Predicting Against The Spread Winners In National Basketball Association Matchups

Jim Caine

November, 2014

DePaul University – CSC 478

Executive Summary

Prior to each game in the National Basketball Association (NBA), Las Vegas odds makers determine a *spread*, indicating which team is believed to win the game and by how many points they will win. If the resulting score of the game results in the favored team (as defined by the spread) winning by more than the spread, then that team is said to have won the matchup *against the spread (ATS).* If not, then the favorite losses against the spread and the underdog win against the spread. In this analysis, data describing each matchup and each team involved in the matchup is leveraged to predict which team in the matchup will win the matchup against the spread. The main objective of the analysis is to build a model that predicts the ATS spread winner for each matchup.

Data is collected from two sources that describe the historical spread, the resulting score, and team performance data for every NBA matchup from the beginning of the 2008 season through Nov. 18, 2014. A series of preprocessing steps are applied to the dataset to convert the game by game team performance data into attributes that represent the ratio between the two teams performance over a given time period – season to date, last 10 games, last 5 games, and last 3 games are used for the time periods. Multiple datasets are created by selecting different sets of features. In particular, a dataset representing all attributes, all season to date attributes, all last 10 game attributes, all attributes selected in backward elimination in the linear regression model, and principal component attributes (at different component lengths).

Linear regression, logistic regression, and decision tree classification algorithms are applied to the datasets to predict the ATS winner of the game. Both numerical prediction algorithms are categorical prediction algorithms are used with the target attribute for numerical prediction being the point differential between the two teams and the target attribute for categorical prediction being a binary variable taking the value 1 when the home team wins and 0 when the away team wins.

The performance of each model is evaluated by looping through each day of the dataset chronologically and for each day, create a train set containing all matchups that occurred before that day and a test set containing all matchups for that day. A model is fit to the train set of data and predictions for that particular day are made for the test set. All models for all datasets prove to be predictive, with an overall accuracy of greater than 60% for all models, and a maximum accuracy of 64.65% for the decision tree model applied to a PCA-transformed dataset containing one principal component. A simulation is ran by betting 5% of the total balance on each game for each day, and it is shown that $1 grows into greater than millions of dollars over a six year period due to exponential growth.

1 The Data

The final dataset used in this analysis contains the results and a description of the participating teams for all NBA matchups from the beginning of the 2008 season through Nov. 18, 2014. Each row in the dataset contains the participating teams, the closing spread for the game (as determined by Las Vegas odds makers), the final result of the games (the points scored by each team), and a series of attributes that indicate the ratio between the away team's performance and the home team's performance. A complete schema of the dataset along with a description of the attributes can be found in Appendix A.

1.1 Data Design Decisions

A few important design decisions are made prior to collecting the data:

1. It is decided that the two participating teams in each matchup are represented by home team and away team. This is contrary to another popular method of representing the teams as the favorite and underdog.
2. The spread for each game is chosen to be negative when the home team is the favorite and positive when the away team is the favorite.
3. The point differential is chosen to be negative when the home team scores more points than the away team.
4. To represent the difference between the two teams that are playing in the matchup, the ratio between the same attributes are taken between the two teams. That is, the away team's statistic is divided by the home team's statistic. Therefore, when attributes are a positive indicator of performance, a value greater than 1 indicates that the away team performs better for that particular attribute.
5. When the attributes are integers (not statistics), then the difference between the away team's attribute and the home team's attribute is taken. Examples of this are win streak, compared to statistics such as points per game that would be compared by taking the ratio as described in (4).

1.2 Data Collection

The first source of data that is used in the analysis is the Sports Data Query Language1 (SDQL). SDQL is an online aggregator of box score data for historical matchups for the NBA, MLB, and NFL. The website supports a custom query language to extract data from its website. Additional details about the query language can be found at the SDQL website. The query used for this analysis can be found in Appendix B. The query collects box score data for each team for every game played by that team for a given season. The box score attributes describe how a team plays in a variety of ways such as how good is the team at shooting, how good they are at rebounding, and overall performance. Example attributes are steals, turnovers, field goals attempted, and field goals made.

The second source of data is oddshark.com2. Oddshark.com is a website that contains historical closing spreads and results for each team in the NBA. The data is scraped using Python and the BeautifulSoup package for Python. This script goes through each season/team website and writes the home team, away team, home team points, away team points, and the closing spread to a csv file. Only the games where the team is the home team are kept to avoid duplication.

Both data sources were spot checked for accuracy using ESPN.com

1.3 Data Preprocessing

A series of preprocessing steps are applied to the two data sets to prepare an aggregated data set representing each matchup and the difference in historical performance between the two teams in the matchup:

- Cleaning the raw SDQL data: each team in the SDQL output is represented by their team name, while each term is represented by their city in the data provided by covers.com. Therefore, each instance of a team name in the SDQL raw data is replaced with the city name. Special consideration had to be made for the Nets as they moved from New Jersey to Brooklyn during the time frame that is analyzed. Special consideration also was made for the Charlotte franchise, as their team name changed from the Hornets to the Bobcats and back to the Hornets during the time frame that is analyzed. In addition, the attribute 'rest' contained entries with a '- ' representing the first matchups of the season. These values were replaced with 0. In the final dataset, the difference between days of rest between the two participants in the matchup is considered, and therefore, the replacement value will not matter as the difference will be 0.
- Aggregating the team attributes: The data from SDQL describe the results achieved by a particular team on a particular day (there are a few exceptions to this rule such as win streak and win streak against the spread) – and therefore fails to describe the attributes of the team before the day. Therefore, it is fair to say that this data is not a fair representation of the team prior to that day. To create a better representation of the performance and attributes of a team, aggregate data for each team is computed by summing the attributes for the specific team for all matchups in that particular season before the day of the entry.
- Normalizing the team attributes: The sum of the attributes are normalized by the number of games that the team has played, thus providing a more fair comparison for each team on a given day (particularly for matchups early in the season when sum of the attributes are more dependent on the number of matchups played).
- Creating derivative attributes: Additional attributes are derived from the existing attributes that are commonly used when evaluating teams. An example of this is

assist to turnover ratio, which is a common indication used to evaluate point guards.

- Reducing the team profile: The attributes for each matchup that represent the total that the team achieved for that day are thrown out. The normalized attributes that are created in the previous section are kept.
- Cleaning the team profiles dataset: Another iteration of data cleaning on the team profiles is done. First, any matchup that is a team's first game of the season are thrown out because there is no historical season data to be used (resulting in inf values for the attributes). Next, all rows with infinite values are thrown out of the dataset.
- Merging the matchups and team profile datasets: Finally, the team profile data set is merged into the matchups dataset, thus providing a snapshot of each team in the matchup that is obtainable before the matchup occurs. Two merges are done, with the first merging the home team attributes to the matchups dataset and the second merging the away team attributes to the dataset. Special consideration had to be taken to ensure that the new columns are labeled accordingly with '_home' or '_away'.
- Calculating ratio attributes between the two participants in each matchup: Instead of using each team's attribute independently in the analysis, attributes are formed to represent the difference between the attribute for the two participating teams in the matchup. For example, the attributes 'average_blocks_per_game_home' and 'average_blocks_per_game_away' are not used in the analysis, rather, the ratio between the two values is used.

The resulting dataset is referred to as *MatchupsMain* throughout the remainder of the paper and the complete schema and description of all of the attributes are shown in Appendix A.

1.4 Attribute Selection

Multiple datasets are created by selecting different features from *MatchupsMain* and are used throughout the analysis. The datasets created are:

Dataset	Description
MatchupsAllAtt	Contains all attributes from *MatchupsMain*
MatchupsSTD	Contains attributes describing season to date performance ratios from *MatchupsMain*
MatchupsL10	Contains attributes describing performance over the last 10 games from *MatchupsMain*
MatchupsBE	Contains attributes that are selected via backward elimination in linear regression on the full dataset
MatchupsPCA1	Contains the first component of the PCA-reduced full data set
MatchupsPCA2	Contains the first two components of the PCA-reduced full data set
MatchupsPCA3	Contains the first three components of the PCA-reduced full data set
MatchupsPCA6	Contains the first six components of the PCA-reduced full data set
MatchupsPCA9	Contains the first nine components of the PCA-reduced full data set

Table 1

The attributes in *MatchupsBE* are selected by performing backward elimination on the full dataset to minimize the RMSE for a linear regression model. The output of backward elimination performed in R containing the selected attributes, their Residual Sum of Squares, and their AIC values can be found in Appendix C.

The number of components in the PCA-reduced datasets was chosen to represent the first three components, then 90% of the variance in the dataset (*MatchupsPCA6)*, and 95% of the variance in the dataset *(MatchupsPCA9)*. More about the principal components is shown in Section 3.3.

2 Data Exploration

A thorough analysis of the dataset is outlined in this section prior to utilizing the dataset for the prediction of against the spread winners. First, a correlation analysis is shown to find which attributes are most highly correlated with the target attributes and which attributes are most highly correlated with other attributes (to possibly avoid multicollinearity in the models). Next, the attributes that have the largest deviation in mean value for home winners and away winners are found, as these attributes are highly informative in fitting some models. Lastly, we explore the shape of the correlations between the attributes with the highest correlation with the target variable and the target variable.

2.1 Target Variables

Two target attributes are used for the analysis, a numerical target attribute and a binary target attribute. The numerical target attribute describes the actual point differential between the two teams in the matchup, as defined as the points scored by the away team minus the points scored by the home team. The binary target attribute takes the value of zero when the away team wins the ATS bet, and one when the home team wins the ATS bet. The numerical target attribute can be converted into a binary prediction by taking the value of zero when the point differential is greater than the spread, and one when the point differential is less than the spread.

The numerical target attribute, point differential, is approximately normal with a mean of -2.88 and a standard deviation of 13.08. The mean of -2.88 indicates that the home team beats the away team by 2.88 points on average. This intuitively makes sense, as the home team should have an advantage due to the crowd and extra rest associated with no traveling. A histogram of the point differentials can be seen in Figure 1. The drop off in the middle of the distribution is due to the fact that no matchup can result in a point differential of zero (matchups that do will go to overtime). Therefore, this bin actually represents a decreased amount of outcomes. The distribution of spreads can also be seen in Figure 1. The spread is also approximately normal with a mean of 0.32 and a standard deviation of 6.95. The positive mean indicates that the spread favors the away team on average. This is not intuitive, as it is expected for the spread to take home court advantage into account – and is possibly an area that can be exploited. The variance of the spread is also less than the variance of the point differential.

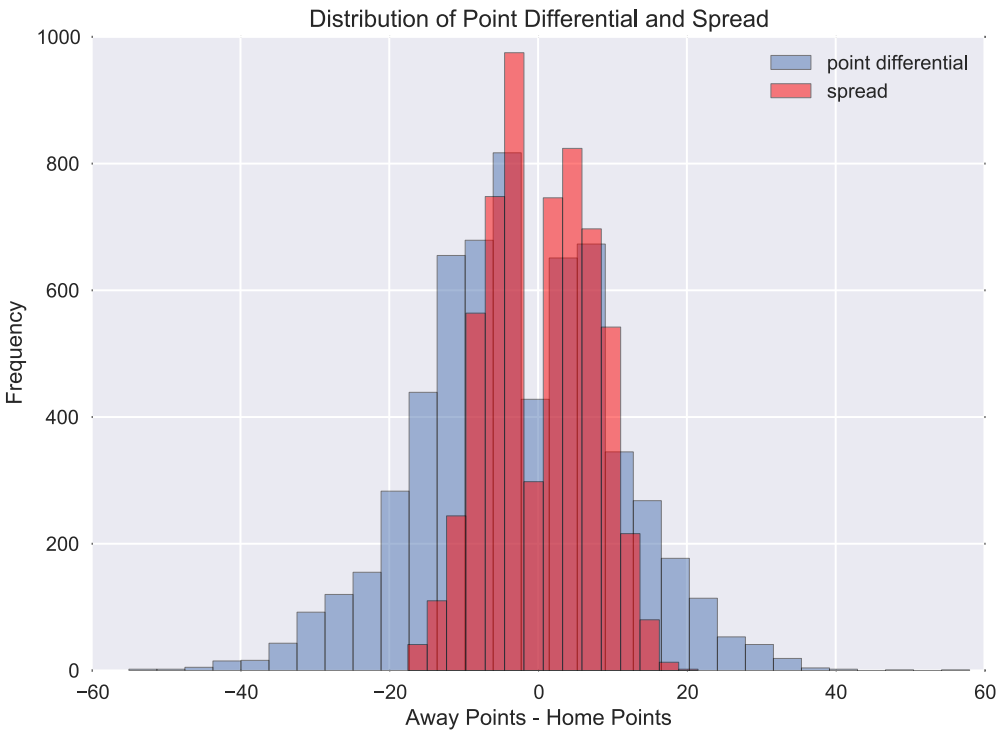

Figure 1

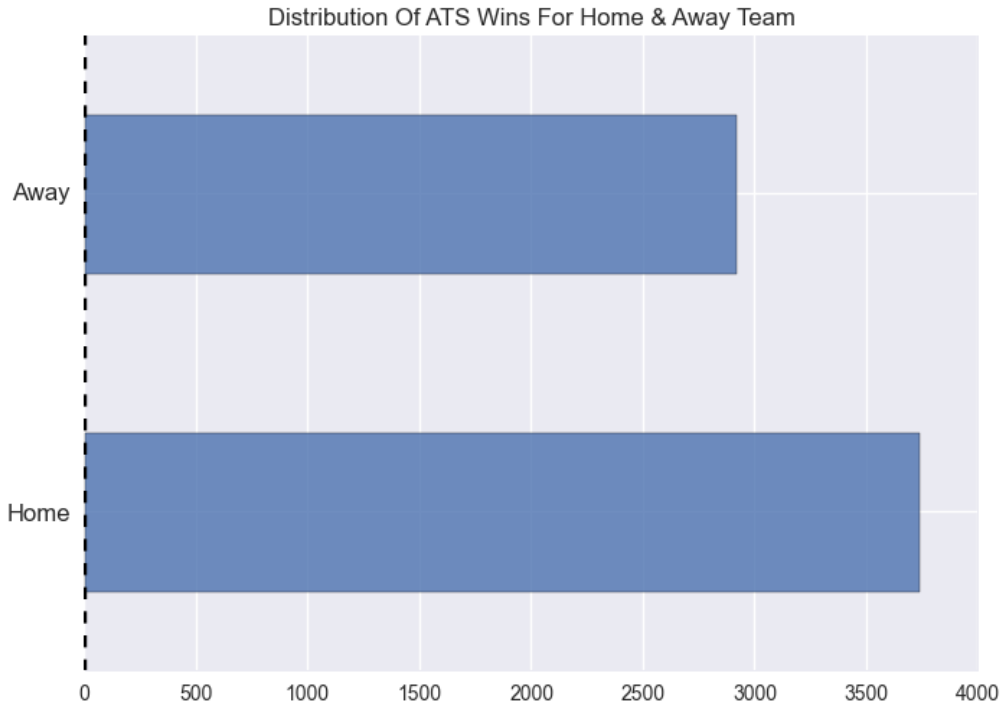

Figure 2

The distribution of ATS wins for the home team and the away team can be seen in Figure 2. The home team beat the spread in 3,741 out of the 6,662 matchups in the dataset, or 56.15%. This is not surprising knowing that the mean spread is 0.32 and the mean point differential is 2.88. This is an indication that the market favors the away team on average, and therefore, models that predict the home team to beat the spread slightly higher more than the away team is desirable.

2.2 Correlation Analysis

Because over 100 attributes are included in the dataset, a complete correlation matrix between all variables is impractical. Instead, this correlation analysis will focus on three mains areas: identifying the attributes that are most correlated with the target attributes and spread, identifying attributes that are highly correlated with each other (to avoid multicollinearity where relevant), and to identify which time frame (season to date, past 10 games, past 5 games, and past 3 games) is most relevant in predicting the ATS winner.

The attributes that are most highly correlated with home_win (a binary variable taking the value of 1 when the home team beats the spread and 0 when the away team beats the spread) are shown below.

Attribute	Corr(home_win)	Corr(point_diff)	Corr(spread)
home_win	1.00	-0.71	0.35
point_diff	-0.71	1.00	-0.03
spread	0.35	-0.03	1.00
biggest_lead_ratio	-0.20	0.31	-0.02
fg_pct_ratio	-0.18	0.27	-0.01
biggest_lead_ratio_l10	-0.16	0.28	-0.02
win_streak_difference	-0.15	0.23	-0.03
fg_pct_ratio_l10	-0.14	0.22	-0.02
biggest_lead_ratio_l5	-0.13	0.23	0.00
fg_pct_ratio_l5	-0.12	0.19	-0.03
tp_pct_ratio	-0.12	0.18	-0.01
def_rebounds_ratio	-0.11	0.18	-0.03
assists_ratio	-0.11	0.14	-0.01
fg_pct_ratio_l3	-0.11	0.16	-0.02
win_perc_ratio	-0.11	0.14	-0.04
fouls_ratio	0.11	-0.10	0.09
assists_ratio_l5	-0.10	0.10	-0.02

Table 2

The following conclusion are made from analysis of the above confusion matrix:

- The point differential and the home_win attributes are negatively related. This observation follows intuition by recalling that the point differential is defined as the number of points scored by the home team minus the number of points scored by the away team. Therefore, the point differential is negative and the home_win attribute is 1 when the home team beats the spread.
- The spread has a significant correlation with home_win and is not highly correlate4d with point_diff. This indicates when the spread is high (the away team is favored), the home team actually is more likely to win. This is a counter-intuitive and a potentially very interesting result.
- The correlation between the explanatory variables and the spread is much lower than the correlation between the explanatory variables and the two target variables – home_win and point_diff. This is an encouraging result as this shows that the spread is not determined using the explanatory variables used in the analysis. Therefore, if the explanatory variables are prove to be predictive of the target variables, the resulting models should have accuracies greater than 50%.

Next, the correlations between the explanatory variables are explored. It is important to note explanatory variables that are highly correlated with each other as this could lead to misinterpretations of some models. The explanatory variables that are most highly correlated are shown in Table 3. As seen in the table, all high correlations between explanatory variables are the same variables but measured over different time frames.

Attribute 1	Attribute 2	Correlation
perc_points_in_paint_ratio	points_in_paint_ratio	0.87
fast_break_ratio_l10	fast_break_ratio_l5	0.87
points_in_paint_ratio_l10	points_in_paint_ratio_l5	0.85
points_in_paint_ratio_l5	points_in_paint_ratio_l3	0.85
blocks_ratio_l10	blocks_ratio_l5	0.85

Table 3

As seen in the table, all high correlations between explanatory variables are the same variables but measured over different time frames.

2.3 PCA Analysis

A dimensionality reduction is done on the dataset via PCA. The total number of components created are 86. The screeplot is shown in Figure 3. The number of components shown in the screeplot are reduced from 86 to 10 because most of the variance is explained by the first few components. 90% of the variance in the data is explained by the first six components, and 95% of the variance in the data is explained by the first nine components.

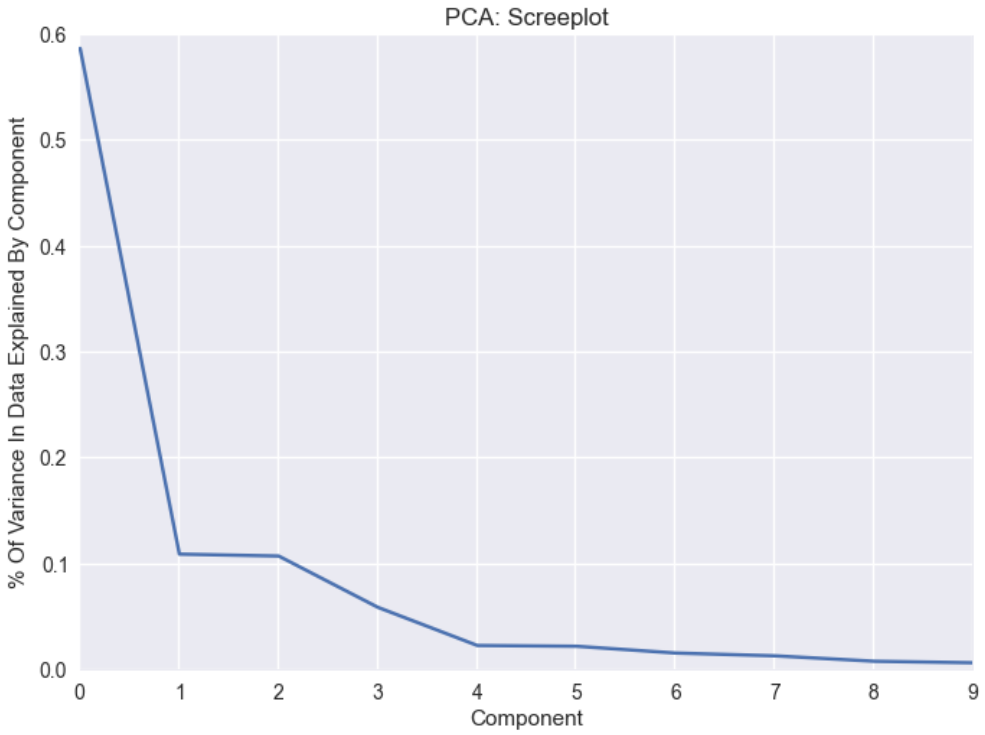

Figure 3

The ten attributes that influence the first principal component are shown in Figure 4. Shooting heavily influences this component. The component is positive when the home team scores more points from shooting the ball and when the away team has increased their scoring more than the home team over the last 10 games compared to the season. Principal component 2 is heavily influenced by turnovers and the difference between 1st and 2nd half scoring. The third principal component is influenced by the days of rest difference and the steals ratio.

Attribute	Principal Component 1
perc_points_by_ft_ratio_l10	31%
perc_points_in_paint_ratio_l10	30%
trend_fgpct_l10_vs_l3_ratio	28%
assists_ratio_l10	20%
blocks_ratio_l10	19%
trend_points_l10_vs_l3_ratio	19%
fast_break_ratio_l10	-15%
fg_pct_ratio_l10	-25%
margin_ratio_l10	-32%
ats_margin_ratio_l10	-33%
trend_points_season_l10_ratio	-43%

Figure 4

Attribute	Principal Component 2
turnovers_ratio_l3	57%
margin_half_vs_full_ratio_l3	56%
steals_to_blocks_ratio_l3	22%
perc_points_in_paint_ratio_l3	17%
fast_break_ratio_l5	16%
perc_points_by_ft_ratio_l3	14%
off_rebounds_ratio_l3	-17%
def_rebounds_ratio_l3	-18%
fast_break_ratio_l3	-20%
fg_pct_ratio_l3	-22%

Figure 5

Attribute	Principal Component 2
steals_ratio	46%
steals_to_blocks_ratio	15%
perc_points_in_paint_ratio	6%
margin_ratio	4%
trend_fgpct_season_vs_l10_ratio	2%
trend_fgpct_l10_vs_l3_ratio	-7%
perc_points_by_ft_ratio	-9%
off_rebounds_ratio	-12%
margin_half_vs_full_ratio	-19%
rest_difference	-83%

Figure 6

3 Models

A variety of machine learning algorithms are applied to the dataset to predict the against the spread winner for a given matchup. Both numerical and categorical prediction algorithms are fit to the data, where the numerical algorithms predict the point differential between the two teams (away score – home score) and the categorical algorithms predict whether the home team or the away team will beat the spread.

Each model is evaluated by simulating ATS predictions for every matchup in chronological order. For each day in the data set, a training set of data is created to represent all matchups before that particular day and a test set of data is created to represent all matchups for that day. The model is trained using the train set (historical matchups) and predictions are made for the test set (today's matchups).

3.1 Linear Regression

A linear regression model is fit to the target attribute point differential to predict the point differential for each matchup. The predicted point differential is compared to the spread to determine which team should be selected to win ATS. If the spread is greater than the predicted point differential, then the away team is selected (remember negative spread and point differential means that the home team is predicted to win). The model is evaluated by evaluating the Root Mean Squared Error (RMSE) and the accuracy from the simulation. The performance for each of the datasets is shown in Table 5.

Dataset	RMSE	Accuracy	Accuracy When Home Is Predicted	Accuracy When Away Is Predicted	Accuracy When Home Favorite	Accuracy When Away Is Favorite
MatchupsAllAtt	44.91	63.87%	59.59%	66.62%	56.63%	70.69%
MatchupsBE	**12.55**	64.14%	59.71%	67.16%	57.60%	70.27%
MatchupsPCA1	13.13	63.89%	59.96%	66.28%	56.42%	70.91%
MatchupsPCA2	13.16	64.00%	**60.12@**	66.35%	56.63%	70.95%
MatchupsPCA3	13.25	63.92%	**60.04%**	66.26%	56.42%	**70.98%**
MatchupsPCA6	13.62	63.80%	59.92%	66.14%	56.25%	70.91%
MatchupsPCA9	13.54	63.59%	59.67%	65.92%	55.94%	70.78%
MatchupsSTD	12.29	**64.25%**	59.91%	**67.19%**	**57.24%**	70.85%
MatchupsL10	16.03	64.06%	59.85%	66.79%	56.94%	70.73%

Table 5

The model with the attributes from the MatchupsBE dataset minimizes the RMSE, and is chosen to explore further. The coefficients with the largest absolute value are shown

in Figure 7. The attribute that contributes the most to the away team scoring points is the change in field goal percentage between the season and the last 10 games. The attribute that contributes the most towards the away team scoring points is the ratio between the fouls over the past 10 games.

Attribute	Coefficient
trend_fgpct_season_vs_l10_ratio	33.42
fg_pct_ratio_l10	33.12
def_rebounds_ratio	17.75
trend_fouls_l10_vs_l3_ratio	14.08
fouls_ratio_l3	13.97
trend_points_season_l10_ratio	7.33
perc_points_by_ft_ratio	6.93
turnovers_ratio	-6.94
fouls_ratio_l10	-17.17

Figure 7

A scatterplot of the actual point differential vs. the predicted point differential for this model is shown in Figure 8. As seen in the scatterplot, the predicted values and the actual values have a slight positive correlation, indicating that the model is predictive in predicting the point differential in a given matchup.

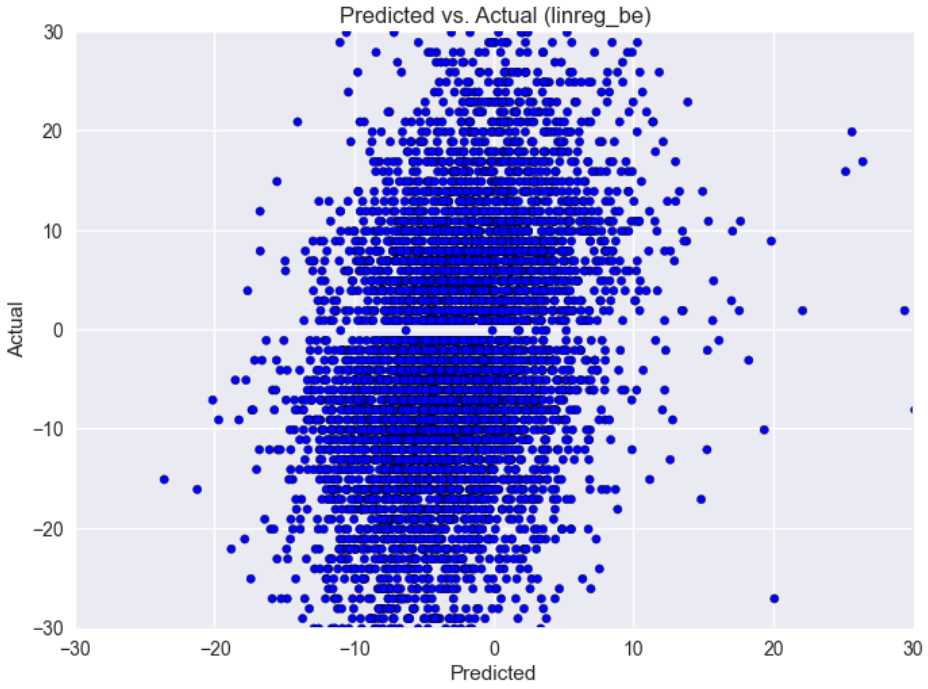

Figure 8

Finally, the residuals for the predicted values are analyzed in Figure 9. The residual is defined as the actual point differential minus the predicted point differential. The residual is highest for negative predictions (predict home wins by a lot) and lowest for positive predictions. Therefore, the absolute value of the prediction is too high compared to the actual amount. Therefore, the linear regression model could likely be improved by transforming point differential, most likely using a logarithmic scale. This is saved for future work.

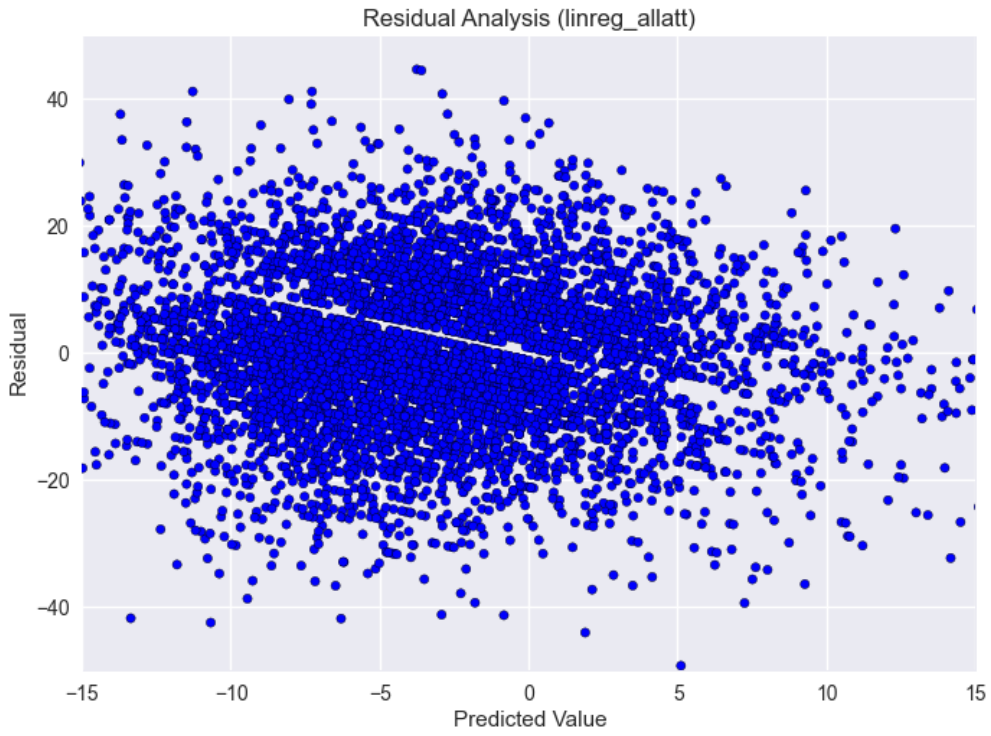

Figure 9

A confusion matrix of the binary-transformed predictions are shown in Figure 10. As shown in the figure, the accuracy when home is predicted is far greater than the accuracy when away is predicted. The model is also significantly better when the home team actually wins the ATS bet.

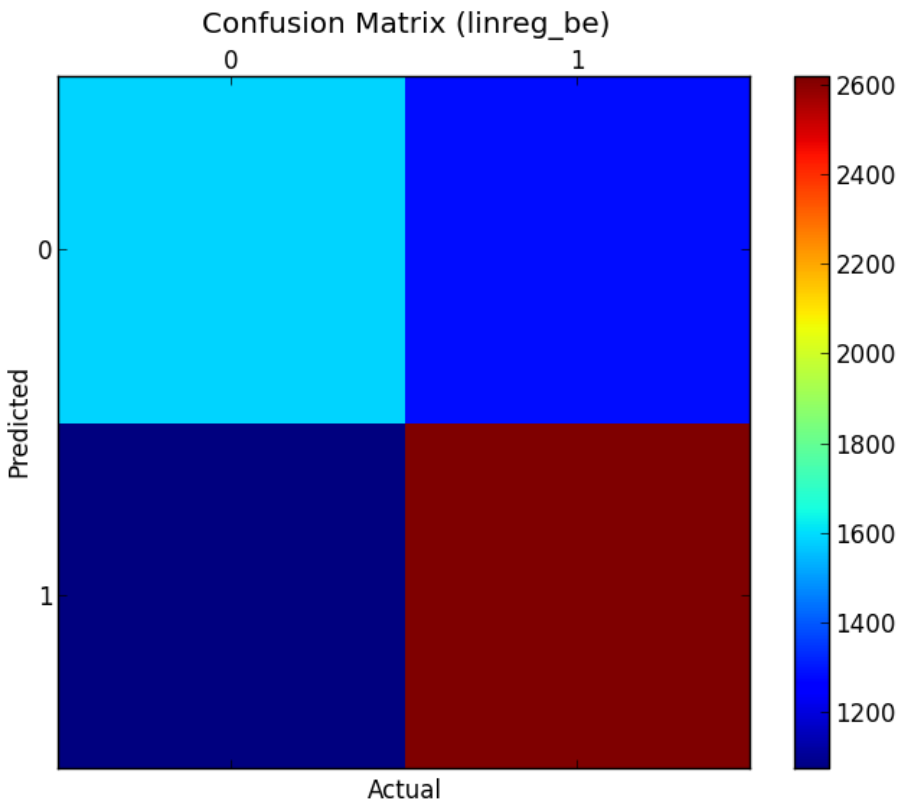

Figure 10

3.2 Logistic Regression

A logistic regression model is fit to the target attribute home_win (1 when home beats the spread, 0 when away beats the spread) to predict the ATS winner for a given matchup. The accuracy for the model over the simulation for all data sets is shown in Table 6.

Dataset	Accuracy	Accuracy When Home Is Predicted	Accuracy When Away Is Predicted	Accuracy When Home Favorite	Accuracy When Away Is Favorite
MatchupsAllAtt	63.64%	58.93%	66.92%	56.19%	70.65%
MatchupsBE	64.08%	59.18%	**67.71%**	57.06%	70.65%
MatchupsPCA1	64.17%	59.77%	67.10%	57.04%	70.88%
MatchupsPCA2	64.20%	59.79%	67.15%	57.11%	70.88%
MatchupsPCA3	**64.20%**	59.78%	67.16%	**57.14%**	70.85%
MatchupsPCA6	64.14%	**59.79%**	67.00%	56.90%	70.94%
MatchupsPCA9	64.02%	59.67%	66.87%	56.77%	70.85%
MatchupsSTD	64.03%	59.27%	67.49%	56.36%	**71.24%**
MatchupsL10	63.38%	58.55%	66.77%	55.30%	70.95%

Table 6

The confusion matrix can be seen in Figure 11. As shown in both the table and the below figure, the accuracy of the algorithm is much higher when home is predicted to win.

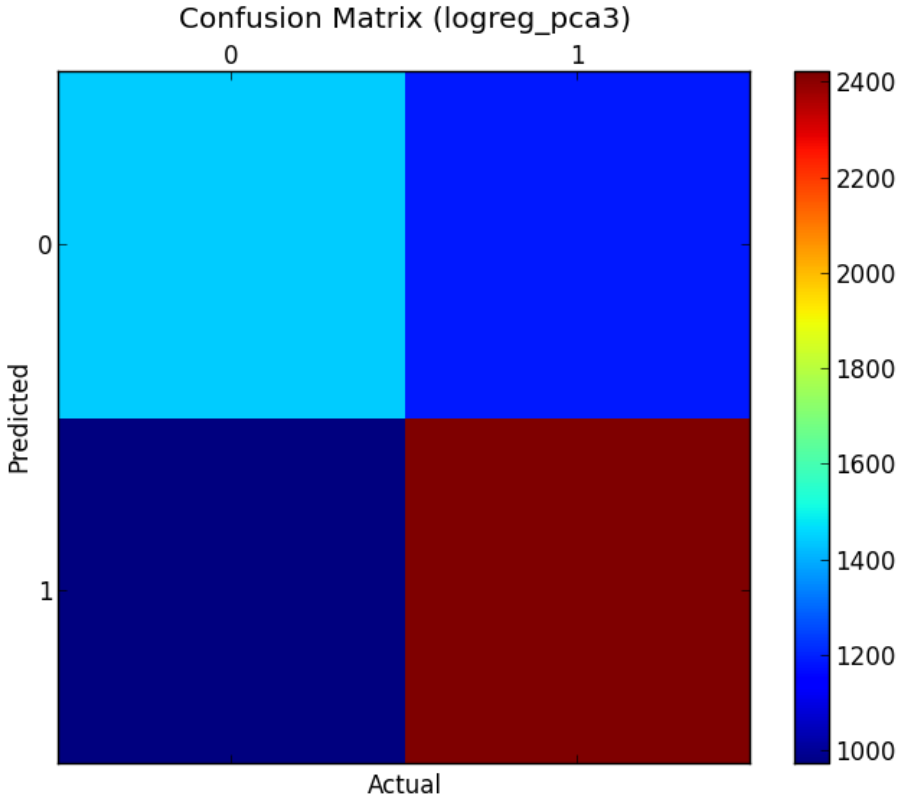

Figure 11

3.3 Decision Tree

A decision tree model is fit to the target attribute home_win to predict the ATS winner for a given matchup. The performance for the model over the simulation for all data sets is shown in Table 7.

Dataset	Max Depth	Accuracy	Accuracy When Home Is Predicted	Accuracy When Away Is Predicted	Accuracy When Home Favorite	Accuracy When Away Is Favorite
MatchupsAllAtt	2	64.60%	57.35%	75.52%	58.13%	71.23%
MatchupsBE	2	64.49%	56.82%	76.17%	58.58%	70.03%
MatchupsPCA1	2	**64.65%**	**57.91%**	72.08%	**58.68%**	**70.27%**
MatchupsPCA2	2	64.65%	57.89%	72.11%	57.78%	70.17%
MatchupsPCA3	2	64.47%	57.61%	72.30%	58.55%	70.04%
MatchupsPCA6	2	64.45%	57.42%	72.87%	58.58%	69.98%
MatchupsPCA9	2	64.52%	56.37%	73.39%	58.44%	70.24%
MatchupsSTD	2	64.09%	56.69%	**74.64%**	58.11%	69.72%
MatchupsL10	2	63.92%	56.70%	73.33%	57.76%	69.69%
MatchupsAllAtt	5	62.56%	56.74%	67.60%	55.56%	69.43%
MatchupsBE	5	63.09%	56.89%	69.17%	55.61%	70.09%
MatchupsPCA1	5	64.22%	57.20%	72.65%	58.44%	69.65%
MatchupsPCA2	5	64.40%	57.43%	72.63%	58.27%	70.17%
MatchupsPCA3	5	63.95%	56.96%	72.36%	58.20%	69.36%
MatchupsPCA6	5	63.21%	56.73%	69.91%	56.63%	69.40%
MatchupsPCA9	5	62.99%	56.56%	69.57%	55.74%	69.82%
MatchupsSTD	5	63.39%	57.25%	69.36%	56.39%	69.96%
MatchupsL10	5	62.97%	57.23%	67.84%	55.53%	69.93%

Table 7

The dataset with just one PCA attribute is the best performing decision tree algorithm, with an accuracy of 64.65%. A visualization of the tree is shown in Figure 12. As the visualization shows, although there is a tree depth of 2, there is only one decision being made. If the value of PCA1 is less than 2.75, then home is predicted. If not, away is predicted.

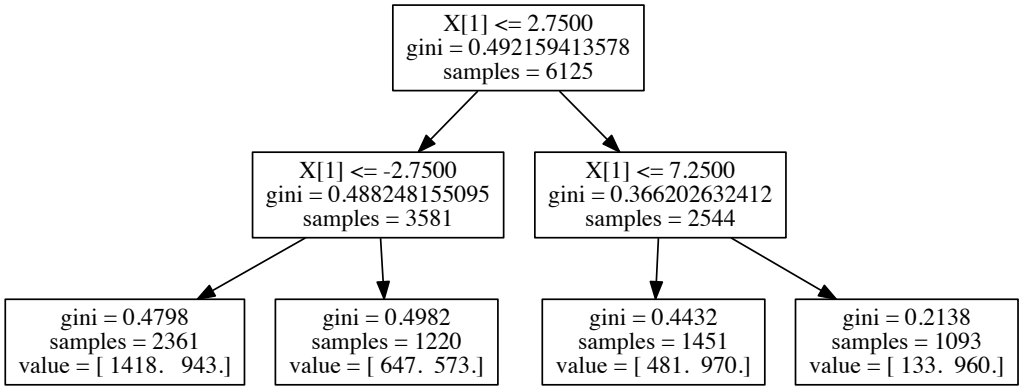

Figure 12

Finally, the confusion matrix for the decision tree is shown in Figure 13. Unlike the previous two models, the decision tree model is more accurate when the away team is predicted. Similar to the past two models, the accuracy is best when the home team actually wins the matchup.

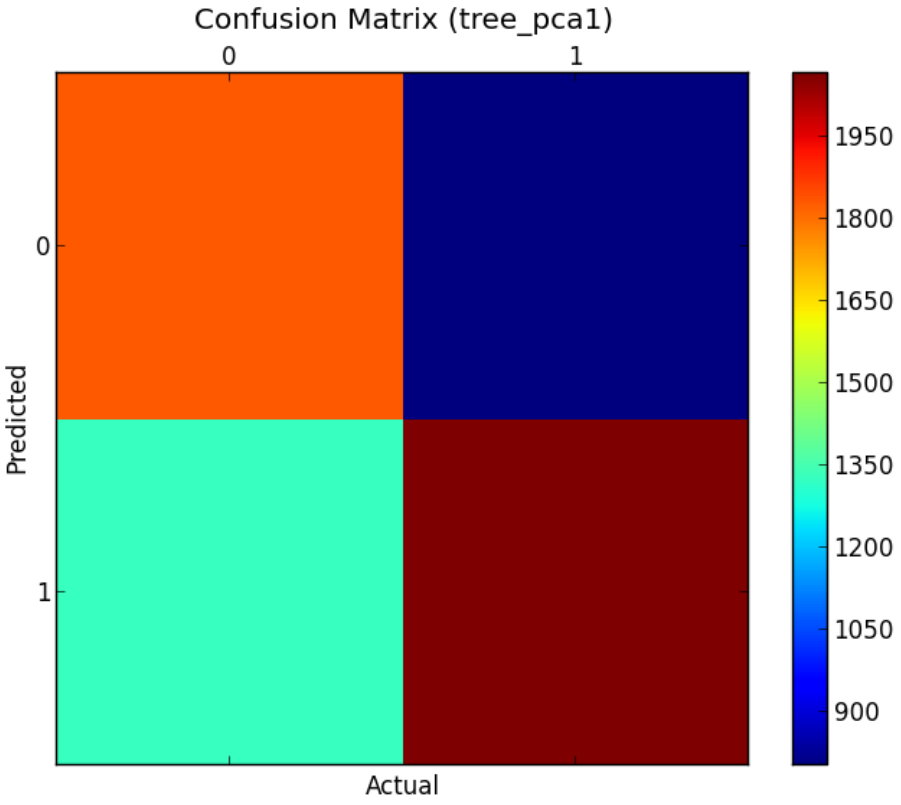

Figure 13

4 Conclusions

Three different models are created to predict the ATS winner for any given matchup in the NBA. The overall accuracy is greater than 60% for all models, which indicates that the models are in fact predictive of the winner of any NBA matchup. All models were evaluated by retraining the model with historical data before predicting the results of the matchups for all days, thus simulating the exact decisions that would be made if this implementation were followed since the beginning of the 2008 season. The best performing model is a decision tree with max depth of 2 on the PCA reduced data set with one component, yielding an accuracy of 64.5%. An analysis is done to calculate the return on the balance if the algorithm is followed exactly from the beginning of the 2008 season through today. With a betting strategy of betting 5% of the overall balance on each game, a \$1 balance grew to \$3.5 x 10^{23}. This return on investment sounds unreasonable, but it is not when you consider that the expected value of a \$1 bet is \$1.22 when the winning percentage is 64%. This results in 1.22% growth on each bet, and therefore, grows exponentially as shown in Figure 14.

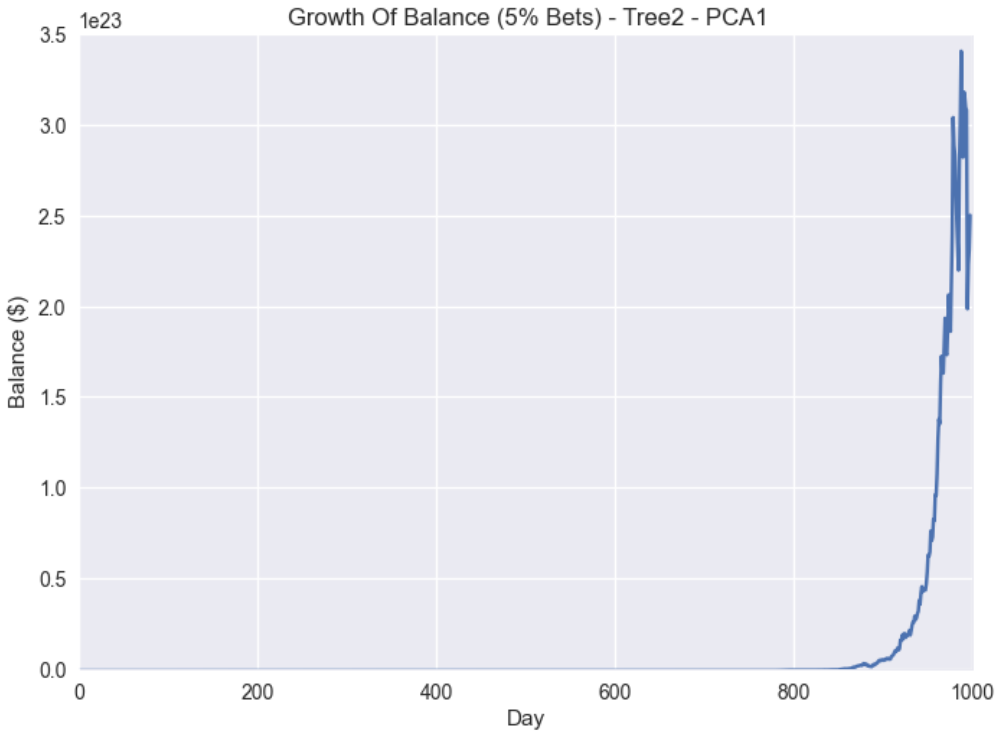

Figure 14

5 Python Scripts

Two main Python scripts are used for the analysis. The first is used to preprocess the data and is called data_main.py. The second runs all simulations to evaluate the performance of the algorithms and is called main.py.

5.1 data_main.py

All code in this module is related to creating the matchups_full.csv file to be used in the analysis. The first method is called 'scrape2csv_gamelogs'. This method iterates through webpages on oddsshark.com and extracts relevant data using the BeautifulSoup module and regular expressions. The second method is called 'clean.sdql' and iterates through the raw_sdql.csv file and change the team names to match the team names extracted from oddsshark.com and saves the results to raw_sdql_clean.csv. The third method is called 'create_team_profiles'. This method iterates through each row in the raw_sdql_clean.csv and transforms the attributes into cumulative attributes as described in Section 2. The reduce_team_profiles method chooses only the most relevant attributes to go forward with. The clean_team_profiles method removes rows containing infinite or NaN values. The merge_matchups_profiles method merges the raw_covers.csv file and the sdql data. Finally, the add_attributes_to_matchups method creates ratio attributes so that each row describes the difference in performance between the two teams and not just the performance of the two teams individually.

5.2 main.py

This module utilizes many third party tools in Python. First, pandas is used extensively to represent the data in DataFrame and Series formats. This is particularly convenient for merging data and iterating through rows. Sci-kit learn (sklearn) is used extensively to apply machine learning models and evaluate these models. Pydot is used to create the decision tree visualization. Matplotlib is used for all plotting. Seaborn is a module that automatically creates better looking plots than the Matplotlib standard plots and is used for all plots. One caveat of the Seaborn module is that is does not do well with confusion matrix plots, and therefore, the script had to be run a second time without using the Seaborn module to extract the confusion matrix plots.

The first section in this module is Data Preprocessing. The first method is called attribute_selection, and goes through a given DataFrame and extracts the relevant features which are chosen based on the parameters that are passed. Finally, this drops all rows with infinite or NaN values. The second module is called partition_test_set and is used to partition a data frame into a training and test set. This method is not used in the analysis, but was used quite a bit in the early parts of the project when the models were evaluated by splitting a train and test set. The next two modules are normalize_min_max and normalize_zscore. Both of these modules take in a DataFrame as input and return the same DataFrame but normalized. If the parameter

targets is equal to True, then the method takes into account target variables. This method will also normalize the target variables if that parameter is marked to be true. Finally, the method create_pca_set takes a DataFrame and the percent of explained variance desired as input and returns the same DataFrame but transformed into a PC-reduced space.

The Data Exploration contains all the methods that are used in Section 2.

Skipping to the Main section, the data is first loaded into a pandas DataFrame. Various other dataframes are created using different types of feature selection. The models of the linear regression and logistic regression models are then saved to a CSV file for further analysis.

Next, the data set and the models to be evaluated are written to a list for modular evaluation. For each combination of data set and machine learning algorithm, the simulate_game_stream method is called and returns a DataFrame consisting of every matchup and the predicted values along with the model. The DataFrame is then sent to the evaluate method and various performance metrics are calculated. If the algorithm is a numerical predictor, then the RMSE is calculated, if not, this is skipped. The performance for each algorithm is saved to a list and then converted into a DataFrame for further evaluation.

Appendix A: Data Schema And Description

Attribute	Description
date	The data that the matchup occured
point_diff	The points scored by the away team minus the points socred by the home team
home_win	Binary classifier – 0 when away team wins ATS, 1 when home team wins ATS
spread	The amount of points the away team is predicted to win the matchup
biggest_lead_ratio*	Average big lead per game for away team divided by average big lead per game for home team
fg_pct_ratio*	Field goal percentage for away team divided by field goal percentage for home team
win_streak_difference*	Away team win streak going into game minus home team win streak going into game
def_rebounds_ratio*	Average defensive rebounds per game for away team divided by average defensive rebounds per game for home team
win_perc_ratio	Season win percentage for away team divided by season win percentage for home team
fouls_ratio*	Fouls per game for away team divided by fouls per game for away team
ats_margin_ratio*	Average margin of victory/loss ATS for away team divided by average margin of victory/loss ATS for home team
ats_win_streak_difference*	Away team win streak ATS going into the game minus home team win streak ATS going into the game
avg_spread_per_game_ratio*	Average number of points away team is favored by divided by average number of points home team is favored by
blocks_ratio*	Average blocks per game by away team divided by average blocks per game by home team
fast_break_points_ratio*	Average fast break points per game for away team divided by average fast break points per game for home team
margin_half_vs_full_ratio*	Ratio of (Average margin at half / average margin for full game) for away team and home team

margin_ratio*	Ratio of average margin per game between away and home team
off_rebounds_ratio*	Ratio of average offensive rebounds per game between away and home team
perc_points_by_ft_ratio*	Ratio of (% points scored by free throws) between away and home team
rest_difference*	Difference between the amount of days off before the matchup between away and home team
steals_ratio*	Ratio of average steals per game between away and home team
steals_to_blocks_ratio*	Ratio of steals to block ratio between away and and home team
total_rebounds_ratio*	Ratio of average rebounds per game between away and home team
tp_pct_ratio*	Ratio of average three point percentage between away and home team
turnovers_ratio*	Ratio of turnovers per game between away and home team
trend_fgpct_l10_vs_l3_ratio	Ratio of (field goal percentage over last 10 games / field goal percentage over last 3 games) between away and home team
trend_fgpct_season_vs_l10_ratio	Ratio of (field goal percentage season to date / field goal percentage over last 10 games) between away and home team
trend_fouls_l10_vs_l3_ratio	Ratio of (fouls per game over last 10 games / fouls per game over last 3 games) between away and home team
trend_fouls_season_vs_l10_ratio	Ratio of (fouls per game season to date / fouls per game over last 10 games) between away and home team
trend_margin_l10_vs_l3_ratio	Ratio of (margin per game over last 10 games / margin per game over last 3 games) between away and home team
trend_margin_season_l3_ratio	Ratio of (margin per game season to date / margin per game over last 3 games) between away and home team
trend_margin_season_l5_ratio	Ratio of (margin per game season to date / margin per game over last 5 games) between away and home team

* Indicates that the ratio is recorded for season to date, over the last 10 games, over the last 5 games, and over the last 3 games.

Appendix B: SDQL Query

date, team, season, game number, assists, ats margin, ats streak, attendance, biggest lead, blocks, conference, defensive rebounds, fast break points, field goals attempted, field goals made, fouls, free throws attempted, free throws made, line, losses, margin, margin after the first, margin after the third, margin at the half, offensive rebounds, ou margin, ou streak, overtime, playoffs, points, points in the paint, quarter scores, rebounds, rest, steals, streak, team rebounds, three pointers attempted, three pointers made, time zone, total, turnovers, wins @ season=2014

Appendix C: Backward Elimination

```
Step:  AIC=30230.59
point_diff ~ biggest_lead_ratio + win_streak_difference + def_rebounds_ratio +
    ats_win_streak_difference + blocks_ratio + fast_break_ratio +
    perc_points_by_ft_ratio + perc_points_in_paint_ratio + steals_ratio +
    turnovers_ratio + trend_fgpct_season_vs_l10_ratio +
trend_fouls_l10_vs_l3_ratio +
    trend_fouls_season_vs_l10_ratio + trend_margin_season_l3_ratio +
    trend_points_season_l10_ratio + fg_pct_ratio_l10 + fouls_ratio_l10 +
    ats_margin_ratio_l10 + blocks_ratio_l10 + tp_pct_ratio_l10 +
    steals_ratio_l5 + steals_to_blocks_ratio_l5 + fouls_ratio_l3 +
    fast_break_ratio_l3 + perc_points_in_paint_ratio_l3 + steals_ratio_l3

                                  Df Sum of Sq    RSS   AIC
<none>                                         871500 30231
- ats_win_streak_difference        1     340.7 871841 30231
- trend_margin_season_l3_ratio     1     375.1 871875 30231
- fast_break_ratio_l3              1     397.5 871898 30231
- ats_margin_ratio_l10             1     423.7 871924 30232
- blocks_ratio                     1     444.5 871945 30232
- perc_points_in_paint_ratio       1     510.9 872011 30232
- steals_to_blocks_ratio_l5        1     586.7 872087 30233
- tp_pct_ratio_l10                 1     767.7 872268 30234
- trend_fouls_season_vs_l10_ratio  1     812.9 872313 30234
- fouls_ratio_l3                   1     998.1 872498 30236
- trend_fouls_l10_vs_l3_ratio      1    1065.4 872566 30236
- trend_points_season_l10_ratio    1    1120.2 872620 30236
- steals_ratio_l3                  1    1233.7 872734 30237
- blocks_ratio_l10                 1    1253.2 872753 30237
- perc_points_in_paint_ratio_l3    1    1368.0 872868 30238
- turnovers_ratio                  1    1518.1 873018 30239
- fouls_ratio_l10                  1    1551.8 873052 30239
- fast_break_ratio                 1    1834.7 873335 30241
- steals_ratio_l5                  1    1862.9 873363 30242
- trend_fgpct_season_vs_l10_ratio  1    1927.9 873428 30242
- steals_ratio                     1    1957.5 873458 30242
- def_rebounds_ratio               1    2968.4 874469 30249
- win_streak_difference            1    3079.1 874579 30250
- perc_points_by_ft_ratio          1    4753.8 876254 30262
- fg_pct_ratio_l10                 1    5924.8 877425 30270
- biggest_lead_ratio               1   17018.1 888518 30346
```