INST 737

Digging into Data

Project Report

Data Mining on 2016 NCAA Basketball Tournament for Predictive Modeling

Submitted by:

Adarsh Srinivas (adarshs1@umd.edu)
Isha Kaur Ratti (ikratti@umd.edu)
Jay Sheth (jaysheth@umd.edu)

MOTIVATION

The National Collegiate Athletic Association (NCAA) Men's Division I Basketball Tournament is a single elimination tournament played each spring in the United States, currently featuring 68 college basketball teams, to determine the national championship of the major college basketball teams. It is informally known as March Madness or the Big Dance, and has become one of the most famous annual sporting events in the United States. The influence of this tournament is significant and has attracted fans all over United States especially college students. Over the years, a lot of studies have been carried out to predict the results of the game, winning team in a tournament and analyze player performance to assist coaches. Though a rich and detailed data is available through collection of data and statistics over the years, it is still very complex to analyze and try to predict a game. In order to counter that complexity, our motivation is to implement a wide range of machine learning methods to achieve better predictions with the data available.

GOAL

In this project, we aim to analyze and predict the winner of each possible matchup that may happen in 2016 NCAA Basketball Tournament. The main objective is to execute different machine learning methods to identify which method gives us a better prediction rate for all the matches in the 2016 tournament. As a secondary objective, we also aim to predict the top 16 and top 32 teams in the tournament.

DATASET

The dataset used for this project is taken from Kaggle for the March Madness Machine Learning competition. The scope of this dataset covers a decade of historical NCAA games. There is data covering seasons, tournament, seeds, tournament slots, teams etc. The final dataset used for our analysis is derived from the different csv files available on Kaggle and consists of three different datasets:

1. **Season Detailed Results 2015:** A detailed set of game results, covering seasons 2003-2015 that includes team-level total statistics for each game (total field goals attempted, offensive rebounds, etc.)
2. **Tournament Detailed Results**: This file contains the more detailed results for tournament games from 2003 onward.
3. **Season Detailed Results 2016:** This file contains the season statistics for 2016.

FEATURE ENGINEERING

In professional basketball, the most commonly used statistical benchmark for comparing the overall value of players is called efficiency. It is a composite basketball statistic that is derived from basic individual statistics: points, rebounds, assists, steals, blocks, turnovers and shot attempts. The efficiency stat, in theory, accounts for both a player's offensive contributions (points, assists) and their defensive contributions (steals, blocks), but it is generally thought that efficiency ratings favor offense-oriented players over those who specialize in defense, as defense is difficult to quantify with currently tabulated statistics.

Efficiency: Score + Rebound + Assist + Steal + Block − Missed Field Goals − Missed Free Throws − Turn Over

The problem with these statistics is that they are all raw numbers, which limits their expressiveness. If a team collects 30 rebounds in total during a game, we cannot know whether to consider this a good result unless we know how many rebounds were there to be had in the first place. 30 of 40 is obviously a better rebound rate than 30 of 60. Similar statements can be made for field goals and free throws, which is why statistics like offensive rebound rate (ORR), turnover rate (TOR), or field goals attempted (FGA) will paint a better picture. Even in that case, however, such statistics are not normalized: 40 rebounds in a game in which both teams combined to shoot 100 times at the basket is different from 40 rebounds when there were only 80 scoring attempts. For normalization, one can calculate the number of possessions in a given game:

Possessions = 0.96 ∗ (Field Goals Attempted − Offensive Rebound − Turn Over + (0.475 ∗ Field Throws Attempted))

Offensive and Defensive efficiencies normalize teams' points scored and allowed per 100 possessions:

Offensive Efficiency = Points scored ∗ 100/ Possessions

Defensive Efficiency = Points allowed ∗ 100/ Possessions

Effective field goal percentage (0.4):

eFG% = FGM + 0.5 *FGM3/ FGA

Turnover percentage (0.25):

TO% = TO/Possessions

Offensive Rebound Percentage (0.2):

OR% = OR /(OR + DROpponent)

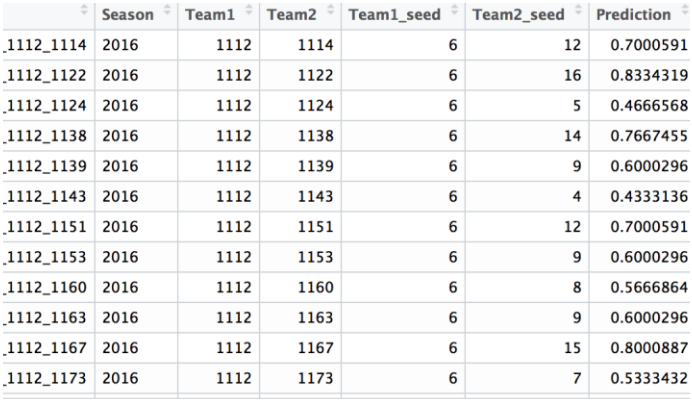
Free throw rate (0.15):

FT R = FTA/ FGA

METHODOLOGY

In this project, we have implemented feature engineering to determine some important features that could be used to build predictive models. As part of the predictive modeling, we have used the following techniques to determine the winners of the matchups in 2016:

1. **Linear Regression:** Initially, Linear Regression was performed which enabled us to achieve a prediction rate for Team 1 winning. The predictors considered were the seed of the two teams. A snapshot of the prediction rate(probability) achieved using this method is shown below:



	Season	Team1	Team2	Team1_seed	Team2_seed	Prediction
_1112_1114	2016	1112	1114	6	12	0.7000591
_1112_1122	2016	1112	1122	6	16	0.8334319
_1112_1124	2016	1112	1124	6	5	0.4666568
_1112_1138	2016	1112	1138	6	14	0.7667455
_1112_1139	2016	1112	1139	6	9	0.6000296
_1112_1143	2016	1112	1143	6	4	0.4333136
_1112_1151	2016	1112	1151	6	12	0.7000591
_1112_1153	2016	1112	1153	6	9	0.6000296
_1112_1160	2016	1112	1160	6	8	0.5666864
_1112_1163	2016	1112	1163	6	9	0.6000296
_1112_1167	2016	1112	1167	6	15	0.8000887
_1112_1173	2016	1112	1173	6	7	0.5333432

2. **Logistic Regression:** Since the variable of interest is a binary dependent variable, it was typical to choose this method to determine the relationship between the categorical dependent variable and one or more predictors.

3. **Support Vector Machines:** A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyper plane which categorizes new examples. The SVM algorithm is based on finding the hyper plane that gives the largest minimum distance to the training examples. The optimal separating hyper plane maximizes the margin of the training data. There were two types of non-linear kernels that were used for this project; Laplacian and Radial Basis function

4. **Decision Trees:** Recursive partitioning is a fundamental tool in data mining. It helps us explore the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcome. Tree methods such as CART (classification and regression trees) can be used as alternatives to logistic regression. It is a way that can be used to show the probability of being in any hierarchical group.

5. **Random Forest:** We chose to implement Random Forest classification, which is an extension to the decision trees. The main advantage of this method, is its high rate of accuracy since it is an ensemble method. There are other advantages of Random forest over Logistic Regression and Boosted Decision trees. These primarily relate to over fitting. Random forest never over-fits as each tree is constructed using randomly selected data with replacement and each node is split using the best split. This allows every tree to be constructed at random and hence is independent of other trees thus overcoming the over fitting problem.

6. **Boosting:** Boosting type decision trees were further used to achieve higher accuracy as compared to general decision trees. Boosting can be used for classification type problems and hence it was an idea fit for this project. Using "bagging" or "boosting" algorithms with decision trees generally improves accuracy while maintaining same simplicity and

speed. We had a large number of training examples and hence boosting algorithm could handle this very well as compared to Logistic regression.

7. **Neural Networks:** ANN's are universal approximators and can be used to approximate whatever function generated the data with an arbitrary amount of accuracy. It can detect complicated features as they learn an arbitrary non-linear boundary. As there was enough training data available, neural networks could give us a better accuracy modelling non-linear functions. Furthermore, Neural Networks can help us explore data sets in search for relevant variables or groups of variables. The results of such explorations can then facilitate the process of building predictive models.

COMPARATIVE ANALYSIS:

Model	Correct Matches	Total Matches	Accuracy
Neural Network	48	67	0.716418
Logistic Regression	48	67	0.716418
Boosting	46	67	0.686567
Random Forest	46	67	0.686567
SVM (Laplacian)	45	67	0.671642
SVM (Radial Basis)	44	67	0.656716
Decision Trees	44	67	0.656716

Decision trees and Boosting:

The accuracy for decision trees is approximately 0.6567 whereas boosted decision trees enabled us to achieve a better accuracy of 0.6865. Out of the 67 matches that took place in the 2016 tournament, decision trees predicted 44 of them correctly while boosted decision trees could predict the results of the additional two matches accurately totaling to 46. Thus, our rationale behind using boosted decision trees was justified as it allowed us to get a better accuracy.

Neural Networks and Support Vector Machines:

Neural Networks technique had an accuracy of 0.7164 which was the highest we achieved among all machine learning methods used in this project. Support vector machines with Laplacian kernel had an accuracy of 0.6716 while radial basis function kernels had a comparatively lower accuracy of 0.6567. Thus, Neural Networks could correctly predict a total of 48 matches out of 67 in the 2016 NCAA tournament while SVM(Laplacian) could predict 45 and SVM (Radial Basis) could only predict 44 correctly.

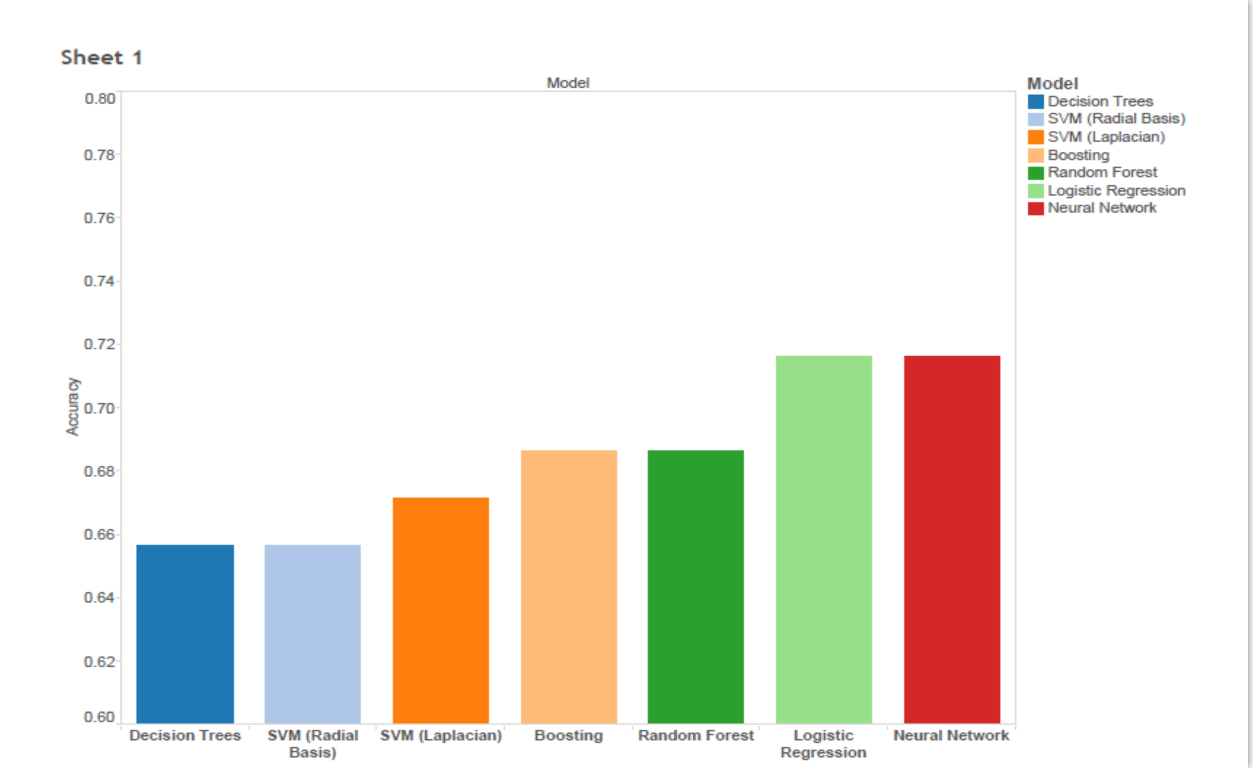
Decision Trees and Random Forest:

The accuracy for decision trees is approximately 0.6567 whereas random forest method had an accuracy of 0.6865. Decision trees contributing in predicting 44 matches out of 67 correctly while the random forest technique provided better results as it was able to correct judge the winners of 46 matches. Random forest was use to overcome over fitting and for its high rate of accuracy since it is an ensemble method. This was evident from the results obtained.

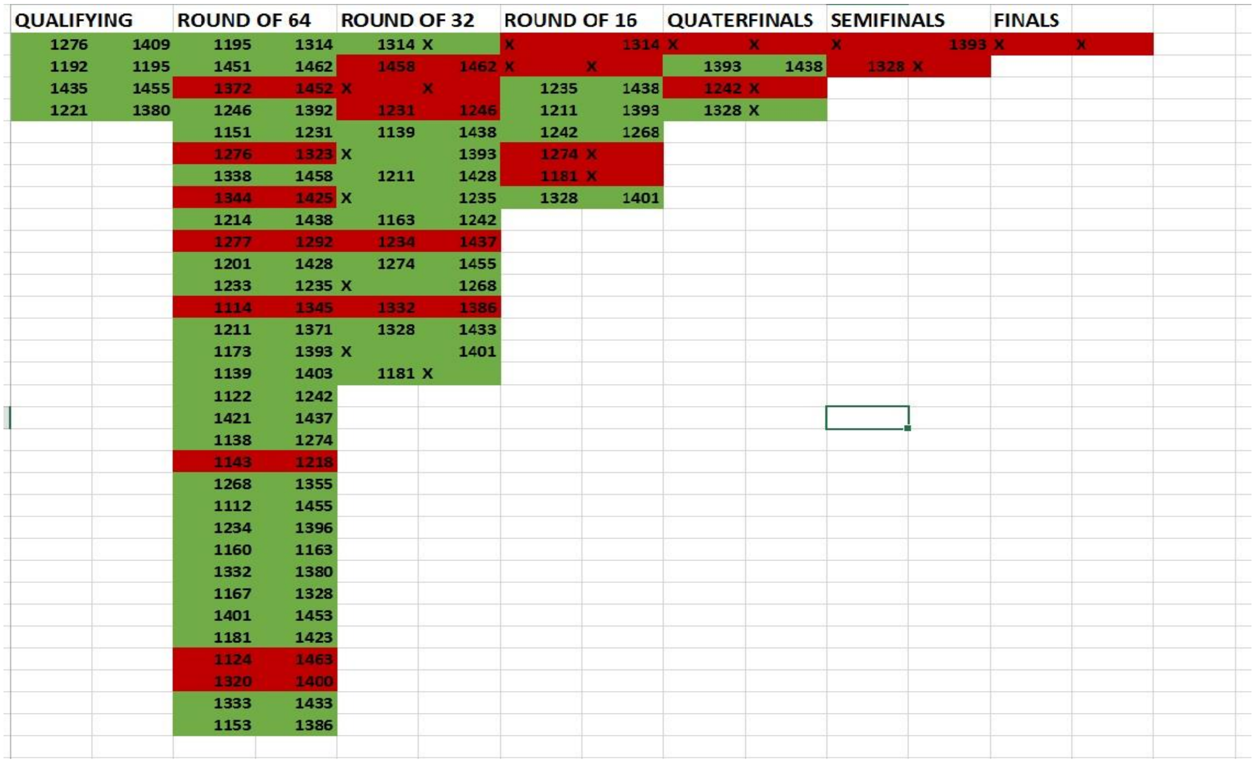
Logistic Regression and Support Vector Machines:

Along with Neural networks, the highest accuracy for our model was obtained through Logistic Regression. This method had an overall accuracy of 0.7164 while SVM(Laplacian) as well as SVM (Radial Basis) had comparatively much lower accuracy in 0.6716 and 0.6567 respectively. This could be summarized in stating that Logisitic Regression was able to tell the winners of 48 matches correctly out of the total 67 while SVM(Laplacian) and SVM (Radial Basis) could only tell the results of45 and 44 matches correctly.

The figure below shows the performance of the various machine learning methods used in this project to build predictive models:



RESULTS



QUALIFYING		ROUND OF 64		ROUND OF 32		ROUND OF 16		QUATERFINALS		SEMIFINALS		FINALS	
1276	1409	1195	1314	1314	X	X	1314	X	X	X	1393	X	X
1192	1195	1451	1462	1458	1462	X	X	1393	1438	1328	X		
1435	1455	1372	1452	X	X	1235	1438	1242	X				
1221	1380	1246	1392	1231	1246	1211	1393	1328	X				
		1151	1231	1139	1438	1242	1268						
		1276	1323	X	1393	1274	X						
		1338	1458	1211	1428	1181	X						
		1344	1425	X	1235	1328	1401						
		1214	1438	1163	1242								
		1277	1292	1234	1437								
		1201	1428	1274	1455								
		1233	1235	X	1268								
		1114	1345	1332	1386								
		1211	1371	1328	1433								
		1173	1393	X	1401								
		1139	1403	1181	X								
		1122	1242										
		1421	1437										
		1138	1274										
		1143	1218										
		1268	1355										
		1112	1455										
		1234	1396										
		1160	1163										
		1332	1380										
		1167	1328										
		1401	1453										
		1181	1423										
		1124	1463										
		1320	1400										
		1333	1433										
		1153	1386										

The NCAA basketball tournament is a knockout tournament where the winner goes to the next round whereas the loser is eliminated from the next round.

The actual bracket for the 2016 tournament and our predictions for it are shown in the figure below. The matchups that are marked in green were correctly predicted by our model, whereas our model failed to predict the matchups in red.

The tournament stared with 68 team as shown in the figure. There were four qualifying matches all of which were correctly predicted by our model.

Then in the round of 64, we correctly predicted the outcome of 24 out of 32 matches (i.e. 75% of the matches). Thus we were able to predict 24 teams of the top 32. The teams which we could not predict moving into the next round are marked by X in the figure. We expected the model to do better in the initial rounds but this year's round of 64 matchups had quite a lot of upsets caused by lower ranked or weaker teams defeating the traditionally stronger teams.

Then in the round of 32 the model correctly predicted the outcome of 11 out of 16 matches. Thus, we were correctly able to predict 11 of the top 16 teams in the NCAA 2016 Basketball tournament.

Similarly, the model correctly predicted 4 of the quarterfinalists.

Out of the 4, the model correctly predicted 2 of the semifinalists (Syracuse and Oklahoma).

The model could not predict the finalists correctly though.

It is understandable that the model doesn't perform too well in the latter part of the tournament because of the effect of previous round's wrong predictions. These statistics are shown in the table below:

ROUND	CORRECT PREDICTIONS	TOTAL TEAMS
Round of 64	64	64
Round of 32	24	32
Round of 16	11	16
Quarter Finals	4	8
Semi Finals	2	4
Finals	0	2

REFERENCES

1. Heit, E., Price, P., & Bower, G. (1994). A Model for Predicting the Outcomes for Basketball Games. Retrieved May 6, 2016 from http://faculty.ucmerced.edu/sites/default/files/eheit/files/basketball.pdf
2. Torres, R. A. (2013). Prediction of NBA games based on Machine Learning Methods. Retrieved May 3, 2016 from http://homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf
3. Zimmermann, A., Moorthy, S., & Shi, Z. (2013). Predicting college basketball match outcomes using machine learning techniques: Some results and lessons learned. Retrieved May 3, 2016 from http://arxiv.org/abs/1310.3607
4. March Machine Learning Mania. (March 2016). Retrieved April 7, 2015 from https://www.kaggle.com/c/march-machine-learning-mania-2016/data