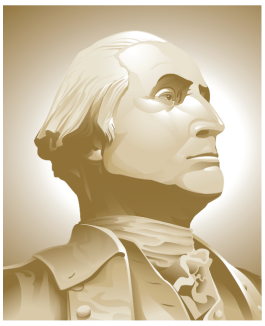

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Data Science Program

Capstone Report - Spring 2022

NBA Game Prediction

Divya Parmar

supervised by
Amir Jafari

Abstract

This paper investigates modelling and forecasting NBA games through feature engineering and model selection. When various model types are tested, they perform extremely similarly and return highly correlated outputs, suggesting that feature choices are more important than model choices. The betting spreads are found to be the most effective feature, displaying the skill of oddsmakers and the wisdom of the betting crowds, and adding additional features beyond this is often not helpful. There is a ceiling of model performance when simply using box score statistics, and to improve there is a need to model both game-by-game player ability and opponent specific characteristics.

Contents

1 Introduction

The history of sports analytics can be traced back for decades. In the 1960's, University of North Carolina head coach Dean Smith realized the value of tracking team statistics not just per game, but on a per possession basis. In baseball, the Society of American Baseball Research (SABR, pronounced "saber") was formed in 1971. However, acceleration began in the 1980's, when baseball fanatic Bill James started developing new methods and pushed sabermetrics into the mainstream. In the 1990's and early 2000's, Ken Pomeroy and Dean Oliver pushed basketball analytics into conventional sports conversation, culminating with John Hollinger and player efficiency rating being used by fans everywhere ("How the 'Idiots Who Believe' in the Analytics Movement Have Forever Changed Basketball").

This analysis has always mattered to super fans and fantasy sports players, but the value has accelerated as sports betting becomes more mainstream. Prior to 2018, sports betting in the United States was largely illegal but was still estimated as a $150 billion industry (Supreme Court Ruling Favors Sports Betting - The New York Times). A 2018 Supreme Court ruling paved the way for legalization of sports betting, and the industry is expected to grow a compounded 10 percent rate from 2021 to 2028 (Sports Betting Market Size & Share Report, 2021-2028).

With this explosion of fan interest and financial incentive, more academic and hobbyist work is being done to predict the outcome of all sporting events, the National Basketball Association included. This provides the opportunity to review the current literature and public work, study its methods, and contribute knowledge to this space.

2 Problem Statement

The goal of this project is to apply and test NBA prediction strategies using known feature sets (team Elo ratings, Four Factors team statistics, NBA 2K video game player ratings) and known models (Random Forest, SVM, Logistic Regression, Multi-Layer Perceptron NN) to find which strategy provides the best forecast of game results from the 2015 through 2020 NBA seasons.

The models will be measured on f1 score, log loss (on their predicted probabilities), and Brier score (as this is a forecasting problem). Furthermore, the models and feature types will be compared on their effectiveness.

3 Related Work

Historically, modeling the outcome of sporting events has been done a few different ways. One way is either modeling the effect of each individual play or game action, such as a Markov chain approach (Bukiet et al., Buttrey, Shi and Song). Another is to model each individual player's effect on winning through player partial effects and to combine such metrics into a larger model (Deshpande and Jensen).

A more common method is using production functions that focus on factors that determine the outcome of a game, such as scoring points or runs, as well as a variable for home field advantage. This second method can turn to creating proxy variables for the strength of each team, such as a power score (Stekler et al.). Proxy metrics include Elo (taken from chess) and Massey rankings, which have been shown to have predictive power (Dabadghao and Vaziri). More recently, dynamic paired comparison models and dynamic state space models have been used (Manner).

One set of commonly used proxy metrics for basketball is the four factors. Introduced by Dean Oliver in his 2004 book Basketball on Paper, the four factors are effective field goal percentage, turnover ratio, free throw rate, and offensive rebounding percentage ("Introduction to Oliver's Four Factors") (Basketball on Paper: Rules and Tools for Performance Analysis: Oliver, Dean: 9781574886887: Amazon.Com: Books).

These four factors (especially effective field goal percentage) are correlated with winning games, both in the regular season and postseason (Teramoto and Cross) (Baghal). Four factors can also be used to inform transition states in a Markov chain (Štrumbelj and Vračar).

Song, Zou, and Shi use the Four Factors and the first half of an NBA season to predict second half of the season results (Song et al.). For the Four Factors, they use both historical averages going into each game as well as an opponent adjusted value, but both have similar amounts of error. The model used was bivariate normal mean regression.

In work that is highly relevant to this work, Ondřej Hubáček , Gustav Šourek, and Filip Železný used two models (logistic regression and CNN) to allocate capital across a range of matches in a single round (Hubáček et al.). They test multiple strategies to allocate capital across matches, including uniform, by confidence level, absolute confidence difference, and relative confidence difference. They combined these strategies to optimize return, and found that this "opt" strategy with the CNN produced the best results. This paper inspired an open-source project that uses a convolutional neural network (CNN) with individual player data across 42 categories, and this model had a positive return in betting over 1000 games (Cheng). In a separate open-source project, Chu found that massive underdogs provide a positive return over time as compared to simply betting on favorites or underdogs (Chu).

Looking at both more academic work and open-source projects yields a wide range of model approaches to sporting event binary classification.

Maral Haghighat, Hamid Rastegari, and Nasim Nourafza studied the application of classification models to various sports including NBA, NFL, and European soccer. They reviewed the use of ANN (MLP), SVM, naïve bayes, decision trees, Fuzzy system, and logistic regression (J. and Rastegari). Fayad looks at predicting win percentage through models such as SVM, KNN, RF, and XGB and features such as shooting, rebounding, assists, and turnovers (Fayad). The different classical models have varying degrees of success, ranging from 68 to 73 percent, but predict much better than random guessing.

Alexandre Bucquet uses two neural networks (MLP and LSTM) to predict a decade of NBA over-under outcomes (Bucquet). Kevin Lane uses logistic regression, SVM, random forest, and a neural network (MLP) to predict NBA game results, and finds that MLP does the best job of classification (Lane). J Wade Johnson uses PCA (principal component analysis) and models such as naïve bayes, decision trees, random forest, KNN, and MLP. His models return $15,000 by betting $100 at a time (Johnson).

Kyle Skompinski uses XGBoost and a neural network to predict both game winners and over/under outcomes (Skompinski). However, this project stands out for its command line interface component. This tool allows a user to enter game information and get predictions as well as expected value (the difference between the model probabilities and the market probability). A creative user interface is not part of this work, but it shows where analytics tools can be improved and may be heading toward.

With the context of the literature, this work focuses on 1) modelling using static models and 2) deploying capital through various strategies. Through this process, this work makes a feature engineering contribution, both in terms of ideas and the code to implement from raw datasets.

4 Solution and Methodology

4.1 Feature Engineering

As the literature largely supports static models with game-specific information, we set out to codify this information into features. These features largely come from the literature or NBA experts.

Four Factors: The idea is that shooting the ball, taking care of the ball, offensive rebounding, and getting to the free throw line are activities that dictate the outcome of the game. These factors are Effective Field Goal Percentage, Offensive Rebounding Rate, Free Throw Rate, and Turnover Rate.

- Effective Field Goal Percentage: (Field Goals Made) + 0.5*3P Field Goals Made))/(Field Goal Attempts)
- Offensive Rebounding Rate: (Offensive Rebounds)/[(Offensive Rebounds)+(Opponent's Defensive Rebounds)]
- Free Throw Rate: (Free Throws Made)/(Field Goals Attempted) or Free Throws Attempted/Field Goals Attempted
- Turnover Rate: Turnovers/(Field Goal Attempts + 0.44*Free Throw Attempts + Turnovers)

In addition, each of these factors is created for the team's home/away games (depending on whether they are home or away for this given game), as well as a moving average for the last 10 games (to emphasize recency). In total, this is 12 features per team per game.

Winning Trends: These are factors that directly measure the team's history of winning and losing.

- Winning streak going into the given game, winning streak in home/away games (some teams may play much better at home than away or vice versa)
- Winning percentage in last 10 games, winning percentage in last 10 home/away games

Point Differential: This has been shown to be highly predictive, as consistently outscoring opponents matters more than winning close games (which can be due to luck).

- Average point differential (points scored minus points allowed) for the season, average point differential in home/away games
- Average point differential in last 10 games, average point differential in last 10 home/away games

Rest: Teams with more rest (days since last game) tend to play better. "Back to backs" (games on two days in a row) are known to be extremely difficult to win.

- Number of days rest. If game is on January 3rd, 2020 and prior game was January 1st 2020, this equals 1 as the game day doesn't count.
- Back-to-back flag: is this game a back-to-back or not?

NBA 2K Ratings: Another approach is to find a value of quality of each player on the team, and turn that into a composite score for team strength. One way to use this is to use the video game NBA 2K, which is released annually. Each player has an overall rating from 1-99. The lowest players are in the 60's, the 70's are average, and only the top players are in the 90's.

- Average team 2K rating: Take the top 10 players on the team (as that is the upper end on number of players used in a non-blowout game) and take the mean of their 2k rating.
- Weighted 2K rating: Take the top 10 players, and do a weighted average that gives the best players more weight. This is because they play more and have more effect on the outcome.
- Best player 2K rating: This is as the best player often has possession of the ball and drives the team.

Elo: This is a zero-sum rating system that was originally used for chess. Teams are given an initial rating, and the rating goes up or down based on game result and the quality of the opponent. This is a system that can capture changes in team quality over time, and it is maintained by FiveThirtyEight.

- Elo value: Team's Elo rating going into the game.
- Elo probability: Team's probability of winning given it's Elo value and opponent's Elo value (fixed formula).

Spread: Lastly, the spread (how many points one team is favored over another) is extremely relevant when it comes to game results. A team favored by 10 points will win most of the time, while a team favored by 1 point will win around half the time. We can use oddsmakers models as our own and gain from the wisdom of the crowds of bettors. For spreads, we use the "closing line" – the final value of the spread right before the game starts.

- Home spread for the given game. For example, -10 means that the home team is favored by 10 points, and +5 means that the home team is a five-point underdog in the given game.
- Spread in last 10 games: The team's average spread value in the last 10 games. For example, -2 means the team has been favored by 2 points on average, and +3 means that on average the team was a 3-point underdog.
- Team coefficient: Using a multi-label binarize technique, do a linear regression with each team as a column and get a coefficient which represents how each team affects the spread.
- Weighted team coefficient: Same as team coefficient, but older games are discounted using an exponential decay.

Ultimately, we will take these features do further analysis before bringing parameters into our models.

4.2 Model Selection

With our features generated, we can apply different model types and see how they compare. Primarily, we will use static models that disregard the sequence of games, as our features such as Elo and rolling averages should capture time-relevant information. However, we will also use an Auto-Regressive Moving Average (ARMA), a dynamic time series model, to serve as a comparison to the static models.

Here are the models we will use:

- Logistic Regression: This is a classical model, and it is mentioned often in the sports forecasting literature. This model is also interpretable, as we can see which features are significant and prominent. However, multi-collinearity is an issue here and thus necessitates dimensionality reduction.
- Logit: This model is associated with the logistic distribution, but it is different in that it is probabilistic. The parameters are estimated via maximum likelihood estimate (MLE), meaning how likely the observed data is given the parameters, over multiple iterations.
- Naïve Bayes: This classical model uses conditional probability and Bayes Rule to estimate parameters.
- Multi-Layer Perceptron: Although the theory and use of MLPs goes back decades, neural networks have come to the mainstream more recently. Thus, we can use an MLP classifier along with our other models.
- Auto-Regressive Moving Average (ARMA): Although time series models are not mentioned much in the literature, they are relevant as this is time ordered data.

For the ARMA modeling, the auto_arima model from statsmodels was used to fit an ARMA for each team-season (i.e. the 2015-16 Phoenix Suns). Facebook's Prophet library was considered, but it was slow to apply in a loop to repeated samples and was not ultimately used.

For an ARMA to be used, the data must be stationary. That check was done by plotting rolling mean and variance (Figure 1), checking the Auto-Correlation Function and Partial Auto-Correlation Function plots (known as ACF/PACF and seen in Figure 2), and doing an Augmented Dickey–Fuller test (Figure 3).

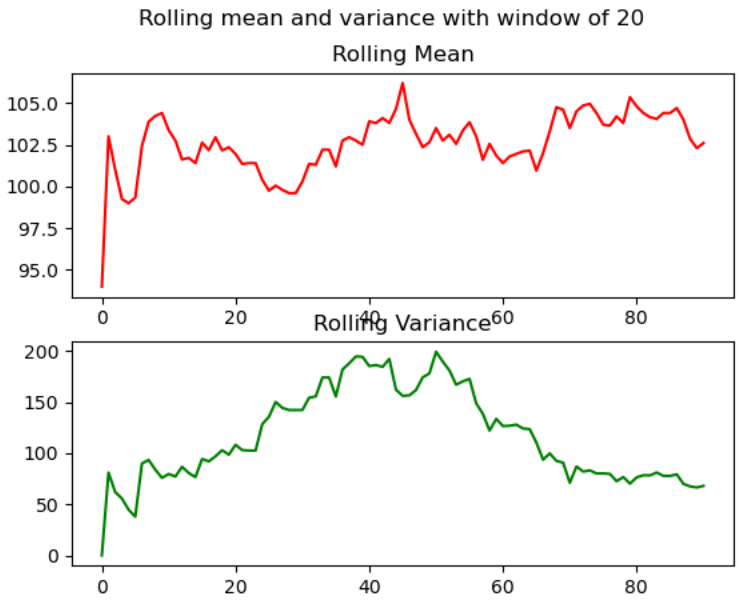

Figure 1: Rolling Mean and Variance of points scored in one team-season

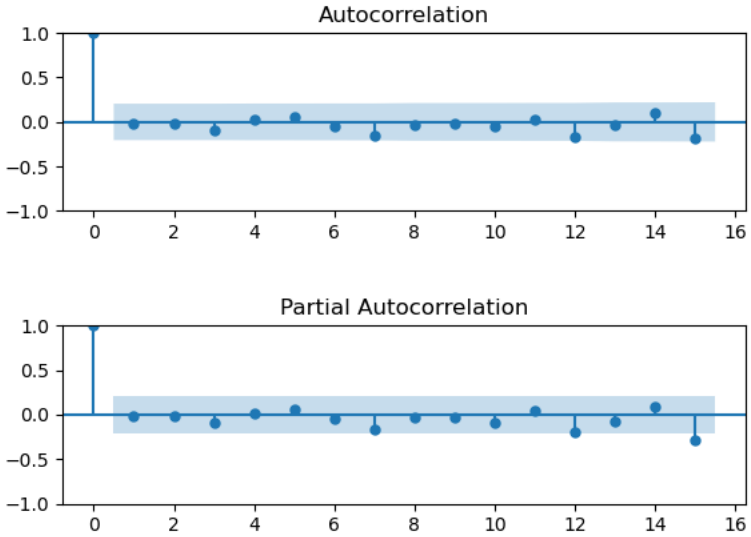

Figure 2: Auto Correlation Function and Partial Autocorrelation Function of points scored in one team-season

ADF Statistic: -9.674276
p-value: 0.000000
Critical Values:
1%: -3.504
5%: -2.894
10%: -2.584

Figure 3: ADF Test on points scored in one team-season

Lastly, we will ensemble the static models above to see if we can find further performance improvements. The probability outputs from the (1) Logistic Regression (2) Logit MLE (3) Naïve Bayes and (4) Multi-Layer Perceptron will be taken as features into (i) a logistic regression model and (ii) a random forest classifier. The ensemble is explained in the diagram in Figure 4.

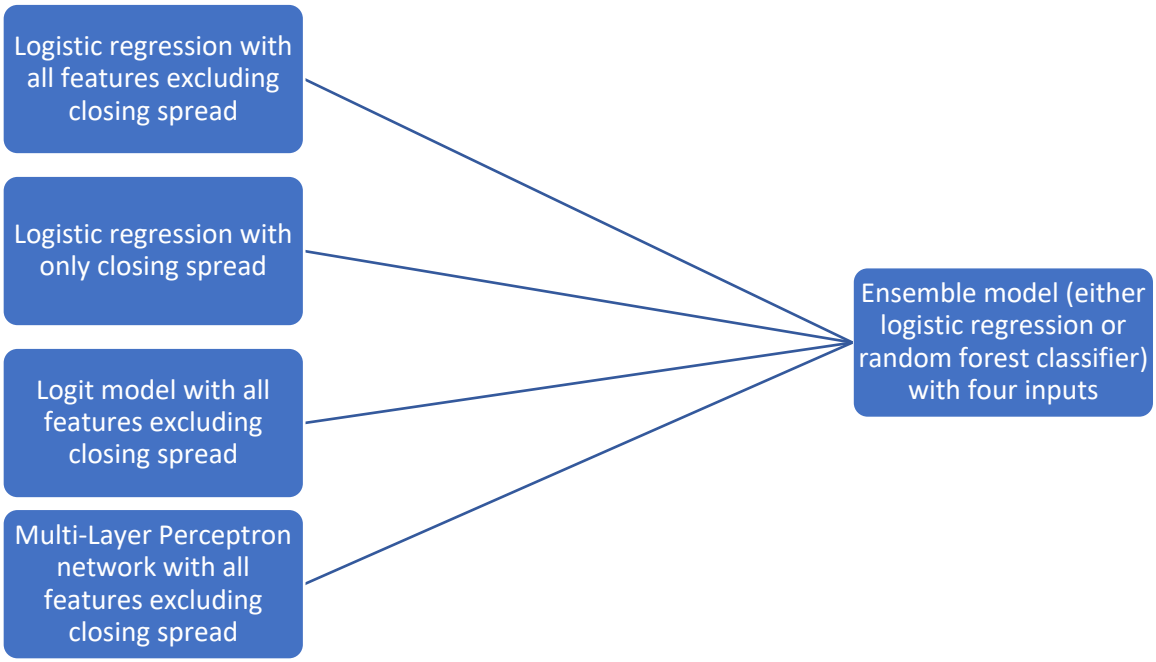

Figure 4: Ensemble Model Architecture

With our features and models defined, we can move to the evaluation phase.

4.3 Model Evaluation

To evaluate our models, we need to set our train-test splits. Three train-test splits were considered.

- Split One: Train on the 2015-16, 2016-17, and 2017-18 seasons. Test on the 2018-19 and 2019-20 seasons (pre-covid stoppable only). The idea here is that a model could be developed on prior seasons and deployed going forward.
- Split Two: Train on the first half of each season (41 regular season games) and test on the second half (41 regular season games). This is done in the literature by Song et al.
- Split Three: Random shuffle of games into train and test.

Ultimately, Split One was used to obtain the metrics reported in the results, as it seems to best mirror how a model would be trained and tested in an actual investment scenario. However, the models were also run using Split Two and Split Three, and the metrics were extremely similar (slightly higher, but within one percent difference on a relative change basis).

5 Results

5.1 Feature Selection and Importance

In order to fit a model such as logistic regression, we must deal with multi-collinearity. First, we can look at our 60 features (30 for home team, 30 for away team). We find that many of the features are highly correlated (Figure 5). This could be items like point differential in the last 10 games and win percentage in the last 10 games.

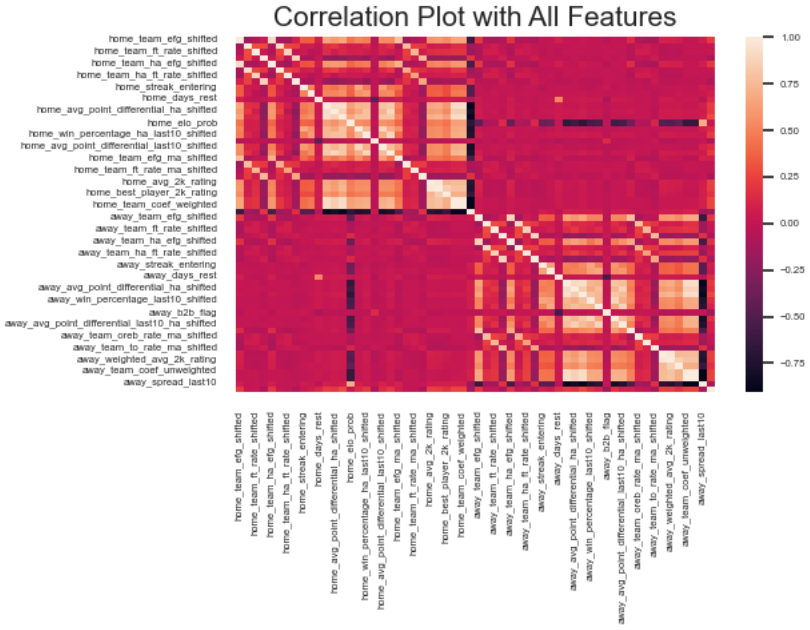

Figure 5: Correlation Matrix of All Features for Static Models

To deal with this, we do dimensionality reduction by dropping features which have a correlation coefficient of 0.8 with another feature. This includes pairings like Effective Field Goal Percentage and Effective Field Goal Percentage Home/Away (Figure 6).

Correlated Pairs
home_team_ha_efg_shifted - home_team_efg_shifted
home_team_ha_oreb_rate_shifted - home_team_oreb_rate_shifted
home_team_ha_ft_rate_shifted - home_team_ft_rate_shifted
home_team_ha_to_rate_shifted - home_team_to_rate_shifted
home_avg_point_differential_ha_shifted - home_avg_point_differential_shifted
home_elo_pre - home_avg_point_differential_shifted
home_avg_point_differential_last10_shifted - home_win_percentage_last10_shifted
home_avg_point_differential_last10_ha_shifted - home_avg_point_differential_shifted
home_avg_point_differential_last10_ha_shifted - home_win_percentage_ha_last10_shifted
home_weighted_avg_2k_rating - home_avg_2k_rating

home_team_coef_unweighted - home_avg_point_differential_shifted
home_team_coef_weighted - home_avg_point_differential_shifted
home_spread_last10 - home_avg_point_differential_shifted
away_team_ha_efg_shifted - away_team_efg_shifted
away_team_ha_oreb_rate_shifted - away_team_oreb_rate_shifted
away_team_ha_ft_rate_shifted - away_team_ft_rate_shifted
away_team_ha_to_rate_shifted - away_team_to_rate_shifted
away_avg_point_differential_ha_shifted - away_avg_point_differential_shifted
away_elo_pre - away_avg_point_differential_shifted
away_avg_point_differential_last10_shifted - away_avg_point_differential_shifted
away_avg_point_differential_last10_shifted - away_win_percentage_last10_shifted
away_avg_point_differential_last10_ha_shifted - away_win_percentage_ha_last10_shifted
away_weighted_avg_2k_rating - away_avg_2k_rating
away_team_coef_unweighted - away_avg_point_differential_shifted
away_team_coef_weighted - away_avg_point_differential_shifted
away_spread_last10 - away_avg_point_differential_shifted
home_team_ha_efg_shifted - home_team_efg_shifted

Figure 6: Column pairs correlated above the 0.8 threshold

We can again look at the correlation plot (Figure 7). We see that there is still correlation, but it falls below the threshold.

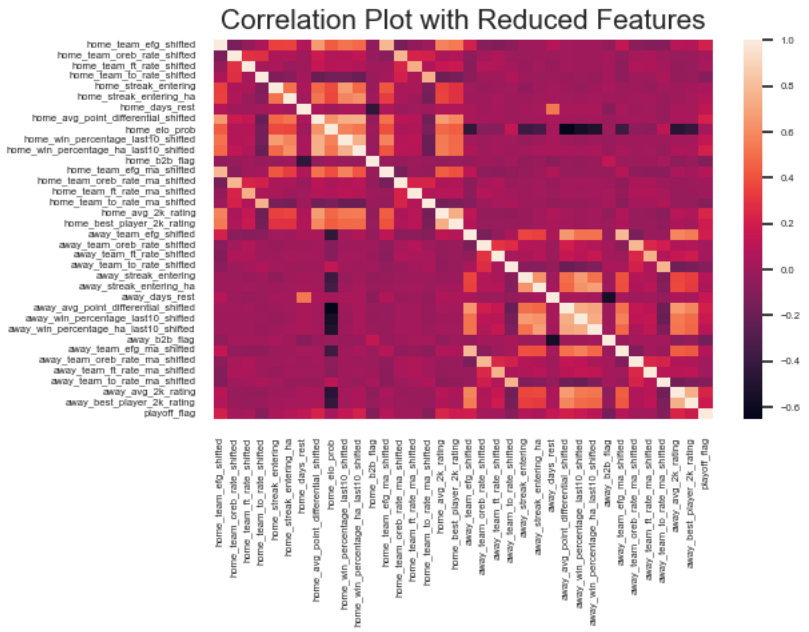

Figure 7: Correlation Matrix of Reduced Features for Static Models

Through this, we have learned that our feature engineering has picked up a lot of the same information. We can now look at the PCA plots to see how much of the explained variance. As seen in Figure 8, around 35 features explain over 90 percent of the variance. Coincidentally, our dimensionality reduction

using correlation gets us to 36 features. In Figure 9, we see that there is a less of a flattening in explained variance.

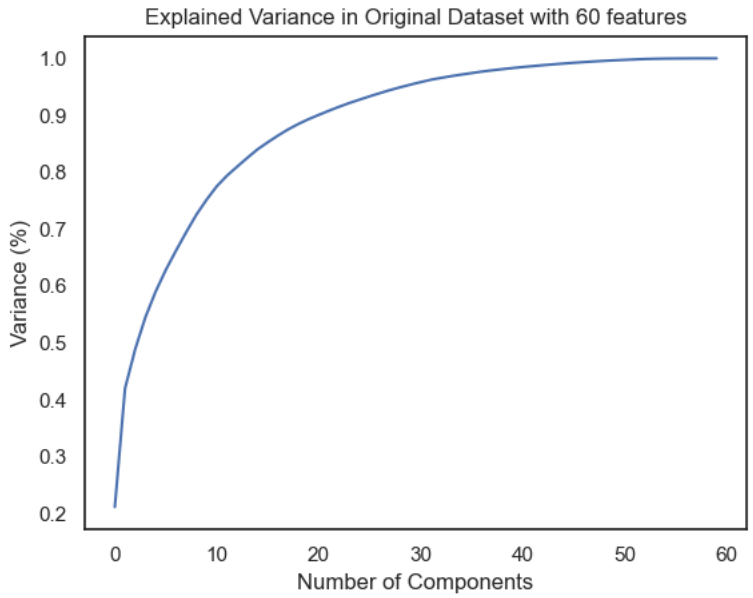

Figure 8: PCA Explained Variance for All Features

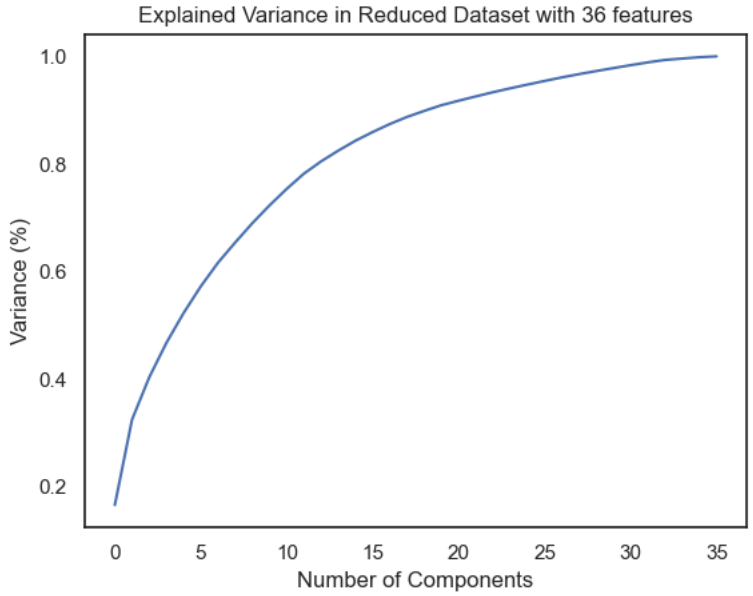

Figure 9: PCA Explained Variance for Reduced Features

Next, we can fit the logistic regression model and use the SHAP library to get feature importance (Figure 10). SHAP takes a model and each observation in the dataset, and it looks at how each feature affects the predicted target value. We can take the mean of the absolute value of effect (we don't care whether

it is positive or negative for the given observation). The home Elo probability has the highest importance, which makes sense as Elo tracks a team's ability over time and should be predictive. Coming in behind are the home team's free throw rate in home games and the home team's average player NBA 2K rating.

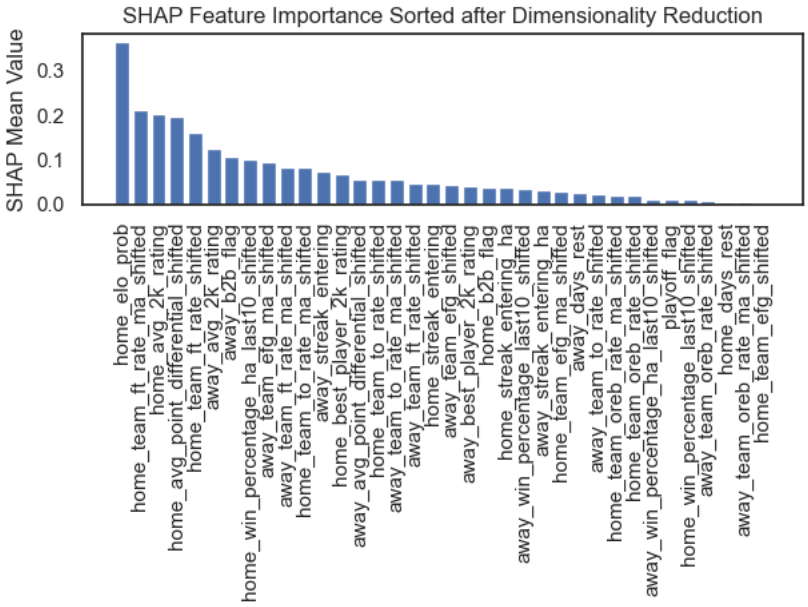

Figure 10: SHAP values for reduced feature set

5.2 Data tables

With the knowledge that a small number of features dominate, in addition to running the models on all features, we can look at individual features and how they would do on their own. To measure how well these feature sets capture the problem, we will look at f1-score, log-loss score, and Brier score. The Brier loss is important as this is a forecasting problem, and the objective is to get the predicted probability as close to the true outcome (1 for home win, 0 for home loss) as possible.

- F1 score: This metric measures accuracy, and it is calculated using precision and recall. It balances true positives, false negatives, true negatives, and false positives in its calculation.

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$

Figure 11: F1 score formula

- Log loss score (also known as binary cross entropy): This metric uses the predicted probability from the model and compares it to the target variable (0 or 1). Log loss is the loss function in logistic regression, whereas this is also called cross-entropy in other contexts.

$$-(y \log(p) + (1 - y) \log(1 - p))$$

Figure 12: Log loss (binary cross entropy) formula

- Brier score: This metric is frequently used in forecasting. It measures the difference in forecasted probability to the true result.

$$BS = \frac{1}{N} \sum_{t=1}^{N} (f_t - o_t)^2$$

Figure 13: Brier score formula

Note that the ARMA model will only have F1 metrics, as in this situation we are using the predicted value from ARMA and assigning the victory to whoever has the higher predicted points. There is no output probability created, as it given from the other models.

Here are the feature set categories:

Table 1: Feature Groups

Feature Group	**Features Included**
Spread Only	(1) Home spread points
Spread Comprehensive	(1) Home average spread in last 10 games, (2) Away average spread in last 10 games
Spread Coefficient	(1) Home team spread-calculated coefficient, (2) Away team spread-calculated coefficient
Spread Weighted Coefficient	(1) Home team spread-calculated coefficient weighted by recency, (2) Away team spread-calculated coefficient weighted by recency
Elo	(1) Home team Elo value going into the game, (2) Home team win probability according to Elo formula, (3) Away team Elo value going into the game Note that away team Elo probability is not included, as it is simply 1 minus home team probability
Four Factors	(1) Home team EFG, (2) Home team Off Reb Rate, (3) Home team FT Rate, (4) Home team TO Rate, (5) Away team EFG, (6) Away team Off Reb Rate, (7) Away team FT Rate, (8) Away team TO Rate
Four Factors Moving Avg	(1) Home team EFG moving average, (2) Home team Off Reb Rate moving average, (3) Home team FT Rate moving average, (4) Home team TO Rate moving average, (5) Away team EFG moving average, (6) Away team Off Reb Rate moving average, (7) Away team FT Rate moving average, (8) Away team TO Rate moving average

NBA 2K video game	(1) Home team average NBA 2K rating, (2) Home team weighted average 2K rating, (3) Home team best player NBA 2K rating, (4) Away team average NBA 2K rating, (5) Away team weighted average 2K rating, (6) Away team best player NBA 2K rating
Point Differential and Rest	(1) Home team days rest, (2) Home team average point differential in all prior games, (3) Home win percentage in last 10 games, (4) Home team back-to-back game flag, (5) Home team average point differential in last 10 games, (6) Home team average point differential in last 10 home games, (7) Away team days rest, (8) Away team average point differential in all prior games, (9) Away win percentage in last 10 games, (10) Away team back-to-back game flag, (11) Away team average point differential in last 10 games, (12) Away team average point differential in last 10 home games, (13) Playoff game flag

Table 2: F1 Score

	Model Type				
Feature Set Name	Logistic Regression	Logit MLE	Naïve Bayes	MLP	ARMA
All Features or N/A	0.733	0.733	0.713	0.720	
Spread Only	0.737	0.737	0.737	0.746	
Spread Comprehensive	0.737	0.737	0.729	0.728	
Elo	0.725	0.725	0.710	0.721	
Spread Coefficient	0.729	0.729	0.730	0.742	
Spread Weighted Coefficient	0.730	0.730	0.733	0.717	
Four Factors	0.723	0.723	0.717	0.720	
Four Factors Moving Avg	0.706	0.706	0.709	0.710	
NBA 2k	0.719	0.719	0.706	0.710	
Point Differential and Rest	0.721	0.720	0.691	0.712	
No Features (ARMA only)					0.604

Table 3: Log Loss

	Model Type			
Feature Set Name	Logistic Regression	Logit MLE	Naïve Bayes	MLP
All Features or N/A	0.613	0.613	1.258	0.614
Spread Only	0.599	0.599	0.600	0.687
Spread Comprehensive	0.605	0.605	0.606	0.605
Elo	0.613	0.613	0.668	0.625
Spread Coefficient	0.609	0.609	0.610	0.639
Spread Weighted Coefficient	0.607	0.607	0.608	0.617
Four Factors	0.644	0.644	0.655	0.654
Four Factors Moving Avg	0.656	0.656	0.663	0.661
NBA 2k	0.637	0.637	0.668	0.635
Point Differential and Rest	0.618	0.618	0.834	0.625

Table 4: Brier Score

	Model Type			
Feature Set Name	Logistic Regression	Logit MLE	Naïve Bayes	MLP
All Features or N/A	0.212	0.212	0.275	0.213
Spread Only	0.206	0.206	0.206	0.247
Spread Comprehensive	0.209	0.209	0.209	0.209
Elo	0.213	0.213	0.228	0.217
Spread Coefficient	0.211	0.211	0.211	0.224
Spread Weighted Coefficient	0.210	0.210	0.210	0.214
Four Factors	0.226	0.226	0.230	0.231
Four Factors Moving Avg	0.232	0.232	0.235	0.234
NBA 2k	0.223	0.223	0.232	0.222
Point Differential and Rest	0.215	0.215	0.256	0.218

Table 5: Root Mean Squared Error (RMSE) of ARMA model prediction of points scored

Group	**RMSE**
Home	12.04
Away	12.13

Table 6: Ensemble model metrics

Metric	**Logistic regression ensemble**	**Random forest ensemble**
F1 score	0.750	0.750
Log loss score	0.592	0.592
Brier score	0.203	0.203

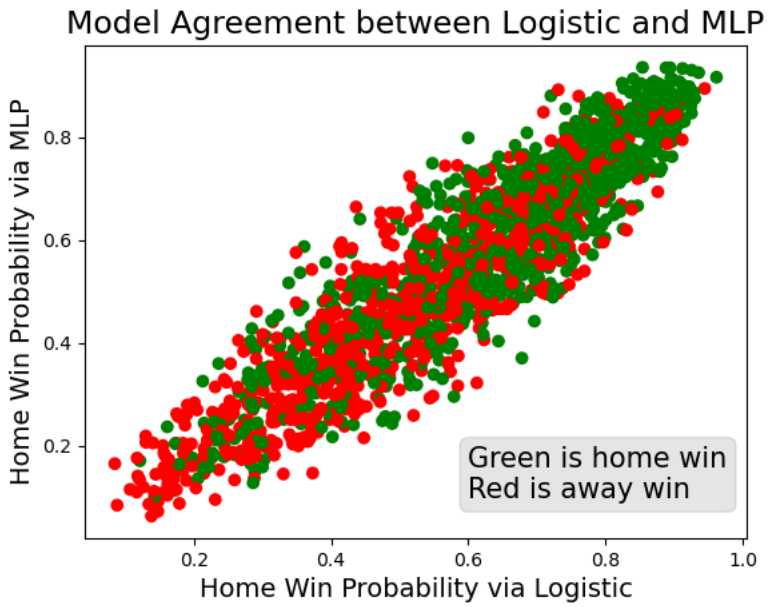

Figure 14: Model agreement between logistic regression and multi-layer perceptron models (r = 0.924)

In addition to the metrics above, we can look at how the models agree (Figure 14). There is a high correlation between the predicted win probability of the home team between the logistic model and MLP

model (Pearson's correlation coefficient equals 0.924). Generally, the home team wins the games that both models predict it highly likely to win, and the opposite occurs for games the models agree the home team will likely lose. However, there are notable errors by both models, making betting with this model difficult, as losses in high confidence situations can be very damaging for financial capital.

6 Discussion

In looking at the model metrics both by model type and feature set, point spread related features do the best across all models. This includes the overall point spread and team coefficients derived using the point spread. Elo, a rating system that accounts for history and team quality, also does well. In comparison, features such as Four Factors that try to measure how well a team does game-related activities did not perform as well. NBA 2K ratings did not do particularly well, but this makes sense as our values are from the start of each season and don't account for player development.

The models seem to matter less, and in some cases, different models return the exact same output metric. There are instances where MLP performs much better on an individual feature set, or naïve bayes performs unusually poorly, but there is no clear pattern to understand why models would do better or worse in terms of metrics.

The ensemble model did slightly better in terms of F1-score and Brier score, but it doesn't provide the massive boost that is sometimes seen in other domains. This is because the closing point spread already captures so much value, and there isn't much room to improve when the model outputs already correlate so much with each other.

One notable finding is that more features is not necessarily better. The "all features" models that take into account over 30 covariates performs worse than a model with just the closing spread. This speaks to the wisdom of the crowds, as well as the fact that individual nuggets of team performance are accounted for by bookmakers who are doing the most advanced modelling in the space. The bookmakers must be ahead of the betters, or the betters will financially exploit them. Thus, their market pricing is very powerful.

The time series approach, using ARMA on points scored, performs the worst in terms of F1 score, and the RMSE of the time series predictions makes them highly likely to be used in a sports betting context. The likely reason time series does not perform well is that there is a lot of noise within points scored, due to variance in shooting and "garbage time" statistics skewing the final values. In addition, there is no clear trend or seasonality in points scored that an ARMA model could pick up on and model. Lastly, to place bets, there is a need to think probabilistically, and time series models generally are not applied with the idea of confidence intervals and the probability of a value being above a certain threshold.

This work came with challenges that made it difficult to improve the performance of the models. There are various factors that have a large effect on the game but are not being picked up by the features above:

- Teams may play well or poorly against certain other teams, and the models above don't account for the specific opponent. Specific players may also struggle against certain teams, something not picked up in the feature set.
- Players may improve or regress throughout a season, and that will have implications for betting on a given team. The NBA 2K ratings in the feature set are static, given at the start of each season, and they could misidentify a player's talent level. Capturing updated NBA 2K ratings throughout the season or having a more dynamic model of a team's roster could be immensely helpful.

- This model does not pick up on trades and injuries. A team's shooting percentages and shot selection may be impacted by a trade or injury, and historical metrics may no longer be relevant.
- Certain referees are known to have negative impacts on a player or team's likelihood of winning, something not captured in the modelling above.

This work also dealt with some data quality issues. Box scores and player logs from earlier seasons such as 2015-16 were not always complete, affecting metrics such as NBA 2K ratings for a small number of games.

Ultimately, none of the features generated here performed better than the betting markets, validating the wisdom of the crowds. There is certainly proprietary work that has discovered insights that can beat the markets, but that is not captured here.

7 Conclusion

One notable finding that the different static model types (logistic regression, logit, naïve bayes, and multi-layer perceptron) all come to approximately the same performance metrics. Despite the different underlying mathematics, these models capture the same level of understanding given identical inputs, and they return highly correlated predicted probabilities. This suggests that feature engineering and pre-processing are just as important as model selection for sports forecasting.

In addition, more features are not necessarily helpful for a model. As the closing point spread captures a large amount of market information and wisdom of the crowd, it predicts the outcome of a game the best of all the features created. A model with just the closing line is better than a model with additional information about the teams themselves. This suggests that markets are highly efficient, and adding correlated features such as winning percentage, point differential, and shooting percentage can actually harm model performance through multi-collinearity and unnecessary complexity.

Although this is time ordered data, doing an ARMA model on a target variable was not effective. There may be a way to do time series analysis on player performance fluctuations, but it's possible that sports data has much more day-to-day variance and model complexity than daily temperatures or airline passengers (datasets commonly modelled with time series).

Given the information captured with this dataset, there seems to be a ceiling of an F1-score of around 75% and floor on the Brier score loss of around 0.2. Ensembling models together helped slightly, but it doesn't get the metrics past this ceiling. At this level, the models are not precise enough to use in betting, as missed classifications are very costly and there is an oddsmaker cut in priced into each bet.

To take the next step in model performance, there is a need for data beyond simple box scores and game statistics. Examples of this would be a better understanding of team-specific matchups (how does Team A match up with Team B), player ability fluctuations within a season, and a method to identify how many minutes each player will play (and to put an outcome-specific weight to their effectiveness). It is likely that this advanced work is already happening, but it is proprietary, as one would not want to lose the financial edge in developing such insights.

8 Bibliography

Basketball on Paper: Rules and Tools for Performance Analysis: Oliver, Dean: 9781574886887: Amazon.Com: Books. https://www.amazon.com/Basketball-Paper-Rules-Performance-Analysis/dp/1574886886. Accessed 17 Jan. 2022.

Bucquet, Alexandre. *The Bank Is Open: AI in Sports Betting*. 2018. 2021. *GitHub*, https://github.com/abucquet/bank_is_open.

Bukiet, Bruce, et al. "A Markov Chain Approach to Baseball." *Operations Research*, vol. 45, no. 1, 1997, pp. 14–23.

Cheng, Caleb. "Predict NBA Games, Make Money — Machine Learning Project." *Medium*, 9 Oct. 2020, https://towardsdatascience.com/predict-nba-games-make-money-machine-learning-project-b222b33f70a3.

Chu, Lambert. "Why You Are Betting on the Wrong NBA Teams." *Medium*, 4 Jan. 2021, https://towardsdatascience.com/why-you-are-betting-on-the-wrong-nba-teams-39e2bf98588.

Dabadghao, S. S., and B. Vaziri. "The Predictive Power of Popular Sports Ranking Methods in the NFL, NBA, and NHL." *Operational Research*, Mar. 2021. *Springer Link*, https://doi.org/10.1007/s12351-021-00630-9.

Deshpande, Sameer K., and Shane T. Jensen. "Estimating an NBA Player's Impact on His Team's Chances of Winning." *Journal of Quantitative Analysis in Sports*, vol. 12, no. 2, June 2016, pp. 51–72.

Fayad, Alexander. "Building My First Machine Learning Model | NBA Prediction Algorithm." *Medium*, 8 Nov. 2021, https://towardsdatascience.com/building-my-first-machine-learning-model-nba-prediction-algorithm-dee5c5bc4cc1.

Feddersen, Arne, et al. "Sentiment Bias in National Basketball Association Betting." *Journal of Sports Economics*, vol. 19, no. 4, May 2018, pp. 455–72. *SAGE Journals*, https://doi.org/10.1177/1527002516656726.

"How the 'Idiots Who Believe' in the Analytics Movement Have Forever Changed Basketball." *The Analyst*, 15 Apr. 2021, https://theanalyst.com/2021/04/how-advanced-analytics-have-changed-basketball/.

Hubáček, Ondřej, et al. "Exploiting Sports-Betting Market Using Machine Learning." *International Journal of Forecasting*, vol. 35, no. 2, Apr. 2019, pp. 783–96. *ScienceDirect*, https://doi.org/10.1016/j.ijforecast.2019.01.001.

"Introduction to Oliver's Four Factors." *Squared Statistics: Understanding Basketball Analytics*, 6 Sept. 2017, https://squared2020.com/2017/09/05/introduction-to-olivers-four-factors/.

J., ACSIJ, and Hamid Rastegari. *A Review of Data Mining Techniques for Result Prediction in Sports*. Nov. 2013.

Johnson, J. Wade. *NBA Sports Betting Model*. 2020. 2021. *GitHub*, https://github.com/garfjohnson/Nba-Sports-Betting-Model.

Lane, Kevin. *DataBall: Betting on the NBA with Data*. 2017. 2022. *GitHub*, https://github.com/klane/databall.

Manner, Hans. "Modeling and Forecasting the Outcomes of NBA Basketball Games." *Journal of Quantitative Analysis in Sports*, vol. 12, no. 1, Mar. 2016, pp. 31–41.

Skompinski, Kyle. *NBA Sports Betting Using Machine Learning* 🏀. 2019. 2022. *GitHub*, https://github.com/kyleskom/NBA-Machine-Learning-Sports-Betting.

Song, Kai, et al. "Modelling the Scores and Performance Statistics of NBA Basketball Games." *Communications in Statistics - Simulation and Computation*, vol. 49, no. 10, Oct. 2020, pp. 2604–16. *Taylor and Francis+NEJM*, https://doi.org/10.1080/03610918.2018.1520878.

Sports Betting Market Size & Share Report, 2021-2028. https://www.grandviewresearch.com/industry-analysis/sports-betting-market-report. Accessed 28 Mar. 2022.

Stekler, H. O., et al. "Issues in Sports Forecasting." *International Journal of Forecasting*, vol. 26, no. 3, July 2010, pp. 606–21. *DOI.org (Crossref)*, https://doi.org/10.1016/j.ijforecast.2010.01.003.

Štrumbelj, Erik, and Petar Vračar. “Simulating a Basketball Match with a Homogeneous Markov Model and Forecasting the Outcome.” *International Journal of Forecasting*, vol. 28, no. 2, Apr. 2012, pp. 532–42. *ScienceDirect*, https://doi.org/10.1016/j.ijforecast.2011.01.004.

Supreme Court Ruling Favors Sports Betting - The New York Times. https://www.nytimes.com/2018/05/14/us/politics/supreme-court-sports-betting-new-jersey.html. Accessed 28 Mar. 2022.

Teramoto, Masaru, and Chad L. Cross. “Relative Importance of Performance Factors in Winning NBA Games in Regular Season versus Playoffs.” *Journal of Quantitative Analysis in Sports*, vol. 6, no. 3, July 2010. *www.degruyter.com*, https://doi.org/10.2202/1559-0410.1260.