NOVA IMS
Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

MARCH MADNESS PREDICTION USING MACHINE LEARNING TECHNIQUES

João Gonçalo Silva Serra Fonseca

Project work report presented as partial requirement for obtaining the Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2017

MARCH MADNESS PREDICTION USING MACHINE LEARNING TECHNIQUES

JOÃO GONÇALO SILVA SERRA FONSECA

MGI

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

MARCH MADNESS PREDICTION USING MACHINE LEARNING TECHNIQUES

João Fonseca

(joaogssfonseca@gmail.com)

Project work report presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Business Intelligence and Knowledge Management

Advisor: Mauro Castelli

Co-advisor: Ivo Gonçalves

November 2017

ABSTRACT

March Madness describes the final tournament of the college basketball championship, considered by many as the biggest sporting event in the United States - moving every year tons of dollars in both bets and television. Besides that, there are 60 million Americans who fill out their tournament bracket every year, and anything is more likely than hit all 68 games.

After collecting and transforming data from Sports-Reference.com, the experimental part consists of preprocess the data, evaluate the features to consider in the models and train the data. In this study, based on tournament data over the last 20 years, Machine Learning algorithms like Decision Trees Classifier, K-Nearest Neighbors Classifier, Stochastic Gradient Descent Classifier and others were applied to measure the accuracy of the predictions and to be compared with some benchmarks.

Despite of the most important variables seemed to be those related to seeds, shooting and the number of participations in the tournament, it was not possible to define exactly which ones should be used in the modeling and all ended up being used.

Regarding the results, when training the entire dataset, the accuracy ranges from 65 to 70%, where Support Vector Classification yields the best results. When compared with picking the highest seed, these results are slightly lower. On the other hand, when predicting the Tournament of 2017, the Support Vector Classification and the Multi-Layer Perceptron Classifier reach 85 and 79% of accuracy, respectively. In this sense, they surpass the previous benchmark and the most respected websites and statistics in the field.

Given some existing constraints, it is quite possible that these results could be improved and deepened in other ways. Meanwhile, this project can be referenced and serve as a basis for the future work.

KEYWORDS

March Madness; NCAAB; basketball; prediction; classification problem; Machine Learning

INDEX

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

CV	Cross-Validation
DM	Data Mining
DT	Decision Trees
LRMC	Logistic Regression \ Markov Chain
ML	Machine Learning
MLP	Multi-Layer Perceptron
NCAA	National Collegiate Athletic Association
NN	Neural Network
RBF	Radial Basis Function
RF	Random Forests
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
WEKA	Waikato Environment for Knowledge Analysis

BASKETBALL STATISTICS

3PA	3-Points Field Goal Attempted
3PAr	3-Points Field Goal Attempted Rate
3PM	3-Points Field Goal Made
APG	Assists Per Game
AST	Assists
BLK	Blocks
BPG	Blocks Per Game
DE	Defensive Efficiency
DRB	Defensive Rebounds
eFG	Effective Field Goal
FGA	Field Goal Attempted
FGM	Field Goal Made
FTA	Free Throw Attempted
FTf	Free Throw Factor
FTM	Free Throw Made
L	Loss
OE	Offensive Efficiency
ORB	Offensive Rebounds
OPPG	Opponent Points Per Game
PF	Personal Fouls
PPG	Points Per Game
PTS	Points
REB	Rebounds
RPG	Rebounds Per Game
SPG	Steals Per Game
STL	Steals
TO	Turnovers
TS	True Shooting
W	Win

1. INTRODUCTION

For this thesis, it was decided to do a work project. Something that would not be boring, but fun, interesting and, if possible, that is able to produce some monetary return. Two fields that I particularly like were joined: sports and data analysis. After some research about possible projects, it appeared to be interesting to come up with a way to predict basketball results, which is a sport so connected to statistics. Since college basketball in the United States is so mediatic, it seemed to me an excellent challenge.

1.1 Background

The object of the study will be the March Madness. Every year since 1939, at the end of the NCAA basketball season, the final phase is played in March by the best American college teams in a country-wide tournament to determine the NCAA champion. As the tournament has grown, so has its national reputation and March Madness has become one of the most famous annual sporting events in the United States, partially because of its enormous television contracts with TV broadcasters, but mainly because of the popularity of the tournament pools.

Originally, the tournament was composed of 8 teams. The last enlargement took place in 2011, when the number of participants rose from 65 to 68 and, instead of one play-in game (to determine whether the 64th or 65th team plays in the first round) there are four play-in games before all 64 teams compete in the first round. It is speculated that the number of teams be likely to increase.

The selection of the teams is quite complex, and it includes a committee who endeavors to select the most deserving teams and to achieve fair competitive balance in each of the four (East, West, Midwest and South) regions of the bracket. The process consists of three phases:

i. Select the 36 best at-large teams, who did not automatically qualify for the tournament (the remaining 32 teams guarantee the right to participate by having won the conference championship);

ii. Seed all 68 teams (from 1 to 68);

iii. Place the teams into the championship bracket. The matchups are determined after the 1 to 16 seeding by region (#1 seed plays #16 seed, #2 seed plays #15 seed, and so on).

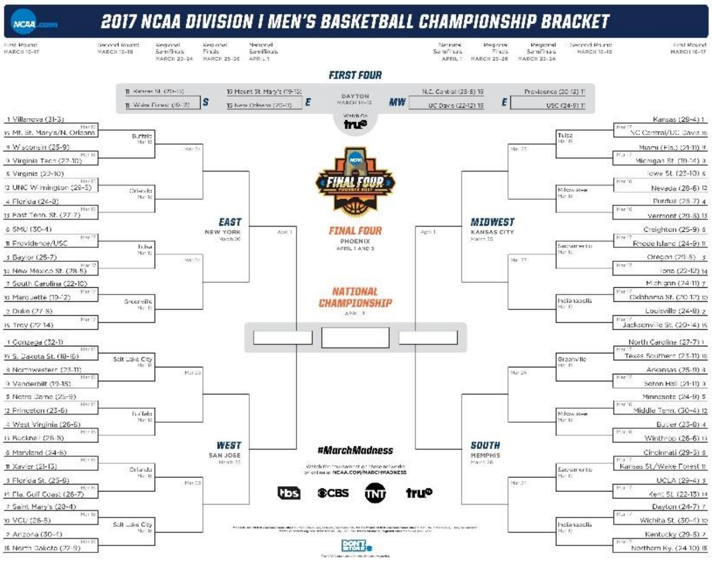


Figure 1 – 2017 March Madness bracket

The initial bracket looks like the one reported in Figure 1 and it is announced on a Sunday, known as selection Sunday. March Madness is a single-elimination tournament where the losers are eliminated, and the winners move on to the next phase. Once the 64 teams that make up the tournament are known, the First Round is played, followed by the Second Round, Regional Semifinals (or Sweet 16), Regional Final (or Elite 8), National Semifinals (or Final 4) and, finally, the National Final where the champion is crowned. Throughout all 6 rounds of the tournament, each game is played at a neutral site rather than on the home court of one team.

1.2 Research problem

This project is a clear challenge against all odds. Ignoring the opening round games, which are not considered in most contests, there is a 64-team pool with 63 games to predict. Given the sporting nature of a basketball game, it also becomes interesting to identify and measure the importance that certain characteristics have on the success of participating in the tournament. Despite being very difficult to reach great accuracies, people continue to research and try their best. Mathematically speaking, perfectly fill a March Madness bracket is one of the most unlikely things on earth:

$$\frac{{}^{63}C_{63}}{2^{32} * 2^{16} * 2^{8} * 2^{4} * 2^{2} * 2^{1}} = \frac{1}{9\ 223\ 372\ 036\ 854\ 775\ 808} \approx 0.000000000000000000108420217248550 44$$

Typically, the goal in these pools is to predict the winners of as many games as possible before the beginning of the tournament. More sophisticated contests incorporate point schemes that award different numbers of points to correct predictions depending on which teams and games are involved: usually, to each round are assigned 32 points. In this sense, picking the teams that play the latest rounds is far more important than picking correctly all first-round results.

Besides this, there is Kaggle: a data science community that has been hosting prediction contests since its inception in 2010. Kaggle contests involve building prediction models or algorithms for specific data questions, often posed by companies that reward the best forecasts. The March Madness contest, called March Machine Learning Mania, started in 2014 and it is divided into two independent stages. The provided data is the same for all participants, but in the course of the contest, many competitors help each other with data sharing, coding, and ideas. In the first stage, Kagglers will rely on results of past tournaments to build and test models, trying to achieve maximum accuracy. The second stage is the real contest where competitors forecast outcomes of all possible match-ups in the tournaments. Contrary to what happens with most tournament pools, in which the winning bracket is the one which successfully predicts the largest number of possible game winners, the goal is to have a greater sum of probabilities for the winners. In the literature review chapter, beyond the results achieved, the methods used by the last winners will also be analyzed.

Another big issue about this topic is the selection of the variables. Pool participants use several sources like specialists' opinions, Rating Percentage Index[1] (a combination of a team and opponent's winning percentage),

[1] Available at http://www.ncaa.com/rankings/basketball-men/d1/ncaa-mens-basketball-rpi

Sagarin's ratings[2] published in USA Today, Massey's ratings[3], Las Vegas betting odds, and the tournament selection committee's seedings. Many researchers are not apologists of these metric and try to fight the subjectivity of the win record and seeding evaluation that are key factors in choosing who receives the at-large bids (Zimmer & Kuethe, 2008; Fearnhead & Taylor, 2010).

1.3 Research Objectives

The main focus of this project is to use several ML algorithms to predict the result of basketball games. Hence, the accuracy of the prediction is a crucial point. Another important part focuses on getting a better insight about variables, trying to overcome results of previous studies by including this knowledge in the formalization of the problem.

To achieve a model with great accuracy is essential to find the best possible combination of variables. Every year, bettors, researchers and pools enthusiasts tend to look at specific metrics such as seeds, team records, and several rankings. In this project, it was tried to build models with a large historical dataset, using previous year's tournament results as input to determine future outcomes of NCAA Tournament.

Decisions must be made as quickly as possible and this challenge of collecting data right after the selection, build solid predictive models and fill brackets must be made by the time of the first first-round game, usually on Thursday, the deadline for most of the tournament challenges. Once the amount of data collected is increasing, another aim must be to find more practical and autonomous methods to extract the raw data and run the algorithms.

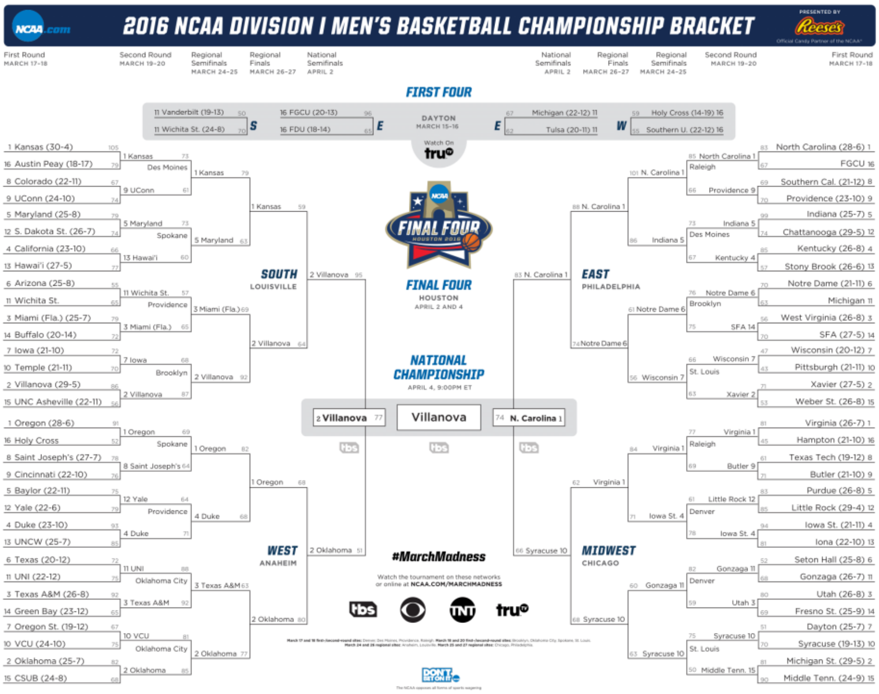


Figure 2 – 2016 March Madness filled bracket

[2] Available at http://www.usatoday.com/sports/ncaab/sagarin/
[3] Available at http://www.masseyratings.com/cb/ncaa-d1/ratings

2. LITERATURE REVIEW

This chapter will review some research and experiments related to the topic. The review is divided in Machine Learning, general basketball predictions and former predictions of the NCAA Basketball Tournament.

2.1 MACHINE LEARNING

CONCEPT

Machine Learning is the subgroup of computer science and artificial intelligence that provides computers the ability to learn, without being explicitly programmed, and perform specific tasks. ML was born in the 50s and 60s from pattern recognition and its focus was the development of computer programs that can change when exposed to new data. The earliest computer scientists like Alan Turing - who invented the Turing Machine, the foundation of the modern theory of computation and computability, and John von Neumann - who defined the architectural principles of a general purpose "stored program computer" on which all succeeding computers were based, had the intention of imbuing computer programs with intelligence, with the human ability to self−replicate and the adaptive capability to learn and to control their environments (Mitchell, 1996).

Ever since computers were invented, there has always been a desire to computers to learn like humans, but Algorithms began to be effective for certain types of learning tasks and many practical computer programs have been developed to exhibit useful types of learning like making accurate predictions, and significant commercial applications have begun to appear on automatic method, without human intervention or assistance. In the field known as Data Mining, ML algorithms, allied to other disciplines such as Probability and Statistics or Computational Complexity, are being used routinely to discover valuable knowledge from large databases (Mitchell, 1997; Schapire, 2008).

A well-defined learning problem requires a well-specified task, performance metric, and source of training experience. Designing a ML approach involves many design choices, including choosing the type of training experience, the target function to be learned, a representation for this target function, and a sequence of computational steps that takes inputs and produces output, usually called algorithms, for learning the target function from training examples (Mitchell, 1997; Cormen, 2009).

With the evolution of IT - cheaper data storage, distributed processing, more powerful computers, and the analytical opportunities available the interest in ML systems that can now be applied to huge quantities of data have dramatically increased (Bucheli & Thompson, 2014).

OBJECTIVES

The primary goal of ML research is to develop general purpose efficient algorithms of practical value and solve a certain problem. The best would be to look for models that can be easily applied to a broad class of learning issue. ML focuses on the construction and study of systems that can learn from data (data driven) and analyze massive datasets (Bucheli & Thompson, 2014).

To a ML algorithm is given a “teaching set” of collected data for a concrete problem, then asked to use that data to answer a question or solve a specific task. For instance, you might provide a computer a teaching set of photographs, some of which say “this is a cat”, some of which say “this is not a cat” and then show the computer a series of unseen photos and it would be able to identify which photos were of cats (Marr, 2016).

The accuracy of the prediction is extremely important, especially in fields like sciences and medicine. Every researcher wants his model to be as accurate as possible, since there are no infallible models. Other relevant concerns are about the amount of data that is required by the learning algorithm and the interpretability of the results: in some contexts, it is essential to find outcomes that are easily understandable in order to support decisions. Briefly, the goal of ML is to develop deep insights from data assets faster, extract knowledge with greater precision, improve the bottom line and reduce risk (Bucheli & Thompson, 2014; Schapire, 2008).

APPLICATIONS

ML algorithms have proven to be of great practical value in a variety of application domains. They are especially useful in automatically discover patterns, explore poorly understood domains where humans might not have the knowledge needed to develop effective algorithms and develop dynamic programs adaptable to changing conditions (Mitchell, 1997). Following are some examples of studies and research carried out in the ML field:

- Customer segmentation and consumer behavior

Faced with constant changes, the market becomes increasingly competitive. In this sense, companies are more and more concerned about customers, mainly on the quality of services provided and their satisfaction, trying to attract, retain and cultivate consumers. Early in 2004, with the support of SVMs, was shown that in the very noisy domain of customer feedback, it is nevertheless possible to perform sentiment classification (Gamon, 2004). Dynamic pricing and ML techniques were also studied. Facilitated by statistical, DM methods and ML models the study sought to predict the purchase decisions based on adaptive or dynamic pricing of a product. The results were encouraging enough to implement the framework completely (Gupta & Pathak, 2014);

- Drive autonomously a vehicle.

Autonomous driving systems which can help decrease fatalities caused by traffic accidents and this kind of everyday tasks are the major challenges in modern computer science. Back in the 90s, ALVINN, a backpropagation network, was developed to autonomously control a Chevy van by watching a human driver’s reactions in several circumstances including single-lane paved and unpaved roads, and multilane lined and unlined roads, at speeds of up to 20 miles per hour (Pomerleau, 1991). More recently, computer vision was combined with deep learning to bring about a relatively inexpensive, robust solution to autonomous driving. Using a large data set of highway data and apply deep learning and computer vision algorithms it was proved that convolutional NN algorithms are capable of reliable performance in highway lane and vehicle detection (Huval et al., 2015);

- Financial sector issues

An overview study (Husain & Vohra, 2017) shown existing applications of ML in the financial sector as loan approvals, asset management, risk profiling, trading or market predictions. Particularly in the fraud detection, ML techniques are useful to identify irregular transactions and some experiments tested ML algorithms and meta-learning strategies on real-world data (Stolfo et al., 1997). This topic was analyzed by Fawcett and Provost (1996) and they combined DM and ML techniques to design methods for detect fraudulent usage of cellular telephones based on profiling customer behavior. Specifically, they used a rule learning program to uncover indicators of fraudulent behavior from a large database of cellular calls and subsequently generate alarms;

- Recognize spoken words.

All of the most successful speech recognition systems employ ML in some form. For example, some Silicon Valley researchers have presented an end-to-end deep learning-based speech system capable of outperforming existing state-of-the-art recognition pipelines in two challenging scenarios: clear, conversational speech and speech in noisy environments (Hannun et al., 2014). In another study, it was shown that the combination of deep, bidirectional Long Short-term Memory RNNs with end-to-end training and weight noise gives state-of-the-art results in phoneme recognition on a specific database (Graves et al., 2013). Using the same database, it was also used a new type of deep NN that uses an SVM at the top layer (Zhang et al., 2015). Both deep recurrent NN and SVM are two ML features;

Besides all these topics, there are many other fields where ML techniques are used such as spam filtering, weather forecast, medical diagnosing and topic spotting (categorize news articles as to whether they are about politics, sports, entertainment). This project is part of one the ML field, called task-oriented studies, that consists of the development and analysis of learning systems oriented toward solving a predetermined set of tasks, also known as the “engineering approach” (Carbonell et al., 1983).

2.2 Predictions

Intuition and data-based decisions

In any sport, people always like to bet and predict who will win a certain event or a particular game. Often, when presented with a decision like filling a March Madness bracket it is usual to have an instinctive sense that one alternative is better than others. Intuition is hard to define, but feelings, experience and the ability to detect patterns, even unconsciously, are definitely part of it. Some top executives are the first to admit that statistical models based on rules typically outperform and are more consistent than human experts’ gut. For them, the major problems leading to bad decisions are:

- The tendency to identify inexistent patterns, what statisticians call overfitting of the data;
- The abundance or paucity of emotion;
- Lack of feedback – without knowing about the mistakes, it is impossible to learn from them.

Despite the abundance of data and analytics at their disposal, experienced managers occasionally opt to rely on gut instinct to make complex decisions (Hayashi, 2001; Matzler et al., 2007; Seo & Barrett, 2007).

On the other hand, this trend tends to reverse and there are good reasons to believe in data-based decision-making. Buzzwords like Big Data, Data Mining or Data Science have more and more importance in business world. Retail systems are increasingly computerized and merchandising decisions were automated. Famous examples include Harrah's casinos' reward programs and the automated recommendations of Amazon and Netflix (Provost & Fawcett, 2013). Its benefits were demonstrated conclusively by Brynjolfsson et al. (2011) who conducted a study of how data-based decision-making affects firms' performance. They showed statistically that the more data-driven a firm is, the much more productive it is and there is a positive association with the return on assets.

In the NBA, in the mid 90's, coaches started to use a PC-based Data Mining application called Advanced Scout. This tool helped staff to discover interesting patterns in their strategies such as shooting performance or possession analysis to determine optimal line-up combinations. These types of analyses could be more enriched and more valuable through by inference rules and the combination with coaches' expertise (Bhandari et al., 1997).

Predictions in basketball

Many studies and experiments have been made to counter decisions based on intuition. Basketball is full of statistics and, specifically in this sport, data are increasingly important and massive amounts of them are collected for every team. There are individual and collective, offensive and defensive stats and entire teams all have an immensity of data that attempt to quantify how any part of their game is performing.

In the early days of sport, the analyses were limited to basic operations such as averages, counts and sums calculations. Over time, statistics experts have begun to deepen and refine this type of analysis. In a period when access to information was still limited, one of the first studies (Zak et al., 1979) approached the topic in an econometric way on a statistical basis. The objects of study were games played by teams from the Pacific Division during the 1976-77 season. Although the sample is based on this five teams (Boston Celtics, Buffalo Braves, New York Knicks, New York Nets and Philadelphia 76ers), the schedule granted the representation of all teams in the league.

The statistical methods used, Cobb-Douglas production functions and the Ordinary Least Squares method, are easy to interpret due to elasticities and it is simple to understand the input variables' impact on the output despite admitting some randomness from match to match, being thus possible to identify game features where the team should improve.

The features used were the ratio of the final scores as output and ratios of shooting (FG% and FT%), offensive and defensive rebounds, ball handling (assists), defense (steals and the difference in number of blocked shots) and negative aspects of the game like personal fouls and turnovers, in addition to a binary variable for location as inputs. The adoption of ratios makes sense because the main goal of sports is to have a better relative performance than the other team. The results of this experiment can be found in the table below:

Variables	**League**	**Boston**	**Buffalo**	**N. Y. Knicks**	**N. Y. Nets**	**Philadelphia**
Constant	-.0016	-.0064	-.0040	-.0084	.0011	-.0001
Log (FG%)	.6136 * (20.395)	.5511 * (8.158)	.6562 * (10.437)	.5500 * (8.494)	.5839 * (8.872)	.6634 * (8.958)
Log (FT%)	.1137 * (8.581)	.0760 * (2.466)	.1677 * (6.295)	.0979 * (3.269)	.1308 * (4.378)	.1260 * (4.378)
Log (Offensive rebounds)	.0812 * (13.144)	.0847 * (6.828)	.0900 * (7.518)	.0554 * (4.174)	.0873 * (6.370)	.0829 * (4.836)
Log (Defensive rebounds)	.0610 * (3.103)	.0839 * (1.958)	.0438 (1.150)	.1182 * (2.571)	-.0034 (-.082)	.0354 (.681)
Log (Assists)	.0116 (1.289)	.0364 * (1.727)	-.0092 (-.509)	.0204 (.087)	.0333 * (1.723)	.0053 (.250)
Log (Personal fouls)	-.1175 * (-12.013)	-.1706 * (-7.179)	-.0952 * (-5.890)	-.1212 * (-4.658)	-.1418 * (-5.773)	-.1196 * (-4.486)
Log (Steals)	.0165 * (3.181)	.0138 (1.441)	.0114 (1.234)	.0400 * (2.710)	.0268 * (1.879)	.0254 * (2.199)
Log (Turnovers)	-.1216 * (-11.018)	-.0908 * (-4.540)	-.1287 * (-6.380)	-.1434 * (-5.149)	-.0752 * (-2.842)	-.1231 * (-4.448)
Home court (=1)	.0067 (1.276)	.0075 (.690)	.0038 (.379)	.0150 (1.113)	-.0013 (-.117)	.0418 (1.304)
Blocked shots [a]	-.0003 (-.388)	-.0024 (-1.357)	.0003 (.310)	-.0015 (-.745)	-.0006 (-.360)	.00004 (.028)
R^2	87,37 %	86,40 %	90,41 %	85,08 %	85,50 %	87,52 %
Number of games	357	77	79	81	78	79

[a] Difference in blocked shots in a game; * Significant at the 5% level (one-tailed test)

Table 1 – Zak et al.'s production function estimates

At the 5% level, most of the coefficients are statistically significant. The largest output elasticities are associated with shooting percentages, particularly FG%, being the elasticity of FT% comparatively lower while rebounds and, in several cases, contribute substantially to output. On the other hand, personal fouls and turnovers reduce output and the difference in blocked shots and the ratio of assists proved to be insignificant. The coefficient on the locational variable is consistently insignificant which may mean that, therefore, this variable does not have an impact on its own, but may have an impact on the remaining inputs.

The study also allowed to develop an estimate of the performance of the team according to its resources, like a power ranking. Taking the logarithm of the production function, equation yields:

$$\ln Y = \ln F(x) + \ln u = [\ln F(x) - \lambda] + [\lambda - v].$$

The team frontier (a limit based on team stats), was calculated using the mean values for all inputs and estimated coefficients. For the investigators, a team's actual performance is a combination of its potential (the frontier output) and its efficiency. Multiplying the frontier output by the level of efficiency yields expected output, and teams can be ranked on this basis.

Variables	League	Boston	Buffalo	N. Y. Knicks	N. Y. Nets	Philadelphia
Frontier output	1.0025	1.0049	.9804	1.0190	.9589	1.0486
Variance (λ)	.00185	.00177	.00127	.00219	.00201	.00154
Efficiency ($2^{-\lambda}$)	.99872	.99877	.99912	.99849	.99861	.99893
Frontier output x efficiency	1.0012	1.0037	.9795	1.0175	.9576	1.0475
Estimated rank	-	3	4	2	5	1
Actual rank	-	3	4	2	5	1

Table 2 – Zak et al.'s estimated production output

The results were identical to the league standings for that season. The next step was to find the marginal productivity of each input, given by:

$$(Marginal\ Productivity)_i = \alpha_i \frac{\bar{Y}}{\overline{X_\imath}}$$

Variables	League	Boston	Buffalo	N. Y. Knicks	N. Y. Nets	Philadelphia
FG%	.6245	.5445	.6637	.5262	.5858	.6449
FT%	.1132	.0733	.1600	.0943	.1238	.1307
Offensive rebounds	.0737	.0636	.0792	.0565	.0734	.0879
Defensive rebounds	.0553	.0753	.0428	.1169	-.0036	.0318
Assists	.0121	.0326	-.0100	.0185	.0398	.0054
Personal fouls	-.1178	-.1590	-.1033	-.1182	-.1182	-.1330
Steals	.0160	.0156	.0093	.0408	.0211	.0244
Turnovers	-.1094	-.0702	-.1129	-.1329	-.0739	-.1140
Home court	.0135	.0147	.0071	.0298	-.0025	.0300
Blocked shots	.0008	77	79	.0011	.0009	.0002

Table 3 – Zak et al.'s estimated marginal products of inputs

In most of the cases, a higher output elasticity implies a larger marginal product.

The last step of the research was to find if a host factor exists. By performing Chow test, the conclusion was that all teams, except the New York Knicks, performed significantly better playing home that away, mainly in shooting and rebounding elements. This research is interesting because it is possible to see that the same combination of factors can have different worth to different teams and by using this logic a team could evaluate players based on their contribution to output and choose those players that increase output.

Outside the NBA world, Ivanković et al. (2010) studied the Serbian basketball league from 2005-06 to 2009-10 seasons, the equivalent of 890 games. In Serbia, the basketball court is divided into eleven positions: six from 2-point shots and five from 3-point shots, and the main goal was to analyze the influence of shooting from different field positions and, after that, the influence of regular basketball parameters on winning.

In the first analysis, the model was composed of variables that cover the type of throw and the area (i.e., p21_percent stood for 2-point shots percentage from position 1) and an output parameter for the final result. The algorithm used was a feed-forward NN with one hidden layer fully connected to all nodes. The results were the following:

Variable	Influence (%)
p1_percent	12.1
p21_percent	2.2
p22_percent	3.5
p23_percent	3.7
p24_percent	2.3
p25_percent	31.4
p26_percent	2.7
p31_percent	8.9
p32_percent	5.8
p33_percent	11.3
p34_percent	6.7
p36_percent	9.6

Table 4 – Results of Ivanković et al.'s 1st experiment

It was visible that the two-points shot from position five, underneath the basket, had the highest influence on winning the game, followed by one-point shots (free throws) and then three-point shots. Midrange shots from other positions had the least influence. The model obtained a 66.4% accuracy, possibly due to lack of other important variables. In the next experiment the regular box-score stats were evaluated:

Variable	Influence (%)
FT%	7.96
2P%	15.58
3P%	15.35
DRB	15.88
ORB	12.14
AST	2.23
STL	12.53
TO	12.39
BLK	5.94

Table 5 – Results Ivanković et al. 2nd experiment

The conclusions are that shooting, and rebounding are the main factors and steals and turnovers could also have a vital role. The accuracy of this model reached almost 81%.

Recently, the evolution of technology, the growing popularity of the NBA and the accessibility of data allowed more complex experiments, particularly in the ML, DM and Data Analysis fields. Loeffelholz et al. (2009) used 2007-2008 season team statistics, box score lines as inputs and a binary variable (0, 1) as output. The researchers used 4 types of NN (feed-forward NN, RBF, probabilistic NN and generalized NN) and two fusions that can help NN to complement themselves: a Bayesian Belief Network (BBN) and a Probabilistic NN Fusion.

In a second phase, a reduction of dimensionality was made. One approach used the Signal-to-Noise-Ratio method that examines the lower level weights of Feed-Forward NN and withdraws the less important features of the dataset. The other was based on using shooting statistics (FG, 3P, and FT), as suggested by different experts which infer a good offense wins basketball games (and good defense championships). A factor analysis showed a high correlation between FG and 3P and that is why the last feature has only 4 variables.

These experiments used a 10-fold CV to provide accurate estimates of the NN performance having 10 different validation sets. The first one is notably important because it contains game played after the rest. The results are given in Table 6 and are compared to experts' predictions:

	Accuracy (%)							
Technique	**V1 Baseline**	**Baseline**	**V1 SNR (TO & PTS)**	**SNR (TO & PTS)**	**V1 Shooting (FG, 3P & FT)**	**Shooting (FG, 3P & FT)**	**V1 Shooting (FG & FT)**	**Shooting (FG & FT)**
Feed-forward NN	70	71.67	70	70.67	80	72.67	83.33	74.33
RBF	66.67	68.67	70	69	73.33	68	70	72
Probabilistic NN	70	71.33	70	69	80	72.33	83.33	73.34
Generalized NN	70	71.33	70	69	80	72.33	83.33	73.34
PNN fusion	70	71.67	70	70.67	80	72.67	76.67	72.67
Bayes fusion	70	71.67	70	70.67	80	72.67	80	74
Experts	**70**	**68.67**	**70**	**70**	**70**	**68.67**	**70**	**68.67**

Table 6 – Loeffelholz et al.'s research results

In NBA Oracle, Beckler et al. (2013) applied ML methods for predicting game outcomes, infer optimal player positions and create metrics to identify outstanding players. Teams' dataset had 30 features each season and players' dataset had 14 basic individual player statistics. Additionally, it was possible to create *per game* and *per minutes* derived statistics. When comparing two teams, investigators normalized teams' stats by taking the ratio of each team's numbers for an easier understanding of relative advantage.

Focusing on the first task, there were applied 4 different ML classification techniques: Linear Regression, SVM, Logistic Regression and Artificial NN with a 100-fold CV and, for each one, a classification performed with previous season stats (P) and previous plus current season stats (P+C). Below is the test sets classification accuracy:

	Accuracy (%)					
Techniques	**1992**	**1993**	**1994**	**1995**	**1996**	**Average**
Linear (P)	69.45	71.10	67.09	68.82	73.09	69.91
Linear (P+C)	69.55	70.70	68.36	69.82	72	70.09
Logistic (P)	67.36	69.10	65.27	67.73	67.73	67.44
Logistic (P+C)	68	70	67.36	68.91	69.55	68.76
SVM (P)	64.45	66.80	65.27	65	68.27	65.96
SVM (P+C)	65.27	69.80	66.55	68.64	69.27	67.91
ANN (P)	64.73	66.01	62.36	64.15	66.64	64.78
ANN (P+C)	63.09	66.20	64	67.54	65.95	65.36

Table 7 – NBA Oracle's results

The main conclusions are that the simplest Linear Regression outperformed other ML algorithms and the inclusion of data from the current year in the models can improve accuracy. The NBA Oracle obtained results show percentages of accuracy similar to previous studies and, in some years, are even better than the experts.

To better understand the importance of each feature in this implementation, researchers tried single feature models at a time. The most dominant feature in the dataset was the win record in the previous season (65.9%), being possible to notice a correlation between the past and future results. Furthermore, in order of decreasing importance: defensive rebounds, points made by opposing team, number of blocks and assists made by opposing team are also noteworthy features that reveal the importance of defense in order to win a game.

Lin et al. (2014) used NBA data from 1991 to 1998 to predict the winners of matchups and determine the most key factors to the outcome of a game without looking at individual player statistics. They began by setting benchmarks to compare with the research's results. The accuracy of the expert predictions is inflated thus should not be considered as a goal.

Method	Accuracy (%)
Team with greater difference between points per game and points allowed	63.5
Team with greater win rate	60.8
Expert prediction (not include games deemed too close to call)	71

Table 8 – Lin et al.'s benchmarks

The variables used were the differences in the teams' stats: win-loss record, PTS scored, PTS allowed, FGM and FGA, 3PM and 3PA, FTM and FTA, ORB and DRB, TO, AST, STL, BLK, PF and, additionally, the recent performance of a team. The discussion about the impact of the recent performance on future results is long and it is usually called Hot Hand Fallacy. Beside some literature who support this theory (Bocskocsky et al., 2014), researchers tried to explain future results based only on the recent performance (between 1 and 20 games) and they achieve an accuracy peak of around 66%.

Generally, all ML models suffered from overfitting and poor accuracy, so it was tried to find a better set of variables using three separate feature selection algorithms: forward and backward search with a 10-fold CV, adding or removing features one by one in order to determine which features result in the highest prediction accuracies, apart from a heuristic feature selection algorithm.

Feature selection algorithms	Forward Search	Backward Search	Heuristic
Variables	Points Scored	Points Scored	Points Scored
	Points Allowed	FGA	FGA
	FGA	DRB	FTM
	DRB	AST	DRB
	AST	TO	AST
	BLK	Overall record	Overall record
	Overall record	Recent record	Recent record

Table 9 – Lin et al.'s feature selection algorithms results

Backward search variables are very similar to the heuristic approach and it is in line with the experts' view of the game, that takes into consideration the offensive, team possessions and scoring potential of a team compared to its opponent.

All experiments used 1997-98 season data as test set and the remaining as training set. Teams for different years were considered independent from each other due to trades, staff changes, retirements or players' development and decline that can cause high variance in the strength of a team from year to year. Therefore, the feature vectors on the team's performance only considered the current season and, consequently, the evaluation of a team's strength would be less accurate at the start of the seasons. The ML techniques used were Logistic Regression, SVM, Adaptive Boost, RF and Gaussian Naive Bayes, all with a 10-fold CV. The first experiment used selected features to pick a winner and the results were:

	Accuracy (%)	
Technique	**Training**	**Test**
Logistic Regression	66.1	64.7
SVM (RBF Kernel, Cost = 10)	65.8	65.1
AdaBoost (65 iterations)	66.4	64.1
Random Forest (500 trees, Depth = 11)	80.9	65.2
Gaussian Naïve Bayes	63.1	63.3
Benchmark	**63.5**	

Table 10 – Results of Lin et al.'s 1st experiment

It is possible to see that the techniques' accuracy outperformed the baseline benchmark by a small margin and some of the algorithms, especially RF, overfitted data. The second experiment tried to explore how the accuracy of win classifications performed over time. Seasons were partitioned into 4 quarters and the algorithms were tested on games occurring within each of these 4 parts.

	Accuracy (%)			
Technique	**Quarter 1**	**Quarter 2**	**Quarter 3**	**Quarter 4**
Logistic Regression	58.8	64.1	66.2	68.8
SVM (RBF Kernel, Cost = 10)	58.8	64.5	65.7	67.8
AdaBoost (65 iterations)	55.9	61.8	62.4	67.8
Random Forest (500 trees, Depth = 11)	55.9	67.3	63.8	64.4

Table 11 – Results of Lin et al.'s 2nd experiment

As expected, accuracy has a trend of improvement over time, reaching nearly 70% in the final part of the season, much higher than the baseline utilizing simply the win-loss record. In the last experiment, it was tested the impact of the win-loss record in the accuracy of the model. Unlike the first experiment, the variable was not applied here.

	Accuracy (%)	
Technique	**Training**	**Test**
Logistic Regression	66.3	64.5
SVM (RBF Kernel, Cost = 10)	66.1	63.7
AdaBoost (65 iterations)	67.2	61.8
Random Forest (500 trees, Depth = 11)	88.6	62.8
Gaussian Naïve Bayes	56.0	59.9
Benchmark	**63.5**	

Table 12 – Results of Lin et al.'s 3rd experiment

Results show that the accuracies obtained from using only box score performs reasonably well, but fall short of the benchmark for some models. This indicates that advanced statistics that go beyond the box score are needed to increase accuracy and win record represents a significant role in this kind of exercise.

Richardson, et al. (2014) tried a different approach to make NBA predictions with the use of Regularized Plus-Minus (RPM). This concept was introduced by Engelmann, and it shares a family resemblance with the Plus-Minus stat, which registers the net change in score (plus or minus) while a player is on the court. The problem here is that each player's rating is deeply affected by his teammates' performance. RPM isolates the unique impact of each player by adjusting for the effects of each teammate and opposing player apart from being able to divide in the offensive (ORPM) and defensive (DRPM) impact (Illardi, 2014).

Using several data sources, investigators built a database containing players, teams and games details. To create the model features, they merged the player's stats from the previous season with the results in the matches of the current season and formed home and away, offensive and defensive statistics, using as weights the players' average minutes per game from the previous year. Moreover, a label was added indicating whether or not the home team won the game. In the experimental phase, researchers used previous seasons as training set and that particular year as test set. To predict games, algorithms such as Linear Regression, Logistic Regression, Naïve Bayes, SVM and DT were used. The results, where it is possible to see the best accuracy of the linear regression, were the following:

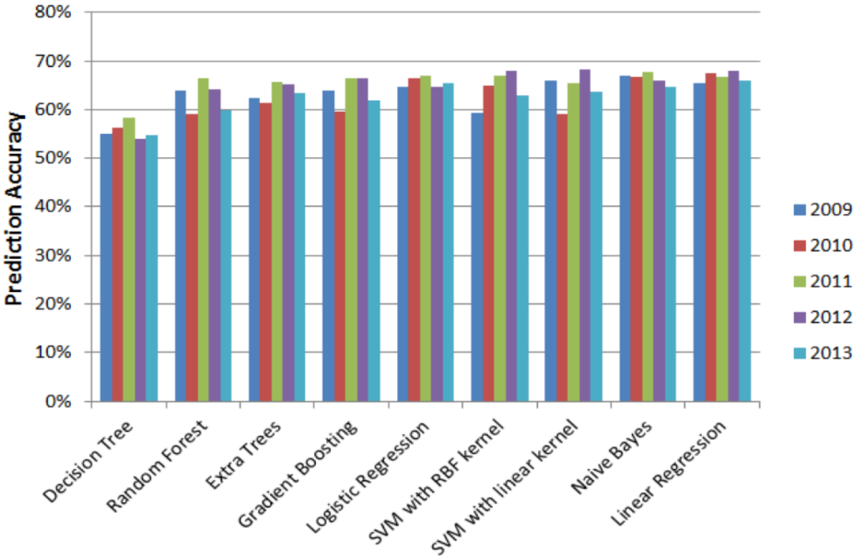


Figure 3 – Richardson et al.'s 1st experiment

All these models used 44 variables and a feature selection was performed to prevent overfitting and thus improve prediction accuracy. For the linear regression model, researchers used Lasso regularization (to minimize the generalized CV error) and a stepwise AIC procedure (where variables are included or dropped according to the upgrading of the model). The results were the following:

	Accuracy (%)					
Techniques	2009	2010	2011	2012	2013	Average
Full model (linear)	65.48	67.4	66.77	67.86	66.02	66.71
Lasso	66.59	68.54	66.26	67.13	66.34	66.97
AIC	66.18	67.89	67.17	68.19	66.02	67.09

Table 13 – Richardson et al.'s 2nd experiment

The stepwise AIC found a model with higher accuracy, although there are no big differences to the original. Another conclusion is the identification of the overall team weighted RPM as the most important predictive feature because the AIC procedure started with the inclusion of home and away RPM to the model. For the investigators, it was clear that home court advantage and the quality of each team are preponderant factors and RPM stats have a greater predictive power because they contain more information. On the other hand, new methodologies like the use of cameras that reveal detailed information every second can make this type of approach outdated.

More recently, Cheng et al. (2016), tried to forecast NBA playoffs using the concept of entropy. The first step was collecting the main statistics of 10 271 games from 2007-08 to 2014-15 seasons and labelling it with a win or loss for the home team. This dataset was used to train the NBA Maximum Entropy (NBAME) model, also known as Log-Linear model, by the principle of Maximum Entropy and predict the probability of the NBA playoffs game home team's win for each season based on probabilities.

Maximum Entropy models are designed to solve the problems with insufficient data like predicting the NBA playoffs. The principle points out the best approximation to the unknown probability distribution, making no subjective assumptions and decreasing the risk of making wrong predictions. It has been widely used for Natural Language Processing tasks, especially for tagging sequential data.

Researchers applied 28 basic technical features (FGM, FGA, 3PM, 3PA, FTM, FTA, ORB, DRB, AST, STL, BLK, TO, PF and PTS for both teams) of the coming game to the NBAME model and calculated the probability of the home team's victory in the game, p(y|x). Since p(y|x) is a continuous value, the model makes a prediction based on a defined 0.5 threshold:

$$f_k(x,y) = \begin{cases} 1\ (win), & p(y|x) \geq 0.5, \\ 0\ (lose), & p(y|x) < 0.5. \end{cases}$$

Assuming that the probability of the home team winning is higher, 0.6 and 0.7 thresholds were also used. In these cases, increasing the level of confidence, there is a decrease in the number of games to predict. The accuracy of the NBAME model was calculated by the following formula:

$$Accuracy = \frac{\#\, correct\ predictions}{\#\, predictions}$$

In the following table is possible to see the prediction accuracy of the NBAME model and the number of predicted games:

	Accuracy (%) & number of predicted games								
Thresholds	**2007-08**	**2008-09**	**2009-10**	**2010-11**	**2011-12**	**2012-13**	**2013-14**	**2014-15**	**Average**
0.5	74.4 (86)	68.2 (85)	68.3 (82)	66.7 (81)	69 (84)	67.1 (85)	65.2 (89)	62.5 (80)	67.71
0.6	77.1 (48)	74.5 (55)	75 (44)	69.8 (53)	73 (26)	71.4 (42)	66.7 (36)	70.4 (27)	72.5
0.7	100 (3)	80 (5)	100 (2)	- (0)	100 (1)	75 (4)	100 (1)	100 (6)	90.91

Table 14 – Cheng et al. 1st experiment

When compared to other ML algorithms in WEKA, the results were the following:

	Accuracy (%)								
Technique	**2007-08**	**2008-09**	**2009-10**	**2010-11**	**2011-12**	**2012-13**	**2013-14**	**2014-15**	**Average**
Back Propagation NN	59.3	60.4	52.4	67.9	56	63.5	57.1	57.5	59.25
Logistic Regression	61.6	57.1	61	61.7	60.7	64.7	62.6	60	61.19
Naïve Bayes	54.7	61.5	56.1	59.3	53.6	58.8	59.3	55	57.3
Random Forest	64	60.4	64.6	64.2	58.3	70.6	62.6	56.3	62.66
NBAME model	**74.4**	**68.2**	**68.3**	**66.7**	**69**	**67.1**	**65.2**	**62.5**	**67.71**

Table 15 – Comparison of Cheng et al. 1st experiment with other models

Overall, the NBAME model is able to match or perform better than other ML algorithms.

FiveThirtyEight is established as one of the most reputable informative websites. It has covered a broad spectrum of subjects including politics, sports, science, economics, and popular culture. In addition to its much-recognized data visualization and electoral forecasts, the site also held forecasts under the NBA scope, both at the level of games, chances of reaching the playoffs or winning the NBA championship as well as daily power rankings.

The website has two ratings: the Elo Rating and CARMELO. Elo Ratings, created by Arpad Elo, was originally used for calculating the relative skill levels of chess players. FiveThirtyEight (Silver & Fischer-Baum, 2015) recreated it (538 Elo Rating) for many other sports and used it in NBA to find the best teams of all time and visualize the complete history of the league. The essential features of Elo Rating are:

- The ratings depend only on the final score of each game and where it was played (home-court advantage) and it includes both regular-season and playoff games.
- Teams' Elo points increase after wins and decrease after defeats. They gain more points for upset wins and for winning by wider margins.
- The system is zero-sum: the gains of a team are balanced with the losses of the opponent.
- Ratings are established on a game-by-game basis.

The long-term average Elo rating is 1500, although it can differ slightly in any particular year based on how recently the league has expanded and only historically teams fall outside the 1300 (pretty awful)-1700 (really good) range. This method has a few NBA-specific parameters to set: The K-factor, home court advantage, margin of victory and year-to-year carry-over.

Elo's K-factor determines how quickly the rating reacts to new game results. It should be set so as to efficiently account for new data but not overreact to it, minimizing autocorrelation. The defined K for the NBA to is 20, implying a relatively high weight to an NBA team's recent performance.

Home-court advantage is set as equivalent to 100 Elo Rating points, the equivalent of about 3.5 NBA points, so home team would be favored if teams were otherwise evenly matched. Some teams (especially Denver and Utah that play at high altitudes) have historically had slightly larger home-court advantages.

Elo strikes a nice balance between rating systems that account for margin of victory and those that don't. This works by assigning a multiplier to each game based on the final score and dividing it by a team's projected margin of victory conditional upon having won the game. The formula accounts for diminishing returns; going from a 5-point win to a 10-point win matters more than going from a 25-point win to a 30-point win.

Instead of resetting each team's rating when a new season begins, Elo carries over three-quarters of a team's rating from one season to the next. Compared to other sports, the higher fraction reflects the fact that NBA teams tend to be consistent. Although having some nice properties, this method doesn't consider offseason trades and drafted players' impact on a team's performance. In the past, the solution was to revert the season before Elo Ratings toward the mean for the preseason ratings, but with FiveThirtyEight's CARMELO projections it is possible to have better priors to account for offseason moves.

The CARMELO (Career-Arc Regression Model Estimator with Local Optimization) algorithm (Silver, 2015), inspired on PECOTA (Silver, 2003), forecasts NBA players' performance identifying historical similar careers (CARMELO Card). Three steps constitute this process:

1. Define each player's skills and attributes statistically. Primary, biographical aspects such as height, weight and draft position, being the most vital age. Then some basketball stats that reflect the weighted average of a player's performance over his past three seasons, considering the minutes played in each season too.

2. Identify comparable players. CARMELO runs a profile for past NBA players since 1976 with the same age and identifies the most similar ones from 100 (perfect similarity) to negative values. By CARMELO standards, many NBA players don't have any comparison with a similarity score above 50. For this calculation, each of the 19 categories has its weight which is as follows:

Statistic	Weight	Notes
Position	3.0	Positions are translated to 1 (Point Guard) to 5 (Center) scale.
Height	3.5	-
Weight	1.0	-
Draft Position	2.5	Taken as a natural logarithm. Undrafted players are treated as 90th pick
Career NBA minutes played	1.5	-
Minutes per game	3.5	Overall record
Minutes played	6.0	For historical players, minutes for seasons shortened are prorated to 82 games
TS%	5.0	-
Usage %	5.0	-

FT %	2.5	-
FT frequency	1.5	-
3P frequency	2.5	The league-average 3P frequency is subtracted from the player's frequency
AST %	4.0	-
TO %	1.5	-
REB %	4.0	-
BLK %	2.0	-
STL %	2.5	-
Defensive plus-minus	2.0	Calculated as a 50-50 split between BPM and RPM
Overall plus-minus	5.0	Calculated as a 50-50 split between BPM and RPM

Table 16 – CARMELO's variables weights

3. Make a projection. CARMELO uses all historical players with a positive similarity score to make its forecasts, usually hundreds of players, each with its contribution, according to the similarity score: a player with a similarity score of 50 will have twice as much influence on the forecast as one with a score of 25, for example. For rookies, the projection is based on college and rely heavily on a player's age and draft position. Projections tend to be more flexible. The unit measure used in these projections is the wins above replacement WAR, that reflects a combination of a player's projected playing time and his projected productivity while on the court. WAR is calculated as follows:

$$WAR = \frac{[plus\ minus] * [minutes\ played] * 2.18}{(48 * 82)}$$

The first version of CARMELO reflected a 50-50 blend of Box Plus/Minus and Real Plus-Minus (RPM). In the second version, CARMELO projections are now based on BPM only due to the lack of data in more distant years which poses a problem for a system that relies heavily on making historical comparisons. In addition to running player forecasts, FiveThirtyEight also released projections for win-loss totals for each franchise, based on a version of the Pythagorean expectation where:

$$Win\ ratio = \frac{[points\ for]^x}{[points\ for]^x + \ [points\ against]^x}$$

After some back testing, the conclusion was that a Pythagorean exponent of 11.5 would produce the most accurate team forecasts when dealing with RPM and BPM based projections. Team projections involve some human intervention, so injuries and other news are considered. Its performance was great in the initial experiment, edging out Vegas along with most other projection systems[4].

Also from FiveThirtyEight, CARM-Elo Ratings can be used to calculate win probabilities and point spreads for every NBA game and determine which teams have the best shot to make the playoffs or win the finals (538 Projections). In this rating system, home team has a standard bonus of 92 CARM-Elo points, and the margin of victory is considered when adjusting team ratings after each game. In addition to these standard adjustments, there are a few other factors such as:

[4] Results available at http://apbr.org/metrics/viewtopic.php?f=2&t=8633&start=255

- Fatigue: teams that played the previous day are given a penalty of about 46 CARM-Elo points (5-percentage point in win probability);
- Travel: teams are penalized based on the distance they travel from their previous game. For a long leg, the traveling team loses about 16 CARM-Elo points (2-percentage points in win probability);
- Altitude: In addition to the general home-court advantage, teams that play at higher altitudes are given an extra bonus when they play at home. Similar to the travel adjustment, this bonus is a linear function of the home-court altitude.

Once the adjustments are made, FiveThirtyEight simulates the regular season 10,000 times to find the average final record of each team and the percentage of simulations that each team makes the playoffs. They use NBA tiebreaking rules to seed teams in the playoffs and then simulate the playoffs 10,000 times to find the winner of the finals. Back tests found them to beat the spread about 51 percent of the time.

March Madness predictions

NCAAB matches might be a predictive challenge even bigger comparing to NBA. Despite all the differences in terms of money, facilities, and national exposure and lopsided results, many upsets happen during the season. Such as in basketball, these predictions have also started based on statistics. Several authors considered the use of Markov models, where the probabilities are evaluated having into account each round individually to predict the winner of the game and the calibration is made based on teams' seeds (Edwards 1998; Schwertman et al. 1991, 1996). Despite seeding may well measure the potential of the teams at the beginning of the championship, this structure of favoritism is unchanged during the course of the tournament, forcing a seed to have an equal relative strength to that same seeds in other regions, and can thus mislead models.

Carlin (1994) extend this approach by considering external information available at the 1994 NCAAB tournament's outset. Some rankings like Rating Percentage Index, Massey's and Sagarin's rating, typically linear functions of several variables (team record, home record, strength of conference, etc.) are updated during the season, providing more refined information about relative team strengths than seeds and enable differentiation between identically seeded teams in different regions.

For the first round of games, point spreads offered (predicted difference of points between the favorite and the underdog) by casinos and sports wagering were collected. In spite of potentially being so valuable by considering specific information as injuries, there is no possibility of having these values beyond the first round. For each first-round match, it was analyzed the differences between teams' seeds (*i* - *j*), differences between teams' Sagarin rating (S[*i*]-*S[j]*) and the expected point spreads (Y_{ij}) obtained prior to the beginning of tournament from a highly-regarded Las Vegas odds maker. These measures were compared with the actual margin of victory, *R*.

Carlin started to develop some regression based on that data. The first fitted regression line used seeds and achieved a good R^2 value of 88,3% and was defined as:

$$\hat{Y}_{ij} = 2.312 + .1\,(j - i)^2, where\ i < j.$$

The second obtain fitted model was:

$$\hat{Y}_{ij} = 1.165\ [S(i) - S(j)], where\ i < j$$

and had a R^2 of 98,1% which suggests that the Sagarin method is a better predictor of point spread than seeds.

The main goal of this study it was to compare these methods with Schwertman methods and assign a probability to each team to win the regional tournament. In order to calculate that probability, Carlin based in some professional football literature that showed that the favored team's actual margin of victory was reasonably approximated by a normal distribution with mean equal to the point spread and standard deviation of 13.86.

$$\mathrm{P}\ (\mathrm{R} > 0) \approx \Phi\left(\frac{Y}{\sigma}\right), where\ \Phi(\bullet)\ is\ the\ cumulative\ distribution\ function\ of\ the\ standard\ normal\ distribution$$

Due to the whole context, this value has been reduced to 10.

The table below compares the ability of five methods regarding all 60 games of 1994 NCAA's regional tournaments. The reference point for the scores is the assumption of every game as a toss-up, where a 50% chance would have a score of -0.693 according to logarithmic scoring rule.

	Scores				
Region	**Schwertman method**	**Seed reg.**	**Sagarin diffs.**	**Sagarin reg.**	**Sagarin reg. + R1 Spreads**
East	-0.116	-0.111	-0.106	-0.101	-0.102
Midwest	-0.134	-0.147	-0.134	-0.134	-0.127
East	-0.154	-0.148	-0.149	-0.152	-0.145
Southeast	-0.114	-0.103	-0.116	-0.114	-0.111
Total	**-0.517**	**-0.508**	**-0.505**	**-0.502**	**-0.485**

Table 17 – Carlin's scores compared with Schwertman method

In conclusion, all tested methods had a better performance than Schwertman's approach, particularly Sagarin Regression combined with Point Spreads for the first round. In this sense, it is possible to notice that Point Spreads are useful and good predictors.

Kaplan and Garstka (2001) focused on the study of office pools, namely types of pools and optimal prediction strategies. This topic was also studied by Niemi (2005) and Wright and Wiens (2016). The first compared the use of Return on Investment over strategies that maximize expected scores and he believed that these contrarian strategies provide high potential, particularly in years when the heaviest favorites do lose. The group of investigators analyzed 200 000 brackets from 2015 and 2016 and found that is vital to correctly pick the champion in order to win a large pool.

For NCAA Tournament prediction, Kaplan and Garstka used three Markov models that do not rely on seeding information. The Regular Season Model was based on regular season records and looks to the tournament as a regular season's extension. This simple model tried to maximize the log-likelihood function assuming the existence of a strength coefficient $s_i \geq 0$ and was used to develop a probability of a team *i* defeat a team *j*. The parameter n_{ij} stands for the number of wins of *i* over *j*:

$$\log L = \sum_{i,j \in NCAA} n_{ij} \log(p_{ij}) \,, where\ p_{ij} = \frac{s_i}{s_i + s_j}$$

The second one – Expert Rating Model, was based on the already known Sagarin ratings and assumes that the points scored by competing teams in the same game are uncorrelated random variables. In this method, λ_i denotes the Sagarin rating on team *I* and p_{ij} the probability of team *i* defeats team *j* in any game (or simply the probability of positive point spread):

$$p_{ij} = P\{X_{ij} > 0\} = \Phi\left(\frac{\lambda_i - \lambda_j}{\sqrt{\lambda_i + \lambda_j}}\right), where\ \Phi(\bullet)\ is\ the\ cumulative\ distribution\ function\ of\ the\ standard\ normal\ distribution$$

The last one used two Las Vegas popular bets (point spreads and point scored by both teams). The parameters of this method were λ_i represented the average scoring rates per game of team *i*, and x_{ij} and y_{ij} the point spreads and point totals posted before the tournament, respectively. From there, were assumed that

$x_{ij} = \lambda_i - \lambda_j$, $y_{ij} = \lambda_i + \lambda_j$ and the equations solved to yield $\lambda_i = \frac{x_{ij}+y_{ij}}{2}$, $\lambda_i = \frac{y_{ij}-x_{ij}}{2}$. In order to estimate the probability of a team to beat the other, researchers used the probability function applied in the second model.

These approaches were illustrated using the 1998 and 1999 NCAA and NIT men's basketball tournaments. The results from all 188 games were compared to predictions based on the tournament seedings.

	Identical picks for winners and Accuracy (both %)			
Methods	**Regular Season**	**Expert Rating**	**Las Vegas Odds**	**Actual Results**
Pick highest seeds	78	78	83	56
Regular Season		81	71	59
Expert Rating			75	57
Las Vegas Odds				59

Table 18 – Kaplan and Garstka's results

Overall, there is not a great increase in the accuracy of these models against the choice of the highest pick and there are cases where they cannot overcome. It is also possible to detect that Las Vegas odds agreed with picking the highest seeds on 156 out of 188 games and Sagarin models agreed with Regular Season Model on 152 out of 188 games. This may mean that Las Vegas rely heavily on seeds, while the expert Sagarin ratings and the regular season method are also closely connected.

Years later this became a hot topic. Zimmermann et al. (2013) identified the problem of data relativity which limits their expressiveness: for instance, collecting 30 rebounds could be a good be a nice stat in a 40-rebounds game but not so nice in a 60-rebounds game. In this sense, investigators normalized advanced statistics regarding pace, opponent's level, and national average, deriving adjusted (offensive and defensive) efficiencies:

$$AdjOE = \frac{OE * avg_{all\ teams}(OE)}{AdjDE_{opponent}};\ AdjDE = \frac{DE * avg_{all\ teams}(DE)}{AdjOE_{opponent}}$$

Several ML techniques were used: DT, Artificial NN (represented by a MLP), a Naïve Bayes and an Ensemble learner. For each experiment run, one season was used as test set and the preceding seasons from 2008 onward as training data, leading to the training and test set sizes shown:

Season	2009	2010	2011	2012	2013
Training	5265	10601	15990	21373	26772
Test	5336	5389	5383	5399	5464

Table 19 – Shi's training and test set sizes per season

In the first set of experiments, the investigators aimed to identify which attributes out of the full set were most useful in predicting match outcomes. Using a Weka's feature selection methods to examine the attribute set down, the results were location first, followed by adjusted efficiencies and the Four Factors.

The Four Factors of Basketball Success theory, introduced by Dean Oliver (2004), one of the most relevant researchers in basketball world, identify four offensive and defensive statistics (and their weights) as being of particular meaning for a team's success: shooting (40%), measured by the eFG%; turnovers (25%), measured by the TO%; rebounding (20%), measured by ORB% and DRB%; and, finally, free throws (15%), measured by the FT factor.

	Accuracy (%)				
Technique	2009	2010	2011	2012	2013
J48	68.39	68.39	69.05	70.42	68.98
Random Forest	68.85	69.42	67.79	71.37	68.81
Naïve Bayes	71.01	71.72	70.28	72.76	71.93
MLP	70.77	72.51	71.6	74.46	72.15

Table 20 – Accuracy using adjusted efficiencies

	Accuracy (%)				
Technique	2009	2010	2011	2012	2013
J48	66.47	66.45	66.22	67.88	65.08
Random Forest	68.01	69.31	69.83	70.2	68.92
Naïve Bayes	71.21	72.02	72.06	73.05	70.81
MLP	70.11	71.65	71.21	73.11	70.92

Table 21 – Accuracy using adjusted four factors

The main conclusions were that MLP and Naïve Bayes gave consistently best results and more training data does not translate into better models. Although it has not been possible to overcome the state-of-the-art, some lessons were learned such as that picking a more complex technique does not guarantee satisfactory results (the simplest classifiers, like Naïve Bayes or Ken Pomeroy's straight-forward Pythagorean Expectation, could perform better than Brown et al.'s LRMC model (2012) but the essence of having good models relies on the choice and quality of variables. Besides this, the researchers thought they had discovered a ceiling for accuracy in NCAA games around 75%, like those for football (77%), American football (79%), NCAA football (76%) and NBA (74%). This unpredictability may be due to intangibles attributes such as experience, leadership or luck and to the non-separation of games by conferences.

Motivated by Kaggle's "March Machine Learning Mania" competition, there are hundreds of participants with different models every year. The next three reviews will be on the winners of the contests in 2014, 2015 and 2016.

The first place in 2014 was for Gregory Matthews and Michael Lopez. Their submission was the combination of two models, a margin-of-victory (MOV) based model and an efficiency model using Ken Pomeroy's data (KP). For first round games, the MOV model used the spread posted in Las Vegas for each game; for the following rounds, they used previous game results to predict a margin of victory. For the KP model, different regression models using different team-wide efficiency metrics were tested and choose the one that minimized the loss function in the training set. At the end, the outcomes were converted into a probability using logistic regression and the final submission used a weighted average of those probabilities.

For most important insights of this participation were the “absolutely incredible” predictive power of Las Vegas line, the good performance of simple models, the importance of having the right data and “a decent amount of luck”. The recommendation made was to train models along with regular season data due to a short sample of NCAA Tournament games.

The winner of 2015 competition, Zach Bradshaw was a sports analytics specialist at ESPN and a former analyst in two NBA teams and made use of his previous experience, particularly in data pre-processing and knowing the techniques to apply. Using a Bayesian framework allowed for the incorporation of prior knowledge or intuition that was not accounted for in the data. However, with the winning entry, Zach manually tweaked a game and successfully predicted the upset. His experience in sports analytics taught him that there are no perfect models and also it takes luck to succeed.

In 2016, Miguel Alomar used logarithmic regression and RF and even tried ADA Boost but did not get very results. For this winner, the key factors were offensive and defensive efficiency, the weight of strength of schedule and penalize teams with easier games throughout the season. After testing these issues, he ended up with two models: one more conservative and another that surprised him by discovering most of the upsets, which eventually won the competition.

The score of each submission follow a log loss function:

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where n is the number of games played, $\hat{y}_i$ is the predicted probability of A beating B, y_i is 1 if A wins, 0 if B wins and log is the natural base e logarithm. The results achieved by the winners are:

Year	Log Loss Score
2014	.52951
2015	.438933
2016	.48131
2017	.438576

Table 22 – Best scores of Kaggle’s March Madness contests

In addition to the work already reviewed, FiveThirtyEight has also developed some effort on this topic. For the 2017 March Madness, FiveThirtyEight.com (Boice & Silver) had permanently a dashboard containing game scores and the probabilities of each team to win that game given by logistic regression analysis. Specifically, play-by-play data from the past five seasons of Division I NCAA basketball is used to fit a model that incorporates:

- Time remaining in the game;
- Score difference;
- Pre-game win probabilities;
- Which team has possession, with a special adjustment if the team is shooting free throws.

These in-game win probabilities won't account, for instance, a key player fouling out but are reasonably good showing which games are competitive and which are not.

The Excitement Index is a measure of how much each team's chances of winning changed during a game and is a good reference for expecting an upset or an exciting game. It is calculated using the average change in win probability per basket scored, weighted by the amount of time remaining in the game (a late-game basket has more influence on a game's rating than a basket near the beginning of the game). Normally, ratings range from 0 to 10.

Like NBA's Elo rating, it relies on the final score, home-court advantage and the location of each game. They also account for a team's conference and whether the game was an NCAA Tournament game. Elo is one of six computer rankings used for predictions. The other five are ESPN's BPI[5], Sagarin's ratings, Pomeroy's ratings[6], Sokol's LRMC ratings[7] and Moore's computer power ratings[8]. In addition, the selection committee's 68-team "S-Curve" and preseason ratings from coaches and media polls compose the eight systems that are weighted equally in coming up with a team's overall rating and tournament predictions. Like in NBA, ratings are adjusted for travel distance and player injuries.

[5] Available at http://www.espn.com/mens-college-basketball/bpi
[6] Available at http://kenpom.com
[7] Available at http://www2.isye.gatech.edu/~jsokol/lrmc/
[8] Available at http://sonnymoorepowerratings.com/m-basket.htm

3. DATA COLLECTION AND DATA MANAGEMENT

3.1 Data sources

The main source of data for this project was the website Sports-Reference.com. It was launched in 2000 and has data from several sports including baseball, basketball, football or hockey both professional and university level. As for college basketball, it is possible to find a wide amount of records, including some residual statistics from the last decade of the 19th century. Using Sports-Reference.com was possible to collect data from teams (both historical and current year) and coaches' stats. Below, there are examples of the collected data that was served as the basis for the data set.

	Team and Opponent Stats																	
	G	**FG**	**FGA**	**2P**	**2PA**	**3P**	**3PA**	**FT**	**FTA**	**ORB**	**DRB**	**TRB**	**AST**	**STL**	**BLK**	**TOV**	**PF**	**PTS**
Team	34	912	1856	730	1340	182	516	608	916	358	928	1286	456	227	142	444	614	2614
Opponent	34	818	2006	602	1351	216	655	439	677	377	757	1134	416	224	98	442	724	2291

Table 23 – 2009/10 Gonzaga Bulldogs stats

		Coach Record						
Season	**School (seasons)**	**G**	**W**	**L**	**NCAA Tournament**	**Final Four**	**Champion**	**W - L**
2007-08	Butler	34	30	4	x			1-1
2008-09	Butler	32	26	6	x			0-1
2009-10	Butler	38	33	5	x	x		5-1
2010-11	Butler	38	28	10	x	x		5-1
2011-12	Butler	37	22	15				
2012-13	Butler	36	27	9	x			1-1
Career	**Butler (6)**	**215**	**166**	**49**	**5**	**2**	**0**	**12-5**

Table 24 – 2012/13 Brad Stevens record

	School History					
Season	**W**	**L**	**NCAA Tournament**	**Final Four**	**Champion**	**W - L**
2016-17	26	8	x			0-1
2015-16	21	14	x			1-1
2014-15	22	11				
2013-14	22	13				
2012-13	26	11	x			2-1
2011-12	15	17				
2010-11	10	20				

Table 25 – 2016/17 Florida Gulf Coast Eagles history

Unfortunately, during the study period (1999-2017), such comprehensive information was not always available. Only after 2009 all data above was available, which turned out to be the biggest limitation of the project.

The NBA's website was also used for search from alternative basketball metrics.

3.2 Data collection

The collected data was saved is an excel file. The observations include several variables that go from the 1999 to the 2017 season. The constitution of the dataset is the following:

Record ID	Team A	Team B	Team A stats	Team B stats	Ratio between teams' stats	Output

The ID part is composed by the year, round and a ID number (for instance, **2013SR1** refers to the 1st second round game of the 2013 tournament). The dataset is sorted according to the bracket. The games in the upper left corner correspond to the first observations and the games in the lower right corner correspond to the last observations. The team above in the bracket corresponds to team A in each observation.

For each team, there were collected:

- Seed
- # W
- # L
- PPG
- OPPG
- FGM
- FGA
- 3PM
- 3PA
- FTM
- FTA
- RPG
- APG
- SPG
- BPG

And get, for predefined formulas (could be seen in the Appendix A):

- W %
- FG %
- 2PM
- 2PA
- 2P%
- 3P%
- 3PAr
- FT %
- FTf
- eFG %
- TS %

Besides all basic and advanced team stats, it was decided to also include coaches' features and teams' historical NCAA data. For coaches, the features included are:

- # Seasons
- # Games (W & L)
- W %
- # NCAA Games (W & L)
- # NCAA Tournaments
- # Final Fours
- # Championships
- Same from previous year

The features for teams are:

- Previous year # W
- Previous year # L
- Previous year W %
- Previous year NCAA
- # NCAA
- # Final Four
- # Championships
- Team from First Four or Opening Round

3.3 Data transformation

A major problem of collecting data from Sports-Reference.com is the data contamination. When selecting archival data to feed into predictive models of sport events, it is critical to ensure that "the past" doesn't contain its own future. As a result, a seemingly successful prediction will, in fact, rely heavily upon anachronous metrics – rendering such models inept for true future prediction.

Yuan et al. (2015) refer to data contamination as being input features that already know the result of what they are aiming to predict and will give much weight to these variables that contain the future: number of games and number of wins are clear examples of this, because a team is not supposed to have much more games than the others during the season, unless it reaches the final stages of the NCAA tournament.

In this sense, to ensure the quality of data, some transformation had to be made, like in the examples below.

	Team Stats												
	G	FG	FGA	3P	3PA	FT	FTA	TRB	AST	STL	BLK	PTS	OPTS
Season	37 (29-8)	960	2012	200	555	531	759	1302	566	265	157	2651	2195
Vs. LIU	-1 (-1 W)	-40	-68	-3	-9	-6	-9	-42	-21	-6	0	-89	-67
Vs. St. Louis	-1 (-1 W)	-25	-46	-4	-10	-11	-17	-29	-12	-2	-1	-65	-61
Vs. Louisville	-1 (-1 L)	-14	-49	-5	-21	-11	-12	-32	-12	-8	-7	-44	-57
Regular Season	34 (27-7)	881	1849	188	515	503	721	1199	521	249	149	2453	2010

Table 26 – 2011/12 Michigan State Spartans stats

		Coach Stats								
Season	**School (seasons)**	**G**	**W**	**L**	**NCAA Tournaments**	**NCAA W**	**NCAA L**	**Final Four**	**Championships**	**Coach previous year**
2011-12	Michigan State	37 (-3)	29 (-2)	9 (-1)						
Career	**Michigan State (17)**	**578**	**412 (-2)**	**169 (-1)**	14 (+1)	37 (-2)	14 (-1)	6	1	X

Table 27 – 2011/12 Tom Izzo pre-NCAA Tournament Stats

Team Stats					
Previous year W	**Previous year L**	**NCAA Tournaments**	**Final Four**	**Championships**	**NCAA previous year**
19	15	25 (+1)	8	2	X

Table 28 – 2011/12 Michigan State Spartan stats

From 2017 on, in future tournament predictions, this issue will no longer be a problem because, by collecting data right after the selection Sunday, it makes the data collection process simpler, faster, without compromising data quality.

The last step regarding data treatment was to duplicate the dataset, making it symmetric. Besides increasing the number of records, this was a great solution to prevent any impact from teams' position in the output. This was due to the fact that the first team of each observation had a higher chance of winning once. In approximately 65% of the observations (798 of 1235) this happened, making the dataset unbalanced.

To the original dataset were added the same observations where, this time, the teams were changed, and the output changed to their respective (0 to 1 and 1 to 0):

Record ID	Team B	Team A	Team B stats	Team A stats	Ratio between teams' stats	Output

4. EXPERIMENT DESIGN

Within the experimentation there is a need to perform some fundamental steps, such as: pre-processing the data, realizing which variables are most important for the problem, defining the algorithms to be used and tuning their parameters, and select metrics to evaluate the results.

To implement all these steps there was a need to resort to Anaconda. The world's most popular Python data science platform[9], was used to support the experiment part of the project. This product of Continuum Analytics, is a virtually complete scientific stack for Python that includes the standard libraries, like Scikit, NumPy or Pandas.

For the experimental part, the ratios among team statistics were used. As analyzed in the bibliographic review, in this type of problems the most import challenge is to understand which team has a relative advantage over the other.

4.1 Data standardization

Standardization of datasets is a common requirement for many ML estimators. For example, in Artificial NN and other Data Mining approaches there is the need of normalizing the inputs, otherwise the network will be ill-conditioned. In essence, normalization is performed to have the same range of values for each of the inputs to the ANN model. This can guarantee stable convergence of weight and biases.

Data standardization was done using the Python's preprocessing function StandardScaler. It was possible to standardize each input variable with center equals to 0 and standard deviation equals to 1.

4.2 Machine Learning techniques

Machine Learning is a buzzword in the technology world right now. ML has several techniques that can be divided in Unsupervised and Supervised Learning. In the first one, algorithms operate on unlabeled examples where the target output associated with each input, is not known by the system and it tries to find a hidden structure. The goal of this type of learning is to explore the data to find intrinsic structures within it using methods like clustering or dimensional reduction. Supervised learning algorithms are trained using labelled examples where the desired output is known. Supervised learning is commonly used in applications that use historical data to predict likely future events, as in this project. Below, there is a review of the used supervised learning algorithms. The definitions below were based on the SciKit-Learn website[10].

Decision Trees

This "divide and conquer" technique creates tree structures where leaves stand for labels and branches for combinations of features. At each step, the algorithm chooses the best variable to split the dataset with respect to the values of the target according its discriminative power. The goal is to have a model that predicts the value of a

[9] https://www.anaconda.com/what-is-anaconda
[10] Available at http://scikit-learn.org

target variable by learning simple decision rules inferred from the data. As advantages, DT requires little data preparation, can handle both numerical and categorical data and it is a very simple model to understand. As disadvantages, DT learners can create over-complex trees that do not generalize the data well (creating overfitting) and can be unstable because small variations in the data might result in completely different trees.

Logistic Regression

Despite its name, this is a linear model for classification rather than regression. It shares some similarities with linear regression, but uses a sigmoid function instead of a linear one. Logistic regression is also known in the literature as logit regression, maximum-entropy classification or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

Multi-Layer Perceptron

MLP belongs to the NN type of algorithms. NN are non-linear statistical data modelling tools, able to model complex relationships between inputs and outputs, or to find interesting patterns. These techniques consist of three main components: the structure of the network, the training method, and the activation function. The main advantage of this method is that it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. As disadvantages, MLP requires hyper parameterization and has more than one solution, depending on the initial weights.

Nearest Neighbors

Nearest Neighbors classification is a type of instance-based learning that stores instances of the training data, known as examples. In the K-Nearest Neighbors model, classification is computed from a simple majority vote of the K-nearest neighbors of each point. The optimal choice of the value K is highly data-dependent: a smaller K could lead to noisy decision boundaries, while a larger K will lead to over-smoothed ones.

Random Forest

RF is a meta estimator that fits different DT and average outputs to improve the predictive accuracy. Besides accounting for particularly complex decision boundaries, it is a fast-to-train method that minimizes the generalization error, proven not to overfit, and computationally effective. These merits make RF a potential tool suited for classification problems (Osman, 2009).

Stochastic Gradient Descent

SGD is a simple yet very efficient approach to fit linear models often applied in problem like text classification and Natural Language Processing. It is particularly useful when the number of samples (and the number of features) is very large. The advantages of SGD are the efficiency and the ease of implementation. Like other algorithms, it requires hyper parameterization and it is sensitive to feature scaling.

Support Vector Machine

SVM is a set of supervised learning methods used for classification, regression and outliers' detection. SVM attempts to find a hyperplane that separate different outputs based on the feature vectors. The major advantages are the effectiveness in high dimensional spaces and the versatility (different Kernel functions can be specified for the decision function). As disadvantage, SVM is likely to give poor performances if the number of features is much greater than the number of samples.

4.3 Model evaluation

Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Despite its simplicity, it can be misleading when in imbalanced datasets. In imbalanced datasets, it may be desirable to consider selecting a model with a lower accuracy because it might have a greater predictive power on the problem. On the other hand, when evaluated in symmetric datasets where values of false positive and false negatives are identical, like in this case, could be a useful measure.

F-Measure

F-Measure (also known as F1-Score or F-Score) is the weighted average of Precision and Recall. Intuitively it is not as easy to understand as Accuracy, but it is usually more useful than accuracy, especially in uneven class distribution datasets.

$$\mathrm{F-Measure} = 2 * \frac{\mathrm{Recall} * \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}$$

Precision (also known as Positive Predictive value) is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Recall (also known as Sensitivity or True Positive rate) is the ratio of correctly predicted positive observations to the all observations in actual class - yes. High recall relates to the low false negative rate.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Holdout method

Hold-out cross-validation is a widely-used CV technique popular for its efficiency and easiness. When evaluating a model, it is important to do it on held-out observations that were not seen during the grid search process. In this method, the data is split into two mutually exclusive subsets: a training set (that could be trained using a Grid Search CV) and an unseen test set to compute performance metrics. The major problem of this technique is that the chosen split heavily affects the quality of the final model. If the dataset is split poorly, the data subsets will not sufficiently cover the data and especially the variance will increase (Reitermanová, 2010). The application of this method in the experiment was possible using the SciKit's Train_Test_Split function.

Cross Validation

CV is a model evaluation method. In the approach used in this project, called k-fold CV, data is split into k subsets of matching size. At each iteration, one of the k subsets is used as the test set, while the other k-1 subsets form a training set. The performance measure reported by k-fold CV is the computed average error across all k-trials in the loop. Compared to the simplest holdout method, this approach is much more efficient in terms of lowering the variance value of the resulting estimate when k increases. While the main advantage is that all observations are used for both training and testing, the disadvantage of this method is that the training algorithm must be rerun from k times, which means more computational effort to make an evaluation.

4.4 Feature selection

When examining a dataset with a large number of variable it is a good practice to reduce the dimensionality of the dataset without sacrificing useful information. The curse of dimensionality describes the problem caused by the exponential increase in volume associated with adding extra dimensions to Euclidean space (Bellman, 1957). Feature selection methods can be useful regarding this kind of problem, automatically selecting variable that contribute most to the output and eliminating those that are redundant. The main benefits of performing a feature selection are: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data (Guyon & Elisseeff, 2003).

This task of was completed using Recursive Feature Elimination, available at SciKit's library. This method is used for ranking feature with recursive elimination: iteratively, it ranks all variables and the less important for the model is not included in the train dataset. This procedure was tested using five different ML techniques (NN, Logistic Regression, SVC, SGD and DT) and the metric used to evaluate the impact of each reduction was the f-measure.

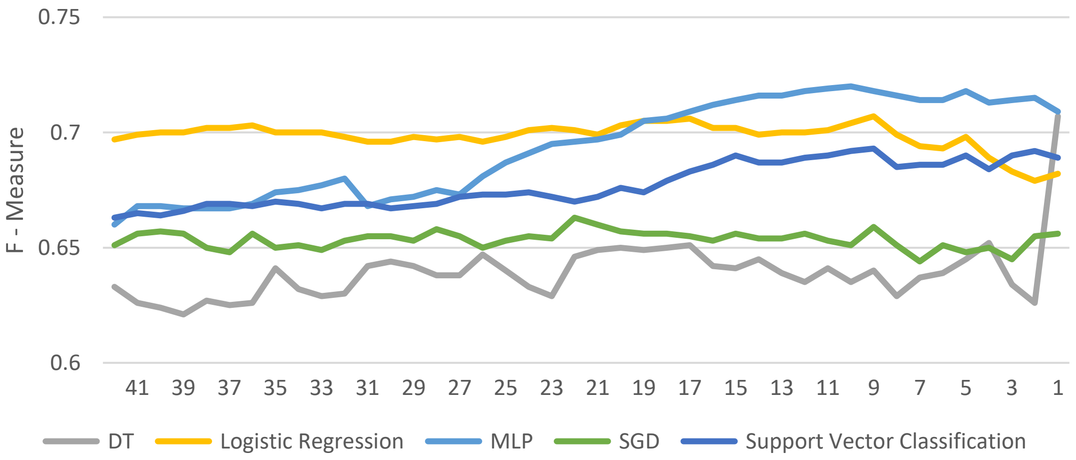


Figure 4 – Evolution of Recursive Feature Selection

According to the results, it is not evident that the elimination of variables improves the performance of the methods. Analyzing in more detail the iterative cycle of the elimination of variables it is possible to draw the following conclusions.

- Coach stats have not proved to be very important once the algorithm removed them in the first third of the process.
- The same thing happened with team statistics. The only variable that proved to be relevant was the ratio of the number of participations in the championship.
- The variables considered most important refer mainly to points ratio, scoring ratios and teams' records.

4.5 Grid Search

Grid Search is an exhaustive examination over parameter values for an algorithm through a manually defined subset of candidates. For each of the combinations, the models are trained and evaluated using a CV. The main purpose is to find the best possible combination of parameters. For being exhaustive, this choice is very time-consuming, and it is not assured that the solution is the best global one. In this sense, many researchers prefer an alternative method called Random Search (Bergstra & Bengio, 2012; El Deeb, 2015). Below it possible to find all the information about the process of parameter tuning for each technique.

Parameters	Definition	Values	Grid Search
criterion	The function to measure the quality of a split. Supported criteria are Gini impurity and information gain.	[‘entropy’, ‘gini’]	‘entropy’

max_features	The number of features to consider when looking for the best split:
	Sqrt: max_features = sqrt(n_features);
	Log2: max_features = log2(n_features);
	None: max_features = n_features.
		[‘log2’, ‘none’, ‘sqrt’]	‘none'

min_samples_leaf	The minimum number of samples required to be at a leaf node.	list (range (1,200))	194
splitter	The strategy used to choose the split at each node. Supported strategies are best split and best random split.	[‘best’, ‘random’]	‘best’

Table 29 – Decision Tree Classifier’s Grid Search parameters

Parameters	Definition	Values	Grid Search
c	Inverse of regularization strength	list (range (1,300))	10
penalty	Used to specify the norm used in the penalization.	[‘l1’, ‘l2’]	‘l1’
solver	Algorithm to use in the optimization problem. For small datasets, ‘liblinear’ is a good choice, whereas ‘sag’ is faster for large ones. For multiclass problems, only ‘newton-cg’, ‘sag’ and ‘lbfgs’ handle multinomial loss. The ‘newton-cg’, ‘sag’ and ‘lbfgs’ solvers support only L2 penalties.	[‘liblinear’, 'newton-cg', 'lbfgs', 'sag']	‘liblinear’

Table 30 – Logistic Regression’s Grid Search parameters

Parameters	Definition	Values	Grid Search
n_neighbors	Number of neighbors to use	list (range (1, 200))	92
algorithm	Algorithm for the choice of neighbors' search. Brute: The naivest implementation uses brute-force computation of distances between all pairs of points in the dataset. KD tree: This is a more efficient approach and its constructions is very fast. Ball tree: This method can surpass inefficiencies of KD Tree in higher dimensions.	['ball_tree', 'brute', 'kd_tree]	'ball tree'
leaf_size	Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree.	[10, 20, 30, 40, 50]	10
weights	Uniform: All points is each neighborhood are weighted equally; Distance: Closer neighbors will have a greater influence than neighbors which are further away.	['uniform', 'distance']	'distance'

Table 31 – K-Nearest Neighbors' Grid Search parameters

Parameters	Definition	Values	Grid Search
activation	Activation function for the hidden layer	['identity', 'logistic', 'tanh', 'relu']	'tanh'
solver	The solver for weight optimization	['lbfgs', 'sgd', 'adam]	'lbfgs'
alpha	L2 penalty (regularization term) parameter	[0.1, 0.01, 0.001, 0.0001]	0.1
learning_rate	Learning rate schedule for weight updates. Constant: constant learning rate given. Invscaling: gradually decreases the learning rate. Adaptive: keeps the learning rate constant as long as training loss keep decreasing.	['constant', 'invscaling', 'adaptive']	'constant'
momentum	Momentum for gradient descent update. Only for 'sgd'	[0, 0.25, 0.5, 0.75, 1}	n/a
power_t	The exponent for inverse scaling learning rate. Used in updating effective learning rate when learning_rate = 'invscaling'. Only for 'sgd'.	[0.1, 0.25, 0.5, 0.75, 1]	n/a

Table 32 – Multi-Layer Perceptron Classifier's Grid Search parameters

Parameters	Definition	Values	Grid Search
n_estimators	Number of trees built.	[100,200, 300, 400, 500, 700, 1000]	700

max_features	The number of features to consider when looking for the best split:
	Sqrt: max_features = sqrt(n_features);
	Log2: max_features = log2(n_features);
	None: max_features = n_features.
		['log2', 'sqrt', None]	'log2'

min_samples_leaf	The minimum number of samples required to be at a leaf node.	list (range (1,200))	100
criterion	Uniform: All points is each neighborhood are weighted equally Distance: Closer neighbors will have a greater influence than neighbors which are further away.	['gini', 'entropy]	'entropy'

Table 33 – Random Forest Classifier's Grid Search parameters

Parameters	Definition	Values	Grid Search
c	Penalty parameter C of the error term.	[10, 20, 30, 40, 50]	30
kernel	Specifies the kernel type to be used in the algorithm.	[‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’]	Poly
degree	Degree of polynomial kernel function.	[1, 2, 3, 4]	3
gamma	Kernel coefficient for RBF, Poly and Sigmoid.	[0.1, 0.01, 0.001, 0.0001]	0.01

Table 34 – C-Support Vector Classification’s Grid Search parameters

Parameters	Definition	Values	Grid Search
alpha	L2 penalty parameter	[0.1, 0.01, 0.001, 0.0001]	0.001
eta0	The initial learning rate for the ‘constant’ or ‘invscaling’ schedules.	[0, 0.1, 0.25, 0.5, 0.75, 1]	0.75
learning_rate	Learning rate for weight updates.	['constant', 'optimal', 'invscaling']	'invscaling'
loss	The loss function to be used.	['hinge', 'log', 'modified_huber', 'squared_hinge', 'perceptron']	‘modified_huber’
penalty	The regularization term to be used. ‘L2’ is the standard regularizer for linear SVM models. ‘L1’ and ‘elasticnet’ might bring sparsity to the model not achievable with ‘L2’.	['l2', 'l1', 'elasticnet']	‘l1’
power_t	The exponent for inverse scaling learning rate. Used in updating effective learning rate when learning_rate = ‘invscaling’. Only for 'sgd'.	[0.1, 0.25, 0.5, 0.75, 1]	0.75

Table 35 – Stochastic Gradient Descent Classifier’s Grid Search parameters

5. RESULTS AND EVALUATION

After a well-structured experiment, the next step consists of evaluate the results. As seen in the experiment design chapter, the models were trained using a 10-fold CV, while the sets where divided using a holdout method. In this chapter, the predictive power of the algorithms will be assessed and compared with more basic approach and with some reference sites.

In the first experiment, the objective was to get an overall overview of the different algorithms' performance. Here, the allocation of the observations was randomly made by the Scikit's function train_test_split. The data was partitioned into 75% of the observation for training and the remaining 25% for test. Below, it is possible to see the results for the first experiment.

	Accuracy (%) & F-Measure	
Technique	**Training Set**	**Test Set**
Decision Tree Classifier	67.9 (.679)	65.9 (.658)
Logistic Regression	69.1 (.691)	68.9 (.689)
K-Nearest Neighborhood Classifier	68.2 (.680)	68.6 (.678)
Multi-Layer Perceptron Classifier	72.2 (.722)	66.7 (.667)
Random Forest Classifier	69.6 (.695)	69.2 (.691)
Stochastic Gradient Descent Classifier	69.6 (.695)	68.7 (.686)
Support Vector Classification	70.8 (.708)	70.2 (.701)

Table 36 – Results from the overall experiment

As the result shown, SVM is the only method that surpasses the 70% accuracy boundary, while the rest have a score between 66% and 69%. In general, the results from the training and test sets were similar.

In the second approach, all data from 1999 to 2016 was used for training the models. The goal of this experiment was to check how good historical data would be in terms of predicting the NCAA Tournament of 2017. Below, it is possible to see the results for each of the techniques.

	Accuracy (%) & F-Measure	
Technique	**1999-2016**	**2017**
Decision Tree Classifier	67.2 (.673)	50.0 (.333)
Logistic Regression	70.0 (.700)	73.1 (.731)
K-Nearest Neighborhood Classifier	67.1 (.671)	64.9 (.637)
Multi-Layer Perceptron Classifier	78.7 (.787)	78.8 (.788)
Random Forest Classifier	68.2 (.682)	50.0 (.487)
Stochastic Gradient Descent Classifier	69.0 (.690)	69.1 (.691)
Support Vector Classification	71.5 (.715)	85.1 (.851)

Table 37 – Results from the experiment for predicting 2017 Tournament

These results were more encouraging than those from the first experiment. In general, this tryout has led to a greater breadth of model performance. On one hand, the performance of the MLP Classifier and SVM Classifier were the ones that stand out the most (with 79% and 85%, respectively). On the other hand, DT and RF have proved to be the method with worst predictive power, with only 50% of accuracy. In this experimentation, a greater variation in the outcomes of the training and test set was noticed.

Another well-known technique to pick winning teams is considering the seed, selecting the best one. Despite all the limitations already seen in the literature review, many people use this approach when filling out the March Madness brackets. Below, it is possible to see the results of picking the highest seed in each game. This analysis only considers game where teams had different seeds (1181 out of 1235).

	Accuracy (% of games)	
Technique	**From 1999 to 2017**	**Only 2017**
Pick the highest seed	71% (95,6%)	77,1% (91%)

Table 38 – Pick the highest seed method results

Considering the games where the teams' seeds are equal as a toss coin the results would decrease to 70.1% for the games from 1999 to 2017 and 74.7% for the games from the 2017 NCAA Tournament. This method is capable of making bold predictions, being almost as good as the ML algorithms used in the 2nd experiment.

Another suitable manner to evaluate the outcome’s quality is to compare the accuracy of predictions with some reputed websites and statisticians. Below there are the accuracies of several predictors considering the 2017 Tournament.

Predictor	**Accuracy (%)**
Massey	74.6%
Pomeroy	74.6%
ESPN BPI	73.1%
538.com	68.7%

Table 39 – Results for 2017 Tournament for comparable predictors

With respect to the tournament of 2017, the algorithms tested in the 2nd experiment showed to have a better predictive power than those of the specialists.

Unfortunately, it was not possible to get the accuracy outcomes from competitors of Kaggle’s contest. It would be very interesting to check if the scores of these models are close to the best competitors.

6. CONCLUSIONS

This project explores the application of ML techniques in the field of college basketball. With the increasing collection of data about the topic, associated with the appeal that is predicting the outcomes of sporting events, this is a perfect project for data enthusiasts. For many, March Madness is considered the greatest sports event in the United States of America, able to move billions of dollars in betting and put millions of people trying to hit the winners of the 68 games in the tournament. The madness comes with all the hype fans feel in every basket, every buzzer-beater, the win-or-go-home feeling and the unpredictable upsets that spoil any bet.

In the literature review is possible to read an overview about ML: how it was born, the importance it has nowadays, the various applications and everything that can be developed in the near future. Still within this chapter, it was possible to verify that, both individuals and organizations, are betting more and more on decisions that are based on data. The most important part, about basketball predictions, allowed to understand that there are a long history and different points of view on which this topic can be approached. Besides that, from the various sources it was possible to gain significant insight to develop the project.

The source of the data used in this project was the website Sports-Reference.com, that collects college basketball stats for a long time. For this project, data from 1999 to 2017 were used. The dataset contains data on the teams' regular season, coaches and teams' stats. After the quality of the dataset was guaranteed, it was possible to proceed to the experimental part.

The experiment was made using the Python language's Scikit library, and began with ensuring one of the main assumptions of the algorithms - the standardization of the data. Also in this section, an overview of the ML techniques to be implemented and the ways in which the results would be evaluated was made. The remaining subjects covered were the feature selection (where it was possible to perceive the importance and insignificance of some variables) and the hyper-parameterization (find the best parameters for each algorithm). An interesting remark about this study is the historical weight of the teams may have, that should be something to consider. Teams like UNC, UCL or Kentucky are more likely to have better players and reach later stages of the competition. On the contrary, for rookie teams there is no such expectation.

In what concerns the results, despite all the limitations (that could be seen in the next chapter), the outcomes achieved can be considered as quite satisfactory. Though the problem of predicting March Madness tournaments seems to be too random for ML to perform extremely well, these models can definitely provide insights into how a tournament will progress. More specifically in the first experiment, where observations were randomly grouped in training set and test, all models achieved an accuracy between 66% and 70%. In the second experiment, whose goal was to make a forecast for the 2017 tournament, training data from previous years, the results were broader: the worst outcomes were obtained from DT and RF classifiers (with an accuracy of 50%), and the best were from SVM (with an accuracy of 85%), MLP Classifier (with an accuracy of 79%) and Logistic Regression (with an accuracy of 70%).

In this sense, the majority of the methods had a better accuracy than the most rudimentary benchmark – flipping a coin, associated with an accuracy of 50%. The best have succeeded in overcoming what is probably the best predictor, the seed of a team, which consists of the knowledge of the experts. When comparing these results with some of the most reputed websites in the forecast area, which usually have an accuracy in the 70 / 75 %, it can be settled that it is possible to achieve and surpass them. The main conclusion drawn from this whole project is that the greatest challenge is to realize what will be the upsets that no one is waiting for them to happen.

7. LIMITATIONS AND FUTURE WORK

Like in any other sports, the outcomes of games usually have a close relationship with the players' ability but, besides all tactics and statistics, there is also a lot of uncertain factors like injuries, unexpected errors from players and referees, the distance traveled, rest and, of course, the luck and the physics of a shot to go in or go out. These immeasurable elements are hard to quantify in a way that cannot be put into a mathematical model, so that no prediction can be fully accurate.

Regarding this project, there is a major issue about unavailable data in the most distant years, which has more focus on the data collection chapter. This limitation makes the models to not consider much of the defensive aspects of a team which, for many basketball experts, is fundamental: most of NBA and NCAA Tournament champions were, at least, fair defensive teams (Williams, 2016; Boozell, 2017). Allied to this, it is proven that good defensive tends to lead to easy offense opportunities and it is easier to practice. Once this limitation was exceeded, it would be expected an increase in the predictive power of the methods.

A notable feature that could improve the efficiency of the data collection would be to create some programming languages scripts, using Ruby, PHP or some Python's specific libraries, following what Cao (2012) did. Ahead of this, it would be interesting to create a software that could automatize all the process and display the evolution of the tournament. Another additional aspect, which could be work if there was no limitation of the data, would be to compare the results of this project with the famed Dan Oliver's theory of the Four Factors.

A challenging approach to March Madness would be to view this problem from the underdog's side and understand what are the main characteristics of the upsets. A project like this would be very relevant because it is quite easy to achieve great accuracies using just the seed to predict the winner of a game. An alternative way to address this challenge could be to consider a different granularity level and verify the impact of some characteristics of the player in the outcomes: age or seniority, physical aspect like weight, height or wingspan, the player efficiency rating and matchups between players are some examples of useful data. It would be also interesting to analyze if there is any kind of influence regarding the history of a team, like the number of participations in the NCAA Tournament which, according to the feature selection made, could be a good predictor.

For any person interested in this topic, it is recommended to follow Kaggle's contests, both from March Madness and other competitions that often appear. There also are some websites that allow anyone to create his own bracket, like Yahoo, ESPN or CBS Sports.

In the future, it is intended that the data be kept up to date so there will be a better basis to train the models and hopefully improve their performance. All project, as well as the dataset and the scripts, can be used for future references in the area of predictive analysis in sport and reused for further research.

8. BIBLIOGRAPHY

Beckler, M., Wang, H., & Papamichael, M. (2013). NBA Oracle. *Zuletzt Besucht Am*.

Bellman, R. E. (1957). Dynamic programming. Princeton, NJ: Princeton University Press.

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research, 13*, 281-305.

Bhandari, I., Colet, E., Parker, J., Pines, Z., Pratap, R., & Ramanujam, K. (1997). Advanced Scout: Data Mining and Knowledge Discovery in NBA Data. *Data Mining and Knowledge Discovery*, *1*, 121–125.

Bocskocsky, A., Ezekowitz, J., & Stein, C. (2014). The Hot Hand: A New Approach to an Old ``Fallacy''. In 8th Annual MIT Sloan Sports Analytics Conference, 1–10.

Boice, J., & Silver, N. (2017, March 13). How FiveThirtyEight Is Forecasting The 2017 NCAA Tournament. *FiveThirtyEight*. Retrieved from https://www.fivethirtyeight.com

Boozell, J., (2017, February 16). College basketball: How good does your defense have to be to win a national title?. *NCAA*. Retrieved from http://www.ncaa.com

Bradshaw, Z., (2015, April 17). Predicting March Madness: 1st Place Winner, Zach Bradshaw. *No Free Hunch*. Retrieved from http://blog.kaggle.com

Brown, M., Kvam, P., Nemhauser, G., & Sokol, J. (2012). Insights from the LRMC Method for NCAA Tournament Prediction. In MIT Sloan Sports Analytics Conference, 1-9.

Brynjolfsson, E., Hitt, L. M. and Kim, H. H. (2011). Strength in Numbers: How Does Data-Driven Decision-making Affect Firm Performance?.

Bucheli, H., & Thompson, W. (2014). Statistics and Machine Learning at Scale: New Technologies Apply Machine Learning to Big Data. Insights From the SAS Analytics 2014 Conference.

Cao, C. (2012). *Sports Data Mining Technology Used in Basketball Outcome Prediction* (master's thesis). Dublin Institute of Technology, Dublin, Ireland.

Carbonell, J.G., Michalski, R.S., & Mitchell, T.M. (1983). Machine Learning: A Historical and Methodological Analysis. *AI Magazine, 4* (3), 69-79.

Cheng, G., Zhang, Z., Kyebambe, M.N. & Kimbugwe, N. (2016). Predicting the Outcome of NBA Playoffs Based on Maximum Entropy Principle. *Entropy, 18* (450), 1-12.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). Cambridge, Massachusetts: The MIT Press.

Edwards, C.T. (1998). Non-parametric procedure for knockout tournaments. *Journal of Applied Statistics, 25(3)*, 375–385

El Deeb, A. (2015, June 22). Parameter Sweeps (or Why Grid Search is Plain Stupid). *Medium*. Retrieved from https://medium.com

Fawcett, T., & Provost, F. (1996). Combining Data Mining and Machine Learning for Effective User Profiling. *Proceedings on the Second International Conference on Knowledge Discovery and Data Mining*, 8–13.

Fearnhead, P., & Taylor, B. M. (2010). Calculating Strength of Schedule, and Choosing Teams for March Madness. *The American Statistician*, *64 (2)*, 108–115.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. *International Conference on Computational Linguistics*, 841–847.

Graves, A., Mohamed, A.-R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *International Conference on Acoustics, Speech and Signal Processing*, (3), 6645-6649.

Gupta, R., & Pathak, C. (2014). A Machine Learning Framework for Predicting Purchase by online customers based on Dynamic Pricing. *Procedia Computer Science*, 36, 599–605.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., … Ng, A.Y. (2014). Deep Speech: Scaling up end-to-end speech recognition.

Hayashi, A. M. (2001). When to Trust Your Gut. *Harvard Business Review, 79(2)*, 59-65.

Husain, R., & Vohra, R. (2017). Applying Machine Learning in the Financial Sector. *International Education & Research Journal*, 3(1), 19–20.

Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., … Ng, A.Y. (2015). An Empirical Evaluation of Deep Learning on Highway Driving.

Illardi, S. (2014, April 16). The next big thing: real plus-minus. *ESPN.com:NBA*. Retrieved from http://www.espn.com

Ivanković, Z., Racković, M., Markoski, B., Radosav, D., & Ivković, M. (2010). Analysis of basketball games using neural networks. *11th International Symposium on Computational Intelligence and Informatics*, 251-256.

Kaplan, E. H., & Garstka, S. J. (2001). March Madness and the Office Pool. *Management Science*, 47 (3), 369–382.

Lin, J., Short, L., & Sundaresan, V. (2014). Predicting National Basketball Association Winners.

Marr, B. (2016, September 30). The Top 10 AI And Machine Learning Use Cases Everyone Should Know About. *Forbes*. Retrieved from http://www.forbes.com

Matthews, G., & Lopez, M. (2014, April 21). Q&A with Gregory Matthews and Michael Lopez, 1st Place in March ML Mania. *No Free Hunch*. Retrieved from http://blog.kaggle.com

Matzler, K., Bailom, F., & Mooradian, T. A. (2007). Intuitive decision making. MIT Sloan Management Review, 49 (1), 13-15.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms* (1st ed.). Cambridge, Massachusetts: The MIT Press.

Mitchell, T. M. (1997). *Machine Learning* (1st ed.). New York, NY: McGraw-Hill.

Niemi, J. B. (2005). Identifying and Evaluating Contrarian Strategies for NCAA Tournament Pools (Master's thesis). University of Minnesota, Minnesota, MN.

Oliver, D. (2004). *Basketball on paper: Rules and tools for performance analysis*. Washington, DC: Brassey's, Inc.

Osman, H. E. (2009). Random Forest-LNS Architecture and Vision. *7th IEEE International Conference on Industrial Informatics (INDIN 2009)*, 319-324.

Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

Pomerleau, D. A. (1991). Efficient Training of Artificial Neural Networks for Autonomous Navigation, *Neural Computation*, *3*, 88-97.

Reitermanová, Z. (2010). Data Splitting. *WDS'10 Proceedings of Contributed Papers, Part I*, 31-36.

Richardson, L., Wang, D., Zhang, C., & Yu, X. (2014). NBA Predictions, 1-10.

Schapire, R. (2008). *COS 511: Theoretical Machine Learning*.

Schwertman, N. C., McCready, T. A. & Howard, L. (1991). Probability models for the NCAA regional basketball tournaments. *The American Statistician*, 45 (1), 35-38.

Schwertman, N. C., Schenk, K. L. & Holbrook, B. C. (1996). More probability models for the NCAA regional basketball tournaments. *The American Statistician*, 50 (1), 34-38.

Seo, M.-G., & Barrett, L. F. (2007). Being Emotional During Decision Making - Good or Bad? An Empirical Investigation. Academy of Management Journal, 50 (4), 923–940.

Silver, N., (2003). Introducing PECOTA. In G. Huckabay, C. Kahrl, D. Pease et. al., (Eds.), *Baseball Prospectus*. (pp. 507-514). Dulles, VA: Brassey's Publishers.

Silver, N., (2015, October 9). We're Predicting The Career of Every NBA Player. Here's How. *FiveThirtyEight*. Retrieved from https://www.fivethirtyeight.com

Silver, N., & Fischer-Baum, R. (2015, May 21). How We Calculate NBA Elo Ratings. *FiveThirtyEight*. Retrieved from https://www.fivethirtyeight.com

Stolfo, S. J., Fan, D. W., Lee, W., Prodromidis, A. L., & Chan, P. K. (1997). Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results. *Association for the Advancement of Artificial Intelligence Workshop: Artificial Intelligence Approaches to Fraud Detection and Risk Management*, 83–90.

Williams, B., (2016, April 7). What Villanova's 2016 National Title Says About Defense In College Basketball. *Forbes*. Retrieved from https://www.forbes.com

Wright, M., & Wiens, J. (2016). Method to their March Madness: Insight from Mining a Novel Large-Scale Dataset of Pool Brackets. *KDD Workshop on Large-Scale Sports Analytics*.

Yuan, L.-H. et al. (2015). A Mixture-of-Modelers Approach to Forecasting NCAA Tournament Outcomes. *Journal of Quantitative Analysis in Sports* 11(1):13–27.

Zak, T. A., Huang, C. J., & Siegfried, J. J. (1979). Production Efficiency: The Case of Professional Basketball. The Journal of Business, *52 (3)*, 379-392.

Zhang, S.-X., Liu, C., Yao, K., & Gong, Y., (2015). Deep Neural Support Vector Machines for Speech Recognition. *International Conference on Acoustics, Speech and Signal Processing*, *(1)*, 4275-4279.

Zimmer, T., & Kuethe, T., (2008). Major Conference Bias and the NCAA Men’s Basketball Tournament.

Economics Bulletin, *12 (17)*, 1-6.

Zimmermann, A., Moorthy, S., & Shi, Z., (2013). Predicting NCAAB Match Outcomes Using ML Techniques – Some Results and Lessons Learned. *ECML/PKDD 2013, Sports Analytics Workshop*.

9. ANNEXES

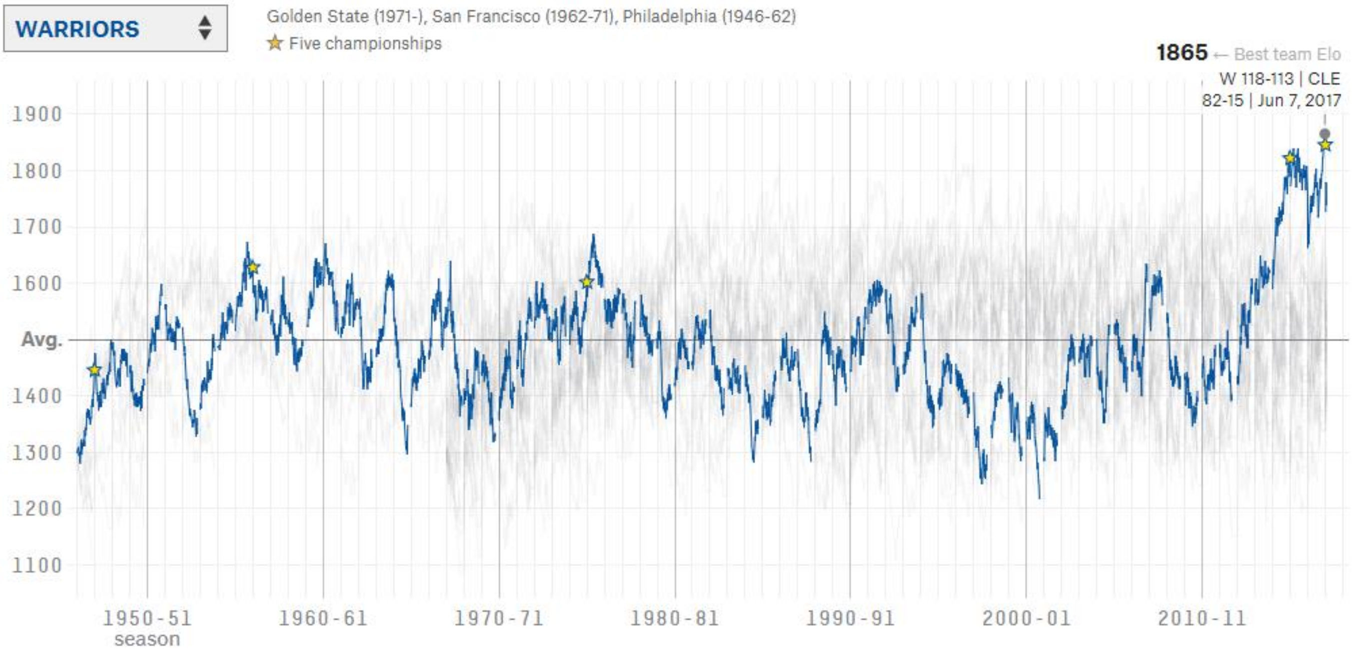
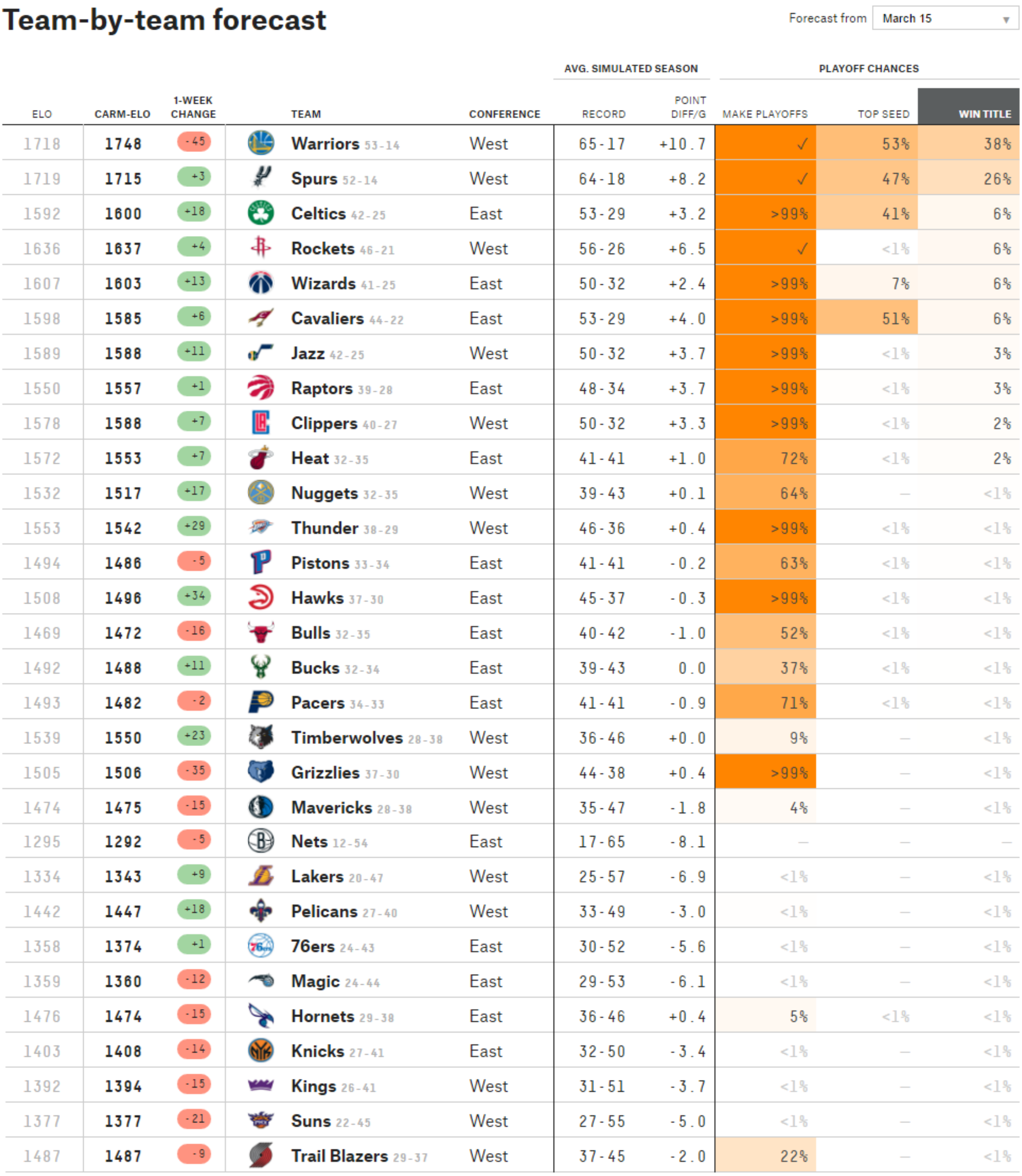


Figure 5 – FiveThirtyEight's Golden State Warriors Elo Rating[11]

[11] Available at https://projects.fivethirtyeight.com/complete-history-of-the-nba/#warriors



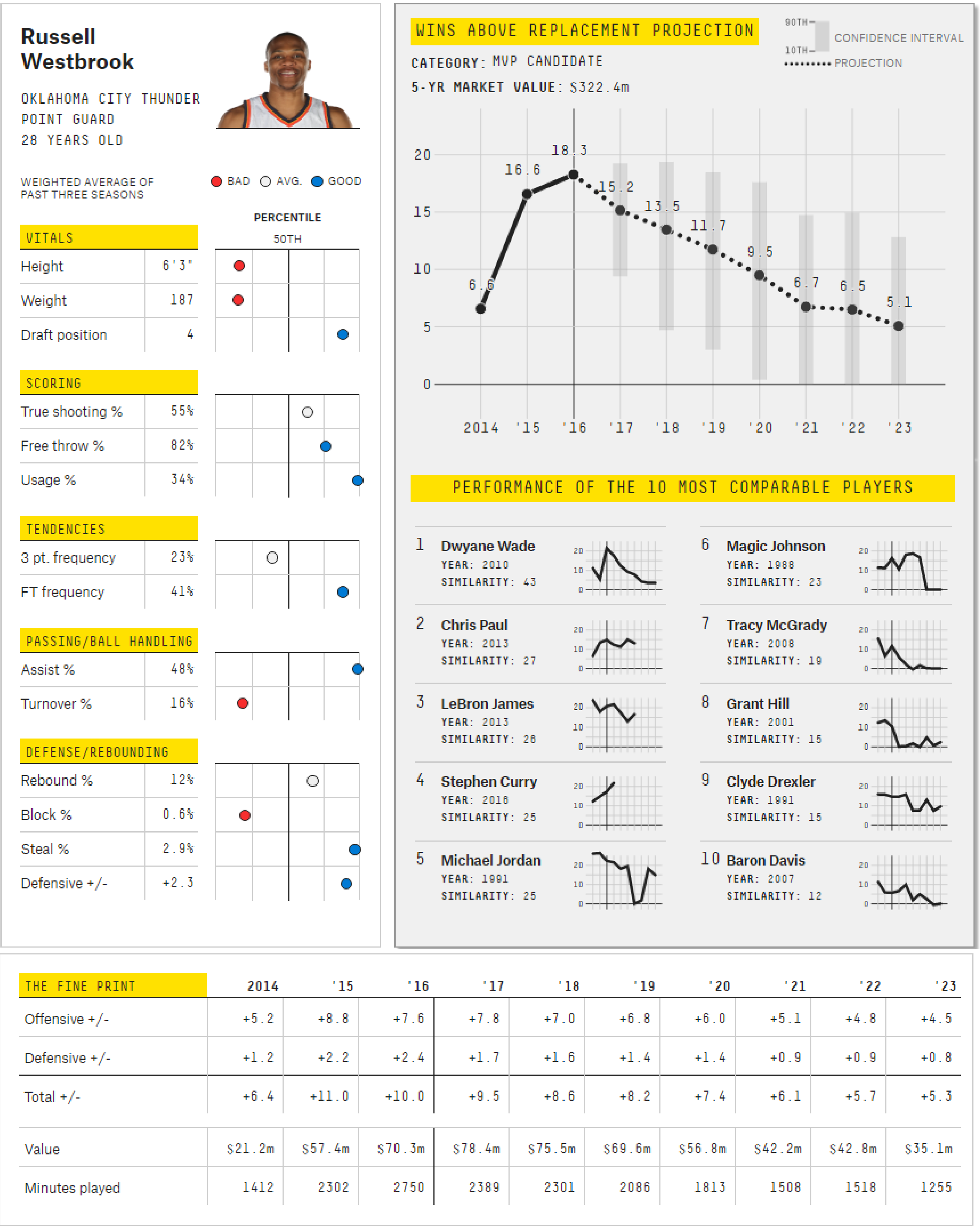
Team-by-team forecast

Forecast from March 15 ▾

					AVG. SIMULATED SEASON		PLAYOFF CHANCES		
ELO	CARM-ELO	1-WEEK CHANGE	TEAM	CONFERENCE	RECORD	POINT DIFF/G	MAKE PLAYOFFS	TOP SEED	WIN TITLE
1718	1748	-45	Warriors 53-14	West	65-17	+10.7	✓	53%	38%
1719	1715	+3	Spurs 52-14	West	64-18	+8.2	✓	47%	26%
1592	1600	+18	Celtics 42-25	East	53-29	+3.2	>99%	41%	6%
1636	1637	+4	Rockets 46-21	West	56-26	+6.5	✓	<1%	6%
1607	1603	+13	Wizards 41-25	East	50-32	+2.4	>99%	7%	6%
1598	1585	+6	Cavaliers 44-22	East	53-29	+4.0	>99%	51%	6%
1589	1588	+11	Jazz 42-25	West	50-32	+3.7	>99%	<1%	3%
1550	1557	+1	Raptors 39-28	East	48-34	+3.7	>99%	<1%	3%
1578	1588	+7	Clippers 40-27	West	50-32	+3.3	>99%	<1%	2%
1572	1553	+7	Heat 32-35	East	41-41	+1.0	72%	<1%	2%
1532	1517	+17	Nuggets 32-35	West	39-43	+0.1	64%	—	<1%
1553	1542	+29	Thunder 38-29	West	46-36	+0.4	>99%	<1%	<1%
1494	1486	-5	Pistons 33-34	East	41-41	-0.2	63%	<1%	<1%
1508	1496	+34	Hawks 37-30	East	45-37	-0.3	>99%	<1%	<1%
1469	1472	-16	Bulls 32-35	East	40-42	-1.0	52%	<1%	<1%
1492	1488	+11	Bucks 32-34	East	39-43	0.0	37%	<1%	<1%
1493	1482	-2	Pacers 34-33	East	41-41	-0.9	71%	<1%	<1%
1539	1550	+23	Timberwolves 28-38	West	36-46	+0.0	9%	—	<1%
1505	1506	-35	Grizzlies 37-30	West	44-38	+0.4	>99%	—	<1%
1474	1475	-15	Mavericks 28-38	West	35-47	-1.8	4%	—	<1%
1295	1292	-5	Nets 12-54	East	17-65	-8.1	—	—	—
1334	1343	+9	Lakers 20-47	West	25-57	-6.9	<1%	—	<1%
1442	1447	+18	Pelicans 27-40	West	33-49	-3.0	<1%	—	<1%
1358	1374	+1	76ers 24-43	East	30-52	-5.6	<1%	—	<1%
1359	1360	-12	Magic 24-44	East	29-53	-6.1	<1%	—	<1%
1476	1474	-15	Hornets 29-38	East	36-46	+0.4	5%	<1%	<1%
1403	1408	-14	Knicks 27-41	East	32-50	-3.4	<1%	—	<1%
1392	1394	-15	Kings 26-41	West	31-51	-3.7	<1%	—	<1%
1377	1377	-21	Suns 22-45	West	27-55	-5.0	<1%	—	<1%
1487	1487	-9	Trail Blazers 29-37	West	37-45	-2.0	22%	—	<1%

Figure 6 – FiveThirtyEight’s NBA Projections[12]

[12] Available at https://projects.fivethirtyeight.com/2017-nba-predictions/



Russell Westbrook

OKLAHOMA CITY THUNDER
POINT GUARD
28 YEARS OLD

WEIGHTED AVERAGE OF PAST THREE SEASONS

🔴 BAD ○ AVG. 🔵 GOOD

PERCENTILE
50TH

VITALS		
Height	6'3"	🔴
Weight	187	🔴
Draft position	4	🔵

SCORING		
True shooting %	55%	○
Free throw %	82%	🔵
Usage %	34%	🔵

TENDENCIES		
3 pt. frequency	23%	○
FT frequency	41%	🔵

PASSING/BALL HANDLING		
Assist %	48%	🔵
Turnover %	16%	🔴

DEFENSE/REBOUNDING		
Rebound %	12%	○
Block %	0.6%	🔴
Steal %	2.9%	🔵
Defensive +/-	+2.3	🔵

WINS ABOVE REPLACEMENT PROJECTION

CATEGORY: MVP CANDIDATE
5-YR MARKET VALUE: $322.4m

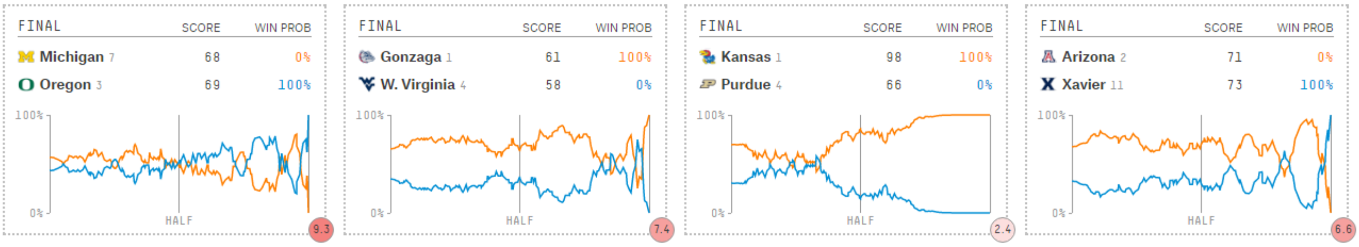
PERFORMANCE OF THE 10 MOST COMPARABLE PLAYERS

	Player	Year	Similarity
1	Dwyane Wade	2010	43
2	Chris Paul	2013	27
3	LeBron James	2013	26
4	Stephen Curry	2016	25
5	Michael Jordan	1991	25
6	Magic Johnson	1988	23
7	Tracy McGrady	2008	19
8	Grant Hill	2001	15
9	Clyde Drexler	1991	15
10	Baron Davis	2007	12

THE FINE PRINT	2014	'15	'16	'17	'18	'19	'20	'21	'22	'23
Offensive +/-	+5.2	+8.8	+7.6	+7.8	+7.0	+6.8	+6.0	+5.1	+4.8	+4.5
Defensive +/-	+1.2	+2.2	+2.4	+1.7	+1.6	+1.4	+1.4	+0.9	+0.9	+0.8
Total +/-	+6.4	+11.0	+10.0	+9.5	+8.6	+8.2	+7.4	+6.1	+5.7	+5.3
Value	$21.2m	$57.4m	$70.3m	$78.4m	$75.5m	$69.6m	$56.8m	$42.2m	$42.8m	$35.1m
Minutes played	1412	2302	2750	2389	2301	2086	1813	1508	1518	1255

Figure 7 – FiveThirtyEight's Carmelo Card[13]

[13] Available at https://projects.fivethirtyeight.com/carmelo/



FINAL	SCORE	WIN PROB
Michigan 7	68	0%
Oregon 3	69	100%

FINAL	SCORE	WIN PROB
Gonzaga 1	61	100%
W. Virginia 4	58	0%

FINAL	SCORE	WIN PROB
Kansas 1	98	100%
Purdue 4	66	0%

FINAL	SCORE	WIN PROB
Arizona 2	71	0%
Xavier 11	73	100%

Figure 8 – FiveThirtyEight's March Madness Predictions (1)

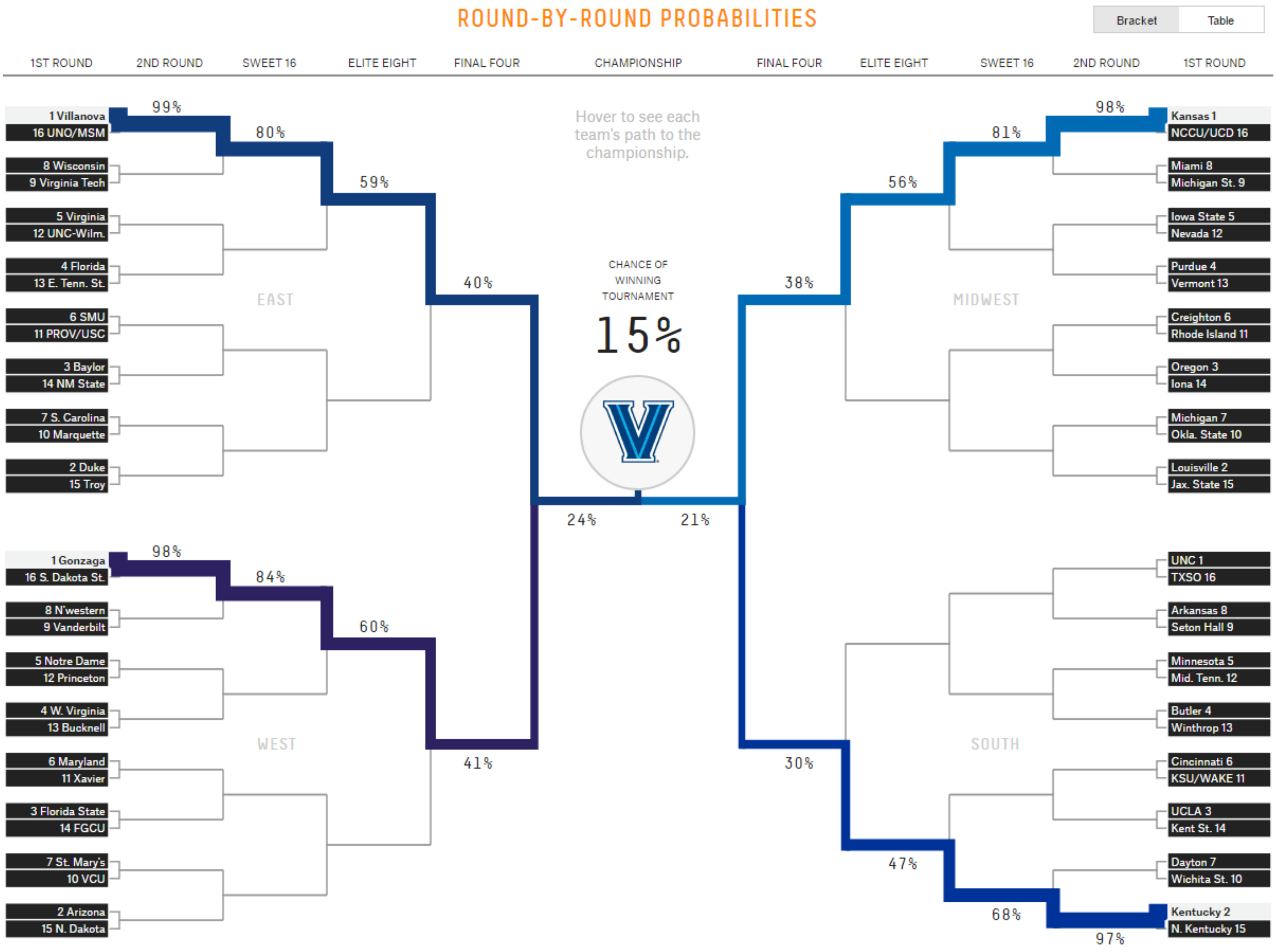


Figure 9 – FiveThirtyEight's March Madness Predictions (2)[14]

[14] Available at https://projects.fivethirtyeight.com/carmelo/

9.1 Appendix A - Dataset

Variable	Type	Formula	Team	Notes
ID	ID			Record ID (Year + Round + ID)
Year	Year			Year of the game
Round	Nominal			NCAA Tournament Round
IDRound	Nominal			Round of the match: 0: Opening Round / First Four, 1: First Round, 2: Second Round, 3: Sweet 16, 4: Elite 8, 5: Final 4, 6: Final
Team1	Nominal		A	Name of team A
Team2	Nominal		B	Name of team B
Coach1	Nominal		A	Coach of team A
Coach2	Nominal		B	Coach of team B
Seed1	Numeric		A	Assigned seed for that NCAA Tournament
W1	Numeric		A	Number of wins in the season (W)
L1	Numeric		A	Number of losses in the season (L)
Wp1	Numeric	W1 / (W1 + L1)	A	Win percentage in the season (W%)
PF1	Numeric		A	Team points per game (PPG)
PA1	Numeric		A	Opponent team points per game (OPPG)
FGM1	Numeric		A	Field goals made per game (FGM)
FGA1	Numeric		A	Field goals attempted per game (FGA)
FGp1	Numeric	FGM1 / FGA1	A	Field goal percentage (FG%)
2PM1	Numeric		A	2-point field goals made per game (2P - FGM)
2PA1	Numeric		A	2-point field goals attempted per game (2P - FGA)
2Pp1	Numeric	2PM1 / 2PA1	A	2-point field goal percentage (2P - FG%)
3PM1	Numeric		A	3-point field goals made per game (3P - FGM)
3PA1	Numeric		A	3-point field goals attempted per game (3P - FGA)
3Pp1	Numeric		A	3-point field goal percentage (3P - FG%)
3PAr1	Numeric	3PA1 / FGA1	A	Percentage of field goal attempts from 3-point range (3P-FGA Rate)
FTM1	Numeric		A	Free throws made per game (FTM)
FTA1	Numeric		A	Free throws attempted per game (FTA)
FTp1	Numeric	FTM1 / FTA1	A	Free throw percentage (FT%)
FTf1	Numeric	FTA1 / FGA1	A	Number of free throw attempts per field goal attempt (FTA Rate)
eFGp1	Numeric		A	Effective field goal percentage (eFG%). This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal.
TSp1	Numeric		A	True shooting percentage (TS%). A measure of shooting efficiency that considers 2-point field goals, 3-point field goals and free throws.
RPG1	Numeric		A	Rebounds per game (RPG)
APG1	Numeric		A	Assists per game (APG)
SPG1	Numeric		A	Steals per game (SPG)
BPG1	Numeric		A	Blocks per game (BPG)
coachnseasons1	Numeric		A	Coach number of seasons
coachW1	Numeric		A	Coach total number of wins
coachL1	Numeric		A	Coach total number of losses
coachgames1	Numeric	coachW1 + coachL1	A	Coach total number of games
coachWp1	Numeric	coachW1 / coachgames1	A	Coach winning percentage
coachncaagames1	Numeric		A	Coach number of NCAA Tournament games
coachnncaa1	Numeric		A	Coach number of NCAA Tournaments
coachncaaW1	Numeric		A	Coach number of NCAA Tournament wins
coachncaaL1	Numeric		A	Coach number of NCAA Tournament losses
coachfinal41	Numeric		A	Coach number of NCAA Tournament Final 4's
coachchamps1	Numeric		A	Coach number of NCAA Tournament Championships
nfinal41	Numeric		A	Team number of NCAA Tournament Final 4's

Variable	Type	Formula	Team	Notes
nchamps1	Numeric		A	Team number of NCAA Tournament Championships
Seed2	Numeric		B	Assigned seed for that NCAA Tournament
W2	Numeric		B	Number of wins in the season (W)
L2	Numeric		B	Number of losses in the season (L)
Wp2	Numeric	W2 / (W2 + L2)	B	Win percentage in the season (W%)
PF2	Numeric		B	Team points per game (PPG)
PA2	Numeric		B	Opponent team points per game (OPPG)
FGM2	Numeric		B	Field goals made per game (FGM)
FGA2	Numeric		B	Field goals attempted per game (FGA)
FGp2	Numeric	FGM2 / FGA2	B	Field goal percentage (FG%)
2PM2	Numeric		B	2-point field goals made per game (2P - FGM)
2PA2	Numeric		B	2-point field goals attempted per game (2P - FGA)
2Pp2	Numeric	2PM2 / 2PA2	B	2-point field goal percentage (2P - FG%)
3PM2	Numeric		B	3-point field goals made per game (3P - FGM)
3PA2	Numeric		B	3-point field goals attempted per game (3P - FGA)
3Pp2	Numeric		B	3-point field goal percentage (3P - FG%)
3PAr2	Numeric	3PA2 / FGA2	B	Percentage of field goal attempts from 3-point range (3P-FGA Rate)
FTM2	Numeric		B	Free throws made per game (FTM)
FTA2	Numeric		B	Free throws attempted per game (FTA)
FTp2	Numeric	FTM2 / FTA2	B	Free throw percentage (FT%)
FTf2	Numeric	FTA2 / FGA2	B	Number of free throw attempts per field goal attempt (FTA Rate)
eFGp2	Numeric		B	Effective field goal percentage (eFG%)
TSp2	Numeric		B	True shooting percentage (TS%)
RPG2	Numeric		B	Rebounds per game (RPG)
APG2	Numeric		B	Assists per game (APG)
SPG2	Numeric		B	Steals per game (SPG)
BPG2	Numeric		B	Blocks per game (BPG)
coachnseasons2	Numeric		B	Coach number of seasons
coachW2	Numeric		B	Coach total number of wins
coachL2	Numeric		B	Coach total number of losses
coachgames2	Numeric	coachW2 + coachL2	B	Coach total number of games
coachWp2	Numeric	coachW2 / coachgames2	B	Coach winning percentage
coachncaagames2	Numeric		B	Coach number of NCAA Tournament games
coachnncaa2	Numeric		B	Coach number of NCAA Tournaments
coachncaaW2	Numeric		B	Coach number of NCAA Tournament wins
coachncaaL2	Numeric		B	Coach number of NCAA Tournament losses
coachfinal42	Numeric		B	Coach number of NCAA Tournament Final 4's
coachchamps2	Numeric		B	Coach number of NCAA Tournament Championships
nncaa2	Numeric		B	Team number of NCAA Tournaments
nfinal42	Numeric		B	Team number of NCAA Tournament Final 4's
nchamps2	Numeric		B	Team number of NCAA Tournament Championships
RatioSeed	Numeric	Seed1 / Seed2		Ratio of seeds between teams
RatioW	Numeric	W1 / W2		Ratio of wins between teams
RatioL	Numeric	L1 / L2		Ratio of losses between teams
RatioWp	Numeric	Wp1 / Wp2		Ratio of win percentage between teams
RatioPF	Numeric	PF1 / PF2		Ratio of points per game between teams
RatioPA	Numeric	PA1 / PA2		Ratio of opponent points per game between teams
RatioPF1PA2	Numeric	PF1 / PA2		Ratio between team A's points per game and B's opponent points per game
RatioPF2PA1	Numeric	PA1 / PF2		Ratio between team A's opponent points per game and B's points per game
RatioFGM	Numeric	FGM1 / FGM2		Ratio between teams' FGM

Variable	Type	Formula	Team	Notes
RatioFGA	Numeric	FGA1 / FGA2		Ratio between teams' FGA
RatioFGp	Numeric	FGp1 / FGp2		Ratio between teams' FG%
Ratio2PM	Numeric	2PM1 / 2PM2		Ratio between teams' PM
Ratio2PA	Numeric	2PA1 / 2PA2		Ratio between teams' PA
Ratio2Pp	Numeric	2Pp1 / 2Pp2		Ratio between teams' PP%
Ratio3PM	Numeric	3PM1 / 3PM2		Ratio between teams' 3PM
Ratio3PA	Numeric	3PA1 / 3PA2		Ratio between teams' 3PA
Ratio3Pp	Numeric	3Pp1 / 3Pp2		Ratio between teams' 3P%
Ratio3PAr	Numeric	3PAr1 / 3PAr2		Ratio between teams' 3PAr%
RatioFTM	Numeric	FTM1 / FTM2		Ratio between teams' FTM
RatioFTA	Numeric	FTA1 / FTM2		Ratio between teams' FTA
RatioFTp	Numeric	FTp1 / FTp2		Ratio between teams' FTP%
RatioFTf	Numeric	FTf1 / FTf2		Ratio between teams' FTf
RatioeFGp	Numeric	eFGp1 / eFGp2		Ratio between teams' FG%
RatioTSp	Numeric	TSp1 / TSp2		Ratio between teams' TS%
RatioRPG	Numeric	RPG1 / RPG2		Ratio between teams' RPG
RatioAPG	Numeric	APG1 / APG2		Ratio between teams' APG
RatioSPG	Numeric	SPG1 / SPG2		Ratio between teams' SPG
RatioBPG	Numeric	BPG1 / BPG2		Ratio between teams' BPG
Ratiocoachnseasons	Numeric	coachnseasons1 / coachnseasons2		Ratio between coaches' number of seasons
RatiocoachW	Numeric	coachW1 / coachW2		Ratio between coaches' number of wins
RatiocoachL	Numeric	coachL1 / coachL2		Ratio between coaches' number of losses
Ratiocoachgames	Numeric	coachgames1 / coachgames2		Ratio between coaches' number of games
RatiocoachWp	Numeric	coachWp1 - coachWp2		Ratio between coaches' win percentage
Difcoachncaagames	Numeric	coachncaagames1 - coachncaagames2		Difference between coaches' number of NCAA Tournament games
Ratiocoachnncaa	Numeric	coachnncaa1 / coachnncaa2		Ratio between coaches' number of NCAA Tournaments
DifcoachncaaW	Numeric	coachncaaW1 - coachncaaW2		Difference between coaches' number of NCAA Tournament wins
DifcoachncaaL	Numeric	coachncaaL1 - coachncaaL2		Difference between coaches' number of NCAA Tournament losses
Difcoachfinal4	Numeric	coachfinal41 - coachfinal42		Difference between coaches' number of NCAA Tournament Final 4's
Difcoachchamps	Numeric	coachchamps1 - coachchamps2		Difference between coaches' number of NCAA Tournament Championships
Rationncaa	Numeric	nncaa1 / nncaa2		Ratio between teams' number of NCAA Tournaments
Difnfinal4	Numeric	nfinal41 - nfinal42		Difference between teams' number of NCAA Tournament Final 4's
Difnchamps	Numeric	nchamps1 - nchamps2		Difference between teams' number of NCAA Tournament Championships
Output	Binary	0 or 1		Result of the game: 1 for A to win, 0 for B to win