Final Report

Abstract

Advanced, shooting and per possession statistics of every NBA player for the 2019-20 season was cleaned, scaled and then trained using various supervised classification models in order to predict the position of the player. Then heights were factored into the data to determine if the addition of player heights improved the performance of each model. Finally, unsupervised clustering techniques were performed on the data to find an intuitive grouping of the players.

Background

Statistics are playing a huge role in the realm of sports in the modern age, ranging from the NBA, to the NFL, and even to sports like tennis. Advanced stats have led to more teams using data models to choose which plays to run during given situations. The current landscape of basketball has also changed within the last 10 years, as players are more versatile and can play with a greater variety of skills. The league has centers winning 3-point contests, 7 foot points guards, and 2 way wings that can produce on offense and defense. With all of these new factors, one of the goals of this project was whether one can accurately predict the position of a given NBA player based on their individual player statistics over the course of a single season. The consensus idea is that in the past, about 2 decades ago, the roles of each position were more scripted and had defined skills. Moreover, with the augmented height range for each position in the modern era, the heights of each individual NBA player was also added to the dataset for positional classification. Finally, an unsupervised classification method was used in order to obtain a clustering of players that may or may not better represent the modern game of basketball. This project can be useful in many realms of the NBA. For instance, if a certain player does not fit the predictive model of certain tests, then that could entail that their skills are more suited for another position. This would be just another assistive tool for a coach to have at their disposal. The results of the unsupervised classification method could show in what category a certain team has depth or is deficient. This method could also be used on datasets that contain both modern and historical players to make data-driven player comparisons.

Dataset

The dataset that we used for position classification was from Kaggle (link: https://www.kaggle.com/datasets/nicklauskim/nba-per-game-stats-201920). The dataset includes five different files that contained stats for NBA players that played during the 2019-20 season, three of these files were used. One of these included player stats that were averaged per 100 possessions, another included data about shot location frequency and percentage made, and other included advanced statistics. We decided to use the per possession data instead of the per game and per 36 minutes data because it adjusts for playing time and pace of play. There were 651 rows in the data, one for each player, and 52 columns after dropping features. 21 features were dropped; most of these were duplicate features since 3 datasets were merged. For the rest of the dropped features, we decided that they were not useful for classifying a player's position. Missing values were filled with zero since most of these were from the 3pt% column (if a player didn't attempt any 3pt shots we thought it appropriate to give them a 0 here). All features were scaled to a mean of 0 and standard deviation of 1. Feature "importance" was found using the ExtraTreeClassifier from the sklearn module. Importance is defined as the mean decrease impurity across all trees in the model. Basically, the importance reflects the degree to which a feature can be used to separate the data into distinct groups. We found that there were not any features that had low importance compared to the rest so we kept them all except those that were dropped initially. Since we were not using linear regression, we didn't transform in any way after scaling. It is our understanding that clustering techniques and SVM don't assume normality. Players with values that did not conform to the standard 5 positions were dropped for the dataset (these players had hybrid positions such as "PF-C"; there were only a few of these players in the entire dataset). Players that did not play a minimum of 500 minutes were dropped from the data. This left a dataset of 373 unique player seasons. There were multiple players that appeared in the data 2 or more times due to being traded at some point during the season. We decided to keep these. Dimensionality reduction in the form of principal component analysis (PCA) using the sklearn module was performed on the dataset. Using SVM, we found that 14 principal components produced the highest accuracy score, so we decided to use 14 components on models where we used the PCA data. Player heights were added to the original dataset after running our models on the scaled and PCA datasets. Adding heights produced a dataset with

Methodology

A k-nearest neighbors (knn) classifier from the sklearn module was trained on the dataset. A 75:25 training/testing split was used. Values for k 2 through 19 were evaluated using 10 random splits for each to find the k-values that produced the highest accuracy. Additionally, both uniform and euclidean distance weights were tested. A support vector machine classifier from the sklearn module was trained on the dataset using a 75:25 training/testing split. Radial basis function kernel was found to produce the best accuracy. Precision and recall for each position was found for each training/testing split. Decision tree and random forest classifiers were used on the dataset with a 75:25 split. A minimum sample split of 12 produced the highest accuracy. All of these classifiers were again used on the principal component data with 14 principal components. Again, after heights were added to the original data, all of the models were tested again using k fold cross validation. A k means unsupervised clustering model was used on the pca dataset. An agglomerative hierarchical clustering model was used on the pca dataset. This model was then used with the fcluster class from scipy to produce an unsupervised clustering of players. PCA results were analyzed to give each component an intuitive label that described the set of skills that contributed most to that component. Mean values for each component were analyzed for each player cluster. Each player cluster was then given an intuitive label that described the type of players in each cluster.

Results

Scaled Dataset:

For the k-nearest neighbors classifier, it was found that a k-value of 11 produced the highest accuracy score of 0.57 across 10 splits. Values of 20 > k > 3 produced similar results. SVM produced an average accuracy score of 0.63 across 10 random splits. Decision tree produced an average accuracy of 0.52 using k-fold cross-validation (min_sample_split of 8 and criterion of 'gini'). The random forest classifier produced an accuracy score of 0.62 using k-fold cross-validation (min_samples = 10, criterion='gini').

PCA:

Models were evaluated using 14 principal components. K-nearest neighbors produced an accuracy of 0.57 across 10 random splits (k = 11). SVM produced an average accuracy of 0.64 across 10 random splits (kernel = 'rbf'). Decision tree had an accuracy of 0.46 using k-fold cross-validation (min_sample_split = 8). Random forest had an accuracy of 0.58 (min_sample_split = 4).

Scaled Dataset w/ Heights:

K-nearest neighbor had an average accuracy of 0.60 across 10 random splits (k = 12). SVM produced an average accuracy of 0.64 using k-fold cross-validation (kernel = 'rbf'). Decision tree had an accuracy of 0.61 using k-fold cross-validation (min_sample_split = 8). Random forest produced an accuracy of 0.69 using k-fold cross-validation (min_sample_split = 4).

Unsupervised clustering:

K Means produced a silhouette score of 0.31 with 2 clusters on the scaled dataset. We decided not to pursue this method further. Agglomerative hierarchical clustering had a cophenetic correlation of 0.60 on the scaled dataset (linkage = 'ward', distance = 'euclidean'). It also produced a cophenetic correlation of 0.60 on the PCA dataset (linkage = 'ward', distance = 'euclidean'). On the PCA dataset, agglomerative clustering produced 12 clusters when max distance was set to 21. The top 10 feature contributors by absolute value for the first 6 principal component are shown below:

Principal Component	Top 10 Feature Contributors
1	['TRB%' '# Dunks' 'ORB%' 'ORB' '0-3 Proportion' 'PER' 'WS/48' '2P Proportion' '3P Proportion' '3PAr']

2	['USG%' 'PTS' 'AST%' 'OBPM' 'AST' '2P Proportion Astd' 'TOV' 'VORP' 'FTA' 'PF']
3	['TS%' 'ORtg' 'TOV%' '2P Proportion' '3PAr' '3P Proportion' '3-10 Proportion' 'TOV' 'WS/48' 'OWS']
4	['STL' 'STL%' 'DBPM' 'DRtg' 'DWS' 'TOV%' '2PA' 'AST' 'PTS' 'VORP']
5	['DRB' 'DRB%' 'ORtg' 'TRB%' 'TS%' '3P Proportion Astd' 'DRtg' 'FTr' '%3PA Corner 3s' 'USG%']
6	['10-16 Proportion' '16-3P Proportion' '3-10 FG%' '10-16 FG%' 'FTA' '3PA' 'FTr' '3P Proportion Astd' '16-3P FG%' 'TOV']

Using these features and their associated values for each component (given by components_ attribute of the PCA class). We gave each principal component a name that we thought reflected the set of skills it was measuring. The first 6 principal components were named "Paint play", "Creation", "Inverse offensive efficiency", "Defensive efficiency", "Offensive/Defensive tendency", and "Inverse mid-range tendency" respectively. Using the mean values of these principal components that each cluster received, and what was known about the players in each cluster, we gave a name that we felt best described the player type in each cluster. Some of these were "Star players", "Traditional centers", "Knock-down shooters", "Stretch fours".

Future Work

If this project were to be continued in the future one of the aspects done in the future would definitely be the utilization of more types of models, working with more intricate data and variables (factoring in region, climate, system, etc). There are also models related only to finding outliers in a dataset and those could have been studied for the data cleaning process of the assignment, as the cleaning was primarily done through descriptive statistics. The use of neural networks for positional classification was the natural next step for the project. Investigating which types of architectures performed best would have been interesting. As for clustering, future work could include a bigger dataset encompassing multiple years, additional models besides agglomerative hierarchical clustering, use of more cutting edge stats like those offered through play-by-play data, and implementation for data-driven player comparisons between eras. Additional instances of future work also include factoring in data from different time periods in order to get a narrative timeline of when the league truly began to change in terms of the positional responsibilities of a player. Furthermore the three point shot has been quite revolutionary in terms of how teams run their offense and there is no real positional responsibility for that shot anymore. More analysis and studies should be performed on that category in order to determine the value of that statistic for salary purpose and positional responsibility. The rise in the importance of the three point shot is a recent product of the Steph Curry era, and therefore would be a fascinating future addition to the study because the role that category plays in the positional role in the league has not been written about at length in the media yet.

Reflection

This project was very interesting to work on because the dataset was easy to work with and familiar. The dataset and the goal of the project was a great start applying machine learning algorithms to predict certain aspects of a dataset. Some of the challenging parts of the project entailed the cleaning of data that had issues, especially when different datasets were merged together. Height data, a separate set, was merged into the original dataset in order to run tests on. This resulted in a lot of duplicate data points, as even some of the heights of the players were different on a year by year basis. The easier parts of the project were running the tests, as a lot of the algorithms were either worked on in class, or available online. Plotting the data was a pain and required constant reference to tutorials and documentation. We are pretty happy with how the results turned out for SVM and random forest in particular. We also found it really interesting how the unsupervised learning algorithm grouped the players. It was fun to use our basketball knowledge to make sense of all of this.

Citations

https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/

https://medium.com/hanman/the-evolution-of-nba-player-positions-using-unsupervised-clustering-to-uncover-functional-roles-a1d07089935c

https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

https://danvatterott.com/blog/2016/02/21/grouping-nba-players/

https://www.analyticsvidhya.com/blog/2021/01/a-quick-introduction-to-k-nearest-neighbor-knn-classification-using-python/

https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e