Can AI Beat Vegas?

System Documentation

Michael Wynn

Dr. Alex Rudniy

May 7th, 2021

Submitted in partial fulfillment of the requirements of CMPS/IT 490 – Computer Projects

Abstract

This project aimed to construct a machine learning model that predicts the outcome of NBA games, and wagers that may be placed on games through online sportsbooks. To achieve this, a myriad of game [5, 7] and sports betting [6] data was collected, processed, and engineered to become acceptable input for classification algorithms [1]. The original thesis was that the inclusion of gambling point spreads, due to their recent abundance in digital form and constantly updating nature, would be helpful in creating a predictive model for game outcomes. This quickly failed, and the objective shifted to optimizing predictive models.

The models, or *classifiers*, were used to predict two things: whether the home team will win the game, and whether they will beat the spread. Classifiers generated were tested on holdout data, then evaluated using accuracy, precision, and recall. Many models outperformed Vegas when predicting outcome outright; the best of which scored an accuracy of 73.89% for win predictions. However, models performed worse when predicting gambling outcomes. The best model for spread predictions scored 57% accuracy, which is better than it sounds, given that Vegas intentionally sets point spreads to encourage equal distribution of bets.

Table of Contents

Introduction

Objectives

As society becomes increasingly digital, the availability and prevalence of data grows rapidly. Data scientists aim to take advantage of this glut of data by creating machine learning models that consume data and predict the future. In this instance, the goal is to predict the outcomes of NBA games and wagers using team statistics and gambling point spreads. Originally, this project hoped to test the hypothesis that gambling spread data would improve models for win predictions. After discovering this to be false, the objective quickly shifted to optimizing two types of predictive models: one for game outcomes, and one for gambling outcomes.

Methods

To make predictions, a variety of historical NBA team statistics and gambling point spread data was collected, processed, and prepared for input into machine learning algorithms. Algorithms used for this project are examples of supervised learning, meaning the machine possesses the actual results of games and wagers during training, allowing the model to learn from incorrect predictions.

The optimization of models was formatted as a series of experiments – controlled selections of a variable dataset or model determine the accuracy of predictions. This format helps to determine which aspect of the data is most influential in determining outcomes and allows a neat comparison of different models' effectiveness in prediction.

Software System

This was a data science project and as such, used software consistent with industry standards. Collected data was stored in .csv (comma separated values) files. Analysis, including data processing and model creation, training, testing, and evaluation, was performed using Python 3.8 within the Jupyter

Notebook IDE. This environment was chosen due to the ease of including descriptive text via markup cells. Additionally, Python packages such as Pandas, Numpy, Scikit-Learn, Matplotlib, Seaborn, TensorFlow, and Keras were used. All datasets and code will be made available on GitHub.

Related Work

In the past, others have used machine learning to predict NBA game outcomes. Most results fall between 66-72% accuracy, with one outlier reaching above 74% accuracy. However, the 74% model was developed and trained specifically for playoff games [2]. This is an important distinction to make due to a clear pattern in professional sporting events – the better team, or the team having the higher win percentage, usually wins. This pattern is even more prevalent in the postseason. In the history of the NBA regular season, the favorite wins 67.9% of games. During the playoffs, this increases to 78% [9]. This means a model that simply predicts the favorite to win would outperform the best existing model for playoff games. This places the best publicly available model for NBA regular season game predictions at 72% accuracy. This project hopes to improve upon known methods to achieve an accuracy of higher than 72% but would be happy with results above the baseline created by predicting the favorite in each game.

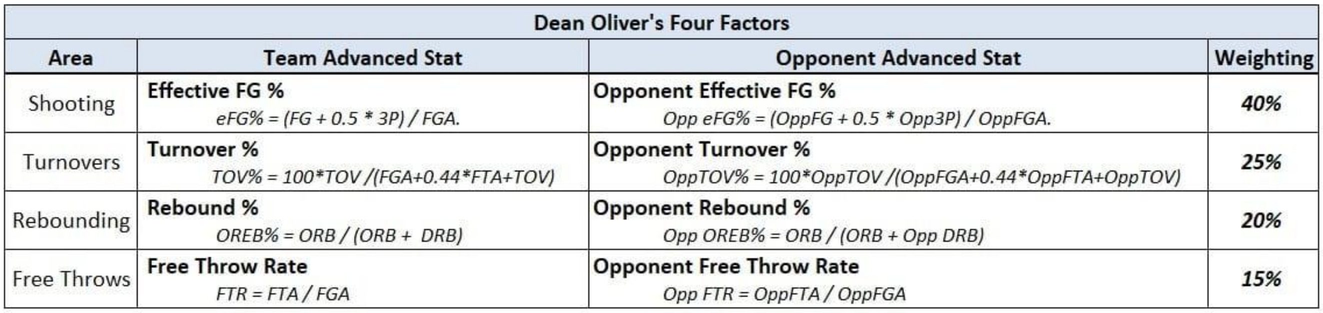
There is substantially less publicly available research into predicting NBA gambling outcomes, likely due to the very recent legalization of sports betting in America. Sportsbooks set gambling lines based on drawing an equal amount of action on either side, resulting in profit regardless of game outcome. Therefore, the gambling point spread isn’t only a prediction of game outcome, it is intended to be the most easily disputed game outcome possible. Setting point spreads like this ensures bets file in equally on both sides of the wager – preventing major downside for sportsbooks. With this in mind, any prediction results more accurate than random chance (50%) would be notable.

Data

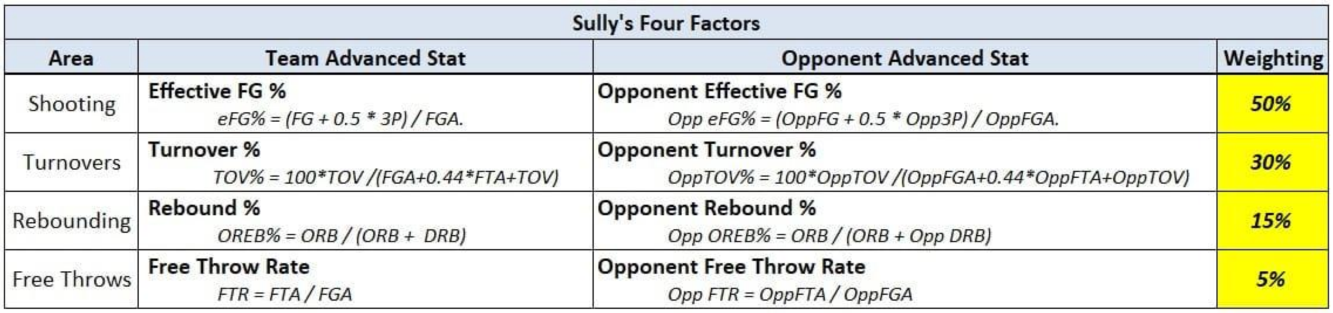
Data Origins

There are 3 main datasets used in this project, hereafter referred to as Basic, Advanced, and Final. The datasets are composed of NBA team statistics and gambling data, spanning across 14 regular seasons, from 2005 to 2018. Basic and Advanced are separated based on a semi-arbitrary grouping of statistics, while Final contains only the data deemed most important during feature engineering, which is explained shortly. The Basic and Advanced datasets lack gambling data but include more games (16,979). The Final dataset contains point spread data, with a tradeoff of fewer games (8,602), due to gambling data only being available from 2012-2018.

Data was gathered from three different sources, two of which were Kaggle datasets, the other a sports gambling website. One Kaggle dataset contained basic game data [5], while the other held gambling data [6]. The gambling data used is an average of the point spreads from 5 different websites: Pinnacle, Bovada, Betonline, Heritage, and 5dimes. The sports gambling website [7] added advanced game data and some interesting features of their own calculation. Website contributor Ryan Sullivan, who famously linked James Harden's road game performance to a city's strip club quality, created a variation of Dean Oliver's Four Factors that was helpful for this project. Dean Oliver's Four Factors is a weighted combination of advanced NBA team stats, which are themselves a combination of basic NBA team stats: effective field goal percentage, turnover percentage, rebound percentage, and free throw rate. Sullivan created a variation on Dean Oliver's statistic to accommodate for the NBA's recent shift toward prioritizing shooting over other facets of the game. Both Dean Oliver and Sully's Four Factors are included in the Advanced dataset.



Dean Oliver's Four Factors			
Area	**Team Advanced Stat**	**Opponent Advanced Stat**	**Weighting**
Shooting	**Effective FG %** *eFG% = (FG + 0.5 * 3P) / FGA.*	**Opponent Effective FG %** *Opp eFG% = (OppFG + 0.5 * Opp3P) / OppFGA.*	***40%***
Turnovers	**Turnover %** *TOV% = 100*TOV /(FGA+0.44*FTA+TOV)*	**Opponent Turnover %** *OppTOV% = 100*OppTOV /(OppFGA+0.44*OppFTA+OppTOV)*	***25%***
Rebounding	**Rebound %** *OREB% = ORB / (ORB + DRB)*	**Opponent Rebound %** *Opp OREB% = ORB / (ORB + Opp DRB)*	***20%***
Free Throws	**Free Throw Rate** *FTR = FTA / FGA*	**Opponent Free Throw Rate** *Opp FTR = OppFTA / OppFGA*	***15%***



Sully's Four Factors			
Area	**Team Advanced Stat**	**Opponent Advanced Stat**	**Weighting**
Shooting	**Effective FG %** *eFG% = (FG + 0.5 * 3P) / FGA.*	**Opponent Effective FG %** *Opp eFG% = (OppFG + 0.5 * Opp3P) / OppFGA.*	***50%***
Turnovers	**Turnover %** *TOV% = 100*TOV /(FGA+0.44*FTA+TOV)*	**Opponent Turnover %** *OppTOV% = 100*OppTOV /(OppFGA+0.44*OppFTA+OppTOV)*	***30%***
Rebounding	**Rebound %** *OREB% = ORB / (ORB + DRB)*	**Opponent Rebound %** *Opp OREB% = ORB / (ORB + Opp DRB)*	***15%***
Free Throws	**Free Throw Rate** *FTR = FTA / FGA*	**Opponent Free Throw Rate** *Opp FTR = OppFTA / OppFGA*	***5%***

Data Description

Field	**Description**	**Dataset**
ID	Unique row identifier	All
GAME_DATE_EST	Estimated game date	All
GAME_ID	Unique game identifier	All
SEASON	NBA Season | (2012-13 season = 2012)	All
HOME_TEAM_ID	Unique team identifier	All
HOME_W%	Current season winning percentage	All
HOME_PS_W%	Previous season winning percentage	Basic, Final
HOME_PS_FG%	Previous season field goal percentage	Basic
HOME_PS_3P%	Previous season three-point percentage	Basic
HOME_PS_FT%	Previous season free throw percentage	Basic
HOME_PS_REB	Previous season rebounds per game	Basic
HOME_PS_AST	Previous season assists per game	Basic

HOME_PS_TOV	Previous season turnovers per game	Basic
HOME_PS_STL	Previous season steals per game	Basic
HOME_PS_BLK	Previous season blocks per game	Basic
HOME_PS_MOV	Previous season margin of victory	Advanced, Final
HOME_PS_PIE	Previous season player impact estimate (team)	Advanced, Final
HOME_PS_TEAM_DOFF	Previous season Dean Oliver’s Four Factors (offensive)	Advanced
HOME_PS_OPP_DOFF	Previous season Dean Oliver’s Four Factors (defensive)	Advanced
HOME_PS_NET_DOFF	Previous season Dean Oliver’s Four Factors (net)	Advanced
HOME_PS_TEAM_SFF	Previous season Sully’s Four Factors (offensive)	Advanced
HOME_PS_OPP_SFF	Previous season Sully’s Four Factors (defensive)	Advanced
HOME_PS_NET_SFF	Previous season Sully’s Four Factors (net)	Advanced
AWAY_TEAM_ID	Unique team identifier	All
AWAY_W%	Current season winning percentage	All
AWAY_PS_W%	Previous season winning percentage	Basic, Final
AWAY_PS_FG%	Previous season field goal percentage	Basic
AWAY_PS_3P%	Previous season three-point percentage	Basic
AWAY_PS_FT%	Previous season free throw percentage	Basic
AWAY_PS_REB	Previous season rebounds per game	Basic
AWAY_PS_AST	Previous season assists per game	Basic
AWAY_PS_TOV	Previous season turnovers per game	Basic
AWAY_PS_STL	Previous season steals per game	Basic
AWAY_PS_BLK	Previous season blocks per game	Basic
AWAY_PS_MOV	Previous season margin of victory	Advanced, Final
AWAY_PS_PIE	Previous season player impact estimate (team)	Advanced, Final
AWAY_PS_TEAM_DOFF	Previous season Dean Oliver’s Four Factors (offensive)	Advanced
AWAY_PS_OPP_DOFF	Previous season Dean Oliver’s Four Factors (defensive)	Advanced
AWAY_PS_NET_DOFF	Previous season Dean Oliver’s Four Factors (net)	Advanced
AWAY_PS_TEAM_SFF	Previous season Sully’s Four Factors (offensive)	Advanced
AWAY_PS_OPP_SFF	Previous season Sully’s Four Factors (defensive)	Advanced
AWAY_PS_NET_SFF	Previous season Sully’s Four Factors (net)	Advanced

SPREAD	Average of 5 sportsbooks' point spreads, refers to home team (negative if home team is favored)	Final
MARGIN	Game Results – not used in prediction (MARGIN = HOME_SCORE – AWAY_SCORE)	All
HOME_TEAM_WINS	Game Results – Classification target If home team won, 1. Else, 0.	All
HOME_TEAM_ BEAT_SPREAD	Game Results – Classification target If (MARGIN > -SPREAD), 1. Else, 0.	Final

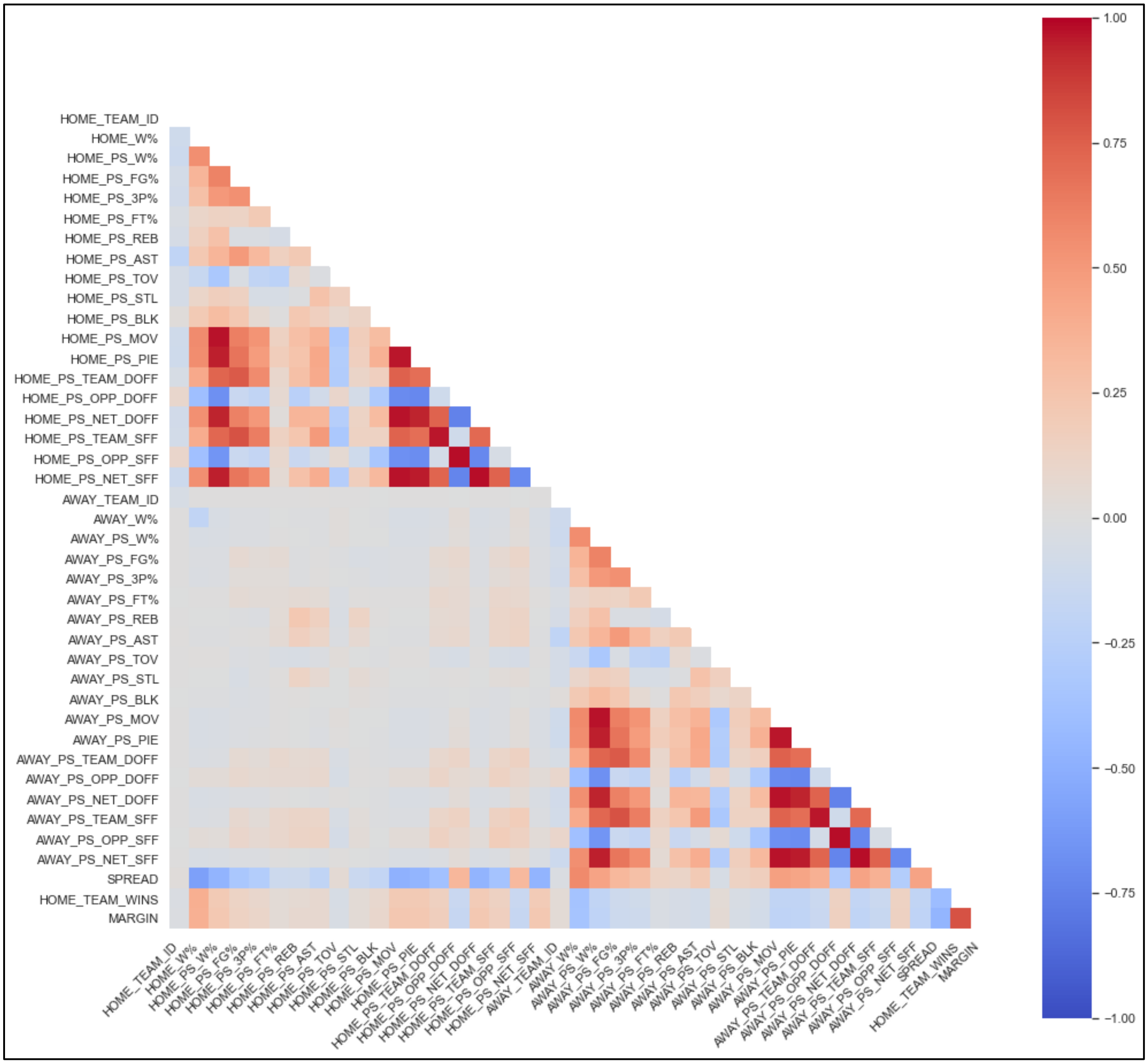
Data Preparation

This project utilized supervised machine learning, which uses standard input data to make predictions, then tunes predictions using the true outcome of each scenario: the target. Targets are essential to supervised learning and were created via simple arithmetic manipulation of game outcomes. In supervised learning, the target provides a concise, numeric distillation of actual results. The machine should not know the outcome during prediction but must know the outcome afterwards to improve.

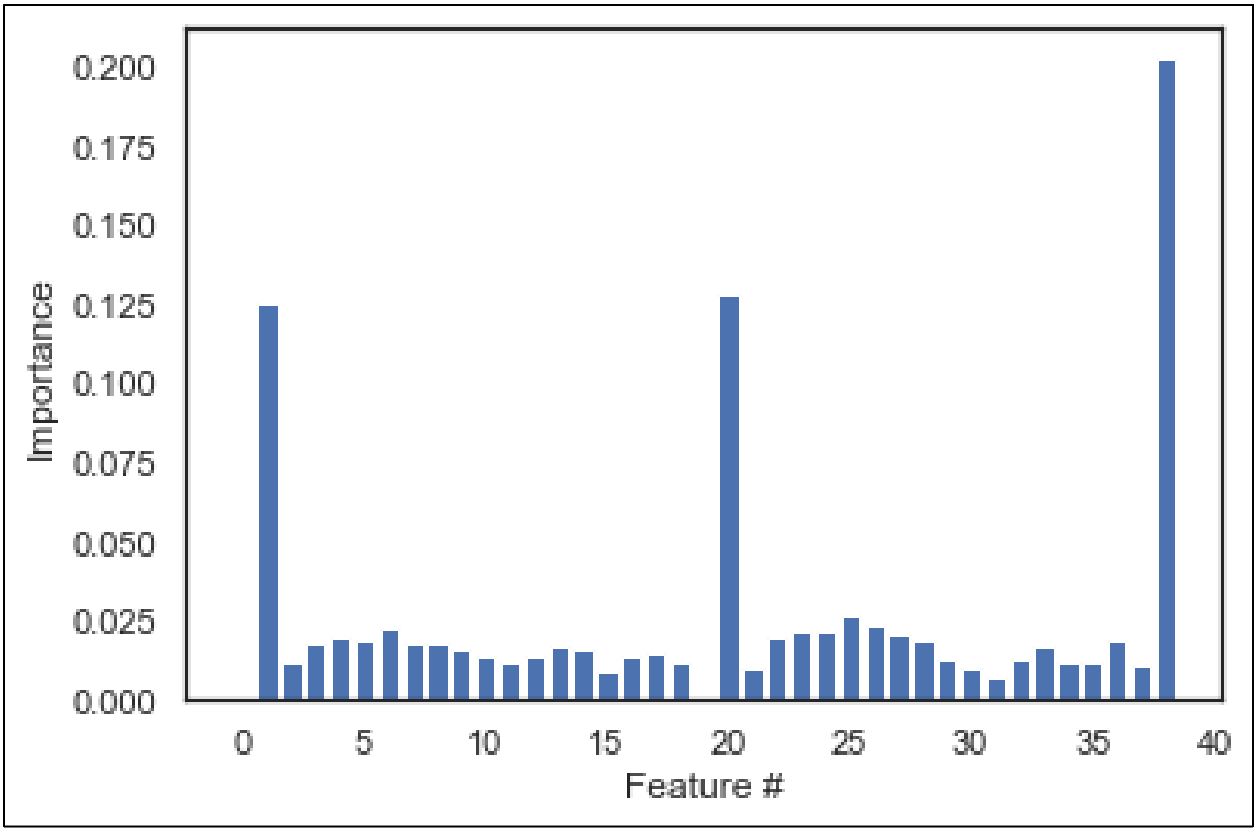
In the same way targets were generated, it was also necessary to calculate benchmarks for success. Two simple algorithms were chosen that fair well when predicting wins: choosing the team with a higher current winning percentage, and choosing the team favored by sportsbooks. For the data used in this project, benchmarks reached 71.3% and 68.7% accuracy, respectively. To be considered a success, this project must improve upon the accuracy of always choosing the better team.

Feature Engineering and Selection

During machine learning, it is important to avoid unnecessary complications such as including non-informative or misleading data. Including such data could potentially confuse the algorithm and cause poor performance. The classic saying goes, "garbage in – garbage out". As such, during training, many descriptive fields such as unique identifiers and unrelated game data were not included. Still, it was necessary to sort through the wide variety of related game data to discover which aspects were most influential. Luckily, there are many methods for this. Creating a correlation matrix for data allowed clear differentiation between the variables that were helpful, indicated by a high correlation to classification targets, and those that were unimportant or redundant.



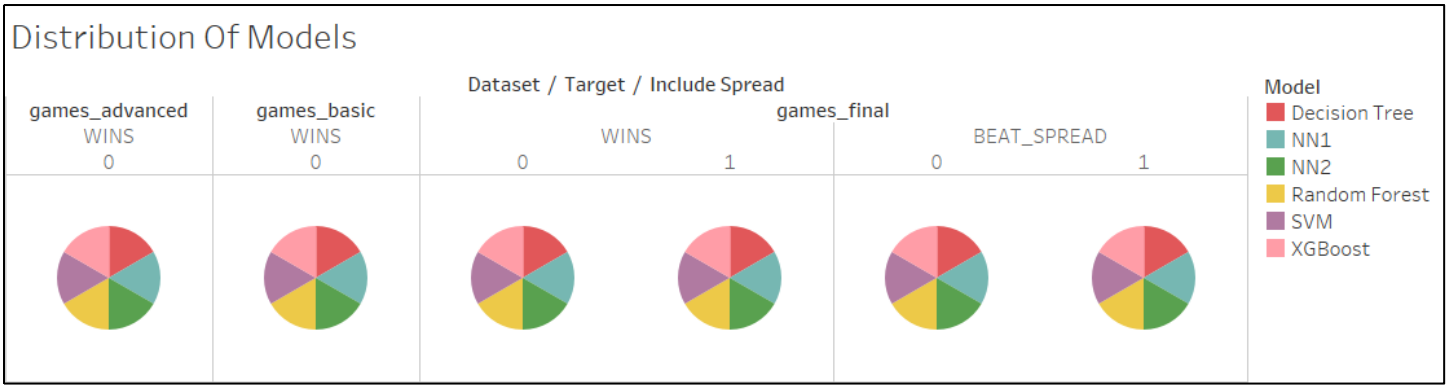
In addition to the correlation matrix, the feature importance mechanism of Scikit-Learn's Decision Tree Classifier was implemented. This generated a clean bar graph indicating which variables split data the best, meaning those which were most indicative of predicting an outcome successfully. Feature #s 1, 20, and 38, which were clearly the most important, are HOME_W%, AWAY_W%, and SPREAD. Viewing variables through these visualizations spawned the Final dataset, which aimed to avoid excess dimensionality by including only the most predictive data.



Modeling

General Information

In total, 36 models of 6 different types were created. 24 predicted game outcomes, while the remaining 12 predicted gambling outcomes. Of the 24 models predicting game outcomes, 25% used the Basic dataset, 25% used the Advanced dataset, 25% used the Final dataset without SPREAD, and 25% used the Final dataset with SPREAD. For gambling outcome predictions, all models used the Final dataset, half with SPREAD and half without.



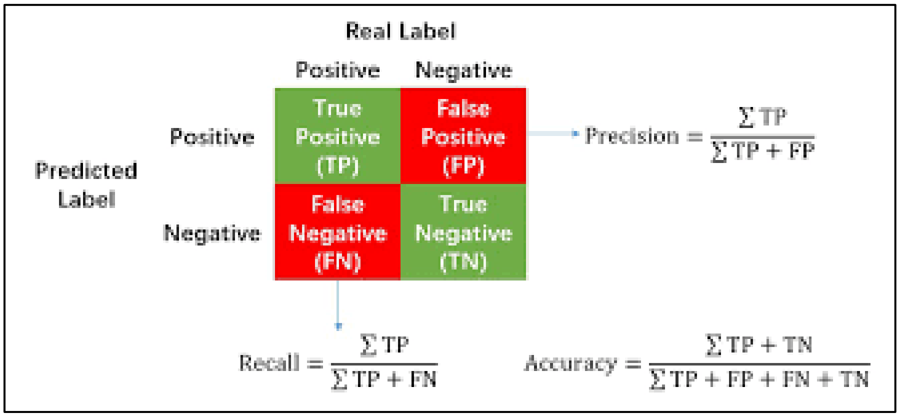
Model Description

Model	Description
Decision Tree	In this algorithm, the sample (population) is split into two or more sub-population sets, decided by the most significant splitter or differentiator in the input variables. The training process resembles a flow chart, with each internal (non-leaf) node a test of an attribute, each branch is the outcome of that test, and each leaf (terminal) node contains a class label. The uppermost node in the tree is called the root node.
Random Forest	An ensemble algorithm that combines purposefully dissimilar decision trees and determines results via a voting mechanism.
XGBoost	Extreme Gradient Boost – An ensemble algorithm that combines gradient boosted decision trees. During training, features that produce good results are strengthened, or boosted. This algorithm is called extreme because it maximizes computer resources.

SVM	Support Vector Machine – This algorithm attempts to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.
NN1	Neural Network 1 – 2 hidden layers, 1 output layer. Trained over 500 epochs.
NN2	Neural Network 2 – 3 hidden layers, 1 output layer. Trained over 100 epochs. Also includes L2 regularization and dropout to avoid overfitting by penalizing simpler solutions.

Model Evaluation

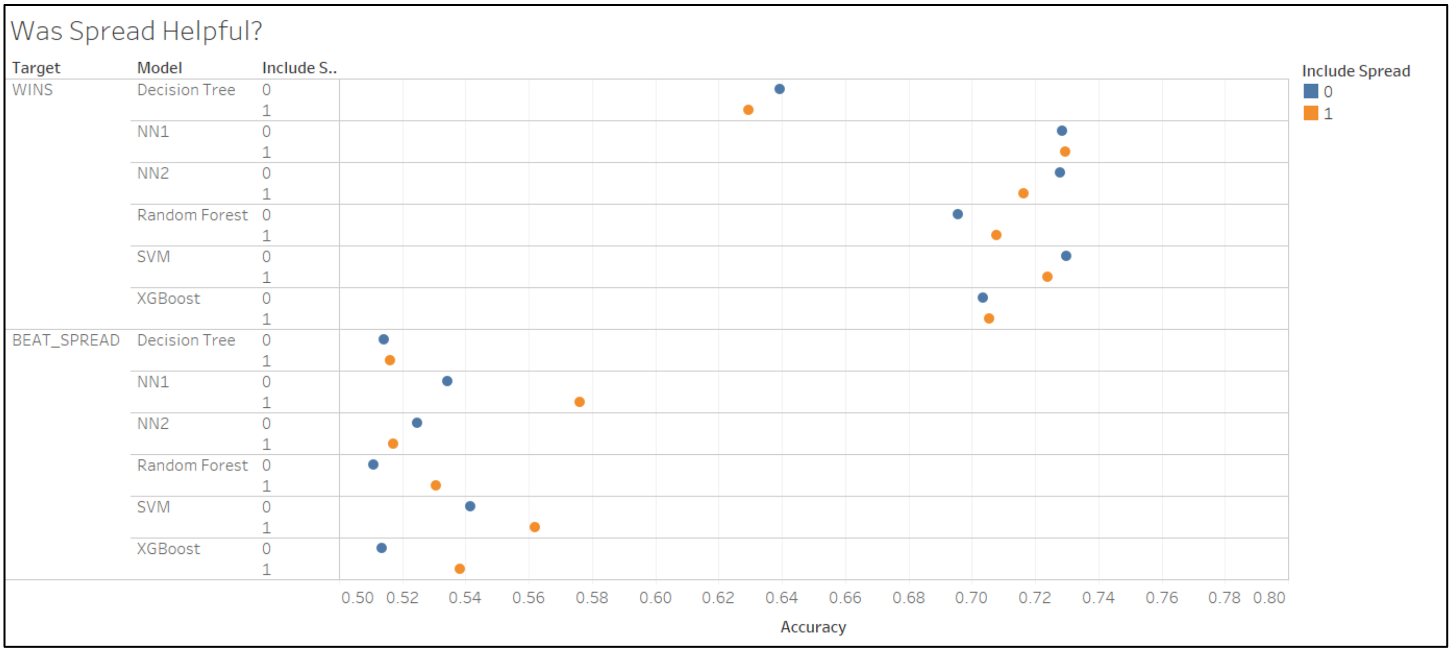
Evaluation of classifiers is based on the distribution of correct vs. incorrect predictions. These values are held in a confusion matrix. From the confusion matrix, metrics such as accuracy, precision, and recall are calculated. This project prioritizes accuracy over other, less informative, evaluation metrics.


Real Label
Positive
Negative
Predicted Label
Positive
Negative
True Positive (TP)
False Positive (FP)
False Negative (FN)
True Negative (TN)
$\text{Precision} = \frac{\sum TP}{\sum TP + FP}$
$\text{Recall} = \frac{\sum TP}{\sum TP + FN}$
$\text{Accuracy} = \frac{\sum TP + TN}{\sum TP + FP + FN + TN}$

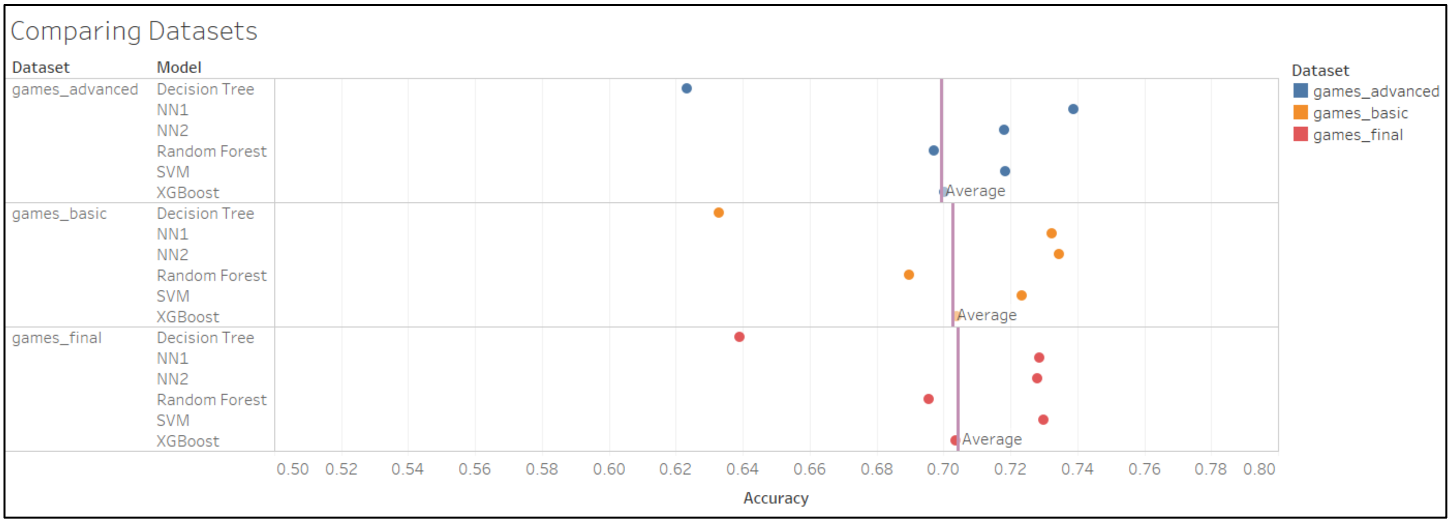
Conclusion

Discussion

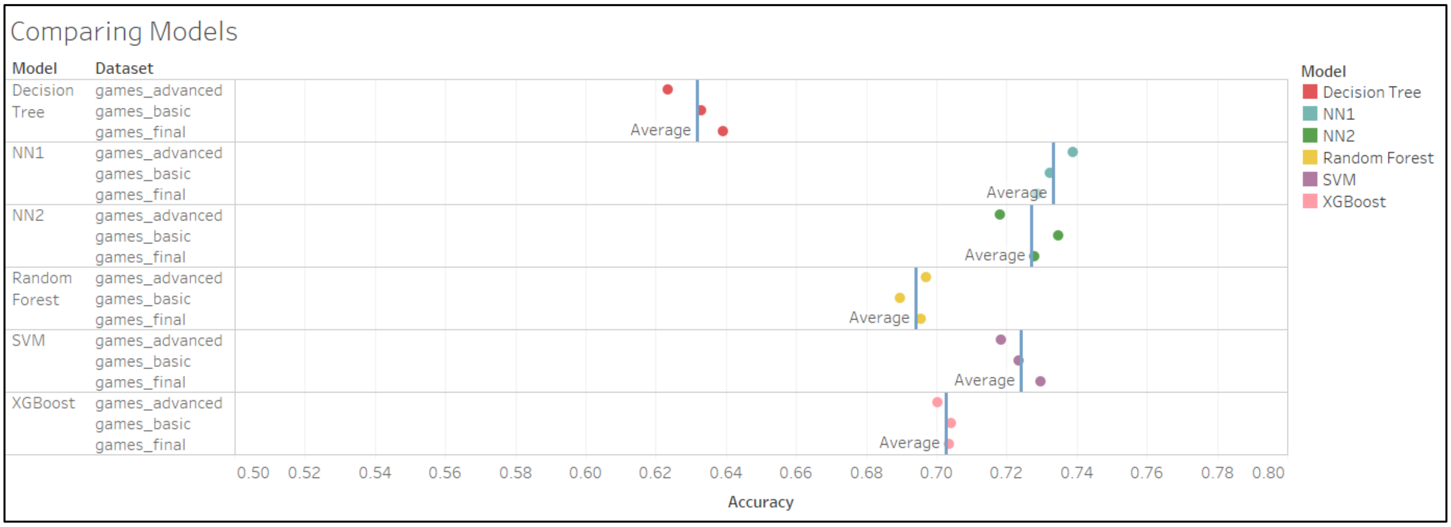
Some datasets performed better than others, and some may have suffered from unintentional error. Adding gambling data to the Final dataset was intended to aid the accuracy of win prediction models but failed to do so. The addition of gambling data improved half of the win prediction models, though only by a negligible amount, while decreasing the performance of the other half substantially. However, for gambling predictions, the inclusion of spread data improved 83.3% of models, most by over 2%. This impressive performance warrants further collection of gambling data.



The Final dataset, which performed the best on average by a slight margin, was not used for creation of the best model. The tradeoff of adding potentially helpful gambling data, at the expense of shrinking the dataset in half, hindered performance when paired with the best models. However, the inclusion of gambling data did raise the bar for the worst performing models, resulting in the highest average performance for a dataset. The Advanced dataset, which had the lowest average performance, spawned the best predictive model of the bunch. This may be an outlier, or possibly neural network 1 works exceptionally well with this dataset, but the result remains regardless.

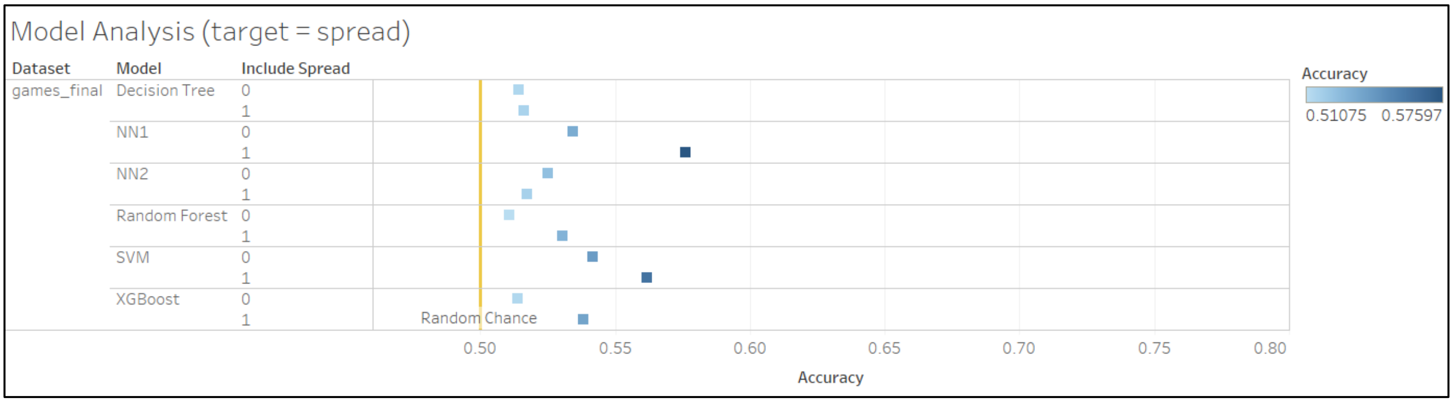


Similarly, some models worked well while others did not. The decision tree performed poorly across the board, and neural network 1 was the opposite. The rest fell somewhere in between.

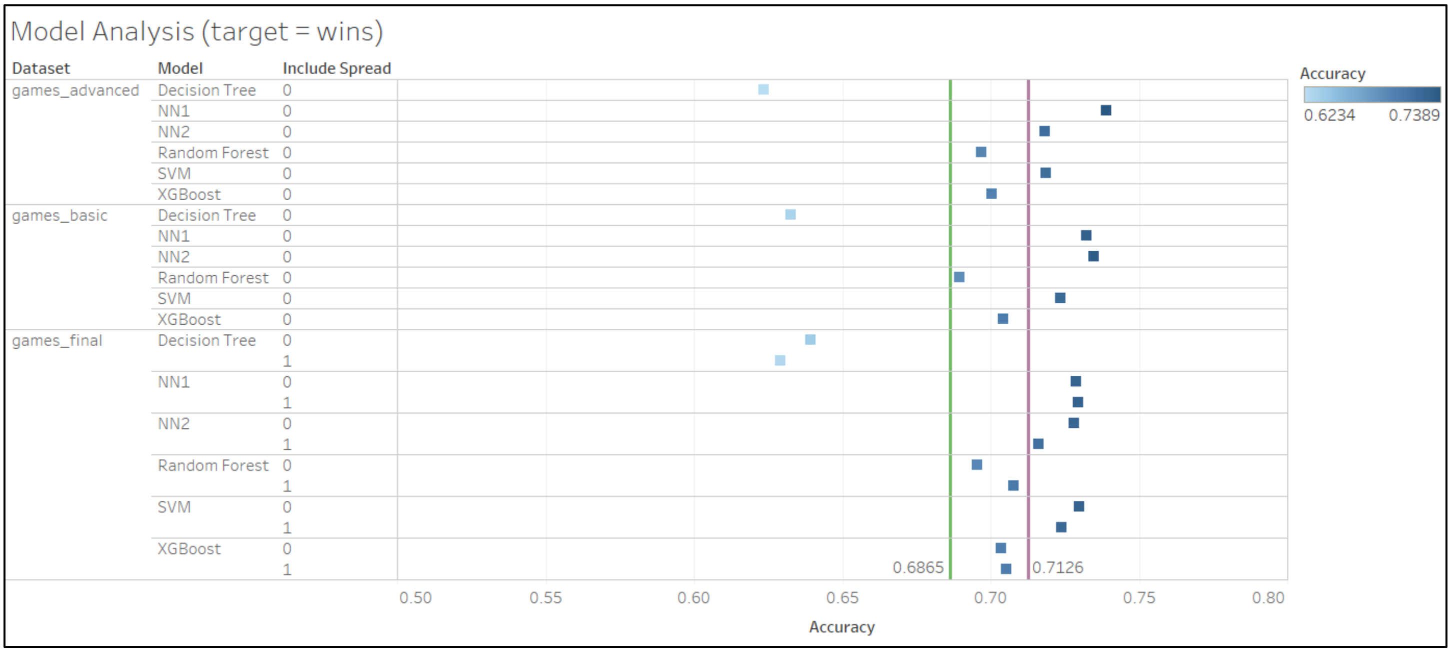


Results

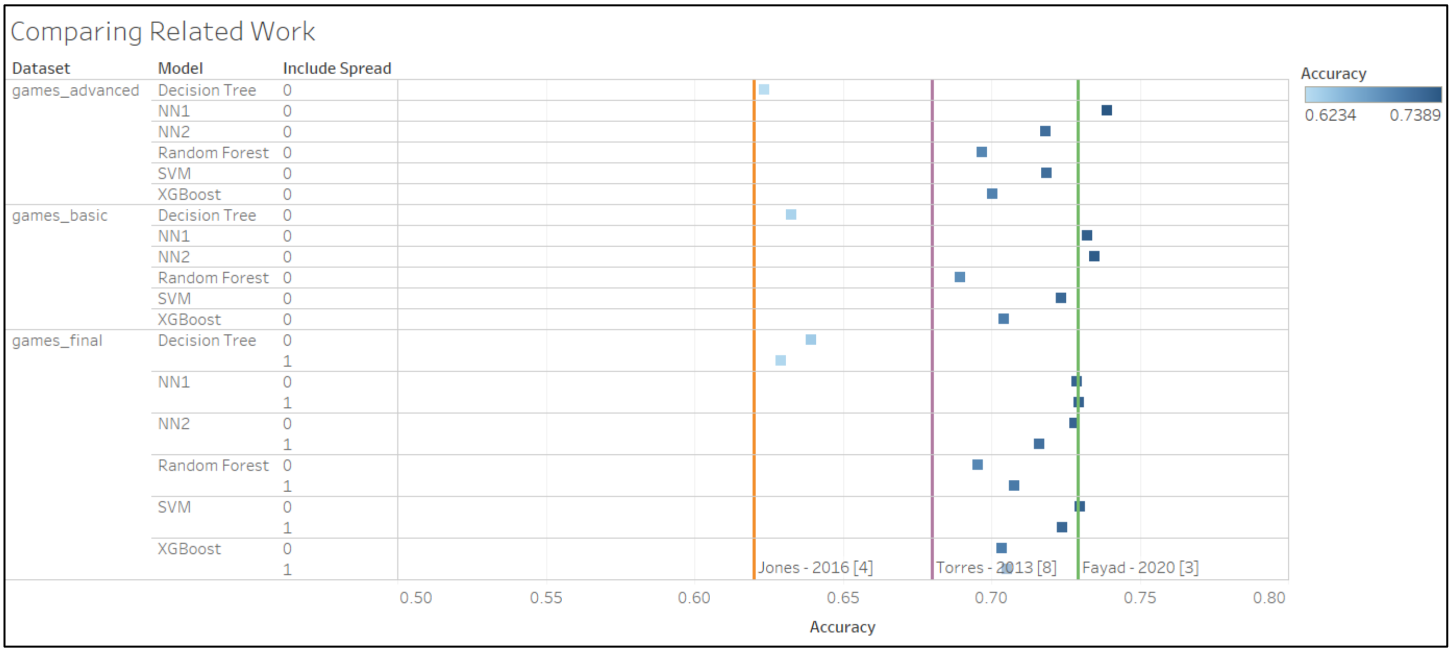
Gambling prediction models performed well; two models scored better than 5% above random chance. In the future, the inclusion of more instances with gambling data will further improve performance.



Win prediction models performed excellently; many outperformed both benchmarks created for the project, and only 16.6% of models fell short of the lower benchmark of 68.6% accuracy.



Some models for NBA regular season win predictions outperformed all related works, the best of which scored an accuracy of 73.89%. Take this with a grain of salt however – data used in this project skews toward the better team winning more often (71.2%) than the historical regular season average (67.9%) – which would mean these games are easier to predict, similarly to playoff games [9].



Evolution

While this project's predictive models performed better than expected, there's always room for improvement. The best way to improve predictive models is to add more data. This could take the form of adding new types of data to each game, or simply gathering more instances (games) of the same data already used. Due to the complications that arose with the Final dataset from adding gambling data and reducing the number of instances, it seems prudent to first collect data on more games, then potentially attempt to add features. Also, another combination of attributes between the Basic and Advanced datasets may have been more predictive than those assembled in the Final dataset. Or maybe the Final dataset would catch up to top performers from other datasets with a few more instances to train from.

There are many potential solutions to this problem, part of what makes data science so intriguing. Ideally, there would be a definitive best dataset to move forward with, but in its absence, a variety of solutions can be tested. In the future, all predictions should be made with neural network 1, the definitive best model for both gambling and win predictions.

Glossary

Artificial Intelligence - the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

Machine Learning - the use and development of computer systems able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data.

Neural Network - Artificial neural networks, usually simply called neural networks, are computing systems vaguely inspired by the biological neural networks that constitute animal brains.

Point Spreads - a forecast of the number of points by which a stronger team is expected to defeat a weaker one, used for betting purposes.

Index

References

[1] Brownlee, Jason. *Machine Learning Mastery*, 19 Sept. 2019, machinelearningmastery.com/.

[2] Cheng, Ge, et al. “Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle.” *Entropy*, vol. 18, no. 12, 2016, p. 450., doi:10.3390/e18120450.

[3] Fayad, Alexander. “Building My First Machine Learning Model: NBA Prediction Algorithm.” *Medium*, Towards Data Science, 9 July 2020, towardsdatascience.com/building-my-first-machine-learning-model-nba-prediction-algorithm-dee5c5bc4cc1.

[4] Jones, Eric Scot. 2016, *Predicting Outcomes of NBA Basketball Games*, library.ndsu.edu/ir/bitstream/handle/10365/28084/Predicting%20Outcomes%20of%20NBA%20Basketball%20Games.pdf?sequence=1&isAllowed=y.

[5] Lauga, N. (2019, December). NBA games data, Version 5. Retrieved February 18, 2021 from https://www.kaggle.com/nathanlauga/nba-games.

[6] Qui, E. (2020, April). NBA Odds and Scores, Version 3. Retrieved February 18, 2021 from https://www.kaggle.com/erichqiu/nba-odds-and-scores.

[7] Sullivan, Ryan. “NBA's Most Valuable Statistic Discovered: How To Predict Team Wins With 95% Accuracy.” *Sports Gambling Podcast*, 20 Apr. 2020, www.sportsgamblingpodcast.com/2020/04/20/nba-most-valuable-statistic/.

[8] Torres, Renato Amorin. 2013, *Prediction of NBA Games Based on Machine Learning Methods*, homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf.

[9] Weiner, Josh. “Predicting the Outcome of NBA Games with Machine Learning.” Medium, Towards Data Science, 7 Jan. 2021, towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bb768f20.