DS-GA 1003 Term Project

The Use of Machine Learning to Predict NBA Scores

David Hamilton (deh284)
Michael Yang (my1492)
Steffen Roehrsheim (sr4376)

1 Introduction

The goal of this project was to accurately predict the outcome of regular-season games in the National Basketball Association (NBA). The simplest approach is to target an absolute winner using a 0-1 loss approach ("Win/Loss"), which most institutional sports books and gamblers ("Vegas" in proper parlance) succeed in getting right around 70% of the time. A bigger challenge is to calculate the expected point spread ("Spread"), the traditional manner in which most people wager on an individual contest.

Standard Machine Learning methods were used to analyze historical league data in an attempt to develop solutions that could routinely beat naïve benchmarks and compete with more advanced wagering metrics employed by professional odds makers. Initial modeling results were then modified with various turn-key and bespoke optimization techniques to improve performance.

2 Raw Data

The primary source of data was the NBA's official statistical website, (stats.nba.com). While there is currently an abundance of open platforms that provide traditional and advanced basketball stats, the decision to use the league site was twofold and relatively simple: it had all the required information in one place, and the data was readily accessible through a customized Python module (github.com/bradleyfay/py-Goldsberry) built to query the source API.

A number of the advanced models being developed today by the scouting departments of league teams and select handicappers make extensive use of individual player statistics, often going so far as to create expected value targets on a possession-by-possession basis. However, project time constraints dictated a more pragmatic approach that relied on five specific categories of team-based figures:

Traditional - numbers found in a customary NBA Box Score (Traditional Statistics)**:**
Field Goal Pct
3-Pt Field Goal Pct
Rebounds
Assists

Advanced - more detailed, "second derivative" stats (Advanced Statistics)**:**
Team Offensive Rating (OFFRTG)
Team Defensive Rating (DEFRTG)
Assist-to-Turnover Ratio (AST/TO)
True Shooting Percentage (TS%)

Four Factors - additional advanced stats, including opponent-focused data (Four Factors)**:**
Opponent's Effective Field Goal Percentage (OPP EFG%)
Opponent's Free Throw Attempted Rate (OPP FTA RATE)
Opponent's Turnover Percentage (OPP TOV%)
Opponent's Offensive Rebound Rate (OPP OREB%)

Miscellaneous - advanced, situation-specific stats (Misc Statistics)**:**
Points off Turnovers (PTS OFF TO)
2nd Chance Points (2ND PTS)
Fast Break Points (FBPS)
Points in the Paint (PITP)

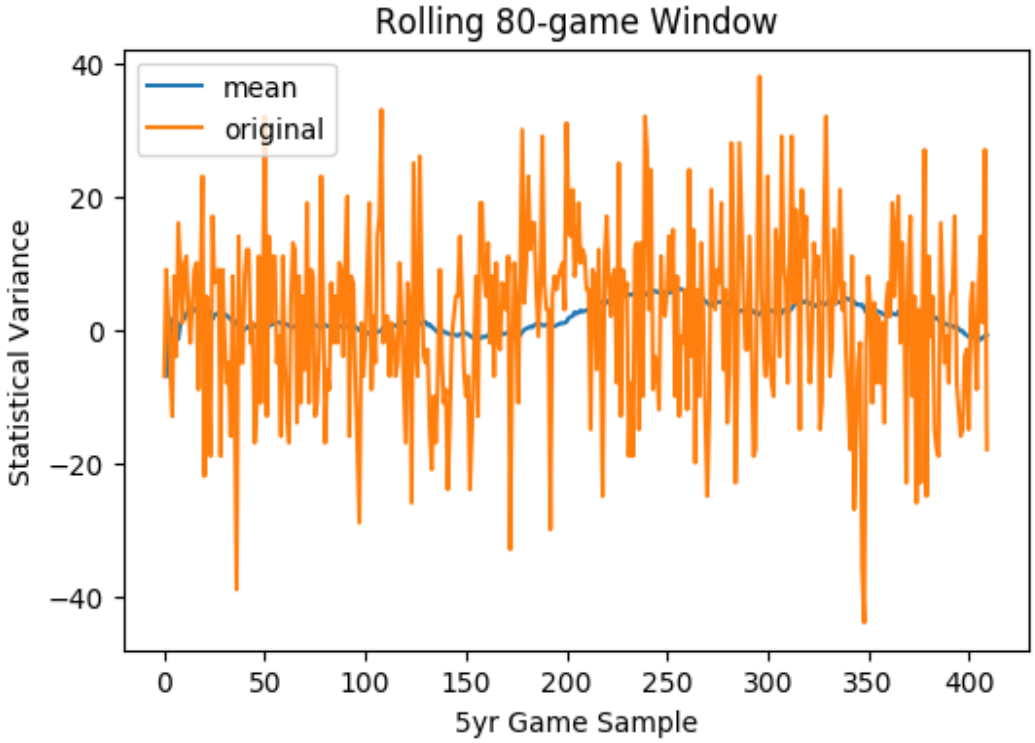
Scoring - advanced shooting-related stats (Scoring Statistics)**:**
Percent of Points (2 Pointers) (%PTS 2PT)
Percent of Points (3 Pointers) (%PTS 3PT)
Percent of Points (Free Throws) (%PTS FT)
Percent of Point Field Goals Made Assisted (FGM %AST)

3 Dataset

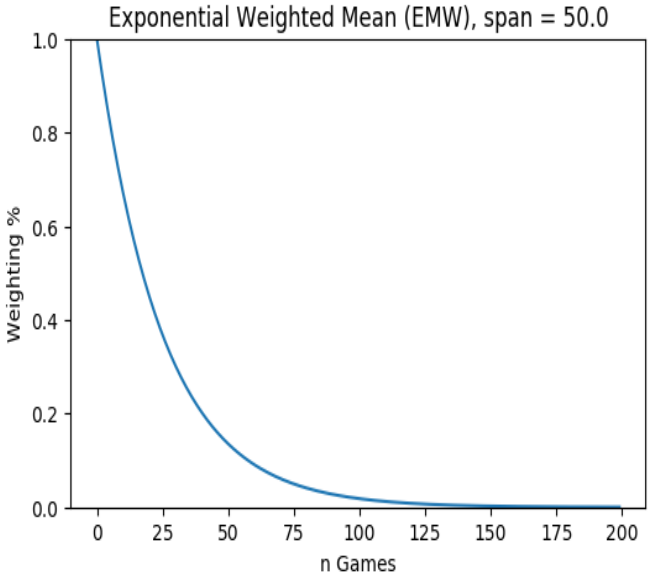
Detailed information for every regular season NBA game over the last five years (2012-13 to 2016-17) was used to build the master dataset. Once downloaded, approximately 100 features for the home and away team in each contest were stacked together, creating a roughly 6,150x100 matrix of individual, head-to-head observances:

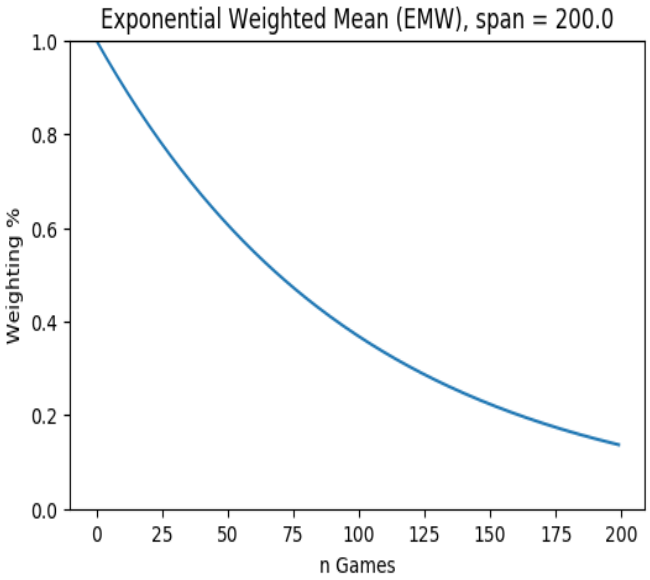
6,150 *Total Games* (1,230 *games/season* x 5 *seasons*) X (≈100 *Features*)

All non-target, numerical features were then scaled to a zero mean with unit variance. Lastly, the entire set was weighted with one of two methods: a rolling average (20, 40 or 80 game window) or an exponentially weighted mean (scan = 50, 100 or 200).



We can clearly see the smoothing effect an 80 game rolling average has on the feature variance of an individual team's 5-season sample. Compare that to exponential weighting and how changes to its decay factor (span) equate to different levels of emphasis being placed on past games.





4 Model Selection and Fitting

When choosing a model to investigate the Win/Loss result or point spread, it is possible to use essentially the same objective loss function and simply adapt the output to correctly interpret what constitutes success or failure in each case:

$$\ell(y, f(x)) = \sum_i (1 - y_i w^T x_i)^+$$

Where x_i is the stacked vector for the i^{th} game:

$$x_i = (\text{HOME team stats.....; AWAY team stats.....})_i$$

In the absolute, Win/Loss framework, y_i is the outcome of the game. In the Spread framework, y_i designates the side of the ultimate outcome, $+1$ if the prediction is on the right side of the actual result, -1 if it is otherwise.

Given this context, looking at Win/Loss seemed to be an obvious starting point, especially considering its greater ease of prediction. A number of regressors (**Linear, Lasso, RandonForest, Logistic** and **XGBoost**), two classifiers (**SVM** and **XGBoost-Classification**) were selected, along with one optimization technique (**PCA**) and a bespoke solution (the "**RIGHT-WRONG**" metric which will be discussed shortly) and applied to the dataset in the following manner:

3/1/1 Train/Validate/Test Split on the 5 individual seasons of figures
TRAIN: 2012-13, 2013-14, 2014-15
VALIDATE: 2015-16
TEST: 2016-17

Training, Validation and Testing for all models with
different Rolling and Exponential Weighted Averages
Rolling Average(n-game window): RollingAvg(20-games), RollingAvg(40-games), RollingAvg(80-games)
Exponential Weighted Mean(EWM): EWM(span=50), EWM(span=100), EWM(span=200)

We also make a point of implementing a naïve model to serve as a benchmark for the fitting process. One solution that can be quickly calculated is the net point differential between any two given teams at the time they play each other):

PPG = Points per game
Point Differential = Avg. PPG - Avg. Opponent PPG

Example:

Team A Point Differential (+5.3) - **Team B** Point Differential (+3.5) = +1.8 (**Team A** is Projected Winner)

In this case the baseline is simply a hard classifier that predicts a winner or a loser. Had the example been used as an estimate for Spread loss, Team A would be seen as a 1.8 point favorite (-1.8 in betting terms) and could be compared to the official Vegas line.

Results on the Validation set were as follows:

VALIDATE						
Standard	RollAvg(20)	RollAvg(40)	RollAvg(80)	EWM(sp=50)	EWM(sp=100)	EWM(sp=200)
Baseline	0.6385817	0.6468770	0.6439493	0.6476903	0.6441119	0.6281718
LinearReg	0.6699187	0.6886179	0.6536585	0.6788618	0.6723577	0.6390244
Lasso	0.6723577	0.6853659	0.6869919	0.6967480	0.6926829	0.6853659
RandomForest	0.6414634	0.6406504	0.6219512	0.6487805	0.6650407	0.6219512
XGBoost(Reg)	0.6658537	0.6788618	0.6747967	0.6813008	0.6804878	0.6512195
XGBoost(Class)	0.6609756	0.6747967	0.6739837	0.6788618	0.6471545	0.6536585
LogisticReg	0.6642276	0.6731707	0.6560976	0.6739837	0.6707317	0.6520325
SVM	0.6723577	0.6796748	0.6869919	0.6869919	0.6747967	0.6731707
MEAN	0.6607170	0.6710019	0.6623026	0.6741523	0.6684205	0.6505743
Optimized						
PCA (SVM)	0.6682927	**0.6910569**	0.6934959	0.6975610	0.6910569	**0.6926829**
PCA (XGB_Reg)	**0.6772358**	0.6739837	0.6634146	0.6861789	0.6861789	0.6780488
PCA (Lasso)	0.6691057	0.6902439	**0.6943089**	**0.7000000**	**0.6991870**	0.6894309
RIGHT-WRONG	0.6699187	0.6804878	0.6894309	0.6861789	0.6869919	0.6829268
MEAN	0.6711382	0.6839431	0.6851626	0.6924797	0.6908537	0.6857724

As an overall class, the top performer was Lasso, initially run on all features in the dataset and then for even better results as an optimized regression on the top 5 principal components in the validation matrix. Intuitively, this appears to make sense- given the large amount of overall features, a model capable of zeroing out non-essential components in favor of more predictive elements should be expected to do reasonably well. This is borne out by a quick check on the sparsity of the Lasso coefficient vector:

```
Fraction of zero coefficients:  0.907216494845
```

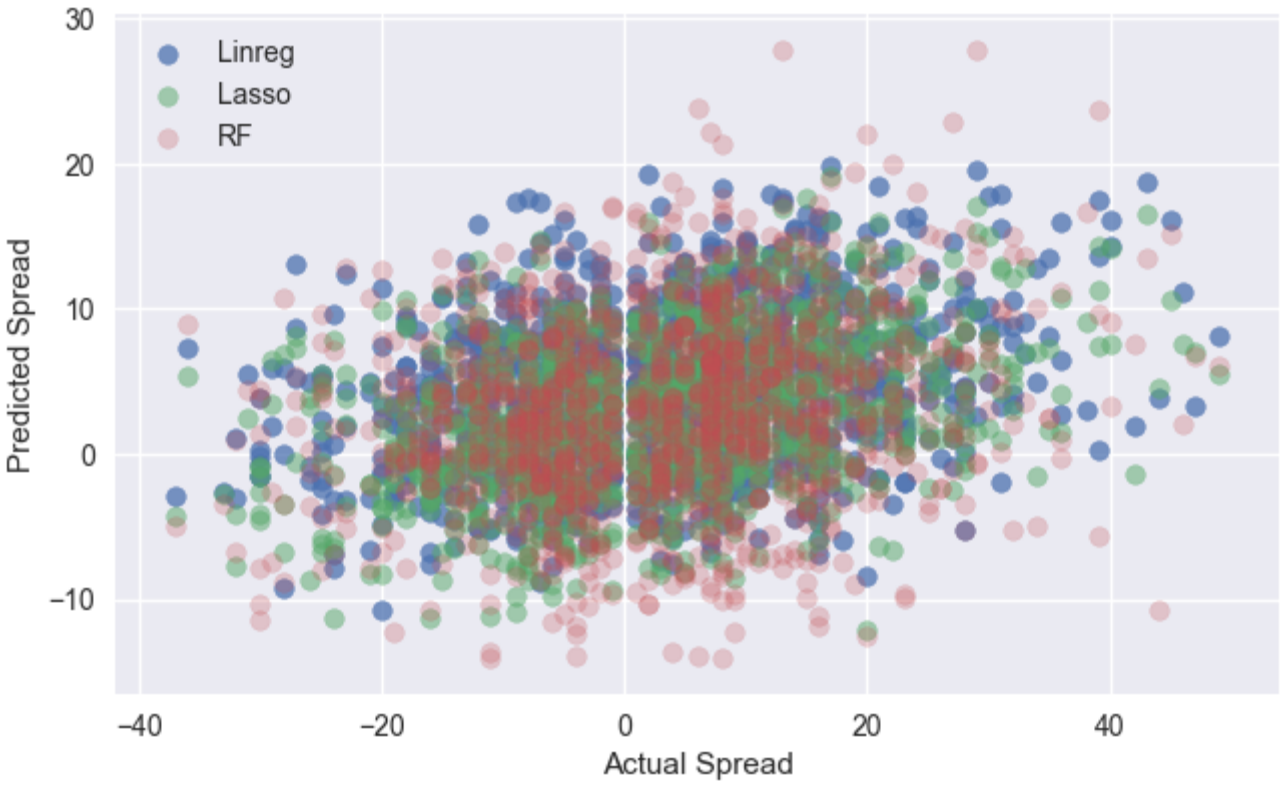
It shows 90.7% (88 out of 97 coeff.) are zero. Other observances of note:

- SVM was the top performing classifier, both on the full feature set with PCA
- The different data weighting schema failed to produce much disparity in terms of average performance, but the slight edge went to shorter-tailed EWM (span = 50, 100) applied to both the standard and optimized models.

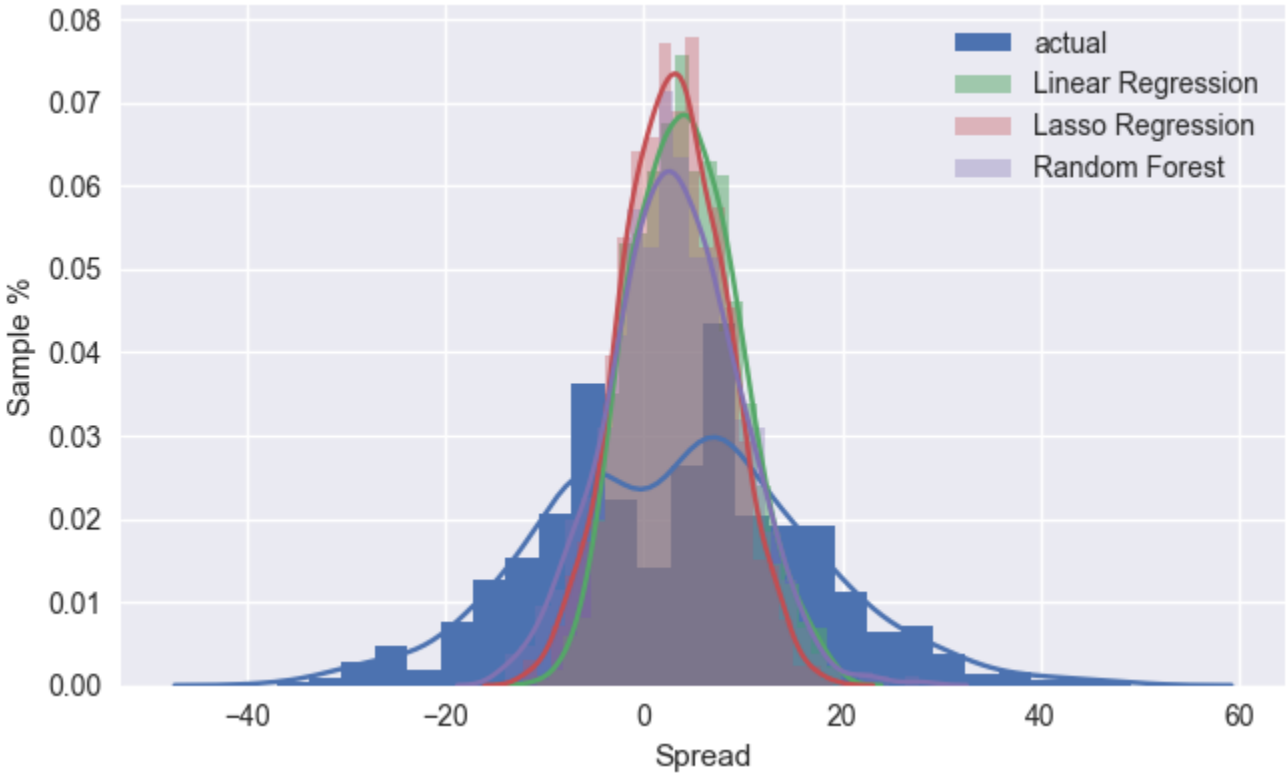
Results on the Test set were as follows:

TEST						
Standard	RollAvg(20)	RollAvg(40)	RollAvg(80)	EWM(sp=50)	EWM(sp=100)	EWM(sp=200)
Baseline	0.6385817	0.6468770	0.6439493	0.6476903	0.6441119	0.6281718
LinearReg	0.6178862	0.6268293	0.6065041	0.6162602	0.6121951	0.5926829
Lasso	**0.6439024**	0.6406504	0.6495935	0.6512195	0.6487805	0.6471545
RandomForest	0.5853659	0.5853659	0.5861789	0.5951220	0.5967480	0.5739837
XGBoost(Reg)	0.6243902	0.6203252	0.6081301	0.6235772	0.6130081	0.6195122
XGBoost(Class)	0.6357724	0.6349593	0.5983740	0.6284553	0.6048780	0.5821138
LogisticReg	0.6154472	0.6317073	0.6325203	0.6341463	0.6203252	0.6016260
SVM	0.6341463	0.6447154	0.6504065	0.6471545	0.6536585	0.6252033
MEAN	0.6244365	0.6289287	0.6219571	0.6304532	0.6242132	0.6088060
Optimized						
PCA (SVM)	0.6406504	0.6471545	0.6463415	**0.6593496**	0.6504065	0.6479675
PCA (XGB_Reg)	0.6406504	0.6382114	0.6325203	0.6479675	0.6365854	0.6512195
PCA (Lasso)	0.6398374	0.6447154	0.6512195	0.6520325	0.6544715	0.6463415
RIGHT-WRONG	0.6422764	**0.6479675**	**0.6601626**	0.6569106	**0.6617886**	**0.6569106**
MEAN	0.6408537	0.6445122	0.6475610	0.6540650	0.6508130	0.6506098

Immediately apparent is the significant drop in performance across the board between the validation and test sets. The results are even more disappointing when compared to our naïve benchmark. Almost all standard models fail to outperform the baseline, while their optimized derivations struggle to match or surpass it. What could be the reason behind this decrease in predictive power? Further examination yields some potential clues as to what the problem might be.



Looking at the above plot, we notice the impact of a unique characteristic found in select sports data. Current NBA rules stipulate that games cannot end in a tie. This means that the realized spread from a game can be positive or negative, depending on the perspective from which it is viewed (favorite or underdog, Home team or Away), **but it will never be zero**. Examining the distribution of results provides additional proof that it is bimodal in nature, as well as something most models will fail to accurately take into account.



The case can be made that such an issue was also present in the training and test sets, thus warranting additional explanation for the drop off in test prediction. Another potential cause could be a concern originally raised in our project proposal: professional sports tend to be more evolutionary in nature, rather than cyclical or mean-reverting (unlike other potential areas of study like financial markets).

Recent changes to the game, including the increased emphasis on wing play and 3-point shooting, decreased home-court advantage due to the transition from commercial to private team flights and more stringent defensive rules tend to be viewed as accepted truths, rather than short-term trends. The obvious effect of these developments is a recency weighting bias that could adversely impact the accuracy of models built and trained on older data. This could account for the validation set performing better, having been trained on the three seasons immediately preceding it, instead of the 1-year gap between the training and test data.

One bright spot among the test results was the ability of an intuitive optimization method, RIGHT-WRONG, to consistently beat the baseline (albeit not by much). To calculate this metric, we examine the training data and determine which teams we predicted correctly and incorrectly most often. The dominant contributing features (both positive and negative) are then ranked by the frequency of their impact on the training set to determine a subset of components. Choosing a subset of the first (top) 20 produced the highest predictive scores for both validation and testing:

```
Index([u'PCT_AST_3PM_HOME', u'AST_RATIO_HOME', u'PCT_UAST_3PM_HOME',
       u'EFG_PCT_HOME', u'TS_PCT_HOME', u'AST_TOV_HOME', u'OREB_PCT_HOME',
       u'PCT_UAST_FGM_HOME', u'DEF_RATING_HOME', u'OPP_EFG_PCT_HOME',
       u'TS_PCT_AWAY', u'PTS_2ND_CHANCE_HOME', u'PCT_PTS_FT_HOME',
       u'PTS_PAINT_AWAY', u'AST_RATIO_AWAY', u'OREB_PCT_AWAY',
```

```
 u'NET_RATING_HOME', u'OPP_EFG_PCT_AWAY', u'OFF_RATING_HOME',
 u'DEF_RATING_AWAY'],
dtype='object')
```

5 Spread Loss

The next step was to take the lessons learned from determining the outright winner of games and apply them to building an accurate prediction estimate for Spread loss. This is the metric most often used by professional handicappers and how the majority of game bets are made. Not surprisingly, the success rate is significantly lower than for Win/Loss, given that we are now trying to forecast **the margin of victory**, rather than just an outright winner. Based on the composite odds offered by institutional sports books, it is estimated that someone who consistently wagers on sports needs to win 52.4% of the time in order to break even.[1] Any signal that can consistently beat this threshold would then be considered capable of producing sustainable profits over the long haul.

To begin, we modify our previous naïve model to create a new, spread-driven baseline:

Example:

Team A Point Diff. (+5.3) - **Team B** Point Diff. (+3.5) = +1.8 (Projected Spread-**Team A** is favored)

Once the updated benchmark is in place, we take the Spread version of our objective loss function and apply it to the data. Results on the Validation set were as follows:

VALIDATE (SPREAD LOSS)				
Standard	RollingAvg (40 gm)	RollingAvg (80 gm)	EWM (span=50)	EWM (span=100)
Baseline	0.488939493	0.482921275	0.495933637	0.491867274
LinearRegression	0.491980964	0.492936083	0.488373984	0.491602882
Lasso	0.497560976	0.494308943	0.494308943	0.486178862
RandomForest(Reg)	0.487519995	**0.509938529**	**0.508677375**	0.504006213
XGBoost(Reg)	**0.515810034**	0.505651398	0.503843612	0.498701170
LogisticRegression	0.505691057	0.508130081	0.500813008	**0.511382114**
MEAN	0.497917086	0.498981052	0.498658426	0.497289752
Optimized				
PCA (Lasso)	0.493495935	0.500000000	0.493495935	0.493495935

Results on the Test set:

TEST (SPREAD LOSS)				
Standard	RollingAvg (40 gm)	RollingAvg (80 gm)	EWM (span=50)	EWM (span=100)
Baseline	0.488939493	0.482921275	0.495933637	0.491867274
LinearRegression	0.481626016	0.468873686	0.472938727	0.470504990
Lasso	**0.518699187**	0.493495935	0.512195122	0.501626016
RandomForest(Reg)	0.497737458	**0.504693635**	**0.512846189**	0.498842620
XGBoost(Reg)	0.50946857	0.497900059	0.510023795	0.492011369
LogisticRegression	0.514634146	0.504065041	0.489430894	0.508130081
MEAN	0.501850812	0.491991605	0.498894727	0.493830392
Optimized				
PCA (Lasso)	0.506504065	0.503252033	0.504065041	**0.509756098**

[1] www.thesportsgeek.com/sports-betting/math/

On average, our models fail to generate significant improvement over the baseline. There are select cases of outperformance, the best of which is a little under 3% *(Lasso, 40-game Moving Avg.)*, but unfortunately none were able to beat the 'sustainability' target of 52.4% *(the same 40-game Lasso regression was closest, coming within 0.50%)*. One encouraging note is the lack of drop off in predictive power between the validation and test sets, as seen in the calculation for Win-Loss. Why that occurred warrants additional investigation.

6 Conclusion and Future Development

As expected, predicting an accurate estimate for Spread loss proved to be more difficult than Win/Loss. This translated into models that would be economically unviable in a real-life sports betting context. However, given more time and the wealth of secondary data sources for individual player statistics and additional team information, it is reasonable to believe that our current results can be improved significantly.

Potential topics to keep in mind for future development:

- **Incorporation of Team Elo Ratings**: a measure of team strength based on head-to-head results and quality of opponent.[2] Unfortunately due to time constraints, they could not be added to the project dataset. Historical scores can be found online, as well as calculated independently.
- **Use of Individual Player Statistics**: It is well known that the most advanced predictive models employed by NBA team scouting departments and professional sports handicappers make extensive use of individual player stats.
- **Additional Models and Optimization Methods**: This could include, but not be limited to, the use of Ridge regression and Elastic Net, Bayesian Methods and advanced forms of feature selection.

[2] projects.fivethirtyeight.com/2017-nba-predictions/

Contributions

David Hamilton

- Wrote project proposal.
- Researched primary sources for data and found Goldsberry repository for querying stats.nba.com.
- Built and cleaned master dataset.
- Coded all primary models and training, validation and testing functions.
- Implemented Rolling Average and Exponential Weighted Mean (EWM) and added them to master dataset.
- Conducted and documented all project training, validation and testing procedures.
- Final edit of project poster.
- Wrote final project report.

Michael Yang

- Contributed to project proposal.
- Coded secondary loss functions, models and optimization methods.
- Design, layout and production of project poster.
- Contributed to final project report.

Steffen Roehrsheim

- Contributed to production of project poster.

Project Repository on GitHub:
github.com/davidehamil/ml_applications_for_the_nba