Can Machine Learning predict NBA Playoff games with enough Accuracy to Overcome the Bookmaker's Advantage?

Ryan Donovan
x16104269
MSc Research Project in Data Analytics

16th August 2017

Abstract

This project uses machine learning (ML) algorithms to predict the outcomes of NBA games as a binary label (Win/Loss) along with the probability of that given label. It aims to use the probabilities found to analyse the results of betting simulations using these probabilities against average bookmaker odds. The algorithms used are: ANN, Bagging, Boosting, kNN, Logistic Regression, Naïve Bayes, Random Forest, and SVM. 10-Fold Cross Validation (CV) is used along with Bayesian Hyperparameter Optimisation (BHO) to tune hyperparameters of each of the algorithms. Four Feature sets are created for each season that are made using NBA boxscore data along with locational data. One feature set is created using feature selection via the GINI index from a random forest algorithm. Seventeen seasons of NBA data is used (2000 - 2016) and 9 years of betting data is used (2000 - 2008). R and Google Sheets are the software that implement all processes in the study. The project follows the CRISP-DM methodology in all steps.

1 Introduction

The NBA is a major sport in America that started in 1946. Most people watch it for entertainment to relax. Some like to talk to it with their friends as well. There are also some that like to place bets on games. Betting is big businees in the United States. (Vaz de Melo et al.; 2008) state that there was $2.4 billion legal sports bets in Las Vegas in 2006 alone and roughly $380 billion in illegal sports bets in 1999. Some people bet on games by guessing and others use data analysis to predict outcomes to get a better edge. These betters believe they can outperform the bookmaker's odds to get a money edge. Machine learning (ML) offers a way to maybe beat the bookmakers at their own game. ML algorithms can be used to predict outcomes by training on past data to predict future unseen data. The higher the accuracy means there is better chance at predicting an outcome. Nine algorithms will be used in this study to train algorithms on past data (regular season) to predict on future data (playoff games) to see if money could be made when comparing the results against bookmaker odds.

It is tough to beat a bookmaker as they skew their odds in their favour. Take a simple example of a person placing a bet on whether a coin toss would be heads or tail. If the bookmaker was fair, he would offer 2 to 1 odds. So if you placed a $1 bet on a heads outcome, you would get paid $2 if you won. You'd get $0 if you lost. In the long run you would break even. The bookmaker makes no money in this situation so he skews it so you do not get paid back a good return. They might offer 1.5 to 1 odds instead. You would lose $0.25 of your $1 bet on average. If a person was smart they would convince the bookmaker to give a higher odds if they could, say 3 to 1. Here a player would make $0.50 each $1 bet over the long haul, but it is doubtful any bookmaker would agree. One way to beat the bookmaker would put an ”unfair” coin in the mix that is skewed to have 70% heads and 30% tails unbeknownst to them (unrealistic, but proves a point). If the bookmaker offered 1.50 to 1 odds payout for heads you would make money in the long run because the odds to break even would be 1.43 to 1 (found by taking 1/70%). A model of NBA predition could possibly predict a game with more accuracy then the bookmaker's odds say and a person could earn money.

In the past it was difficult to gather NBA data to analyse because it was not easily accessible and had to be scraped from various websites. Thankfully, the NBA releases all stats from every season for free on a publicly accessible website. With the NBA data and betting data hopefully a positive return on the virtual betting will be positive. All the studies found in literature have only predicted the outcomes of games. None have tried to use their predictions to actually see if they would give a return when betting. Cheng et al. (2016) concluded their study by pointing out it would be interesting to see how well predicted labels could perform against a bookmakers odds. Note that if the bookmaker's used the same process for prediction as is in this study in would be nearly impossible to win because they would skew their payout odds in their favour.

2 Related Work

Machine Learning (ML) has been used in many sports like basketball, football, cricket and many others. It has also been used for prediction in finance and science fields. This review focuses on past studies from many domains that used prediction via ML algorithms used in this project. Most will focus on past NBA studies as this is the domain this project analyses. This review will begin with a brief history of early analysis of the NBA, followed by review of studies that use the algorithms from this project, and finishing up with studies that use feature selection.

2.1 Introduction

A few early studies in looked for features that explained how teams could reach their full potential Zak et al. (1979) and Hofler and Payne (2006). Bhandari et al. (1997) searched for patterns in NBA data using their Advanced Scout software using a technique called Attribute Focusing. It finds interesting/meaningful patterns in data for coaches to use to lead their teams more effectively. Though none of these early studies predicted outcomes of games, they added benefit by highlighting import features that could influence team performance as will be discussed in in feature selection later on.

2.2 Algorithms

There are nine ML algorithms used in study to predict the outcomes. Six will be discussed independently and three will be discussed together as they are all ensembles. This section will finish with a review of the algorithm Bayesian Hyperparameter Optimisation (BHO) of which is used to tune the hyperparameters all the algorithms.

Artificial Neural Networks (ANN) McCabe and Trevathan (2008) used an ANN modelled by a multi-layered perceptron with back propagation to predict outcomes of football and rugby games. Using features from recent team performances and standings in league they predicted outcomes with accuracy ranging from 54.6% to 65.1%. This is of use as division and conference standings will be used in this project. Since ANNs can have high compute time, Zhang (2000) warned that important features should be selected and unimportant ones should be eliminated to reduce compute. Loeffelholz et al. (2009) used ANNs on NBA data and reduced their features to only four and found lower compute and accuracy as good as when many more features were used. They used feed forward, radial basis, probabilistic and generalised regression neural nets and got an average accuracy of 74%. Cheng et al. (2016) used a back propagation ANN and got from 50.6% to 66.0% accuracy over eight seasons. This disagrees with the results given by Loeffelholz et al. (2009). They used an average of last 10 games for their features as opposed to the last six games as done by Cheng et al. (2016). Also, Loeffelholz et al. (2009) used one season as training (using CV) and the other used regular season and playoffs for each set. Beckler et al. (2013) used a feed forward ANN with a 65% accuracy without optimisation as will be used in this project (BHO). They also noted the problem of ANN overfitting. Cheng et al. (2016) stated the ANN tended overfit because of small amount of examples in each run (roughly 1,200 games).

Decision Trees (D-Trees) Lantz (2013) state that D-Trees are very useful because they can be easily understood as opposed to black box methods like ANN or SVM. If positive results surface they might be able to be used to better understand what patterns predict game outcomes. Joseph et al. (2006) used D-Trees in football prediction but found they didn't perform as well as kNN, but they did achieve a 67.1% accuracy. Zdravevski and Kulakov (2010) used them for two NBA seasons, the first being used for training and tuning of hyperparameters and the second for testing. They used a few different D-Tree algorithms and got accuracies ranging from 61.2% to 71.2%. They did find Logistic Regression to be a better performer at 72.8% accuracy. This project will do divide train and testing sets for regular season and playoffs season as done by Cheng et al. (2016). Soto Valero (2016) used them to predict baseball games with 57.86% accuracy which was slightly under SVM accuracy at 58.9%. Hasbun et al. (2016) used D-Trees to predict the binary outcome of whether a student would dropout of school or get a degree with a 79.3% accuracy. Cheng et al. (2016) stated they have similar problem of ANN as they can overfit because of the limited size of the input data set.

K-Nearest Neighbours (kNN) Witten et al. (2016) states that normalisation should be done using kNN to avoid attributes with higher ranges to overpower attributes with smaller ranges. Lantz (2013) also state that normalisation should be done and they use Max/Min Normalisation as this project will do. Joseph et al. (2006) used kNN to predict the outcome of football games with 77.36% accuracy over 2 seasons for one football club. They had three labels: Win, Loss, or Tie but stated kNN would still work on binary outcomes. They found kNN predicted better when comparing to Naïve Bayes and D-Trees Also, they and Lantz (2013) used Euclidean distance because all the data was numeric. Watcharapasorn and Kurubanjerdjit (2016) predicted a positive and negative mortality rate using kNN and found it worked better than D-Tree implementations at an accuracy of 88.75%. Zhang et al. (2017) used multiple different k-values in their research to get the lowest RMSE in order to choose optimal k. They used grid search which is slower than using BHO as is used in this project.

Logistic Regression Zdravevski and Kulakov (2010) found that logistic regression performed best out of all their classifiers at 72.8% accuracy when predicting NBA outcomes. This gives good reason to see how well it works in this project. Beckler et al. (2013) found that it worked better than SVM and ANN with a 68% accuracy on NBA outcomes. Cheng et al. (2016) found that it sometimes worked better or worse to Naïve Bayes, ANN, and Random Forest over a 7 year period but it was not a clear winner. They also stated that it was the most stable algorithm because accuracy varied little over the seasons. This could be useful in betting as it might reduce variance in betting returns.

Naïve Bayes (Bayes) Joseph et al. (2006) used Bayes in football prediction with a 61.83% accuracy. Kampakis and Thomas (2015) used Bayes to predict cricket matches by reducing 50 features into 4 by using recursive feature selection. They found the reduction helped Bayes perform better and further warrants use of feature selection. Cheng et al. (2016) used Bayes and found that it performed the worst as compared to logistic regression, ANN, and D-Trees. They noted that the lack of independence between variables probably caused the poor performance thus PCA might benefit this algorithm. Miljković et al. (2010) and Hofler and Payne (1997) also noted the problem of independence. Lantz (2013) note that independence usually is not a major issue though.

Support Vector Machines (SVM) Soto Valero (2016) used a SVM to classify baseball games and found they were able to predict baseball games with a 58.92%, which was better than 1NN, ANN and D-Tree. Demers (2015) used SVM to predict probabilities of wins in hockey. These probability predictions will be used similarly in this project. Miljković et al. (2010) tried SVMs and used normalisation. They found it performed worse than Bayes, but still predicted at 67% accuracy. The SVM model used by Beckler et al. (2013) predicted NBA outcomes with a 66% accuracy. Cao (2012) stated they got 67% accuracy predicting NBA games using them.

Ensemble Methods: Bagging, Boosting and Random Forest Bock (2016) used bagging to predict probability of a turnover in the NFL as this project will do. They also used 10-Fold CV along with averaging of ten trials. This will be used in this project as well. Schapire and Singer (1999) stated that boosting algorithms work well big data sets but small data sets can overfit. This project will watch out for this issue. Cheng et al. (2016) used random forests in their prediction of NBA games and found it had the second best accuracy as compared to their Maximum Entropy model. Pathak and Wadhwa (2016) found random forests had accuracy prediction of 60.02% in cricket matches which was close to SVM and Bayes.

Bayesian Hyperparameter Optimisation (BHO) BHO is used in this project to tune all the hyperparameters as opposed to Grid search. Lévesque et al. (2017) used BHO to tune a kNN, SVM, D-Tree, Random Forest, adaBoost and other algorithms as will be done in this project. Eggensperger et al. (2013) tested BHO on Logistic Regression and

ANN and found the hyperparameters were not significantly different from the actual best hyperparameters that are found through an exhaustive grid search. Snoek et al. (2012) used it on a SVM and convolutional neural networks. They found it performed faster than grid search by beating a state of the art model by 3% better error reduction. Feurer et al. (2015) stated it worked well on small amount of hyperparameteres and on SVM and Random Forest and others. This is positive because all algorithms in this project have less than three hyperparameters.

2.3 Feature Selection

Feature selection can be very important in ML as it can reduce compute time, noisiness of data and the dependence between features. Other research has found some of the features that can influence the outcome of games. Zak et al. (1979) found that home teams won more than average than road teams and this project will have feature set organized to reflect home and away teams. Hofler and Payne (2006) found that shooting, rebounding, steals and blocked shots increase a team's winning potential, while turnovers do the opposite. Zdravevski and Kulakov (2010) winning streak, fatigue, home and away records all influenced a team winning or not. Fatigue might be caused by distance travelled between games and by the number of rest days between the games, so these are added to this project. They took average of last 10 games of season and eliminated the first 10 games because they did not represent last 10 games. Kampakis and Thomas (2015) found reduction of data via PCA and other feature selection techniques allowed their Bayes to perform better. Zhang (2000), Page et al. (2005), Kampakis and Thomas (2015) and Loeffelholz et al. (2009) all used PCA to reduce the amount of features. Beckler et al. (2013) found features that predicted a team to lose often were a small amount of defensive rebounds, high amount of points blocks, and assists made by opposing team. They also noted that defensive stats were very important in win, like defensive rebounding and opposing team field goal percentage. Merritt and Clauset (2014) and Waggoner et al. (2014) found that winning streaks have some predictive power so this will be used as an feature. Hofler and Payne (2006) found that shooting percentage, rebounding, steals and blocked shots increase a team's winning while turnovers do the opposite. Strength of schedule is important in predicting the outcome of games as reported by Gumm et al. (2015). Therefore winning percentage against above 500 teams and division and conference will be made into features.

All these studies point to the features that will be created in this project. This project will take all these ideas to make feature sets. There will be four feature sets in this project and one will use random forest selection as Saraswat and Arya (2014) and Pancerz et al. (2016) used in their studies.

3 Methodology

The CRISP-DM method (CRoss-Industry Standard Process for Data Mining) methodology model will be used in this study as shown in Figure 1 from Azevedo and Santos (2008). The six steps named by Chapman et al. (2000) will be discussed in next six sections.

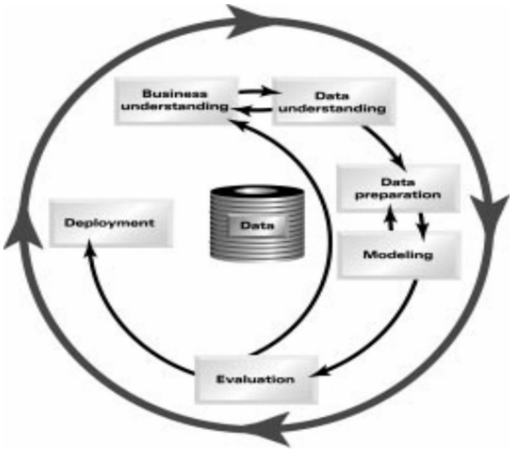


Figure 1: CRISP-DM Methodology

3.1 Business understanding

Business understanding points the project in a distinct direction by stating the overarching question to be answered, along with other smaller questions that might be needed to answer. These questions point the project towards the data needed to answer them. The main question of this project is:

> *Can Machine Learning using Bayesian Hyperparameter Optimisation predict NBA playoff game outcomes with high enough Accuracy to produce a Positive Return on Investment against a Bookmaker's Payout Odds?*

Other smaller questions to be answers are stated in following questions. How well do algorithms predict a game outcome? Do different feature sets predict with higher accuracy? Are ensembles better predictors than algortihms on there own in this project? The evaluation methodology section will go over actual case studies that will be performed to answer some of these questions.

NBA box score data will be used to create all NBA features as previous studies have done. They will be used to create four feature sets to predict wins and losses of a home team. All this data is freely available. The seasons from 2000 to 2016 (17 seasons) were chosen for box score data. Betting data giving odds for each game is needed in order to get a bookmaker's odds for each game. Betting data was available for seasons starting in 2008 to 2016 (9 seasons). Locational data for each stadium is also needed as the distance travelled by a team feature is needed. These data sets are gathered to facilitate the next step, Data Understanding.

3.2 Data Understanding

This part involves exploration of the data to get a better understanding of it. It can help in finding errors, patterns or highlights. It can also point the project in some new and interesting directions if an exciting pattern is found. Errors that were found are cleaned up using various techniques. The data that was gathered was put into same form as used by Cheng et al. (2016). They took the average of last 6 games of a season for their features and used 14 of the 15 features found in boxscores from NBA website. An early investigation sought to check if taking different averages over previous games could aid in better prediction. Averages for games from 6 to 10 were chosen and performed on all nine algorithms discussed in Related Work. The last six were found to be the best performer and this pointed the rest of this project to use the last 6 game average in all feature sets. This early exploration also eliminated D-Trees and Boosting algorithms as they performed 10% worse on average as compared to the other 7 algorithms. Note that all the exploration was done on the regular season and did not touch the playoffs.

3.3 Data Preparation

This phase involves cleaning and transforming data into its final form to be used in the ML models. It also involves the selection of features to be use for those models.

Cleaning Cleaning involved checking for errors in data-type and ranges. The ranges tell if there are any outliers. For instance, a team scoring 1200 points or a bet offering 100 to 1 odds would definitely be wrong as they never happen. If these were found, they were corrected by checking original sources to see if there was a difference. If they both agreed then another outside source would be used to find correct information.

Transformations The NBA data sets had many transformations. The original data had the features shown in Table 1 (along with other matching data used in merging with betting data). All these were transformed by finding the average of the last six games in every season, except for the label. Other attributes were then generated using these original features as shown in Table 2. Note that the first six games of every season is discarded because the feature would not be properly represented and it only makes up roughly 7% of each season. All the features were normalised using Max-Min normalisation as some ML algorithms require this for optimal performance. The last step in process is to put each game in one row of data with both teams. They are merged using Data and Home/Away team information. At the end of all the transformation each row has 53 features (26 each for home and away team and 1 for both) and 1 outcome label. At this point a tabular NBA set for all 17 seasons is available in CSV form for processing by ML algorithm process.

FGM	3PM	FTM	OREB	AST
FGA	3PA	FTA	DREB	PTS
STL	BLK	TO	PF	WL Label

Table 1: 14 Original Features and Label from NBA Boxscores used to generate new ones.

Team Game Number	Days Rest	Miles Travelled
Day of Season	Current Streak	Home Win Pct
Division Win Pct	Conference Win Pct	Away Win Pct
Win Pct vs Above 500 teams	Last 6 Win Pct	Win Pct
Win Pct vs Below 500 teams		

Table 2: 13 Created Features from original features

The betting data has five attributes: Home Odds, Away Odds, Home Team, Away Team, and Date. All the odds are set by the bookmaker and the ones in this project are the average odds over multiple bookmakers. The Odds basically tell you how much you make on a single bet. Say a person bets $100 on a home team to win at 3.15 (3.15 to 1). If home team wins, they get paid $315 (profit of $215). If home team loses then person loses their $100. Bookmaker's odds can be used to get a bookmaker's probability of win by taking the inverse of the odds ($1/Odds$). For example, if a bookmaker offers 2.00 odds then the bookmaker probability would be 50%. If this was the actual probability of team to win then a person betting on this would break even in the long run. As noted earlier, the bookmaker would alter the odds in their favour. So if an outcome had a "break even" odds of 3.25, the bookmaker would change it to a reasonable number below that so they would have to pay less when the person has a winning bet.

Now that the NBA and Betting data are put into tabular form, they can be merged into a master table that is stored so it can be used for feature selection and modelling in following stages.

Feature Selection Four data sets are created using feature selection in order to compare and contrast them in evaluation phase. One will use all features from the master table. Another will use random forest's GINI index for feature selection. The other two will use features used by previous studies that got promising results in label prediction.

The first feature set selected features using random forest with the GINI index in a similar way as Saraswat and Arya (2014) and Pancerz et al. (2016). After random forest is run an index (GINI index) of the importance of each feature in prediction. The higher the GINI index, the higher the effect. This is done for every season of data to get a subset of most important features. The top 50% most important features are selected and stored. This is done to see if there is a subset is as good as predicting as using more features because it saves compute time in high throughput algorithms like ANN or Bagging. Running the algorithm is explained as follows in accordance with Figure 2: BHO chooses a hyperparameter (number of trees here) and a 10-Fold CV is run to produce an accuracy of the predicted WL labels. Note some algorithms don't give the same result every run so an average of 5 runs of algorithm is performed in each fold to get better result. Thus, random forest algorithm is run 50 times for each hyperparameter chosen. BHO also runs 15 times on random hyperparameters and then 10 more times in optimisation phase to find the hyperparameter that finds the highest accuracy. Thus the random forest is run 1,250 times (50 x 25) and a maximum hyperparameter is found (Note: this same process is used in all the ML algortihms). This is then run on random forest again using 10-Fold CV and averaged 20 times. The average GINI index is taken over all of these trials and then each feature is ranked accordingly. The random forest feature set will have 26 features and a label. PCA was attempted on the data set with and it it took 33 components to represent the 53 features in the data set. This number was bigger than the 50% of features selected by random forest so it was disregarded as it would not reduce compute time as much as random forest set.

The second feature set was adapted from Cheng et al. (2016) for comparison and has 28 features and a label. The third feature set uses all 53 features and label. The final feature set was adapted from Loeffelholz et al. (2009) for comparison and has 4 features and a label. If the fourth one performed well it would be useful as it would reduce compute time as it only has four features.

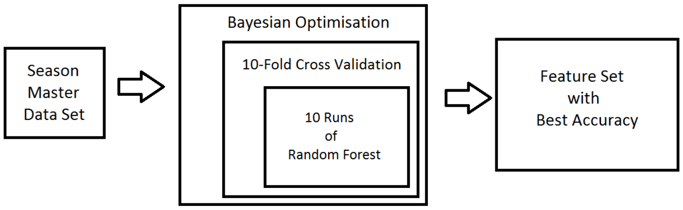


Figure 2: Random Forest Feature Selection Process

3.4 Modelling

There will be 7 algorithms used get labelled and probabilty outcomes. Each of these algorithms will be run on the four feature sets. These are run on all the 17 seasons and use the same process as done in Figure 2 when concerning BHO, 10-Fold CV and averaging over the trials (11 used here). Logistic regression and Naïve Bayes won't use averaging as they produce the same results in every trial. The labels in averaging phase will be made by majority vote and the probabilities will be the average over all runs. Each feature set id split into CV set (regular season) and a test (playoffs). At the end of this process there will be 17 seasons of predictions using 7 algorithms with 4 feature sets making 28 results sets for every season. Bayesian Hyperparameter Optimisation (BHO) is an algorithm that is used to find the optimal hyperparameters of algorithms as opposed to grid search because it can reduce computer complexity. All the algorithms in this study use hyperparameters except for Naïve Bayes. As stated previously, BHO is run 15 times randomly and then 10 times to search feature space for optimal hyperparameters. Once all the algorithms are run the outputs will be saved to CSV for evaluation phase of the study.

Artificial Neural Networks (ANN) ANN is a classifier that is similar to how a human brain works. It uses a network of decision nodes (neurons) to produce an output from given input Lantz (2013). It has an activation function that only fires if a certain threshold is reached. The network can have multiple hidden layers with any number of neurons. Weights on nodes can be updated a few different ways but standard back propagation is chosen as it performed relatively well in the study by Loeffelholz et al. (2009). There are three hyperparameters in an ANN that will be tuned this project: number of neurons in hidden layers, number of iterations (weight updates), and learning rate (rate of algorithm convergence). The fourth feature set only has 4 predictors, so only one hidden layer will be used with a neuron range from 1-8. The other three datasets will have two hidden layers with 15-25 and 2-10 neuron ranges. The iterations range from 500-1500 and the learning rate ranges from 0.1 to 1.0. Lantz (2013) further state that an ANN makes no assumptions about feature relationships which is good as winning percentages showed some correlation. They do state that they are very slow in compute so if they perform poorly they should maybe not be used in future studies.

Bagging Lantz (2013) states that Bagging (Bootstrap Aggregation) is an ensemble technique that takes a number of sample training sets (bags)and then runs a user-set algorithm on all samples. It then takes an average or majority vote to produce results.

This project uses D-Trees in bagging as they performed well in exploration phase. The number of bags is hyperparameter that will be tuned to the range of 25-100.

k-Nearest Neighbours (kNN) A kNN is a classifier that labels data using a distance metric to get the closest neighbour Lantz (2013). This project uses a Euclidean distance as all the data is numeric. kNN has one hyperparameter, k (or nearest neighbours),which ranges from 1-100. kNN also makes no assumptions about the relationships between features and trains very quickly.

Logistic Regression Lantz (2013) state that Logistic Regression can only classify binary outcomes but this works as this study is only concerned with wins or losses. It can also output a probability of that result. It uses three link functions which will be the hyperparameters. They can be Logit, Probit, and Complementary Log Log. It does not need averaging and has very quick runtime. It performed over 1,000 times faster that ANN.

Naïve Bayes (Bayes) Naïve Bayes uses Bayes' theorem to classify as stated by Lantz (2013). Thy also note that it is very fast algorithm that works well with data that has a good deal of noise. Though it does better with independent features, it still works well even if some are correlated. It has no hyperparameters and does not need to be averaged. If this was top performer, it would be great to use in future because its speed.

Random Forest A random forest is another ensemble method that takes subsets of all the features and runs a D-Trees on each Lantz (2013). It combines them at the end using averages and majority votes to get final output. As stated before, it can be used for feature selection. The number of trees is the hyperparameter that ranges from 50-500. It works well on most problems and deals well with noisy data.

Support Vector Machine (SVM) Lantz (2013) state that a SVM classifies binary outcomes by finding a maximum hyperplane (boundary) between two labels. It is good for noisy data and does not overfit a great deal. It is also another black box model and takes a good deal of compute. The two hyperparameters used are the Kernel and Cost function. Five of the top kernels used are Radial Basis (Gaussian), Polynomial, Linear, Laplacian and Bessel. In early exploration the Laplacian and Bessel gave very low accuracies in relation to the other 3, so they were disregarded to reduce computation time. The cost function adds a penalty to misclassified example and will range from 0-10.

3.5 Evaluation

3.5.1 Evaluation Metrics

There are two evaluation metrics that are used in this study: Accuracy and Return on Investment (ROI). Accuracy is the number of correctly predicted wins and losses over all the examples as shown in following equation: $\frac{TP+TN}{TP+TN+FP+FN}$. TP and TN are the correctly predicted wins and losses. FP is incorrectly labelled win and FN is incorrectly labelled loss. ROI is defined as the percentage of return or loss per unit bet in this study. For example, if someone bets \$100 on an outcome and they lose then ROI is defined as -100%. If they won and the odds were 1.45, then the ROI would be 45% as they would have won $45. ROI can range from -100 to any positive values (though you would not see odds over 30 to 1 odds too often. The highest odds in betting data was 28.8 to 1).

3.5.2 Case Studies

There are two major evaluation themes in this project as will be discussed in following two sections. The first evaluation theme involves comparing CV accuracy on the regular season with the testing accuracy on the playoffs. The second evaluation theme involves analysing the ROI of betting using probabilistic predictions from the predictor sets. A predictor/results set will be defined as the set of labels/probabilities created from each algorithm. They can be singular or an ensembled as well.

Comparison of CV Accuracy with Testing Accuracy Cross Validation (CV) is used to get a better estimate of how an algorithm will perform on the testing data. The average performance for each algorithm (over all seasons and feature sets) will be analysed to see the relative comparisons between them. Each season will have 28 predictor sets and the one that had the highest CV accuracy will be the top performer. The worst performer would be the results set with worst CV accuracy. The best performer will be compared to the worst performer and average performer. It will also be compared to the "actual" best and worst performer (the best and worst performers on testing set). There will also be comparisons on how the algorithms performed on average over the four feature sets and seasons. Ensembles of the top performers will also be performed. A Top 10 performer is defined as an ensemble of result sets of the top 10 accuracies achieved in CV. There are 28 possible Top Performer for each season (7 algorithms x 4 feature sets).

ROI using Betting Data Bookmakers set odds in their favour to make money. Table 3 shows a few examples of betting lines from this project. Notice that the summed probabilities is greater than 100%. This is a reason why bookmakers make money. Using some simple math it could be shown that the Bookmaker profit zone is the margin they need to win. If the true probability of an event lies in this zone it is impossible to beat them.

Home Odds/Prob	Away Odds/Prob	Summed Prob	Bookie Profit Zone
1.36/73.5%	3.15/31.7%	105.2%	68.3%-73.5%
1.36/73.5%	3.20/31.3%	104.8%	68.7%-73.5%
1.27/78.7%	3.85/26.0%	104.7%	74.0%-78.7%
1.13/88.5%	5.75/17.4%	105.9%	82.6%-88.5%

Table 3: Bookmaker Betting Odds and Probabilities

The first case study in the ROI comparison is to analyse what the ROI if labels were used to bet on outcomes. The label instructs a bet to be placed on home or away team. The bookmaker takes or gives money back depending on the game odds and result of a game. There will be a check to see if predicting home team wins are better than predicting away team wins. The second case study uses probability from the predictor sets to see how well they work against the bookmaker's odds/probabilities. Using probabilities bets would only be placed on games where the predictor set thinks it has an advantage over the bookie. For example, say bookmaker states the probability of home team win is

60%. If result set predicts a probability of win is greater than 60%, a bet is placed. The same goes for losses. There are examples were a bet would not be placed if the predicted probability of a home win or away win was lower than bookmaker's probabilities on both. For example, say a bookie states home win team will with 60% and the away team will win with a 45% probability. If the predictor set has 57% chance of home win (43% Away Win) then a bet would not be placed as probability of home and away team win is less than bookmaker probabilities. These 2 previous case studies will only analyse 14 predictor sets for simplicity. Seven are the algorithm result sets, four are the feature set results and three are ensembles of the top performers (1, 10, and 28).

One more study will involve seeing if their are hot zones were bets win more often. For example, bets may have an average positive ROI when a predictor probability lies in 50-60% range as compared to the 60-70% bookmaker probability range. This will be accomplished by betting home team win every time and away every time and seeing if there are patterns. This one will use an ensemble of all 28 predictor sets for simplicity.

3.6 Deployment

The final step of CRISP-DM is deployment. This entails the final report being showcased along with a VIVA presentation. If positive results come to fruition it would also entail being implemented for future NBA seasons for further analysis and hopefully money won.

4 Implementation

Data was gathered from NBA Stat Website[1], Betting Odds website[2], and a github repository[3]. Google sheets and R were the two major pieces of software used in this study. This section briefly explains the main ideas, but the Appendix gives a more detailed explanation of the step by step process in this study

4.1 Google Sheets (GS)

GS was used to get data for NBA and Betting into a tabular form to be exported as a CSV to be processed in R. Both data sets were manually copy and pasted into GS. The NBA data was structured in a tabular format and once all seasons were collected it was exported as CSV. Betting data was slightly unstructured so it needed some processing in Google Sheets using functions like "IF", "CONCATENATE", "SPLIT" and other functions to transform into tabular format. GS was also used to create code used in R for quicker creation. Some early basic data analysis and graphing was done in GS. Finally, the three heat maps were created with GS.

4.2 R

R was used as a major part of the project as it transformed data, ran ML algorithms, and was used to visualise most of the charts and graphs made. The R version was 3.3.2. There were 9 packages that had implementations of all the algorithms used in this study

[1] NBA Box Scores: `http://stats.nba.com/`

[2] Betting Odds: `http://www.oddsportal.com/basketball/usa/nba/results/`

[3] Locational Data: `https://gist.github.com/Miserlou/c5cd8364bf9b2420bb29`

and they are listed in the Appendix. All the hyperparameters discussed in methodology section are applicable to all algorithms used. There was an easy use to implementation of Bayesian Optimisation (rBayesianOptimization) where the number of trials could be chosen with hyperparameter ranges for every algorithm. Cross Validation was done using the caret package throughout whole study. The jsonlite package was used to parse the locational data from JSON format to tabular. The majority of the cleaning of the data was done in R. Basic functions like range, mean, sd, hist, boxplot and others were very useful in finding errors and during early exploration. All the data was manipulated using data frames,vectors, and lists. The master feature set was subsetted using random forest for feature selection and the subset function for other feature sets and stored in multiple CSVs to be processed later. There were varying speeds on all the algorithms. ANN, SVM, Random Forest and Bagging were very time consuming as opposed to kNN, Logistic Regression and Bayes.

5 Evaluation

5.1 Comparison of CV Accuracy with Testing Accuracy

When taking the average accuracy over all feature sets and seasons, a statistical difference was found between CV training accuracy and the testing accuracy at a 95% confidence level (Note: this confidence level is used from here on). A t-test was used to prove this (t = 10.465, df = 1311.9, p-value = 2.2e-16). The data was not all normally distributed but the number of measurements were very large and the central limit theorem states these assumptions can be disregarded at high number of measurements. The CV training had an average accuracy of 64.0% and the test a 61.9% average. Figure 3 shows how each algorithm performed on average over CV training and testing with their average variance as well. The test data had a statistically higher average variance as well (t = -24.719, df = 25.356, p-value = 2.2e-16).

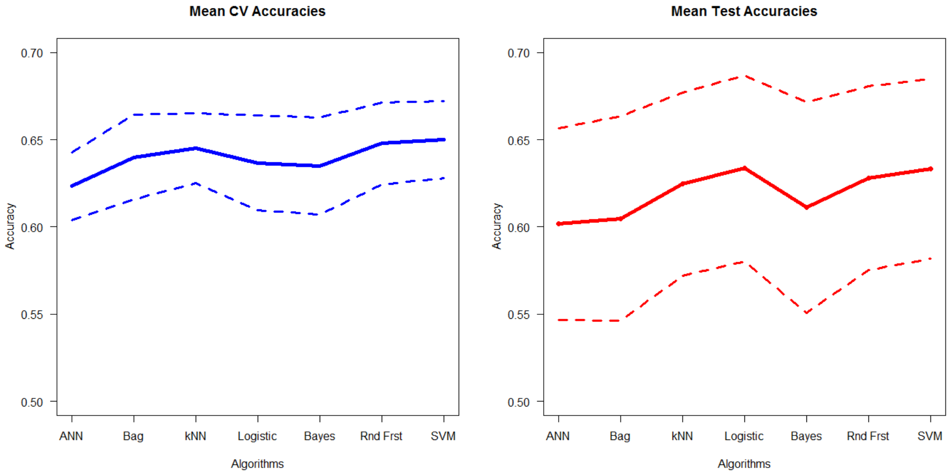


Figure 3: Shows the average performance of each algorithm over all feature sets and seasons. The dashed line represents the first standard deviation from the mean

Figure 4 shows how well the chosen best and worst performers on CV training set compared to the training set. The best chosen performer was significantly lower on the test set as compared to the CV training set (t = 14.546, df = 31.603, p-value = 1.492e-15). Similarly, the worst chosen was significantly lower on test than testing set (t = 3.365, df = 27.626, p-value = 0.002261). There was also a statistical difference between the best and worst testing sets which shows that choosing best CV accuracy can help choose a more optimal result on the test set

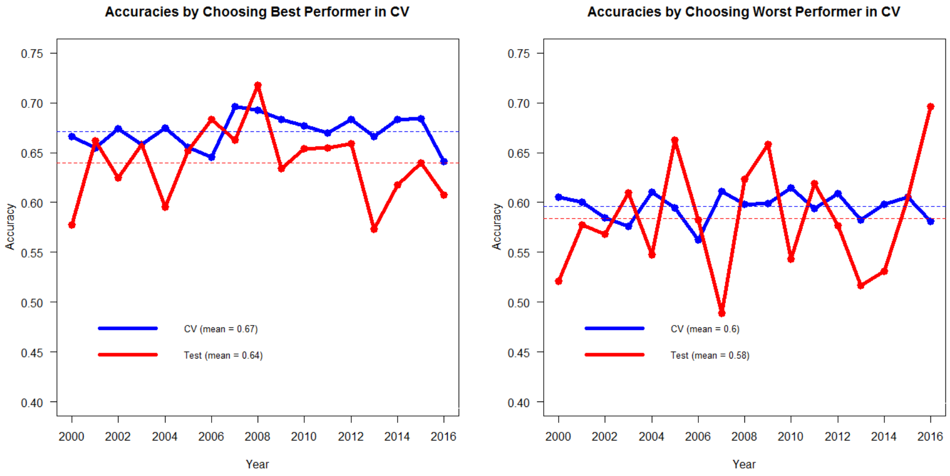


Figure 4: Shows the yearly performance for best and worst CV training sets as compared to the testing sets

The Figure 5 shows what the actual best and worst performers achieved on testing set compare to what the best chosen performer and average performers achieved. Notice that the chosen one is closer to actual best performer as opposed to the actual worst. The chosen had an average accuracy of 64.0% as compared to the average performer at 61.9%. Unfortunately there was not a statistical difference at 95% confidence level (t = 1.6618, df = 31.915, p-value = 0.1064). Tables 4 and 5 shows how many times an algorithm and feature set was chosen (best performer on CV) over the 17 as compared to the actual best and worst performers.

Algorithm	Chosen	Best	Worst
kNN	6	4	2
SVM	5	4	0
Bayes	2	1	6
Rand Frst	2	2	0
Bag	1	3	3
Log	1	1	0
ANN	0	2	6

Table 4: Algorithms use over 17 seasons by chosen, actual best, and actual worst performer

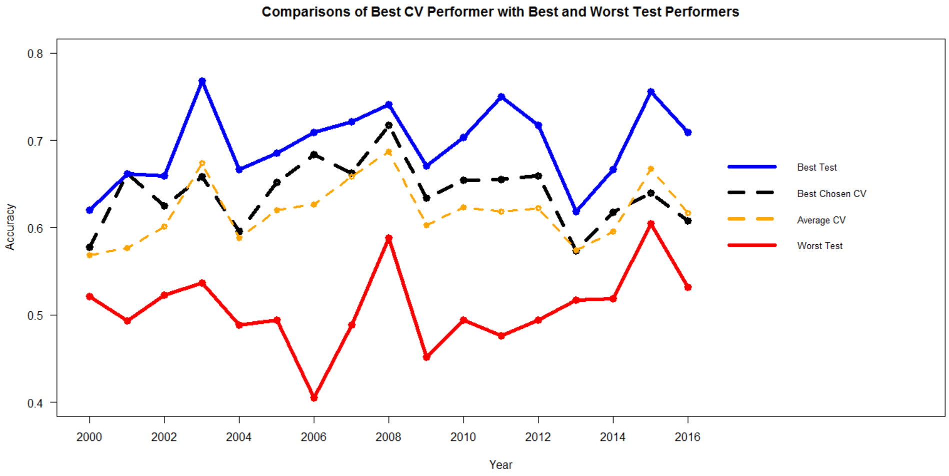


Figure 5: Compares the best chosen predictor set and average predictor set accuracies with that of the actual best and worse testing performers

Feature Set	Chosen	Best	Worst
Set 3	8	5	1
Set 4	5	2	1
Set 1	4	4	6
Set 2	0	6	9

Table 5: Feature Set use over 17 seasons by chosen, actual best, and actual worst performer

Figure 6 shows the mean accuracies of each algorithm for all seasons by feature set. There was no statistical difference between the four feature sets (df = (3,472), F= 1.578,p-value = 0.194) if algorithmic accuracies are averaged. There was a statistical difference in the feature sets 2 and 3 for the Naïve Bayes algorithm (df = (3, 64), F = 3.551, p-value = 0.0192).

When the top 28 ensemble predictor sets were tested ther was no statistical difference. The accuracies for each ensemble and first standard deviation can be seen in Figure 7. The average accuracy for all ensembles was 64.4% as compared to the 64.0% accuracy found by best prediction. Visually it can be seen that there wasn't much deviation from the mean of all ensembles.

5.2 ROI using Betting Data

The first case study took the probability labels for each result set to see what the ROI by betting according to them. Note again that the there were 14 result sets (7 algorithm ensembles, 4 feature set ensembles, and top n ensembles (n=1, 10, 28)). Every prediction set used on actual labels (real game results) returned a negative investment as shown in Figure 8. There were no statistical differences between all the result sets using a t-test once again. The distributions were all positively skewed but a t-test was able to be used as there were 752 observations in each group.

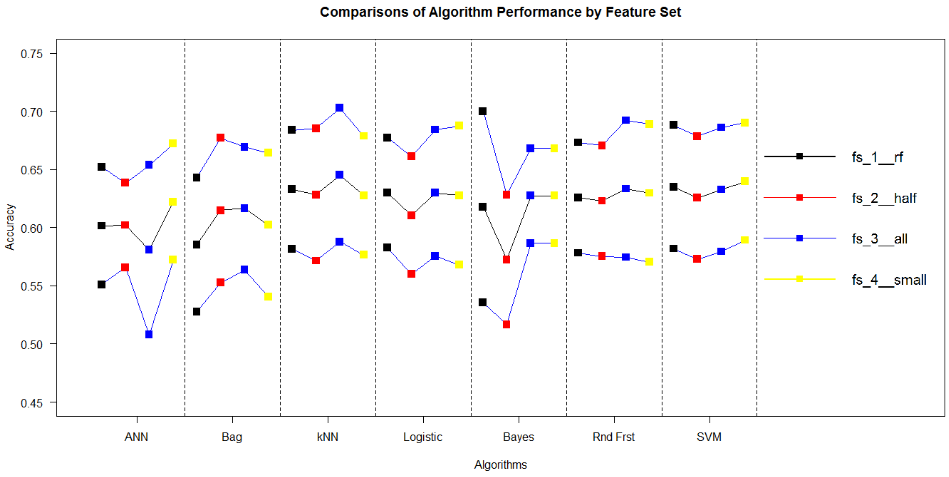


Figure 6: Compares the test sets accuracies for each algorithm by the four datasets and averaged over 17 seasons

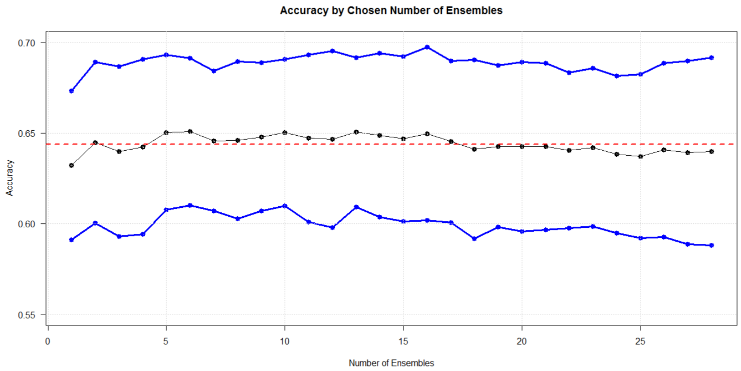


Figure 7: Show the mean accuracies of top 28 ensemble predictor sets

Figure 9 shows the same results except that they are separated by when each predictor set bets on home team versus the away team. There was a statistical difference between the home predictions and away predictions ($t = 2.7269$, $df = 2720.2$, p-value $= 0.006435$) and ($W = 8593800$, p-value $= 0.02112$). The home win predictions had a mean ROI of -4.0% as compared to away predictions at -10.3%. Neither achieved a positive return.

The next study looked at how well the ROI fared using probabilities form the 14 result sets. Note again that this model only bets on games it thought it would win. On average, 11.1% of the games in the result sets were not used as it did not think it had an advantage over the home or away probability of the bookmaker. Figure 10 shows the negative ROIs gained in this process. There were no statistical differences in each of the results found by t-tests. Also, notice that these returns fared much worse then ROI using only labels in Figure 8

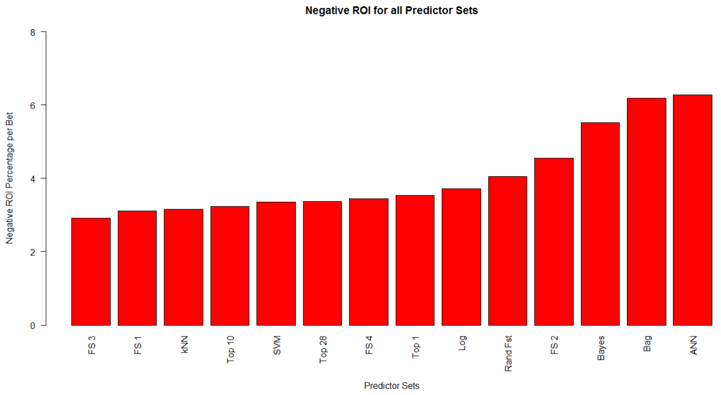


Figure 8: Shows the negative ROI when betting using labels against the bookmaker odds

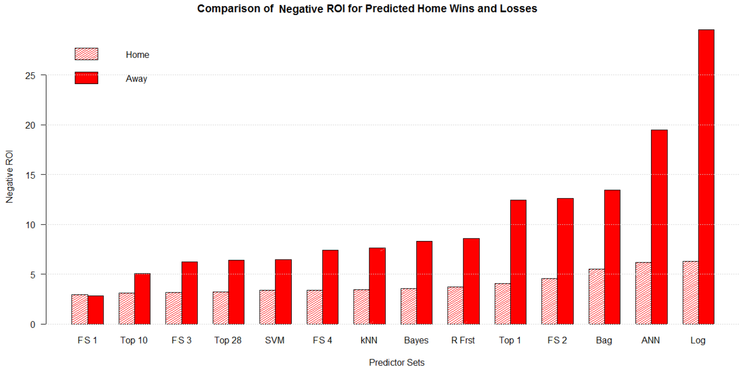


Figure 9: Shows the negative ROI when betting on home wins and away wins separately against the bookmaker odds using labels

The final study was to investigate the ROI zones of bookmaker probability versus predicted probability. Figure 11 shows what ROI would look for all zones if only bet on home team to win every game. Figure 12 shows what ROI would look for all zones if only bet on away team to win every game. In Figure 11 it appears that bets should be made on home team to win if bookmaker probability is 40-45% and the ensembled result set's probability is above 45%. The bottom right of Figure 12 seems to show a zone where the away team should be bet on for positive ROI. Note that this only shows average ROI and does not give an indication of the count of bets made in each zone.

Figure 13 shows a combination of Figures 11 and 12. It takes each zone and chooses the ROI that was best of the Home and Away Betting. In the top heatmap the blue means the best course of action was to bet on home team. Yellow means the same except to choose away instead. Neutral signifies that both had same ROI or that there were no bets appearing in those zones. The bottom heatmap shows the ROI corresponding to the bet choice.

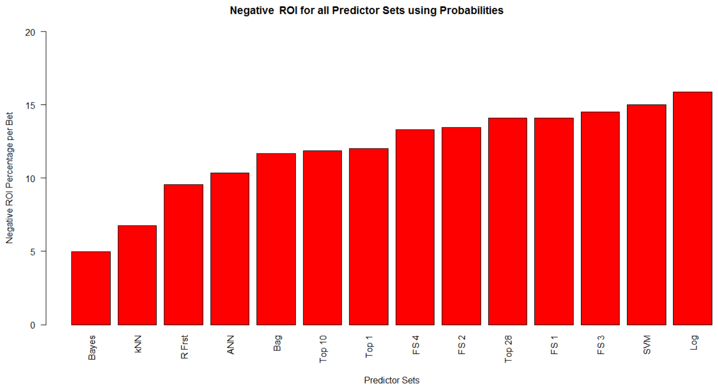


Figure 10: Shows the negative ROI using probability predictions of result sets on bookmaker probabilities

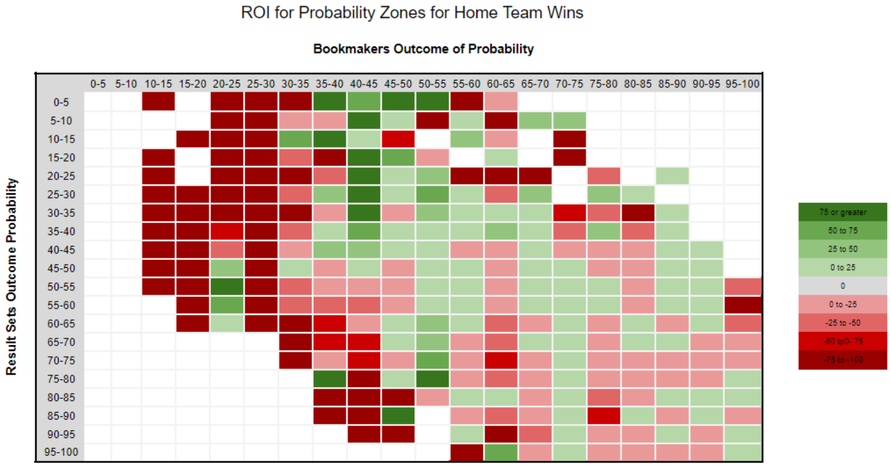


Figure 11: Shows an ROI zones when choosing Home Team to win every game

5.3 Discussion

This study was hoping to find a obvious way to predict games to get a positive ROI, but as seen in the results this was mostly the opposite. The two heatmaps in Figures 11 and 12 point to possible strategies to bet on future games in new seasons. It would only work if all the seasons act similarly as the nine in the project and this is not guaranteed as it could have happened by chance alone. In every other case study there was a negative ROI 3% or over, which makes betting using that strategy would be a bad investment. Using Figure 13 would most likely give better ROI than the previous two as well, but once again it can not be guaranteed.

In most machine learning a training set is used in order to predict some result. Most of the time the testing set will get a lower accuracy. Cross Validation is used to get a better estimate as was used in this study. The CV training set performed better than the testing set way more times on average. Choosing the best performer on the CV training

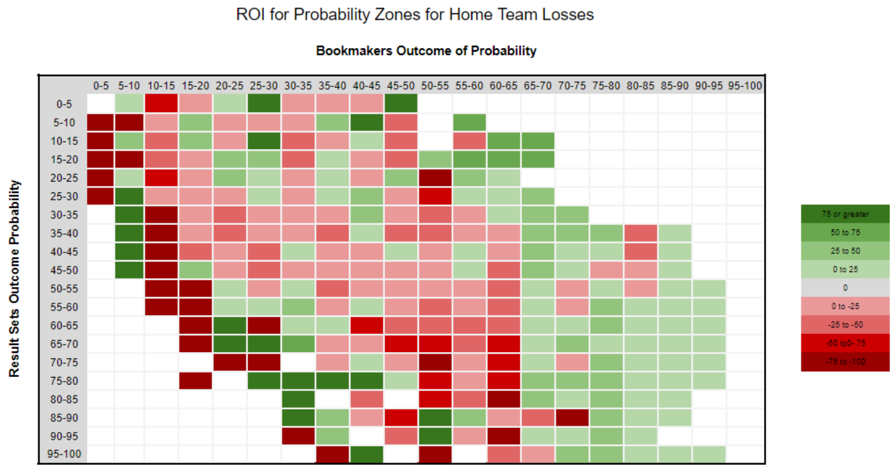


Figure 12: Shows an ROI zones when choosing Away Team to win every game

did produce a result better than average (though not at 95% significance) and always better than the worse actual test set. Therefore, it did do better than choosing result set randomly. Note that if someone chose home team to win they would be accurat 63.9% on average for these 17 seasons. Using labels did perform significantly better then predicting using probabilities as can be seen in Figures 8 and 10. The first didn't go beyond -10% investment but in the latter 11 out of 14 were over this amount.

In conclusion, it is best to only use the results from the last case study as some positive ROI were shown and it might point to zones that are good to bet on. Note that if all the algorithms were to be run again the results could be different as algorithms do not produce same exact results each run. The averaging trials tried to account for this.

6 Conclusion and Future Work

Beating the bookmaker at his own game using machine learning models in this study did not prove victorious. The last case study could offer a chance at positive ROI and it would be interesting to see another study repeat this study on same data multiple times to see the performance. It would also be interesting if it was tested on more seasons of NBA and betting data. This study only concerned with straight out wins but betting on spreads or over/unders could maybe offer better results. Since the data in this study used the average bookmaker odds it didn't represent the full range of values a bet could be. Also how can one know what the average bookmaker probability is until the books close at the start of a game. One problem in prediction could have been because bookmakers might be using similar strategies in creating their odds along with their skewing of the payout odds. If there prediction had same accuracy as our model there still would be no way to win because they would adjust the odds a little in order to get a positive ROI for themselves. Cheng et al. (2016) used Maximum Entropy in their study and got higher accuracies on average then all the algorithms used in this study. This algorithm was not used in this study because the lack of expertise to implement it but it would be interesting to see how well their model would work when compared with betting data.

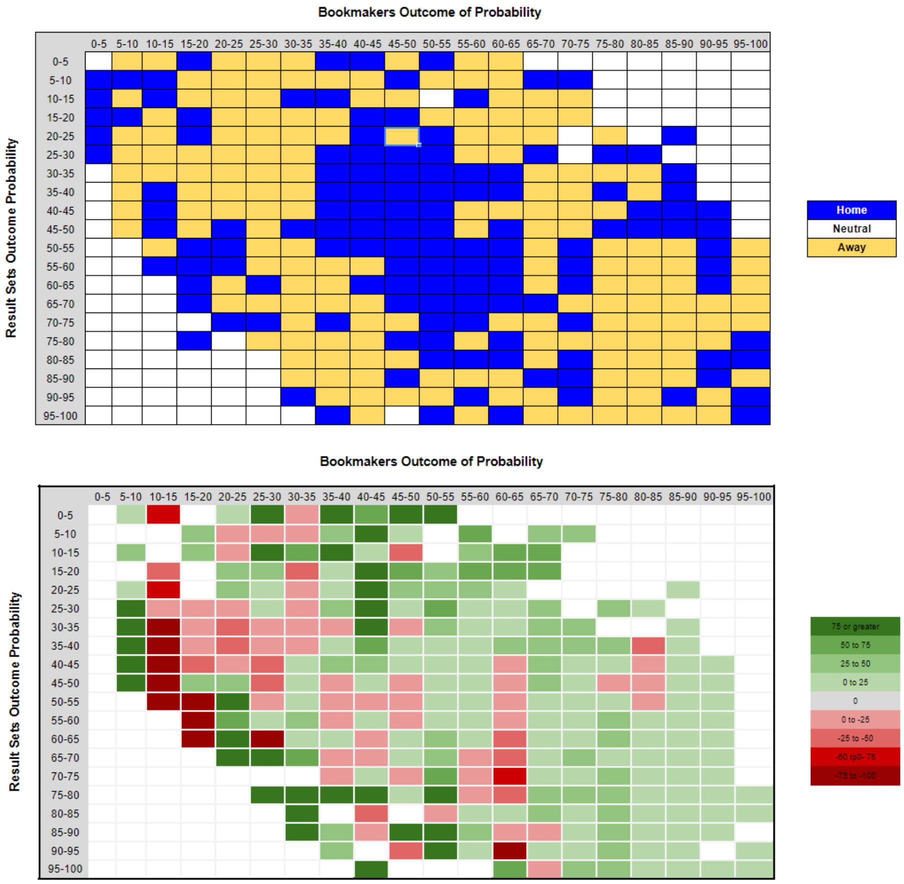


Figure 13: Shows best ROI zones by choosing the best ROI from home teams and away teams

Finally, this study did not take into account anomalies like injuries in a game or player changes in teams at all. If these these games were eliminated or processed in a way to represent reality better maybe the results could be more positive as there was no doubt these occurred over the 17 seasons.

Acknowledgements

I would like to thank my supervisor, Professor Paul Laird, for his support and feedback in moving this project in a positive direction. His input on using Bayesian Optimisation, Heat Maps and ROI strategies helped shape parts of the project. I would also like to thank fellow classmates and NCI professors for their feedback on my project.

References

Azevedo, A. I. R. L. and Santos, M. F. (2008). Kdd, semma crisp-dm: a parallel overviee, *IADS-DM* .

Beckler, M., Wang, H. and Papamichael, M. (2013). Nba oracle, *Zuletzt besucht am* **17**(20082009.9).

Bhandari, I., Colet, E., Parker, J., Pines, Z., Pratap, R. and Ramanujam, K. (1997). Advanced scout: Data mining and knowledge discovery in nba data, *Data Mining and Knowledge Discovery* **1**(1): 121–125.

Bock, J. R. (2016). Empirical prediction of turnovers in nfl football, *Sports* **5**(1): 1.

Cao, C. (2012). Sports data mining technology used in basketball outcome prediction.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. (2000). Crisp-dm 1.0 step-by-step data mining guide.

Cheng, G., Zhang, Z., Kyebambe, M. N. and Kimbugwe, N. (2016). Predicting the outcome of nba playoffs based on the maximum entropy principle, *Entropy* **18**(12): 450.

Demers, S. (2015). Riding a probabilistic support vector machine to the stanley cup, *Journal of Quantitative Analysis in Sports* **11**(4): 205–218.

Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H. and Leyton-Brown, K. (2013). Towards an empirical foundation for assessing bayesian optimization of hyperparameters, *NIPS workshop on Bayesian Optimization in Theory and Practice*, Vol. 10.

Feurer, M., Springenberg, J. T. and Hutter, F. (2015). Initializing bayesian hyperparameter optimization via meta-learning., *AAAI*, pp. 1128–1135.

Gumm, J., Barrett, A. and Hu, G. (2015). A machine learning strategy for predicting march madness winners, *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on*, IEEE, pp. 1–6.

Hasbun, T., Araya, A. and Villalon, J. (2016). Extracurricular activities as dropout prediction factors in higher education using decision trees, *Advanced Learning Technologies (ICALT), 2016 IEEE 16th International Conference on*, IEEE, pp. 242–244.

Hofler, R. A. and Payne, J. E. (1997). Measuring efficiency in the national basketball association, *Economics letters* **55**(2): 293–299.

Hofler, R. A. and Payne, J. E. (2006). Efficiency in the national basketball association: a stochastic frontier approach with panel data, *Managerial and Decision Economics* **27**(4): 279–285.

Joseph, A., Fenton, N. E. and Neil, M. (2006). Predicting football results using bayesian nets and other machine learning techniques, *Knowledge-Based Systems* **19**(7): 544–553.

Kampakis, S. and Thomas, W. (2015). Using machine learning to predict the outcome of english county twenty over cricket matches, *arXiv preprint arXiv:1511.05837* .

Lantz, B. (2013). *Machine learning with R*, Packt Publishing Ltd.

Lévesque, J.-C., Durand, A., Gagné, C. and Sabourin, R. (2017). Bayesian optimization for conditional hyperparameter spaces.

Loeffelholz, B., Bednar, E. and Bauer, K. W. (2009). Predicting nba games using neural networks, *Journal of Quantitative Analysis in Sports* **5**(1).

McCabe, A. and Trevathan, J. (2008). Artificial intelligence in sports prediction, *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, IEEE, pp. 1194–1197.

Merritt, S. and Clauset, A. (2014). Scoring dynamics across professional team sports: tempo, balance and predictability, *EPJ Data Science* **3**(1): 1–21.

Miljković, D., Gajić, L., Kovačević, A. and Konjović, Z. (2010). The use of data mining for basketball matches outcomes prediction, *Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on*, IEEE, pp. 309–312.

Page, G. L., Fellingham, G. W., Reese, C. S. et al. (2005). *Using box-scores to determine a position's contribution to winning basketball games*, PhD thesis, Brigham Young University. Department of Statistics.

Pancerz, K., Paja, W. and Gomuła, J. (2016). Random forest feature selection for data coming from evaluation sheets of subjects with asds, *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*, IEEE, pp. 299–302.

Pathak, N. and Wadhwa, H. (2016). Applications of modern classification techniques to predict the outcome of odi cricket, *Procedia Computer Science* **87**: 55–60.

Saraswat, M. and Arya, K. (2014). Feature selection and classification of leukocytes using random forest, *Medical & biological engineering & computing* **52**(12): 1041–1052.

Schapire, R. E. and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions, *Machine learning* **37**(3): 297–336.

Snoek, J., Larochelle, H. and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms, *Advances in neural information processing systems*, pp. 2951–2959.

Soto Valero, C. (2016). Predicting win-loss outcomes in mlb regular season games–a comparative study using data mining methods, *International Journal of Computer Science in Sport* **15**(2): 91–112.

Vaz de Melo, P. O., Almeida, V. A. and Loureiro, A. A. (2008). Can complex network metrics predict the behavior of nba teams?, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 695–703.

Waggoner, B., Wines, D., Soebbing, B. P., Seifried, C. S. and Martinez, J. M. (2014). hot hand in the national basketball association point spread betting market: A 34-year analysis, *International Journal of Financial Studies* **2**(4): 359–370.

Watcharapasorn, P. and Kurubanjerdjit, N. (2016). The surgical patient mortality rate prediction by machine learning algorithms, *Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on*, IEEE, pp. 1–5.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann.

Zak, T. A., Huang, C. J. and Siegfried, J. J. (1979). Production efficiency: the case of professional basketball, *Journal of Business* pp. 379–392.

Zdravevski, E. and Kulakov, A. (2010). System for prediction of the winner in a sports game, *ICT Innovations 2009* pp. 55–63.

Zhang, G. P. (2000). Neural networks for classification: a survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **30**(4): 451–462.

Zhang, N., Lin, A. and Shang, P. (2017). Multidimensional k-nearest neighbor model based on eemd for financial time series forecasting, *Physica A: Statistical Mechanics and its Applications* **477**: 161–173.

Configuration Manual

A Compute and Software Specifications

All the work was done on a Hewlett-Packard Laptop with an Intel Core i3 2GHz processor and 8 GB of RAM. The operating systems was a 64-bit Windows 10 OS. Google sheets had the functionality current to date of this project. The R version was 3.3.2. The table 6 lists all the important packages that were used along with the main function used in R. All the hyperparameters discussed in methodology are available in these functions. jsonlite (version - 1.5) was another package that was used to parse the JSON location data using the function fromJSON.

Algorithm	R-package	Version	Function
Bayesian Optimisation	rBayesianOptimization	1.1.0	BayesianOptimization
Cross Validation	caret	6.0-76	createFolds
ANN	RSNNS	0.4-9	mlp
Bagging	ipred	0.9-6	bagging
Boosting	adabag	4.1	boosting
Decision Trees	C50	0.1.0-24	C5.0
kNN	class	7.3-14	knn
Logistic Regression	R-imbedded		glm
Naïve Bayes	e1071	1.6-8	naiveBayes
Random Forest	randomForest	4.6-12	randomForest
SVM	kernlab	0.9-25	ksvm

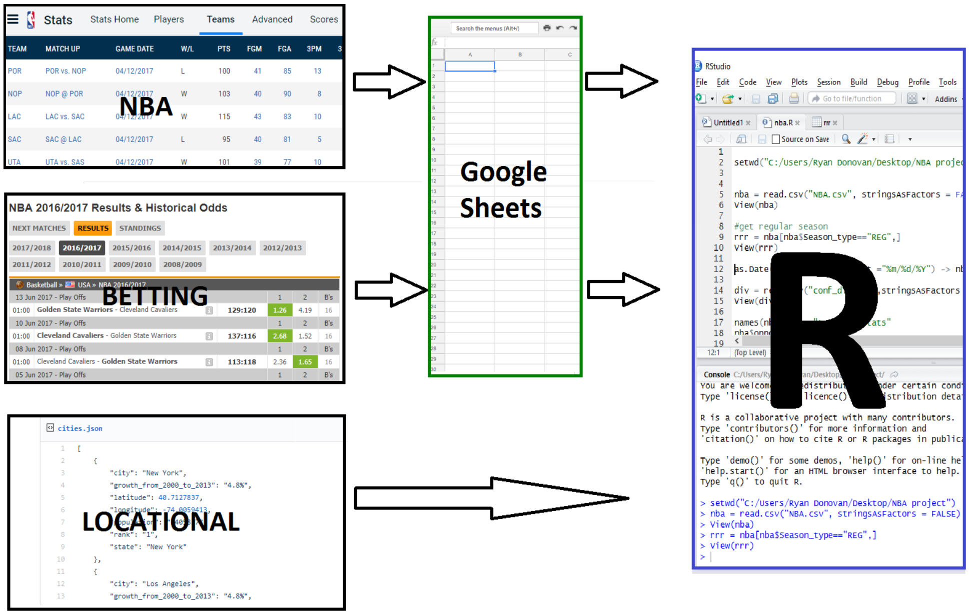
Table 6: R algorithms used

B Data Sourcing and Transformations

Data was gather from 3 websites using copying and pasting. This would be more efficient using web scraping as the data was structured for the most part, although the author did not have expertise in this area. The following shows some more details on how features were formed for 3 datasets. Figure 14 shows an high level view of the process.

NBA Data After copy and pasting of data into Google Sheets it was exported as CSV to be processed in R. All the features were transformed in different ways except for the Win/Loss Label. All preseason games were filtered out as they were not used in this study.

- The following 14 features were all averaged for the previous n (6 - 10) games first: FGA (Field Goal Attempts), FGM (Field Goal Made), 3PA (Three Points Attempt), 3PM (Three Point Made), FTA (Free Throw Attempts), FTM (Free Throw Made), DREB (Defensive Rebounds), OREB (Offensive Rebounds) , AST (Assists), PTS (Points), STL (Steals), BLK (Blocks), TO (Turnovers), PF (Personal Fouls) and Last ”n” Games Winning Percentage. Next the 3 variables FGP (Field Goal Percentage), 3PP (Three Point Percentage) and FTP (Free Throw Percentage) were found by dividing total made (M) by total Attempts (A) for each.



Stats | Stats Home | Players | Teams | Advanced | Scores

TEAM	MATCH UP	GAME DATE	W/L	PTS	FGM	FGA	3PM	3
POR	POR vs. NOP	04/12/2017	L	100	41	85	13	
NOP	NOP @ POR	04/12/2017	W	103	40	90	8	
LAC	LAC vs. SAC	04/12/2017	W	115	43	83	10	
SAC	SAC @ LAC	04/12/2017	L	95	40	81	5	
UTA	UTA vs. SAS	04/12/2017	W	101	39	77	10	

NBA 2016/2017 Results & Historical Odds

NEXT MATCHES | RESULTS | STANDINGS

2017/2018 | 2016/2017 | 2015/2016 | 2014/2015 | 2013/2014 | 2012/2013
2011/2012 | 2010/2011 | 2009/2010 | 2008/2009

Basketball » USA » NBA 2016/2017

			1	2	B's
13 Jun 2017 - Play Offs			1	2	B's
01:00	**Golden State Warriors** - Cleveland Cavaliers	**129:120**	1.26	4.19	16
10 Jun 2017 - Play Offs			1	2	B's
01:00	**Cleveland Cavaliers** - Golden State Warriors	**137:116**	2.68	1.52	16
08 Jun 2017 - Play Offs			1	2	B's
01:00	Cleveland Cavaliers - **Golden State Warriors**	**113:118**	2.36	1.65	16
05 Jun 2017 - Play Offs			1	2	B's

cities.json

```
[
    {
        "city": "New York",
        "growth_from_2000_to_2013": "4.8%",
        "latitude": 40.7127837,
        "longitude": -74.0059413,
        "population": "
        "rank": "1",
        "state": "New York"
    },
    {
        "city": "Los Angeles",
        "growth_from_2000_to_2013": "4.8%",
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools

Go to file/function | Addins

Untitled1 | nba.R | rrr

☐ Source on Save

```

setwd("C:/Users/Ryan Donovan/Desktop/NBA projec


nba = read.csv("NBA.csv", stringsAsFactors = FA
View(nba)

#get regular season
rrr = nba[nba$Season_type=="REG",]
View(rrr)

as.Date(                 ="%m/%d/%Y") -> nb

div = re      ("conf_d      ,stringsAsFactors
View(di

names(n                  tats"
nba$onn
```

12:1 (Top Level)

Console C:/Users/Ry   an/Desktop/   ct/

```
You are welcome    edistribut    nder certain cond
Type 'license(     licence()     istribution deta

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publica

Type 'demo()' for some demos, 'help()' for on-line he
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> setwd("C:/Users/Ryan Donovan/Desktop/NBA project")
> nba = read.csv("NBA.csv", stringsAsFactors = FALSE)
> View(nba)
> rrr = nba[nba$Season_type=="REG",]
> View(rrr)
>
```

Figure 14: Shows the software use in process and how the data flowed from sources

- The next set of features are created by taking difference from previous game to current game: Days Rest and Miles Travelled. So days rest would be how many games off a team had since last game. Similarly the miles travelled is the distance a team would travel between games.

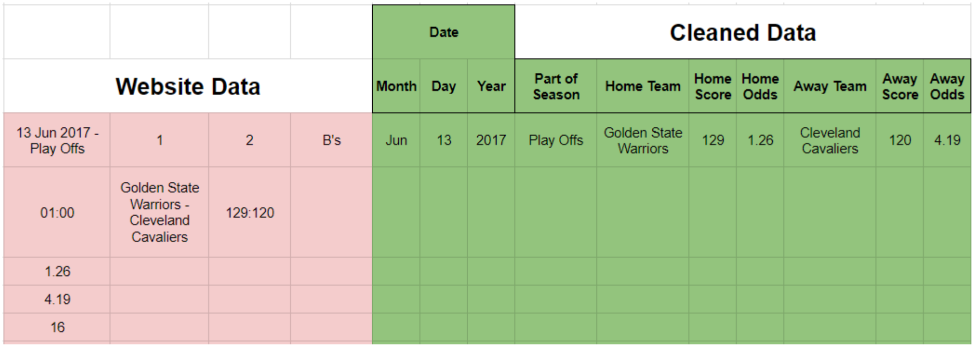
- The next attributes are found by aggregating game data from the beginning of the season: Home Winning Percentage, Away Winning Percentage, Division Winning Percentage, Conference Winning Percentage, Winning Percentage Against Above 500 teams, Winning Percentage Against Below 500 teams, and Winning Percentage.

- Team Game Number represents the game number in the season and the Day of Season represents how many days have passed from beginning of season.

- The final feature calculated was a teams current streak. It represents the current games either won or lost in a row. For instance, if team A's last 5 games were (W, W, W, L, C) and team B's last 5 games were (L, W, W, W, C), then team would have -1 value and team B would have +3 value.

After all these transformations are done the data is checked for errors.

Betting Data The betting data was a little more complex in setting up but not too difficult to implement. More manual work needs to be done as opposed to automating it. When the data is copy and pasted into Google Sheets the data looks like the pink section on the left in Figure 15. The date can be parsed by using SPLIT function provided by Google Sheets and a self-created SUBSTRING FUNCTION. First the data is split by the ”-” character and then SUBSTRING gets the month, day and year as all the dates are in same format on the betting website. Getting the Part of Season uses part of the split just explained to get it's value. The Scores and Teams are found similarly. The Odds are the same in each pasting and can be found in lower left. This worked for most of the data from the website but sometimes there was different set ups and this would be need to be accounted for in an iterative process. At the end of this process the data is checked for errors. There were some NAs found in the betting data but they were only for regular season games that would not be used in this project. The data is then sent to R in CSV form. In R, two new features are created: Home Probabilities and Away Probabilities. Both are just the inverse of the odds.



				Date			Cleaned Data						
Website Data				Month	Day	Year	Part of Season	Home Team	Home Score	Home Odds	Away Team	Away Score	Away Odds
13 Jun 2017 - Play Offs	1	2	B's	Jun	13	2017	Play Offs	Golden State Warriors	129	1.26	Cleveland Cavaliers	120	4.19
01:00	Golden State Warriors - Cleveland Cavaliers	129:120											
1.26													
4.19													
16													

Figure 15: Shows an example how data from Betting Website is parsed in Google Sheets

Locational Data The locational data was parsed using "fromJSON" function in the jsonlite. It translates the data into a tabular format. The data was only for US cities so the two Canadian teams, Toronto and Vancouver had to have their coordinates manually added. The "distm" function from the geosphere package was used to find the distance from each team's city. Once this was done the data was checked for errors.

Combination of Data sets The NBA and Betting data is combined by using merge function by the features "Date" and "Home Team". Since NBA data was for 17 years of data and Betting was 9 years, there were 8 years of data that had NAs for the Betting part of the data. The locational data was parsed before the transformations in the NBA data and it is used when the distance travelled attribute is being created. Once again, the data is checked for data to maintain integrity of the data.

C Feature Selection

There are four feature sets created in this study to be processed by the machine learning algorithms. One uses random forest for feature selection, one uses all the features, and two sets use subsets of the full feature set that previous studies have used.

- Feature Set 1: A random Forest is run on each season of data using Bayesian Optimisation, Cross Validation and multiple trials in order to find the best performer for each season. When random forest is run it produces a GINI index for each feature. This index indicates the power of each feature for each year. These are ordered and then the top 50% is chosen to be in the input feature set for the machine learning process. These feature sets are stored in a CSV as the following 3 feature sets are.

- Feature Set 2: This used the 28 features that Cheng et al. (2016) used in their study.
- Feature Set 3: This feature set uses all the features that have been created.
- Feature Set 4: This uses the 4 features hat Loeffelholz et al. (2009) used in their study.

D Machine Learning Process

The overview of the process takes four feature sets and then runs seven algorithms on them for all 17 seasons of NBA data. Bayesian Hyperparameter Optimisation and Cross Validation is used with 10 trials in each fold. The following gives an idea of each step in this process. The labels, probability of labels, season, and CV training accuracy are saved at the end of the process

- Choose Feature Set (of 4)
- Choose Algorithm (of 7)
- Choose Season (of 17)
- Split data into CV Train (Regular Season) and Test (Playoffs) sets.
- Use CV Train set in Bayesian Hyperparameter Optimisation (BHO) process
- BHO chooses 15 random hyperparameters in range of values the user specifies. These are all run and after BHO does ten more in optimisation process. 10-Fold CV is run for each hyperparameter. 10 trials are run inside each of the 10 folds to reduce variance in results
- BHO issues the best hyperparameters
- Use these hyperparameters and run the algorithm for 11 trials on all the CV train (regular season) data
- Use the average results of the 11 trials to predict the label and probability of label for the test data (playoffs)
- Store results in data frame
- Repeat for all feature sets, algorithms and season.
- When all combinations finished, store all the data in a CSV to be used in the betting comparison process later on

E Betting Data Comparisons

There are two basic comparisons done in this study: (1) compare using only labels and (2) compare using prediction probabilities. Both Used 14 result sets as shown in following bullet points:

- Four result sets take the average of all the algorithms for each Feature Set
- Seven result sets that take the average over the 4 feature sets for each algorithm
- One results set that uses the result set that had the highest accuracy on the CV Training sets
- One result set that takes an ensemble average of the top ten performers in the CV Training sets.
- One set that takes the average over all the 28 results sets for each season

Analysis is done by getting the ROI from betting on games. In the first study, the home team is bet on if a win is predicted for home team and on the away team if home team is predicted to lose. In the second study a team is bet on if predicted probability from the results set is greater than the bookmaker's probability. The return on investment is found by comparing the label and probability to the actual outcome and bookmaker payout odds. Using result sets with betting data, each bet calculates the money won or lost in a bet using all bets an average. ROI is found by summing all money won or lost by the total number of bets made. A payout works as follows: if a predicted win for team ends up in a loss, then person loses the full bet. If predicted team wins the person gets paid there original bet plus their ROI. The following shows an example: If the odds are 3.45 and the predicted team wins on a bet of $100, the person gets paid $100 and $245 in ROI. If opposite happens they lose the $100.