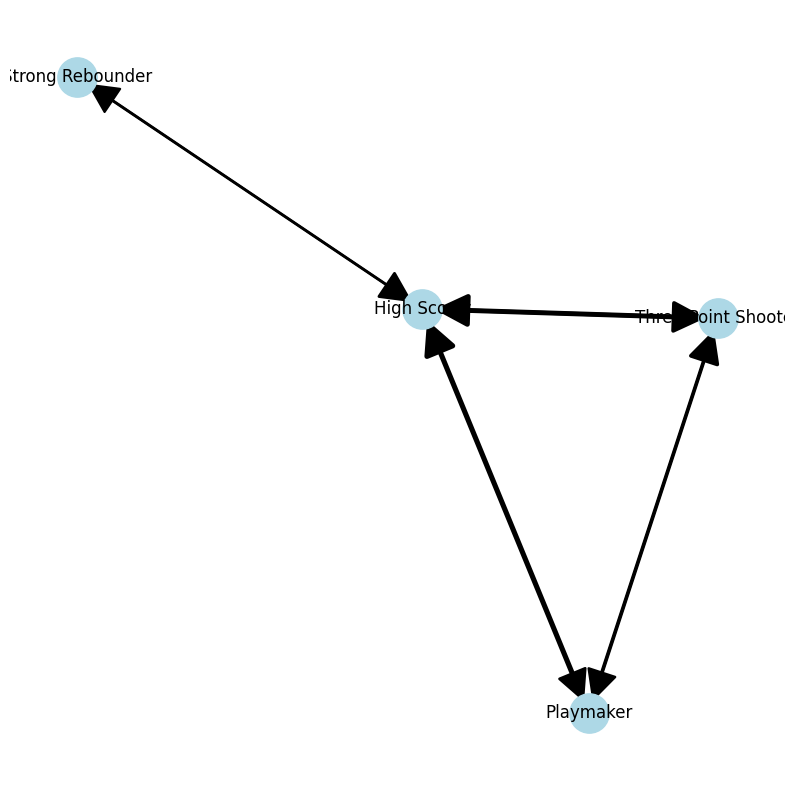
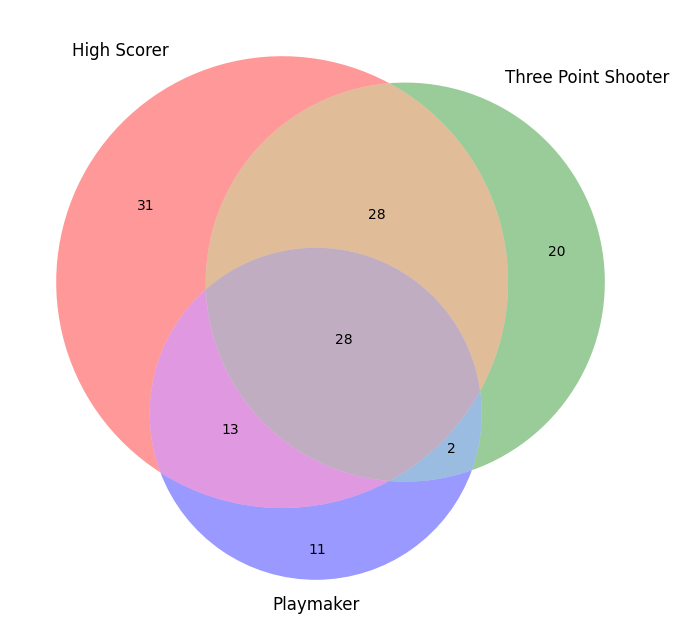
The primary goal of this task was identify patterns that frequently co-occur among players and extract meaningful insights about their playing styles. The process began with some data preprocessing where we transformed numerical performance statistics (such as Points Per Game (PTS), Assists (AST), and Three-Point Percentage (3P%)) into binary attributes. Players were classified into different categories: high scorer (PTS > 15), three-point shooter (3P% > 2.0), playmaker (AST > 5), strong rebounder (TRB > 6). This classification was essential for association rules mining and clustering, as it transformed numerical statistics into binary categories. We, then, applied the Apriori Algorithm with a minimum support threshold of 0.02 and a confidence threshold of 0.03 to ensure meaningful but not overly rare patterns were captured. Below is a table of the association rules that were generated by our Apriori algorithm. It displays the support, confidence, and lift values, along with the number of players that are associated with each rule.

| **Antecedents** | **Consequents** | **Support** | **Confidence** | **Lift** | **Player Count** |
| --- | --- | --- | --- | --- | --- |
| **Three-Point Shooter** | High Scorer | 0.081522 | 0.705882 | 4.810458 | 60 |
| **High Scorer** | Three-Point Shooter | 0.081522 | 0.555556 | 4.810458 | 60 |
| **Playmaker** | High Scorer | 0.057065 | 0.736842 | 5.021442 | 42 |
| **High Scorer** | Playmaker | 0.057065 | 0.388889 | 5.021442 | 42 |
| **Strong Rebounder** | High Scorer | 0.047554 | 0.421687 | 2.873717 | 35 |
| **High Scorer** | Strong Rebounder | 0.047554 | 0.324074 | 2.873717 | 35 |
| **Playmaker** | Three-Point Shooter | 0.04212 | 0.54386 | 4.709185 | 31 |
| **Three-Point Shooter** | Playmaker | 0.04212 | 0.364706 | 4.709185 | 31 |
| **Three-Point Shooter** | High Scorer, Playmaker | 0.039402 | 0.341176 | 5.978711 | 29 |
| **Playmaker** | High Scorer, Three-Point Shooter | 0.039402 | 0.508772 | 6.240936 | 29 |
| **High Scorer** | Playmaker, Three-Point Shooter | 0.039402 | 0.268519 | 6.375149 | 29 |
| **Playmaker, Three-Point Shooter** | High Scorer | 0.039402 | 0.935484 | 6.375149 | 29 |
| **High Scorer, Three-Point Shooter** | Playmaker | 0.039402 | 0.483333 | 6.240936 | 29 |
| **High Scorer, Playmaker** | Three-Point Shooter | 0.039402 | 0.690476 | 5.978711 | 29 |

The findings indicate a crucial bidirectional relationship between High Scorers and Three-Point Shooters, where 70.6% of the Three-Point Shooters are also High Scorers, and 55.6% of High Scorers are also Three-Point Shooters. This suggests that proficient three-point shooters tend to be top scorers, emphasizing the importance of perimeter shooting in scoring efficiency. In addition to this, Playmakers show a strong link to High Scoring, with 73.6% of Playmakers also being High Scorers and high lift score 5.02, indicating that elite facilitators are also highly likely to contribute to scoring.

There were more advanced multi-feature patterns captured in this algorithm as well. A particularly strong rule is that players who are both Playmakers and Three-Point Shooters have a 93.5% likelihood of also being High-Scorers, with a lift of 6.37. This indicates that such players are almost guaranteed to be elite scorers. Another key finding is that many High Scorers and Three-Point Shooters also take on a Playmaking role, reinforcing the idea that modern high-volume scorers are often also facilitators, which is similar to elite guards and forwards. Therefore, such results helped define player archetypes, such as sharp-shooting playmakers who may influence the game beyond the scoring statistics.

To better understand these associations, we visualized them using network graphs, to show how different traits are connected (i.e with a thicker edge representing a stronger relationship) and venn diagrams (to display the overlap between different player traits, specifically the large intersection between High Scorers and Three-Pointer Shooters).

As we can see from the above network graph on the left, there is a directional relationship between key basketball traits, showing how certain attributes influence others. Notably, High-Scorers emerges as the central node, receiving strong connections from Three-Point Shooter and Playmaker. This aligns with the results achieved in the Apriori algorithm earlier, where the rules (e.g. Three-Point Shooter -> High Scorer and Playmaker, Three-Point Shooter -> High Scorer) were identified with high confidence values. The thickness of the edges in the network graph reinforces the strength of these associations. The Venn diagram on the right, on the previous page, was observed to further strengthen these findings by illustrating the overlap between different player traits. As we can see, there are large intersections between High Scorer and Three-Point Shooter (28 players), which confirms the rule from Apriori that these two attributes frequently co-occur. Moreover, 28 players are identified as having all three traits, High Scorer, Three-Point Shooter, and Playmaker, supporting the 85% confidence rule discovered earlier. There are also smaller, more specific intersections (e.g. 2 players being both Playmakers and Three-Point Shooters without being High Scorers). This highlights interesting outliers that merit further analysis. These insights provided a solid foundation for further player profiling and performance evaluations.

* **FP Growth**

The FP-Growth (Frequent Pattern Growth) algorithm was chosen as the next step after Apriori because of its computational efficiency and ability to handle large datasets without generating an overwhelming number of candidate sets. Unlike the Apriori Algorithm, which uses a generate-and-test approach that scans the dataset multiple times to create frequent itemsets, FP-Growth constructs a compact tree structure that helps it to mine frequent patterns more effectively by reducing redundant sans. Given that we were hoping to scale, we used the FP-Tree. The FP-Growth results, as seen in the below table, matched closely with those from Apriori, reinforcing the reliability of the discovered associations previously.

| **Antecedents** | **Consequents** | **Support** | **Confidence** | **Lift** | **Player Count** |
| --- | --- | --- | --- | --- | --- |
| **Strong Rebounder** | High Scorer | 0.048 | 0.422 | 2.874 | 35 |
| **Playmaker** | High Scorer | 0.057 | 0.737 | 5.021 | 42 |
| **Playmaker** | Three-Point Shooter | 0.042 | 0.544 | 4.709 | 31 |
| **Playmaker, Three-Point Shooter** | High Scorer | 0.039 | 0.935 | 6.375 | 29 |
| **High Scorer, Three-Point Shooter** | Playmaker | 0.039 | 0.483 | 6.241 | 29 |
| **High Scorer, Playmaker** | Three-Point Shooter | 0.039 | 0.69 | 5.979 | 29 |
| **Playmaker** | High Scorer, Three-Point Shooter | 0.039 | 0.509 | 6.241 | 29 |
| **Three-Point Shooter** | High Scorer | 0.082 | 0.706 | 4.81 | 60 |
| **High Scorer** | Three-Point Shooter | 0.082 | 0.556 | 4.81 | 60 |

Some of the strongest rules identified included High Scorer ↔ Three-Point Shooter, Playmaker ↔ High Scorer, and (Playmaker + Three-Point Shooter) -> High Scorer. The rule (Playmaker + Three-Point Shooter) -> High Scorer had the highest confidence (93.5%) and the strongest lift (6.37), which indicates that almost every player with both attributes is highly likely to top a scorer. Additionally, while strong rebounders were not central in the Apriori algorithm, FP-Growth found that they had some association with scoring ability. Specifically, the rule (Strong Rebounder -> High Scorer) appeared, but with moderate confidence and lift (~2.87). Hence, some major players who dominate the boards are also elite scorers, and, yet, this is not a universal rule which means that most high scorers still tend to be perimeter players rather than dominant rebounders.

After applying the FP-Growth algorithm, the next logical step was to validate and interpret the discovered patterns by examining specific players who fit these rules. The filtering step was done to extract players who had the most highly identified combination of traits– Player, Three-Point Shooter, and High Scorer – since FP-Growth showed that these attributes frequently co-occurred. The below table organizes the players under this trait along key attributes relevant to scoring and playmaking (e.g. PTS, 3P%, AST, etc.) and is sorted by Point per Game (PTS) in descending order, meaning the highest scorers are listed first. For easier explainability, we have only showcased the top ten players in this filtered table.

|  | **Team** | **Pos** | **PTS** | **3P%** | **AST** | **TRB** |
| --- | --- | --- | --- | --- | --- | --- |
| **Luka Dončić** | DAL | PG | 33.9 | 0.382 | 9.8 | 9.2 |
| **Jalen Brunson** | NYK | PG | 28.7 | 0.401 | 6.7 | 3.6 |
| **Devin Booker** | PHO | PG | 27.1 | 0.364 | 6.9 | 4.5 |
| **De'Aaron Fox** | SAC | PG | 26.6 | 0.369 | 5.6 | 4.6 |
| **Donovan Mitchell** | CLE | SG | 26.6 | 0.368 | 6.1 | 5.1 |
| **Stephen Curry** | GSW | PG | 26.4 | 0.408 | 5.1 | 4.5 |
| **Anthony Edwards** | MIN | SG | 25.9 | 0.357 | 5.1 | 5.4 |
| **Tyrese Maxey** | PHI | PG | 25.9 | 0.373 | 6.2 | 3.7 |
| **LeBron James** | LAL | PF | 25.7 | 0.41 | 8.3 | 7.3 |
| **Trae Young** | ATL | PG | 25.7 | 0.373 | 10.8 | 2.8 |

As we can see, Luka Dončić, Stephen Curry, Damian Lillard, and LeBron James are among the first names, suggesting that the rule captures elite offensive players. Since the players are widely recognized as top scorers and playmakers, the table confirms that the FP-Growth rules are meaningful and not random. We also notice that while many Point Guards (PG) and Shooting Guards (SG) appear, there also some Forwards (PF/SF) like Lebron James, indicating that the pattern applies across positions.

* **Closed Association Rule Mining (CHARM-Approach)**

As seen above, FP-Growth outputs many rules, but we want the most crucial and non-redundant ones. FP-Growth provided us with many frequent patterns, but not all were meaningful in terms of new insights—many were subsets of larger itemsets, making them redundant. The CHARM algorithm helps by identifying closed itemsets, which are the largest itemsets that are not subsets of others. We learned about the CHARM algorithm through the following scholarly articles: [“Closed Itemset Mining and Non-redundant Association Rule Mining”](https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_66#:~:text=An%20(frequent)%20itemset%20is%20called,as%20sup(A%20%E2%87%92%20B)) and [“CHARM: An Efficient Algorithm for Closed Itemset Mining”](https://epubs.siam.org/doi/epdf/10.1137/1.9781611972726.27). One major fact that we learned from these two articles was that unlike traditional itemset mining approaches that rely on a bottom-up or top-down traversal, CHARM simultaneously explores the itemset space using an IT-tree (Itemset-Tidset Tree).

For our implementation of this algorithm, we broke it down into three basic steps: calculating the length of each itemset and storing these values in a new column, looping through all frequent itemsets and checking whether each itemset is a subset of any larger itemset encountered before, and returning only closed itemsets. This was a brute force approach to closed itemset mining because our dataset was easier to deal with, and we did not want to use the SPMF library because that was a Java-based library although it could have increased computational efficiency. Below is a table of the Closed Itemsets that were generated.

|  | Closed Itemsets |
| --- | --- |
| 0 | (High Scorer) |
| 1 | (Strong Rebounder) |
| 2 | (Playmaker) |
| 3 | (Three-Point Shooter) |
| 4 | (Strong Rebounder, High Scorer) |
| 5 | (Playmaker, High Scorer) |
| 6 | (Playmaker, Three-Point Shooter) |
| 7 | (Three-Point Shooter, Playmaker, Highscorer) |

As we can see from these results, both FP-Growth and the Closed Itemsets confirmed the strong connection between High Scorers and Three-Point Shooters. We also see that redundancy is reduced here because instead of listing both (Playmaker → High Scorer) and (Playmaker, Three-Point Shooter → High Scorer), it only retains the most informative sets. Therefore, we understand how this CHARM-like approach helped us fulfill a post-preprocessing step after frequent pattern mining, which made results meaningful.

* **Content Based Filtering (Cosine Similarity)**

We wanted to explore other algorithms and ideas that we have learned in class as we worked with NBA statistics. This code is a player recommendation system based on the similarity of their statistics. For example, to find players who are similar to Stephen Curry, it uses cosine similarity to compare players with Curry’s stats: points scored, 3-point percentage, assists, and rebounds. The player statistics are first standardized using StandardScaler. This normalizes the data so that each feature has a mean of 0 and a standard deviation of 1. After scaling, cosine similarity is computed between every pair of players. The cosine similarity value measures how similar two players' performances are, with 1 meaning they are identical in terms of their performance profile, 0 meaning no similarity, and values in between indicating varying degrees of similarity. The similarity values are then stored in a dataframe where each cell represents the similarity between two players. The recommend\_players() function then uses this matrix to find the most similar players to Stephen Curry by selecting the players with the highest cosine similarity scores. Below, the top 20 players are displayed.

| Players | Cosine Similarity |
| --- | --- |
| CJ McCollum | 0.997208 |
| Herbert Jones | 0.996115 |
| OG Anunoby | 0.991355 |
| Bogdan Bogdanovic | 0.989549 |
| Mikal Bridges | 0.985731 |
| Paul George | 0.980678 |
| Grayson Allen | 0.979539 |
| Tyler Herro | 0.978702 |
| Jalen Suggs | 0.978421 |
| Anfernee Simons | 0.977283 |
| Desmond Bane | 0.973279 |
| De’Anthony Melton | 0.970385 |
| Bojan Bogdanovic | 0.966640 |
| Brandon Miller | 0.965441 |
| Devin Vassell | 0.965112 |
| Cameron Johnson | 0.960826 |
| Donte DiVincenzo | 0.959157 |
| Jalen Green | 0.956773 |
| Buddy Hield | 0.956127 |
| Klay Thompson | 0.953825 |

**Conclusion**

Overall, implementing these multiple algorithms allowed us to gain more insight into how we can analyze and visualize data that is scraped from a website. It also helped us understand how each algorithm works best in situations depending on the user’s goal. For the Apriori algorithm and association rules, in class, we worked with a given dataset and used the libraries for the assignment. Since this was a project, we wanted to implement the algorithm manually and put our knowledge to the test. We were able to accurately generate our foundational association rules to work with. Next, by using FP-Growth we wanted to experiment beyond the abilities of what the Apriori algorithm could do. Compared to the previous algorithm, the FP Growth algorithm can handle large datasets and does not create an overwhelming amount of candidate sets. It also uses a different approach to create frequent itemsets by constructing a compact tree structure that helps it to mine frequent patterns. We compared the results with our Apriori algorithm to test its effectiveness and efficiency and found that the final results were very similar. Next, although FP-Growth reduced the amount of rules generated, we wanted to be able to identify the most crucial and non-redundant ones. The CHARM algorithm helped us identify closed itemsets by using an IT-tree. Its meaningful results allowed us to focus on the informative rules, giving us a post-preprocessing step after frequent pattern mining. Lastly, we wanted to explore working with an algorithm based on a concept we learned in class. By using Cosine Similarity, we built a content-based filtering algorithm that worked as a recommendation system to recommend similar players to Stephen Curry. After implementing all these algorithms, it was informative to see how each algorithm worked and how we could use the results to generate various visualizations. Also, it was a good way to explore the many ways a user can work with HTML data and learn its process.

**Code File:** [CSCI 185 Project Code](https://colab.research.google.com/drive/1Cma0oFIEPGMtxOB4OP6cAl4aoWdrccvn?usp=sharing)