SENG 474 Midterm Report

NBA Shot Expected Point Value Prediction

Michael Reiter
Faculty of Engineering
University of Victoria
Victoria, Canada
mreiter@uvic.ca

Spencer Vatrt-Watts
Faculty of Engineering
University of Victoria
Victoria, Canada
asvw@uvic.ca

Erik Reppel
Faculty of Engineering
University of Victoria
Victoria, Canada
ereppeld@uvic.ca

Chinyere Ibelegbu
Faculty of Engineering
University of Victoria
Victoria, Canada
chinyereibelegbu@uvic.ca

***Abstract*—This project intends to predict the expected point value (0 - 3) of a basketball shot in an NBA game given known features. Data of shots taken during the 2014-2015 NBA season is sourced from Kaggle and includes 120,000+ data points. The group scraped further data to provide context of the player taking the shot. The data was normalized and rescaled where necessary to produce better predictions. For each model tested, the training set was shuffled and models were retrained 100 times to demonstrate training features and predicted label correlation. Multilayered perceptrons were found to be the more accurate model tested so far. The group will continue to tune parameters and features to produce a most optimal model.**

Keywords—software engineering; data mining; machine learning; basketball;

I. Project Background

This project intends to predict the expected point value (0 - 3) of a given a basketball shot in an NBA game. This metric allows for a clear comparison between shot classes, taking into account the quality of the shot. The dataset does not include free throws because they are far more straightforward to predict and depend on fewer variables. Since shots can be worth either 2 or 3 points, the prediction will be a value in the continuous interval between 0 and 3 inclusive. We believe it will be interesting to determine which features have the greatest effect on predicting shot quality.

II. Data Sourcing and Details

Data of shots taken during the 2014-2015 NBA season is sourced from Kaggle [1] and includes 120,000+ data points. The data contains the following features: shot number, period, game clock, shot clock, dribbles, touch time, shot distance, points type, closest defending player, game location, and shooting player id.

Shot number is the nth shot a player has taken in a game. Some players improve over the course of a game, while others perform worse as the game progresses. Period is the quarter of the game (1, 2, 3 or 4). This could be of interest as some players are considered “clutch”, meaning they shoot well very late in the game. Game clock indicates how much time remains in a given period. Shot clock is a 24 second countdown that resets whenever possession of the ball changes teams. Dribbles indicates the number of times the player bounced the ball before shooting. This implies movement around the court. Touch time indicates the time that has elapsed since the player took possession of the ball. Shot distance is the distance from the player shooting to the hoop. Nearer shots have a much higher probability of being made, but are limited to 2 points, while shots taken from beyond the three point line are less likely to go in, but are worth 3 points. Points type indicates whether the shot is a 2 pointer or a 3 pointer. This number sets the upper bound on expected value. Closest defending player measures the distance from the shooter to the nearest defender. Clearly a small value will decrease a shot’s expected point value since it is likely to be blocked. Game location specifies whether the game is played at home or away. Many people believe a home court advantage exists. We do not believe this will have a significant effect on an individual shot, although it could affect shots late in the game due to crowd pressure. Shooting player id allows us to include player specific context into predictions using supplementary player data.

Additional data was scraped from the NBA stats REST API [2]. It includes features such as player id, team id, age, player height, player weight, field goal percentage, and three point percentage. These further details about players could allow for more in depth shot analysis based on the season’s trends for a given shooter. Since the player data we scraped and the original Kaggle dataset were both sourced from stats.nba.com, the player id attributes match without any manipulation.

The dataset does not consider free throws. 73.5% of shots are 2 pointers, while the remaining 26.5% are 3 pointers. Of the 2 pointers, 48.8% are made. Of the 3 pointers, 35.2% are made.

III. Data Manipulation

Spencer trained various models using numerous subsets of these features, but so far no final feature set has been selected. It is clear, however, that some features appear to have a stronger correlation with shot quality than others.

Shots with similar features could vary from player to player. For example, some players rarely shoot three pointers, so even if they had a high distance to the nearest defender, the expected point value would still be low. Field goal percentage and three point percentage are advanced features that are especially of interest, since they indicate a player’s shot quality over the season. We hypothesize this information could lead to a more verbose model that produces a better prediction based on player context.

One data related concern the group has addressed so far is greatly varying data points. For example, FG% is a value between 0 and 1, but for a feature like number of dribbles, the player may have dribbled around the court for a while leading

to a significantly higher number. We have used data normalization to improve our data and combat the large variance between feature values. A simple feature rescaling formula (1) has been applied to certain attributes in order to constrain the data range between 0 and 1.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

The group found that rescaling the features improved prediction accuracy between 5 to 10% on average when compared to non-normalized data on the same model.

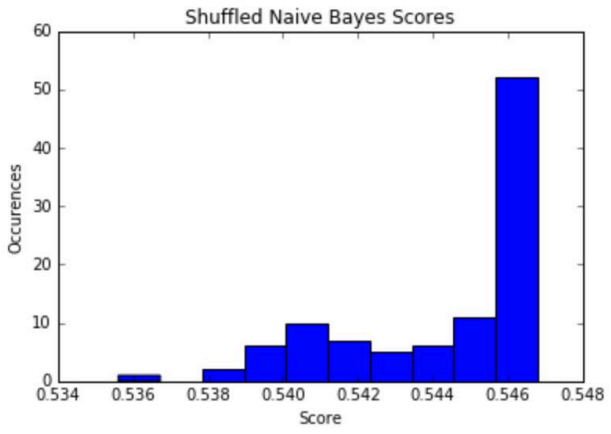
IV. Models Explored

So far, the group trained four different models and evaluated their performance. Each model was fit and scored. The training labels were then randomly shuffled, and the models were refit and scored 100 times. The shuffled scores are plotted as histograms to demonstrate the correlation between the training data and the expected point value. One would expect the unshuffled score to be clearly higher, but this was not always the case.

A. Naive Bayes

Michael trained and compared a Gaussian and Bernoulli Naive Bayes classifier [3]. After tuning some parameters, the Gaussian model scored a modest 0.46. Meanwhile the Bernoulli model scored around 0.53. This result was unexpected since Bernoulli assumes all features to be binary, which is not an accurate reflection of the dataset. Neither model was exceptionally accurate, and more research will need to be conducted before drawing any conclusions.

Fig. 2 below is a histogram plotting 100 scores for a Gaussian Naive Bayes classifier trained on shuffled data. For unclear reasons, the unshuffled score was lower than shuffled scores at 0.46. The shuffled scores ranged in the low 0.50s. Most shuffled models scored around 0.546.



2. Shuffled Naive Bayes Scores

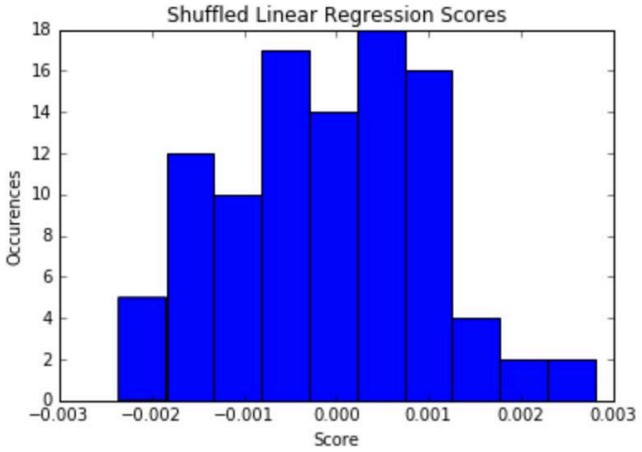
B. Linear Regression

Michael also trained a linear regression model. Even after normalizing the dataset, the model scored only a paltry 0.03 to 0.04. This is likely because linear regression is producing a continuous expected point value, rather than 0, 2 or 3. In sklearn, LinearRegression's score function returns the coefficient of determination R^2 (3) of the prediction [4].

$$R^2 = 1 - \frac{sum\left(\left(y_{true} - y_{pred}\right)^2\right)}{sum\left(\left(y_{true} - mean\left(y_{true}\right)\right)^2\right)} \quad (3)$$

Since the variance is high, the score is low. It is even possible for the score to be negative if the predictions are especially inaccurate.

Fig. 4 below is a histogram plotting 100 scores for a Linear Regression classifier trained on shuffled data. As expected, the unshuffled score was markedly higher than shuffled scores at 0.035. The shuffled scores were close to zero, sometimes even negative.

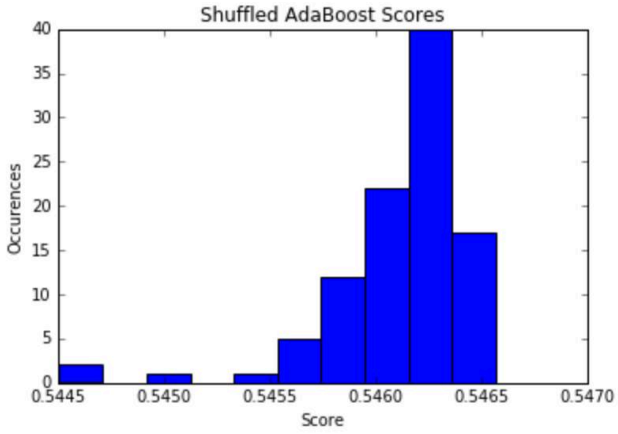


4. Shuffled Linear Regression Scores

C. AdaBoost

Spencer experimented using the AdaBoost classifier. He expected to receive strong results since AdaBoost combines many weak learning algorithms to create a final boosted classifier [5]. Since AdaBoost is sensitive to outliers in the dataset, the data was normalized before training. The best score he was able to achieve was 0.5463.

Fig. 5 below is a histogram plotting 100 scores for an AdaBoost classifier trained on shuffled data. The score of Adaboost was in the top 5% of scores when compared to random.

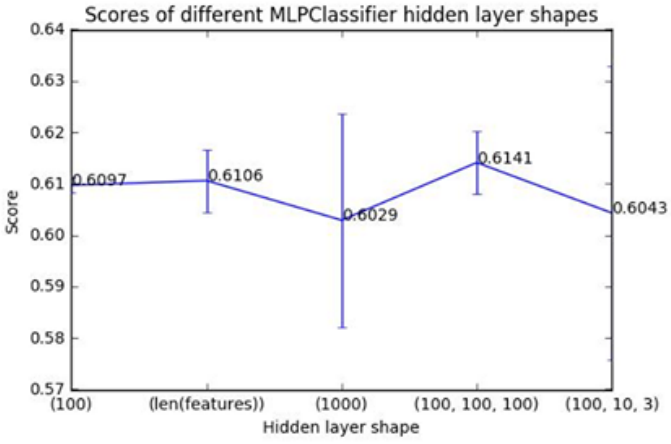


5. Shuffled AdaBoost Scores

D. *Multilayered Perceptron*

Erik explored neural networks in an effort to improve accuracy. Using a multilayered perceptron [6], he scored the highest of any model the group has tested yet, 61%. We hypothesize that the hidden layers allow more room for error to account for edge cases. In other words, more parameters to tune means outliers in the dataset have a smaller negative effect on accuracy.

Fig. 6 below plots scores of sklearn's Multi Layer Perceptron Classifier using logistic sigmoid as an activation function and tuning the shape of the hidden layers. The x axis shows a tuple where the ith item represents the number of neurons for the ith layer.



6. Multilayered Perceptron layer shapes vs. scores

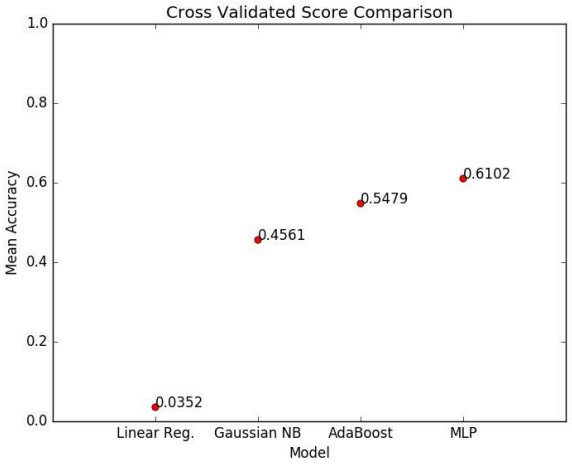
So far, it appears that (100, 100, 100) is the most accurate layer shape. We believe that (len(features), len(features), len(features)) may provide an even better result and shorter training time, but further testing is required.

V. Related Work

Basketball statistical analysis is a highly monetizable area of research, so there are a number of websites and papers devoted to it. There is also the annual Sloan conference that is dedicated to sports analytics [7]. In regard to our problem domain, predicting expected shot value, there is very little work that has been done aside from people on Kaggle trying things with the dataset. There is Yisong Yue's fascinating work [8] on predicting how likely a player is to pass or shoot based on where they are in relation to the hoop and the other players on the court. However, this is a different problem. Yue's work deals with how likely a shot is to occur, rather than the expected value of points from a given shot.

VI. Conclusions Drawn

To perform cross validation, the group randomly shuffled and then split the testing dataset. Afterwards, the models were trained and tested on different parts of the random split. After repeating this process five times, the mean accuracy of each model was calculated. These results are visualized in the accompanying graph, with each model plotted against its mean accuracy.



7. Comparing accuracy of tested models

In the original project proposal, the group postulated that the most accurate scores would result from a linear regression model trained using gradient descent. Conversely, it is now clear that linear regression is significantly less accurate than other models tested.

Additionally, in the proposal the group expected that using regression to produce a real number would give the best results. However, comparing the linear regression model to other classifier models shows that computing the expected value of a classification rather than simply using the real number produced by a regression may provide superior accuracy.

VII. Next Steps

After testing various models, the multilayered perceptron is emerging as the most promising model. The group will continue to optimize model, tuning parameters and features in an effort to produce more accurate predictions. These parameters include trying other activation functions such as relu, and tanh, other hidden layer shapes, and other optimizers such as SGD.

The group will also start trying to incorporate more advanced features such as player field goal percentage and player three point percentage, which could have a significant positive impact on results.

The group may also write a function that takes the results of the classifier (a vector representing the probability of each class) and multiplies each probability by the class value in order to give us an expected value for a given shot.

The group is still on track for the original estimated timeline. Remaining tasks are as follows.

Tune features and parameters for tested algorithms - November 14

Decide on final algorithm and perform any necessary tweaks and implementations - November 21

Finalize and submit report - December 2

References

1. "NBA shot logs | Kaggle", Kaggle.com, 2016. [Online]. Available: https://www.kaggle.com/edwardyun/d/dansbecker/nba-shot-logs/the-best-and-worst-defenders. [Accessed: 03-Nov-2016].
2. "NBA.com/Stats", Stats.nba.com, 2016. [Online]. Available: http://stats.nba.com. [Accessed: 03-Nov-2016].
3. "sklearn.naive_bayes.GaussianNB — scikit-learn 0.18 documentation", Scikit-learn.org, 2016. [Online]. Available: /react-text http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html react-text: 2015 . [Accessed: 05-Nov-2016].
4. "sklearn.linear_model.LinearRegression — scikit-learn 0.18 documentation", Scikit-learn.org, 2016. [Online]. Available: /react-text http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html react-text: 2022 . [Accessed: 05- Nov- 2016].
5. "sklearn.ensemble.AdaBoostClassifier — scikit-learn 0.18 documentation", Scikit-learn.org, 2016. [Online]. Available: /react-text http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html react-text: 2029 . [Accessed: 05- Nov- 2016].
6. "sklearn.neural_network.MLPClassifier — scikit-learn 0.18 documentation", Scikit-learn.org, 2016. [Online]. Available: /react-text http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html react-text: 2036 . [Accessed: 05- Nov- 2016].
7. "MIT Sloan Sports Analytics Conference", MIT Sloan Analytics Conference, 2016. [Online]. Available: /react-text http://www.sloansportsconference.com react-text: 4822 . [Accessed: 06- Nov- 2016].
8. Y. Yue, P. Lucey, P. Carr, A. Bialkowski and I. Matthews, "Basketball Pass & Shot Prediction", Projects.yisongyue.com, 2016. [Online]. Available: /react-text http://projects.yisongyue.com/bballpredict/ react-text: 4829 . [Accessed: 06- Nov- 2016].