Machine Learning to Predict NBA Point Spreads

Ryan Won
Computer Engineering
Drexel University
rhw37@drexel.edu

***Abstract*—The goal of this project is to analyze which statistics of NBA games contribute the most to point spreads, and to create a model that will accurately predict the point spread of NBA games using machine learning techniques in conjunction with a combination of cumulative data over a season and specific metrics over the last few games played.**

I. Background

A. Problem Interest

With the recent lifting of the federal ban on sports betting in 2018, the popularity of sports betting has been growing quickly. Consequently, betting data for sports has slowly become more accessible, enabling more projects related to betting data and sports. This project focuses on predicting the point spread of an NBA game, utilizing spread data that has only recently become easily and publicly accessible.

B. Mathematical Background

The point spread prediction of an NBA game is a prediction of the outcome of the score of a game. The spread is given as a positive or negative number for a specific team, which represents the 'handicap' required for the team to equalize the points. For example, image the following scenario, where the spread for the team is given in the last column.

Location	**Team Name**	**Point Spread**
Home	Philadelphia 76ers	-7
Away	Golden State Warriors	+7

The spread for the home team, the Philadelphia 76ers, is -7. This means the team is expected to win by 7 points, or that their score would need to be subtracted by 7 to be equal to that of the opponents score. From the perspective of the away team, the Golden State Warriors, their spread is +7. This mean this team is expected to lose by 7 points, or that their score needs to have 7 points added onto it to be equal to that of the opponents score.

When betting on spreads, you need to bet for one side of the point spread. Betting to cover the spread means that you are betting on the favorite team to have more points than the away team despite the handicap. In our example, this means that you would expect the 76ers to win by more than 7 points. On the other side, you have betting against the spread. This means that with the handicap, you expect the underdog to have more points. This would happen if the 76ers won by less than 7 points, or if the Warriors won the game by any margin. In the case that the final result matches the point spread, which in our case would be the 76ers winning by 7 points, it is called a push. This results in no winner, with all bets returned. Spreads are often given in 0.5 increments, such as -5.5 or 3.5, to avoid pushes.

To measure the accuracy of predicted point spreads, the average deviation from actual spread is measured. To calculate this, you find the distance from your predicted spread to the actual spread. To reference the above example, if the model predicted a spread of -7 for the 76ers, and the actual result had the 76ers winning by 2, which would be a spread of -2, then our model has a deviation of 5 points for that specific game. The deviation of each game is average to produce a final average deviation score, where a lower number indicates a model that produces predictions that are fairly close to actual results.

C. Machine Learning Models

The two regression feature analysis methods used in the project are F-test and Mutual Information. F-Test is a statistical test that captures linear correlation between a feature and the result through testing of a difference between a model with just a constant and a model with a constant and the feature. While good for determining general relationships, F-test has drawbacks in that correlation does not always indicate relation, and it can only capture linear relationships. Covering these weaknesses is the Mutual Information method, which measures the dependence of a feature to a result and that result to the feature. This approach works well at capturing non-linear relationships, unlike the F-Test.

The two regression models used in the project are a simple Linear Regression model with Cross Validation, and a K Nearest Neighbors model that uses Grid Search Cross Validation to calculate the k value. In the Linear Regression model, we use cross validation to split the data into 5 sections, and run test and train splits with all the section, rotating until all sections have been used. This is then run over 20 different randomized sets of the data, and the mean of each calculated point spread variance is used as the final average point spread variance. For the KNN model, a Grid Search Cross Validation method is used to obtain the best k value, or the number of results to compare your prediction to. This k value is then used to run the KNN model to obtain the average point spread variance.

II. Related Work

This project is partially inspired by a 2013 Stanford paper called "Predicting the Betting Line in NBA Games". The project had a similar goal of predicting point spreads, but instead of predicting the point spread value itself, they focused on binary classification of determining whether a game would be cover the spread or fail to beat the spread. This project was also limited by the data available at the time, as there were not any publicly available betting data that covered multiple seasons. As a result, their results only used one season's worth of data, which equates to 1,230 NBA games.

I also drew inspiration from Dean Oliver's Four Factors of Basketball Success in deciding which features to utilize in our models. While common statistics such as point per game and rebounds per game were being calculated on a seasonal cumulative basis, I needed statistics to measure's a team's recent performance. Mr. Oliver named his Four Factors of Basketball Success as the keys to how basketball teams win games. He identified the following four factors, Shooting (40%), Turnovers (25%), Rebounding (20%), and Free Throws (15%) as the major pieces that affect a team's success. Following these ideas, I decided to capture a team's recent performance in each of these categories. For shooting, I used Effective Field Goal Percentage(eFG%)(1), which is a formula which accounts for the fact that 3-Point shots are tougher shots and worth more. I captured Turnover Percentage(TOV%)(2), or the ratio of turnovers to total possessions, for the turnover metric. For Rebounding, I recorded the team's Offensive Rebound Percent(OREB%)(3), which is a measure of a team's offensive rebounds against the opposing team's defensive rebounds. Finally, for free throws, I found the team's Free Throw Factor(FTF)(4), which measures how often a team shoots free throws, and more importantly, how often they make them. For each game, the average of these four factors were calculated over the team's prior five games.

III. Methodology

A. Data Collection

TABLE I
BETTINGDATA

Home Team	Away Team	Date
Home Spread	Away Spread	Over Under

TABLE II
BOXSCORES

Home Team	Game Number	Date
Away Team	Season	Home Points
Home FG Made	Home FG Attempted	Home FG %
Home 3PT Made	Home 3PT Attempted	Home 3PT %
Home FT Made	Home FT Attempted	Home FT %
Home Rebounds	Home Off. Rebounds	Home Assists
Home Steals	Home Turnovers	Home Blocks %
Home Fouls	Away Fouls	Away Points
Away FG Made	Home FG Attemtped	Home FG %
Away 3PT Made	Away 3PT Attemtped	Away 3PT %
Awaye FT Made	Away FT Attempted	Away FT %
Away Rebounds	Away Off. Rebounds	Away Assists
Away Steals	Away Turnovers	Away Blocks %

There are two major components to the data collection for the project. The first is a web scraper to collect betting data from Odds Shark, which has both point spread and over/under available from all games since the 2014 NBA season. The layout for the SQL table containing betting data is displayed in Table 1 The second component of the data collection was to use a Kaggle dataset which contains all box score data for the 2014-2018 NBA seasons. This dataset is imported into a SQL table, whose columns are shown in Table 2.

B. Data Processing

From the two tables created with the Kaggle dataset and the web scraper, the goal is to output data that can be used by the regression models. The content of this table is shown in Table 3. The per game statistics, such as points per game or rebounds per game, are all on a cumulative season basis. This is done by calculating the average of each of these statistics up until the game. In the case that there are less than five games played in the season before the current game, the entire season's average is used, which is stored in a table shown in Table 4. For the Four Factor statistics, these are calculated based on the team's performance in the past five games, calculating the eFG%(1), TOV%(2), OREB%(3), and FTF(4) over the previous five games. Again, in the case that the current game is one of the first five games of the season, the entire season average for these metrics are utilized instead. There is no table for these statistics, however, as they are calculated and used solely inside the script.

TABLE III
GAMEDATA

HT Points/Game	HT Rebounds/Game	HT Assists/Game
HT Steals/Game	HT Blocks/Game	HT Turnovers/Game
HO Points/Game	HO Rebounds/Game	HO Assists/Game
HO Steals/Game	HO Blocks/Game	HO Turnovers/Game
Home Offensive Rating	Home Defensive Rating	Home eFG%
Home TOV%	Home OREB%	Home FTF
AT Points/Game	AT Rebounds/Game	AT Assists/Game
AT Steals/Game	AT Blocks/Game	AT Turnovers/Game
AO Points/Game	AO Rebounds/Game	AO Assists/Game
AO Steals/Game	AO Blocks/Game	AO Turnovers/Game
Away Offensive Rating	Away Defensive Rating	Away eFG%
Away TOV%	Away OREB%	Away FTF

*HT/HO is Home Team/Opponent. AT/AO is Away Team/Opponent.

TABLE IV
SEASONSTATS

Season	Team	Points/Game
Rebounds/Game	Assists/Game	Steals/Game
Blocks/Game	Possessions/Game	Turnovers/Game
Opp. Points/Game	Opp. Rebounds/Game	Opp. Assists/Game
Opp. Steals/Game	Opp. Blocks/Game	Opp. Turnovers/Game
Opp. Possesions/Game	Offensive Rating	Defensive Rating

C. *Statistic Calculation*

Some of the statistics require specific formulas in order to calculate them. This list includes all of the four factors, and formulas are required to calculate Offensive Rating(5) and Defensive Rating(6) as well. They are shown in the equations section below.

$$eFG\% = (FGM + 0.5 * 3PM)/FGA \tag{1}$$

$$TOV\% = TOV/(FGA + 0.44 * FTA + TOV \tag{2}$$

$$OREB\% = OREB/(OREB + Opp.DREB) \tag{3}$$

$$FTF = FT/FGA \tag{4}$$

$$ORTG = 100 * (PTS/POSS) \tag{5}$$

$$DRTG = 100 * (Opp.PTS/Opp.POSS) \tag{6}$$

IV. Experiments and Results

A. *Feature Importance*

To determine which features were the most important in calculating the data, I used two regression feature importance methods. The first method used was F-Test, which generally is better suited for finding linear relationships, and the second method used was Mutual Information(MI), which excels at finding non-linear relationships.

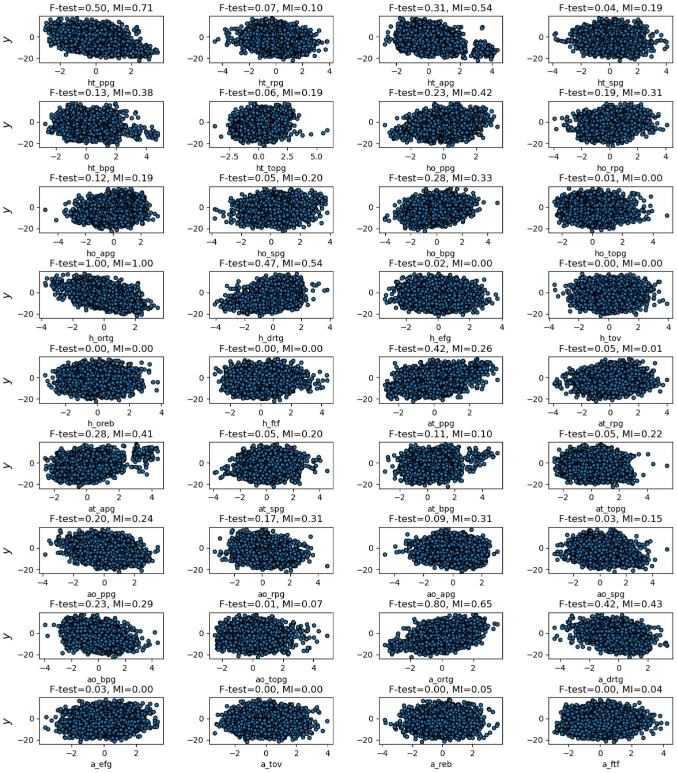

Fig. 1. Feature to Spread Correlation with F-Test and MI

Shown in Fig. 1. is a plot of each feature compared to the resulting point spread, along with their F-Test and MI scores, which are scored from zero to one. A zero indicates a feature that has little effect on the point spread, while a one indicates that a feature is important to the result.

Unsurprisingly, the metric related to Offensive Rating and Points Per Game had the highest F-Test and MI scores, with Home Offensive Rating receiving a score of 1.00 for both methods. This makes sense as generally, scoring more points is directly correlated to winning and the points spread. Interestingly, I found that the home team's Assists per Game was also a large factor, earning the same MI Score as the home team's Points per Game. Looking at Fig. 2., we can also see that apart from rebounds and points, the remainder of the per game statistics are more important from the home team's opponents for both F-Test and MI. This behavior is also similar, with slight differences, for the away team.

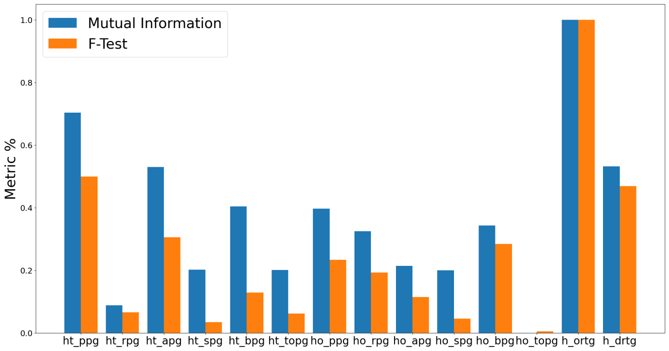

Fig. 2. The home team's MI and F-Test scores for cumulative season statistics.

A surprising finding was that the four factors calculated off of the performance of the last five games had little to no effect on the point spread results. All eight of these features (four for the home team, four for the away team) scored extremely poorly in both F-Test and Mutual Information, with the highest score being achieved in these metrics being a 0.06 in F-Test by away FTF. As we can see in Fig. 3., all of the features had very low scores, displaying their relative unimportance.

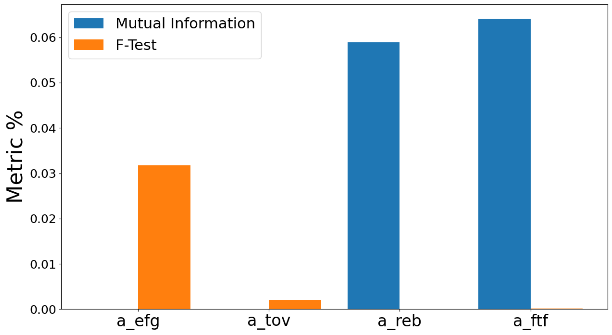

Fig. 3. The away team's MI and F-Test scores for four factor statistics.

B. Regression Models

a) Linear Regression: The Linear Regression model used was the sklearn Linear Regression library. I also implemented the sklearn KFold class to perform cross validation. I did five splits for the KFold Cross Validation over 20 seed values and calculated the average over the twenty runs as the final result.

To find the average deviation from the point spread, I found the deviation between each individual point spread prediction from the actual result. For example, a predicted point spread of +5 for the home team and an actual spread of -2 for the home team would be a deviation of 7. The average deviation is the mean of all deviations from all twenty runs.

To calculate how often the spread was correctly predicted, I counted a spread prediction that was below that of the Las Vegas spread to be a bet against the spread, and a spread prediction above that of the Las Vegas spread to be a bet to cover the spread. I then applied the same logic to the actual spread result and recorded a correct prediction if it was identical to result of my spread prediction.

I used a similar system to calculate whether or not my spread correctly predicted the game outcome. Instead of comparing my spread prediction and the actual results to the Las Vegas prediction, I checked if it was positive or negative. A positive spread for the home team indicates that the home team is expected to lose, while a negative spread for the home team indicates that the home team is expected to win. Therefore, when my predicted spread was positive, I recorded it was a prediction for a home win, and when the spread was negative, I recorded it as a prediction for a home loss. I used the same idea for the actual spread and compared it to my results to get the accuracy of win or loss prediction.

- Average Deviation from Actual Spread: **9.764**
- Percentage of Spreads Corrrecly Predicted: **65.167%**
- Percentage of Games Correctly Predicted: **66.417%**

With the original and main intention of the model being focused on minimizing the average deviation from the actual point spread rather than predicting the spread or game result, I was suprised by how accurate my model was using Linear Regression.

b) K Nearest Neighbor Regression: The K Nearest Neighbor Regression model used the KNeighborsRegressor class from sklearn.neighbors. To determine the optimum k value to be used in the model, I utilized the GridSearchCV class from sklearn. The GridSearchCV is a form of cross validation that uses a grid search preamble to find the optimal k value.

I used the same method I did in Linear Regression modeling for finding the average deviation, spread prediction percentage, and win prediction percentage. Similar to the Linear Regression model, I simple extrapolated the full data of predicted and actual averages from the KNN model, then used the above mentioned three methods to retrieve the spread deviation, spread prediction percentage, and game outcome percentage.

- GridSearchCV Optimal k Value: **23**
- Average Deviation from Actual Spread: **9.843**
- Percentage of Spreads Corrrecly Predicted: **67.733%**
- Percentage of Games Correctly Predicted: **63.608%**

I expected the K Nearest Neighbor Regression to perform slightly worse than the Linear Regression model, as I believed that I did not have enough data for K-Nearest Neighbor Regression to be fully effective. I was surprised to find that it was only approximately 0.1 worse than Linear Regression, and actually better when it came to correctly predicting the spread. However, again, since the main focus of the models are to predict the spread, I did not place much importance into the results of the spread and win predictions.

c) Vegas Spread Predictions: Using the Vegas data pulled from the Odds Shark website and the actual spread results, I also calculated the average deviation for the Vegas selected spread number. This deviation is the average deviation over the all the data in the project, which is four seasons worth of data, or 4920 NBA games.

- Average Deviation from Actual Spread: **9.345**

Unsurprisingly, the Vegas spread had a lower average deviation, but I was still impressed by how close a simple Linear Regression model came to the Vegas deviation.

V. Conclusions

I was overall very surprised and impressed by how close the Linear Regression and KNN Regression models got to the Vegas average deviation. The Linear Regression was only 0.419 points away, while the KNN Regression was only 0.498 points away. I was also impressed by the accuracy of both models when it came to predicting the spread and predicting the game outcome, especially because the model wasn't trained for either of these purposes, and trained to get as close to the spread as possible. In order to get closer to the Vegas prediction, I conclude that perhaps more data is required, and better features are required, as I found that using the four factor statistics from recent games has very little impact on the average deviation.

VI. Future Work

In the future, I hope to be able to create a model that can also utilize a team's strength of schedule and injuries. A team's cumulative season stats can easily be affected by the quality of opponents they have faced so far, especially early in the season. Injuries to star players can also hurt a team's expected performance coming into a game. I would also hope to create a more predictive way to create stats for a team's first few games of the season instead of utilizing season stats, possibly by using player individual stats from the past season.

VII. Acknowledgements

I would like to thank Dr. Edward Kim and the TAs, Paras Wadekar and Yigit Alparslan for their tremendous efforts in teaching everything related to Machine Learning, especially amidst the many situations the university has faced this quarter.

References

[1] B. Cheng, K. Dade, M. Lipman, and C. Mills, "Predicting the Betting Line in NBA Games." cs229.standford.edu.

[2] "Four Factors." Basketball Reference. n.d. basketball-reference.com.