Ruslan Davtian, Howard Liu, Ryan Moore
Professor Stanchev
CSC 466
December 3, 2017

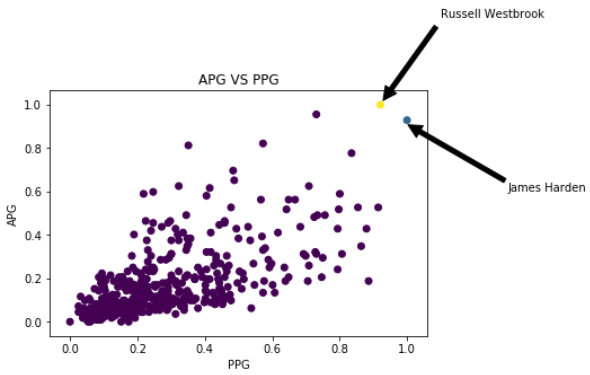
Final Project: K-Nearest Neighbors on NBA Data

For our group project, we decided to focus our algorithm on something we are all passionate about: basketball. We used the 2016-2017 NBA season and answered a few intriguing questions. Our algorithm goes as follows:

Our output includes a prediction of what position a player is, whether the player is a front court or back court player, and the MVP for this year's season. The output is intriguing because we can predict with moderately good accuracy what position a player is, which is basically one of the only classifiers a basketball player can have. These type of statistics and predictions can improve your understanding of how well basketball teams do, as well as your fantasy team. Also, the current selection of how the MVP (most valuable player) is determined is very tricky and subjective. Our prediction of MVP is straight forward and is hardly subjective. We won't know how accurate it is until the awards come out, but for now, if the MVP is Lebron James, we were correct.

The data we gathered is from an NBA statistical website and it contained 2016-2017 NBA players regular season offensive and defensive metrics. Also, we needed the current NBA season to predict the MVP at the end of the season based on who is closest in terms of metrics from last year's NBA MVP, Russell Westbrook. Since the range of all quantitative metrics have variability, we decided to standardize all these variables before doing anything else. To answer the first question of who is the closest to Russell Westbrook (Regular Season MVP), we needed to compute the Euclidean distance against him and every other player and sort in ascending

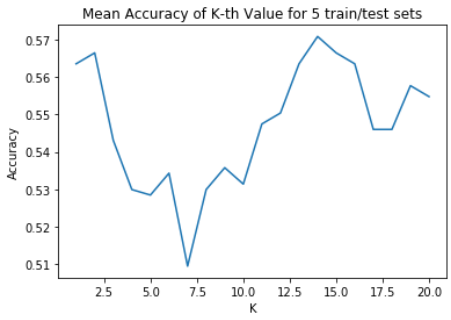
order. The smallest distance from Westbrook is James Harden, which makes sense because he took second place in the MVP race for the 2016-2017 season. The scatter plot below plots assists per game versus points per game which are the two main offensive metrics in basketball. It makes sense that Russell Westbrook and James Harden are very close to each other.



Our first question was simple, but we will now use the KNN algorithm to classify the five positions in the NBA. Typically, the five positions are PG, SG, SF, PF, and C but these positions are represented as 1,2,3,4,5. We will use the numbered positions to compute accuracy. Our data was split randomly into 70% train and 30% test. For each new instance of a player in the testing set, we calculate the distance from that player and every player in the training set and sort distances from smallest to largest. We take the closest k neighbors or distances and find the most frequent class to classify each player from the testing set.

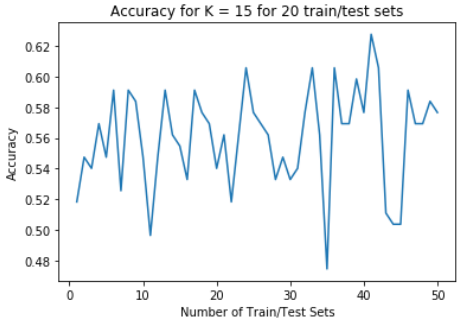
To find what value should we use for k, we considered k-values from 1 to 20 and iterated the algorithm for each k. We used a form of bootstrapping where iterated through the data and random sampled different train/test sets of 70% train and 30% test. Each iteration of the data sets formed different, unique train and test sets. Ideally, we would simulate this many times but because of heavy computation and run-time, we could only use 5 iterations of the data. However, these iterations through randomly selected train/test sets would give us a good estimate if some k-values performed much better than others. Also, because NBA players in today's generation

have similar skills regardless of position, we defined a successful classification if the predicted position number is within +- 1 of the actual position since adjacent positions are much too similar to each other.

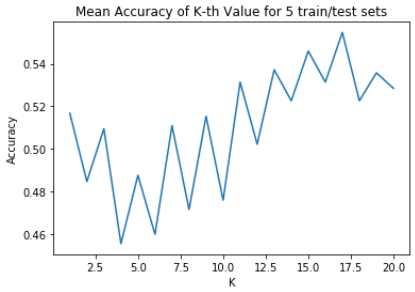


The chart above shows the mean accuracy of K-th value over five iterations of randomized train/test sets. What is interesting is that k = 1 produces some of the most accurate results even though accuracy only ranges from 51% to 57%. Also, K values between 14 and 16 seem to perform the best as well. We understand that these results are not very accurate but in today's NBA game, the positions are less specialized, and everyone seems to be able to have good shooting and offensive metrics regardless of position.

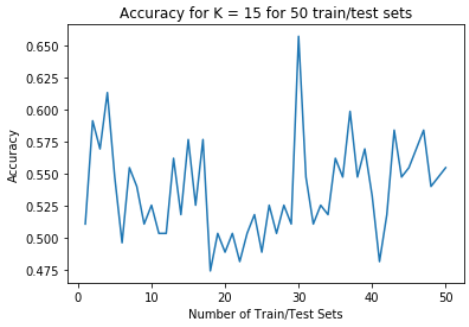
Since K = 15 had on average the most accurate results, we decided to iterate through the data fifty times and each time, have different train/test sets and run the algorithm look at the distribution of accuracies over the fifty iterations. The plot below shows that depending on how lucky the train/test sets were randomized, the accuracy ranges from as low as 0.48 to as high as 0.62. The average accuracy across fifty trials for k = 15 is 0.56.



To achieve better accuracy, we tried to decrease the number of classes from five to two We combined point guards and shooting guards into the frontcourt positions and small forwards, power forwards, and centers into backcourt positions. In basketball, the 1 and 2 positions can be grouped and the 3,4,5 positions can be made into another group separating the passers and scorers from the rebounders and blockers in traditional basketball. We repeated the same s analysis as before, but our mean accuracy numbers did not improve.



Over an iteration of five train/test sets for each k from 1 to 20, the highest mean accuracy percentage occurred around k = 15 or k = 17 so we again used fifty randomly selected train/tests sets and found the accuracy each time for k = 15.



Again, the accuracies have some variation, as low as 0.475 and as high as 0.650 over fifty different train/test sets. From our application of the KNN algorithm to this specific data, it is unclear which K is optimal as accuracies seem to have a random pattern between 0.5 and 0.6. This was expected because of today's style of basketball in the NBA where regardless of

position, everyone has similar skills that produce statistical metrics that make it difficult to distinguish positions.

Lastly, our last research question is to make a prediction of this year's NBA regular season MVP. We have data of about the first twenty games for each player in the 2017-2018 NBA season. Our method was to find a player from the current season with the closest distance between him and last year's MVP (Russell Westbrook) and this player will be our prediction to win the most valuable player award. Using Euclidian distance, we found LeBron James to be the closest in terms of distance to Russell Westbrook from last year. Interestingly, Russell Westbrook from this year is the second closest NBA player to Russell Westbrook of last year but our prediction for this year's MVP is LeBron James.

In conclusion, we tried to different class labels thinking that the smaller class labels of only frontcourt and backcourt would give use better accuracy, but both seemed on average to produce similar accuracies between 50% and 60%. If we had more time, we could find which metrics were most useful in classifying and limit to those instead of finding differences for all twenty metrics which made computations very expensive and time consuming. Also, we could have made our code more efficient and could iterate over many more train/test sets to get better estimates of the true accuracy rate for each k from 1 to 20. Lastly, we could also implement a better cross validation using a hold out testing set and iterating through each fold of the training set and computing the average accuracy rate. Overall, I believe this project gave us interesting results and experience with observing an application to the KNN algorithm.

Work Log:

- Initial Idea: Rus and Ryan
- What kind of data to use: Rus and Ryan
- Getting started: Howard and Rus
- Writing the code: Rus (60%) Ryan (20%) Howard (20%)
- Explanations: Rus, Howard, Ryan
- Visualizations: Ryan
- Organization: Rus, Ryan, Howard
- Data: https://www.nbastuffer.com/2016-2017-nba-regular-season-player-stats/

  https://www.nbastuffer.com/2017-2018-nba-player-stats