Jaleel Attah

Predicting the Outcome of a Shot

12/5/2020



1. **Introduction**

Basketball's history dates to December 1891 where the first game of basketball was played at a YMCA with its earliest rules written by Dr. James Naismith. Since then, the game has been revolutionized many times, with the National Basketball Association (NBA) being the most prestigious league. With the growth of the league came deep diving of box score analytics to draft better, make their teams shoot efficiently, and focus on the core strength of a team to win the NBA championship year in and year out. Like many professions, there are arguments between fans and players debating who the best player is. With the abundance of data, many can come to their conclusions using different ratings.

One of the most prominent figures of basketball all over the world was Kobe Bryant. Among many, Kobe was the player who got people of all ages to tune into the sport and watch how he impacted the game while admiring his basketball expertise.

Unlike typical analysis of rating players and their statistics, I plan to use the shot of Kobe Bryant throughout his 20-year career to predict the shot-making based on features of a shot after training a few machine learning algorithms and comparing the prediction accuracy of each one. I will be using Decision Tree, Perceptron, Stochastic Gradient Descent, Support Vector, and Random Forest classification algorithms to compare which algorithm would yield better results.

Before the analysis, making a scatterplot will help us visualize the dataset.

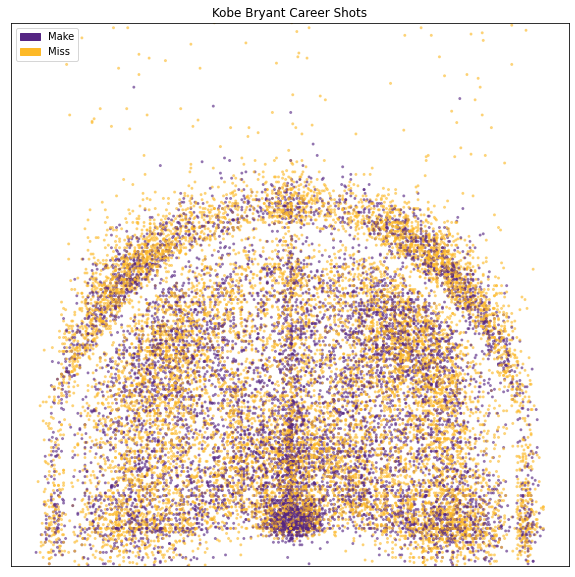


Figure 1: Shots taken in 20-year career

1. **Data Mining Task**

The dataset was acquired through a Kaggle competition named "Kobe Bryant Shot Selection."

The plan is to explore the dataset's features and clean the dataset, preparing it to split and run and test the different models. At the time of loading in the data, we had 30,697 different shots with 25 features, and after cleaning the dataset, we end up with 25,697 total shots and 16 features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| old dataset | |  | new dataset | |
| index | column |  | index | column |
| 0 | action\_type |  | 0 | action\_type |
| 1 | combined\_shot\_type |  | 1 | combined\_shot\_type |
| 2 | game\_event\_id |  | 2 | loc\_x |
| 3 | game\_id |  | 3 | loc\_y |
| 4 | lat |  | 4 | period |
| 5 | loc\_x |  | 5 | playoffs |
| 6 | loc\_y |  | 6 | season |
| 7 | lon |  | 7 | seconds\_remaining |
| 8 | minutes\_remaining |  | 8 | shot\_distance |
| 9 | period |  | 9 | shot\_made\_flag |
| 10 | playoffs |  | 10 | shot\_type |
| 11 | season |  | 11 | shot\_zone\_area |
| 12 | seconds\_remaining |  | 12 | shot\_zone\_basic |
| 13 | shot\_distance |  | 13 | shot\_zone\_range |
| 14 | shot\_made\_flag |  | 14 | age\_in\_days |
| 15 | shot\_type |  | 15 | game\_type |
| 16 | shot\_zone\_area |  |  |  |
| 17 | shot\_zone\_basic |  |  |  |
| 18 | shot\_zone\_range |  |  |  |
| 19 | team\_id |  |  |  |
| 20 | team\_name |  |  |  |
| 21 | game\_date |  |  |  |
| 22 | matchup |  |  |  |
| 23 | opponent |  |  |  |
| 24 | shot\_id |  |  |  |

Figure : features of the dataset

The datasets will be split into two, creating the training set and testing set. The labels of each dataset will then be extracted (shot\_made\_flag) into their feature list.

The datasets will be standardized using the training set as the primary model; then, I apply the same standardization on the testing set separately. Once completed, the models will then be trained on the training set.

1. **Technical Approach**

During the cleaning process, the first step was to drop all NA columns in the dataset since they would not help train the algorithm or determine if the model's prediction would be correct or not since the tuple did not have a shot\_made\_flag value.

The next step was to drop columns that have no importance for the features of a shot. The details of what was dropped can be reviewed in section II.

Some of the existing columns were renamed or combined to decrease the number of features to avoid overfitting.

* matchup was cleaned and renamed to game\_type : {away,home}
* game\_date was changed to become age\_in\_days using the date of the game and Kobe Bryant's birthdate
* seconds\_remaining was a merge of minutes\_remaining and seconds\_remaining

The dataset was then split into a training set (80%) and a testing set (20%) and prepared for standardization and dummy-fication of categorical features.

* loc\_x and loc\_y on the training set were standardized using StandardScaler() from the sklearn package
* loc\_x and loc\_y was then standardized on the testing set using the previously made StandardScaler() to ensure that we would be using the same standardizing logic so that the algorithm makes correct assumptions on the same features
* action\_type, combined\_shot\_type, season, shot\_type, shot\_zone\_area, shot\_zone\_basic, shot\_zone\_range, and game\_type on the training dataset were then dummified on both the training dataset and testing dataset separately
* using the setdiffld() function, we then ensure that both datasets have the same dummified columns so that I would not have mismatched columns when training and predicting with the models

Once all the preprocessing was completed, each of the models was then trained using the training set. Then we made predictions using the models on the testing dataset.

1. **Evaluation Methodology**

For every model, I used a training set of 20,557 tuples. I ran many iterations (1 iteration to 5000 iterations) except for Random Forest, where I used the number of tree classifiers (1 tree to 5000 trees) instead.

On each iteration or number of tree classifiers, I recorded the models' prediction accuracy on the testing set along with the time complexity of each algorithm.

1. **Results and Discussion**

Chart, line chart

Description automatically generated

Figure : Plot of algorithm accuracy

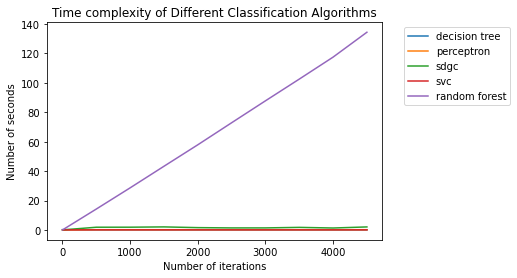
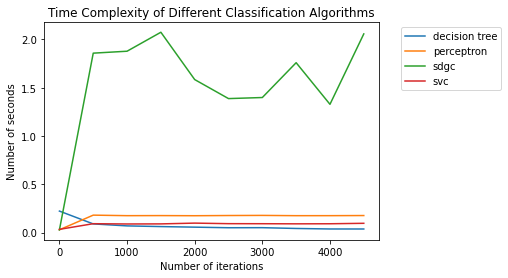
After running each algorithm many times, we conclude that SDGC was probably the worst classifying algorithm for this problem. SVC and the Decision Tree managed to give us the best results, with SVC consistently giving the better prediction after many iterations.

|  |  |
| --- | --- |
| Algorithm | Prediction Accuracy |
| Decision Tree | 67.59% |
| Perceptron | 55.89% |
| SDGC | 56.56% |
| SVC | 67.61% |
| Random Forest | 65.58% |

Figure : Prediction accuracy of algorithms

The results tell us that with the number of iterations that the Support Vector Classification gave us the best prediction accuracy at 67.61% at its peak.

While running each of these algorithms, I thought it was essential to consider the number of seconds each algorithm took for each iteration or tree count in the Random Forest.



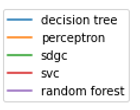


Figure 5: Plot of time complexity of algorithms

From the results, we see that as the number of tree increase, the time the Random Forest algorithm takes also increases making it inefficient to use for this problem. The most efficient algorithms turn out to be the decision tree and the SVC algorithm. Both the Decision Tree and the SVC give at least 17%+ more of an accurate prediction than random guessing making it worthwhile to use both methods to predict whether a shot is made or missed.

1. **Lessons Learned**

Trying to predict a shot in the NBA is very difficult to do. There are many variables involved with trying to make a shot that our dataset could not portray. Things like a defender being in the way of a player, how close they are, fatigue, health, mental state, injuries, and many others may affect a player's shot-making. With our algorithm, we managed to get a solid 67~% accuracy at our peak without accounting for those variables that we do not have access to.

An idea that could improve the model results would be adding to the data and finding the variables that are missing (listed earlier), and see how it affects the prediction of the models. The only potential issue that may come from this is adding more features that could lead to overfitting.

Another idea will be to see if the model is being trained on a balanced amount of made and missed shots and ensure that the training set contains at least one of every categorical feature when dummified.

1. **References**
   * Dataset -- <https://www.kaggle.com/c/kobe-bryant-shot-selection>
   * History of Basketball -- <https://www.basketballforcoaches.com/basketball-history/>
   * Set difference for dummy-fication -- <https://medium.com/@rodwan.bakkar/pandas-set-difference-fde7f4381b53>