Predicting Basketball Positions using 2013-2014 NBA Data

Dhru Patel, Sumit Chandra, Austin Joiner, Chris Leonard

May 7th, 2019

1 GitHub

Github Link: https://github.com/sumitschandra/COMP562-Final-Project/blob/master/ Basketball(_)FinalProject.ipynb

2 Abstract

The goal of this project is to find the best method available to model an NBA player's position based on the parameters present in our data set. Through our research, we found that in each of our models the variables birth(_)place, birth date, college, and *insert variables not used* brought down the accuracy of the projections in each method, so we decided to exclude them from the data set. To predict the player's position, we used a combination of support vector machines, random forest classifiers, and neural networking. After maximizing the prediction rate of each method, we found that the neural network model was the most consistently accurate, as well as being the most accurate method overall.

3 Problem

Our goal was to create a model that predicted a players position based on their stats from the 2014-2015 NBA season. Our data contained the season statistics of every active player in the NBA during the season. It included their minutes and games played, their number of made and missed field goals from 2 point and 3 point range, the amount of time they have been in the NBA, as well as other statistics. Our motivation in creating the model was to find out how well intuitive predictors (like height and weight) could predict a players position if the model also contained shooting statistics and other potentially useful predictors. For example, we wanted to see if our model could correctly classify players like Kevin Durant, who is tall enough to be a center, but who is better at shooting three pointers and lighter than most centers. To best predict a player's position, we used several algorithms and tested the effectiveness of each of them: support vector machines, random forest classifiers, and neural networking.

4 Survey-Related

After we determined what the problem we were trying to solve was, we needed to go about finding good data to use for our model. We decided to use a data set from the 2014-2015 NBA season, which included over 20 different features for individual players. Features included statistics like field goals attempted and made, blocks, and steals as well as other pieces of information like height, weight, and collage. While most of the statistics could be used as features in our data, some had no effect on the model and had to be dropped for the data set when running the models.

After we had processed the data-set, we researched different algorithms to use to predict the position of each player. For our purpose, we decided that the right approach was to use supervised learning, where we could give the model a select amount of feature variables and it predict a single response variable. In addition, we decided to use models that act as classifiers, which lead us to support vector machines, random forest classifiers, and some neural network algorithms.

5 Approach

We used multiple methods in approaching a solution to model the data. We measured the levels of accuracy on Support Vector Machine classifiers, random forest classifier, and a neural network. We first imported all the player statistics from the 2013-2014 season and separated the data set into response and feature variables. This allowed us to use standard/feature scaling to standardize our variables and get optimized results. We used a Random forest classifier, which lets the algorithm average out the impact of several decision trees and improve overall prediction. We then form the confusion and normalization matrices as metrics to evaluate performance.

We then used support vector machine classifiers to best predict a player's position. We were able to get a better classification of our data by using this technique because it was able to differentiate the data points with a ideal hyper plane. Additionally, the support vector machine algorithm uses subsets of training points/support vectors, which makes it more memory efficient. The SVM classifier solves the following primal problem where C ¿ 0 is the upper bound, and Q is the kernel. The algorithm maps the training vectors in some dimensional space which is mapped by phi. (Following Equation From Smola, Schölkopf)

$$\begin{aligned} \min_{w,b,\zeta,\zeta^*} \frac{1}{2}w^T w + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*) & \\ \text{subject to } y_i - w^T\phi(x_i) - b \le \varepsilon + \zeta_i, & \\ w^T\phi(x_i) + b - y_i \le \varepsilon + \zeta_i^*, & \\ \zeta_i, \zeta_i^* \ge 0, i = 1, ..., n & \end{aligned} \tag{1}$$

We also used a neural network to model our data and test its accuracy. This approach allows the program to reuse the solution of a previous call and redo this process. The Multi-layer Perceptron classifier in the library supported would help train the data points

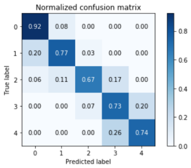


Figure 1: CM for All Stats as Features

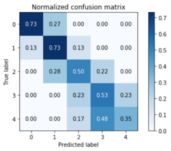
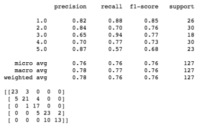


Figure 2: CM Features: Height and Weight



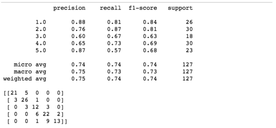
```
              precision    recall  f1-score   support

         1.0       0.82      0.88      0.85        26
         2.0       0.84      0.70      0.76        30
         3.0       0.65      0.94      0.77        18
         4.0       0.70      0.77      0.73        30
         5.0       0.87      0.57      0.68        23

   micro avg       0.76      0.76      0.76       127
   macro avg       0.78      0.77      0.76       127
weighted avg       0.78      0.76      0.76       127

[[23  3  0  0  0]
 [ 5 21  4  0  0]
 [ 0  1 17  0  0]
 [ 0  0  5 23  2]
 [ 0  0  0 10 13]]
```

Figure 3: Metrics for RFC: All Stats



```
              precision    recall  f1-score   support

         1.0       0.88      0.81      0.84        26
         2.0       0.76      0.87      0.81        30
         3.0       0.60      0.67      0.63        18
         4.0       0.65      0.73      0.69        30
         5.0       0.87      0.57      0.68        23

   micro avg       0.74      0.74      0.74       127
   macro avg       0.75      0.73      0.73       127
weighted avg       0.75      0.74      0.74       127

[[21  5  0  0  0]
 [ 3 26  1  0  0]
 [ 0  3 12  3  0]
 [ 0  0  6 22  2]
 [ 0  0  1  9 13]]
```

Figure 4: Metrics for RFC: Height and Weight

to increase accuracy. Because of this efficient approach, the neural network had the largest accuracy of all our models.

6 Results

Through running the NBA statistics through our different models we were able to achieve an accuracy of 78(%) with the random forest classifier, an accuracy of 74(%) with the neural network, and an accuracy of 76(%) with the SVM. Another interesting result to look at is the recall values of the different models with the random forest classifier.... This shows that we were able to achieve some degree of success in predicting NBA positions based on player data. The result of this project is a decent and efficient predictor of position and will more than likely give you the right position than not. Something interesting when looking at the results of the data is the difference in accuracy between positions. For point guard we were able to achieve an accuracy of 94(%) while for small forward we were able to achieve an accuracy of 67(%). Point guards were easy to predict because their stats were more close to the average while small forwards were hard to predict with their weights. Using SVM, random forest classifiers, and a neural network we made the algorithm much more effective than using linear regression or a decision tree, which is what we first looked at. Some things that were holding our accuracy from improving was using data that didn't work well in predicting their position, that we eventually left out, like their date of birth. We also looked at prediction with just using a couple stats like height and weight. This made the accuracy of random forest classifiers go down to 60(%), the accuracy of the SVM go down to 72(%), and the accuracy of the neural network stays at 74(%). As we can see, with different data points we were able to achieve different levels of accuracy with different models, which means that different stats work better with with the SVM vs with the random forest classifiers vs with the neural network. This shows us that if we were trying to use different sets of data, the better solution would have to be determined based on what data you are using.

7 Impact

With the NBA becoming a more increasingly offensive game (less defense, players shooting more threes, small ball lineups), it has become tougher to classify players positions' based on their season statistics. As mentioned before, players like Kevin Durant are taller than other centers, but scores and shoots at a much higher rate. Most teams today will start a game with a center but slowly move towards playing with a smaller lineup, resulting in opposing teams struggling to find the right match ups. By being able to predict a players true position based on their statistics, coaches can better find lineups on their own team to match up against their opponents through. For example, Kevin Durant is classified as a small forward for the Warriors, but eventually when they move to playing in a smaller lineup he plays the Power Forward or the Center position. Logically a center on the opponents team would guard whoever is playing the center, in this case Kevin Durant. But because he is lighter, faster, and a better shooter than most centers, making this a tough match up for the opposing team and coach. This prediction model would help the coach by classifying the position Kevin Durant truly plays, even though he is on the court as a "center" and switch to his own small ball lineup, helping with their defensive mismatches.

Additionally, this predictive model can help GM's and coaches when scouting college prospects and help with decision making during the NBA drafts. In college a player might have the role of a shooting guard because their team had a smaller point guard, but point guards of small size in college rarely make it to the league. In such a case, a GM or coach might wonder if that particular shooting guard could play point guard for their team. They could match their college prospect with a player in the current NBA with similar statistics and see what position they play. As mentioned before, the NBA has moved to much smaller lineups, so this model gives teams a better idea of what a player's potential is and whether they have to flexibility to transition to another position. This assists GM's on draft day when their top players get taken off the board and are forced to make quick decisions on which player they want to draft.

References

[1] Alex J. Smola, Bernhard Schölkopf *A Tutorial on Support Vector Regression.* Statistics and Computing archive Volume 14 Issue 3, August 2004, p. 199-222.

[2] Omri Goldstein *NBA Players Stats - 2014-2015 Points, Assists, Height, Weight and other personal details and stats.* Kaggle, 2015.

[3] Scikit-Learn Tutorial — Machine Learning With Scikit-Learn — Sklearn — Python Tutorial — Simplilearn
`https://www.youtube.com/watch?v=0Lt9w-BxKFQ`

[4] 3.3. Model evaluation: quantifying the quality of predictions
`https://scikit-learn.org/stable/modules/model_evaluation`

[5] 1.4. Support Vector Machines
`https://scikit-learn.org/stable/modules/svm.html`

[6] sklearn.metrics.Precision Recall Fscore Support
https://scikitlearn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html.