2016 NCAA Basketball Tournament Predictions

Adarsh Srinivas | Isha Ratti | Jay Sheth

About NCAA Basketball Tournament

- 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.

- Informally known as March Madness or the Big Dance, and has become one of the most famous annual sporting events in the United States.

Objective

- To predict the winner of each possible matchup that may happen in 2016 NCAA Basketball Tournament
- To predict the top16, top 32 teams in the tournament

Data

- Source : https://www.kaggle.com/c/march-machine-learning-mania-2016/data
- The final dataset consists of:
  - **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.)
  - **Tournament Detailed Results** : This file contains the more detailed results for tournament games from 2003 onward
  - **Season Detailed Results 2016:** This file contains the season statistics for 2016
- New features developed to better predict the winner
- Weighted Average of statistics calculated for every team
- The combined dataset consists of 72088 entries

Challenges

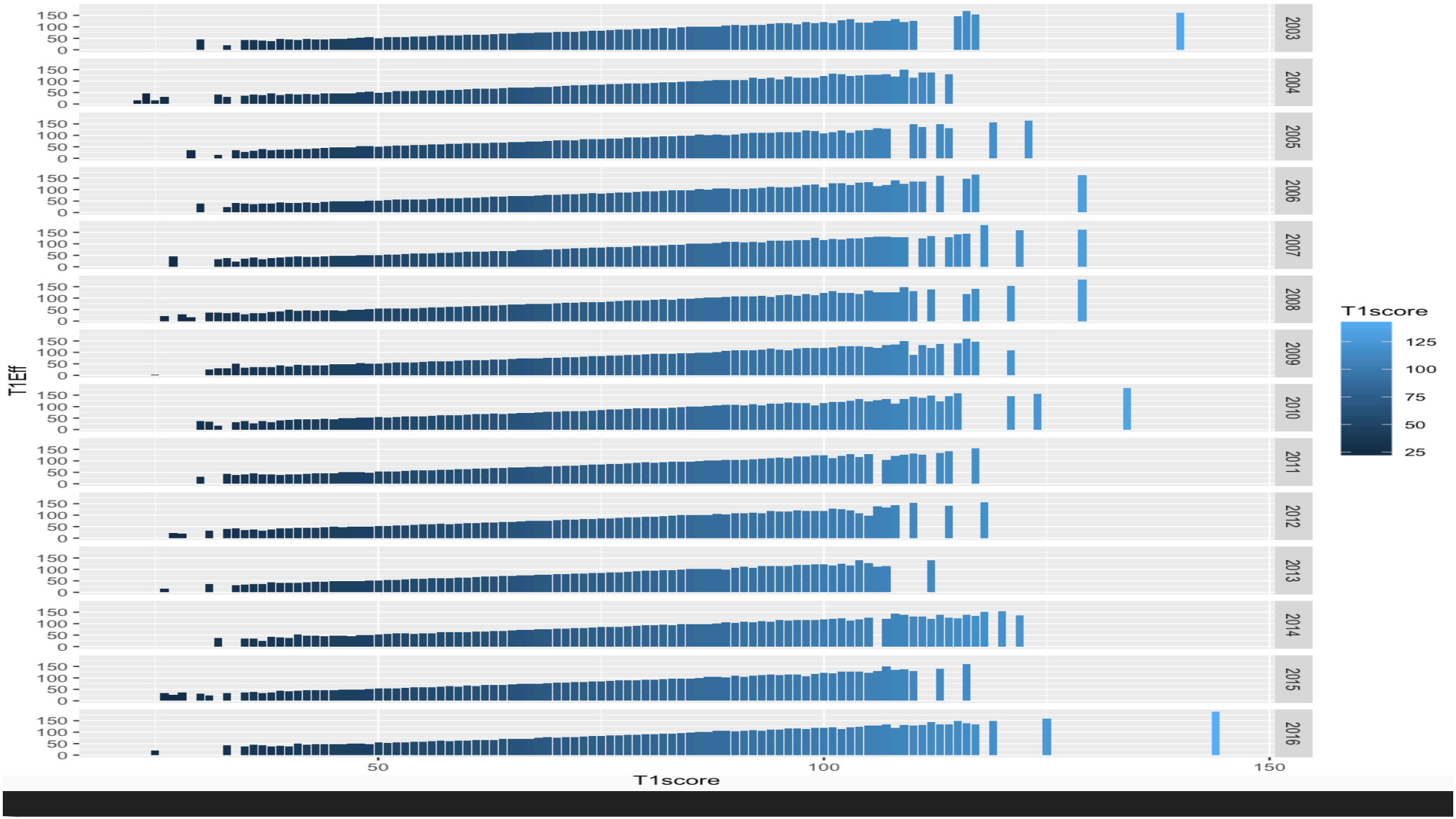
- No dependent variable in the dataset. To build predictive models, an output variable ***'Result' is created***. This is a dichotomous variable with values 0 or 1 indicating whether Team 1 won the match or not.

- The dataset includes statistics of every match that has been played in a season and the NCAA tournament since 2003. Prediction of matches in 2016 for every possible matchup, required the statistics of individual teams for all the matches they had played till 2015. This required a lot of data wrangling and we did this using aggregate functions, weighted arithmetic mean, merge functions etc.

Season	Daynum	Wteam	Wscore	Lteam	Lscore	Wloc	Numot	Wfgm	Wfga	Wfgm3	Wfga3	Wftm	Wfta	Wor	Wdr	Wast	Wto	Wstl
2003	134	1421	92	1411	84	N	1	32	69	11	29	17	26	14	30	17	12	5
2003	136	1112	80	1436	51	N	0	31	66	7	23	11	14	11	36	22	16	10
2003	136	1113	84	1272	71	N	0	31	59	6	14	16	22	10	27	18	9	7
2003	136	1141	79	1166	73	N	0	29	53	3	7	18	25	11	20	15	18	13
2003	136	1143	76	1301	74	N	1	27	64	7	20	15	23	18	20	17	13	8
2003	136	1163	58	1140	53	N	0	17	52	4	14	20	27	12	29	8	14	3

Feature Engineering

- Efficiency: Score + Rebound + Assist + Steal + Block – Missed Field Goals – Missed Free Throws – Turn Over
- Possessions = 0.96 ∗ (Field Goals Attempted – Offensive Rebound – Turn Over + (0.475 ∗ Field Throws Attempted))
- 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


T1Eff
150
100
50
0
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
50
100
150
T1score
T1score
125
100
75
50
25

Logistic Regression

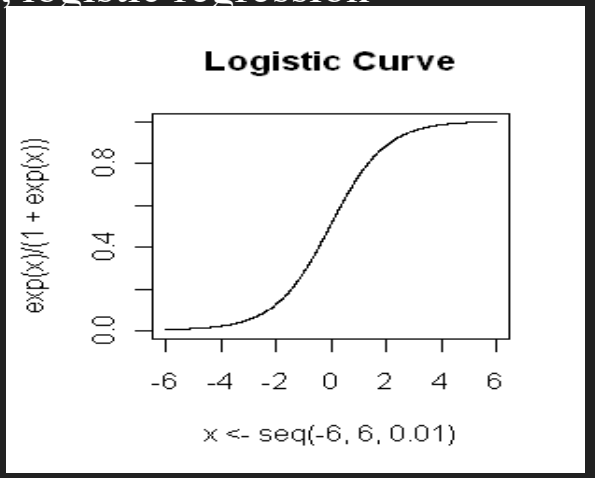
- Logistic regression is a method for fitting a regression curve, y = f(x), when y consists of proportions or probabilities, or binary coded (0,1--failure, success) data.

- When the response is a binary (dichotomous) variable, and x is numeric, logistic regression fits a logistic curve to the relationship between x and y.

- The logistic function is:

$$y = [\exp(b0 + b1x)] / [1 + \exp(b0 + b1x)]$$

b0 and b1= the regression coefficients

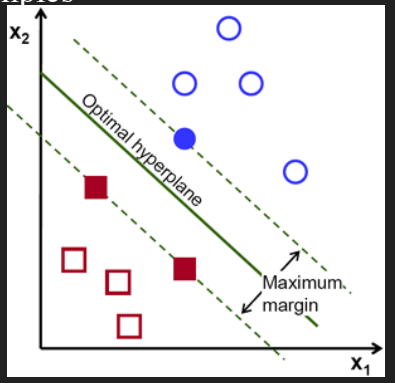


Logistic Regression

Feature Combination	Accuracy
T1score+T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2score+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6716**
T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.716418**

Support Vector Machine

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



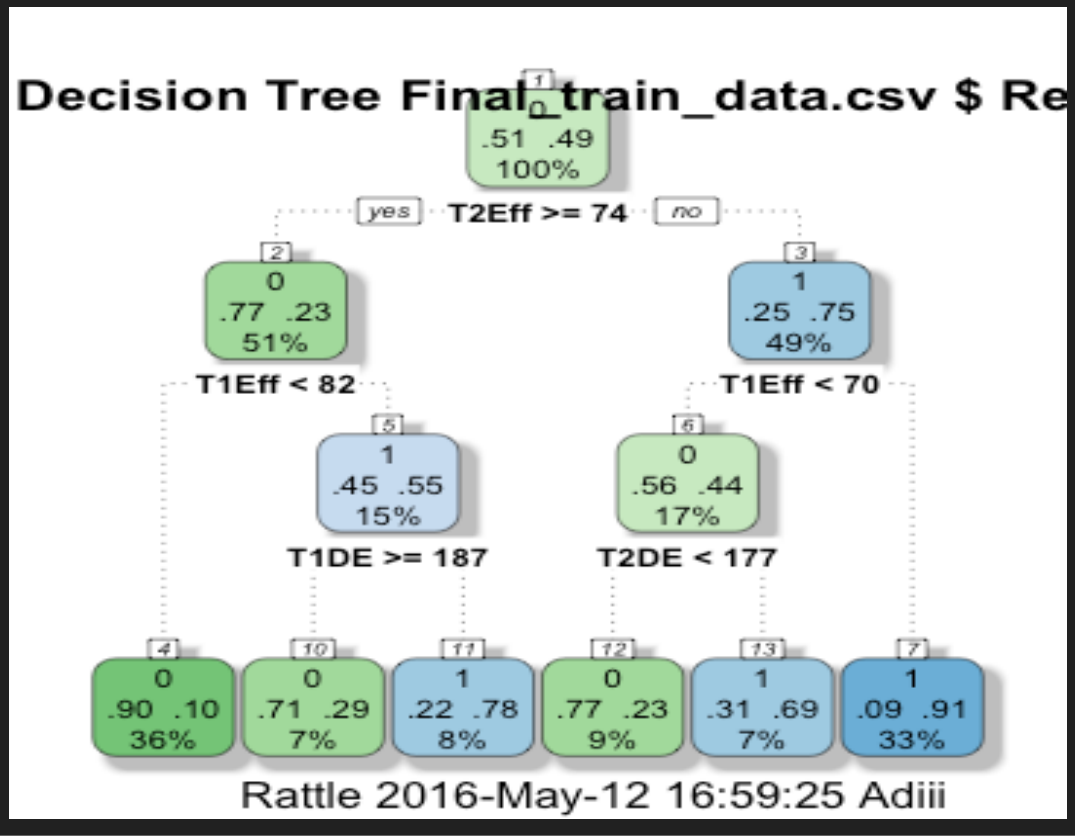
SVM (Laplacian)

Feature Combination	Accuracy
T1score+T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2score+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6665**
T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6716**

Decision Tree

- 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.
- Boosted Trees are used for this classification problem

Decision Tree



Decision Trees

Feature Combination	Accuracy
T1score+T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2score+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6417**
T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6567**

Random Forest

- A **Random Forest** classifier uses a number of decision trees, in order to improve the classification rate
- It uses multiple models for better performance than just using a single tree model
- In addition because many sample are selected in the process a measure of variable importance can be obtain and this approach can be used for model selection

Random Forest

Feature Combination	Accuracy
T1score+T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2score+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6567**
T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.6865**

Neural Network

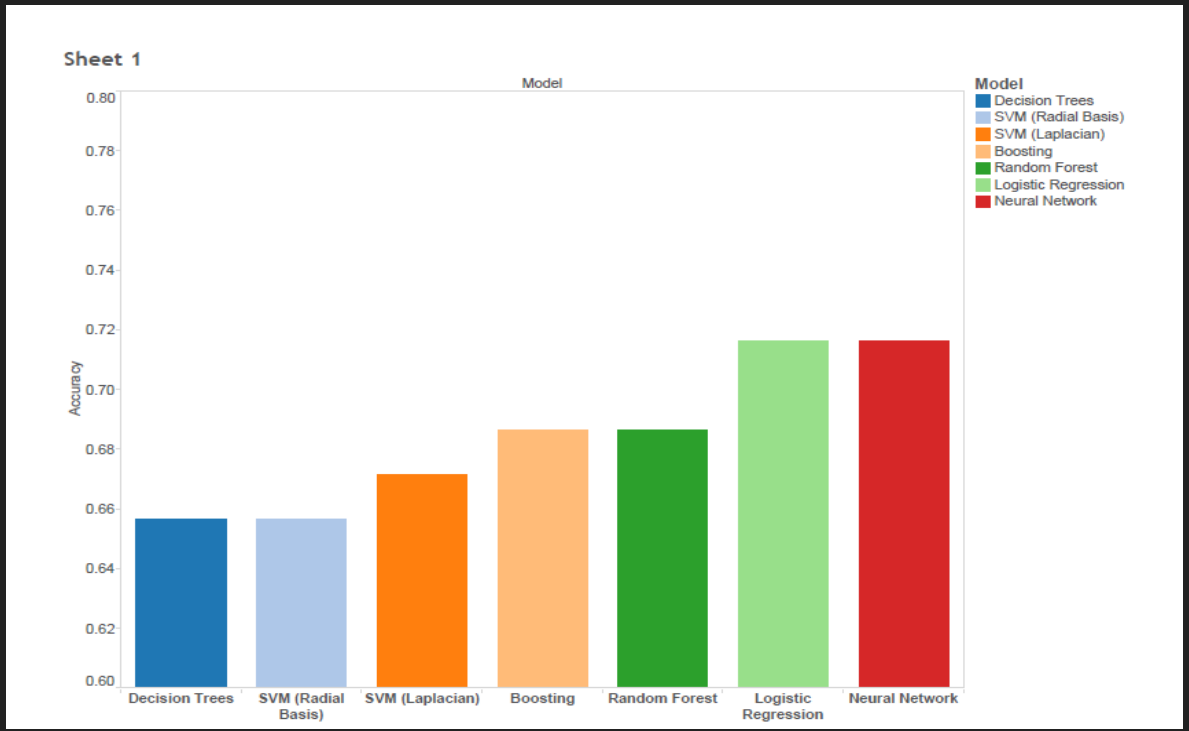
- A neuron defines a relationship between the input signals received from dendrites and the output signal
- Input signals are summed and passed to activation function f
- Training means learning the values for the weights that will best approximate the output labels in our training set (x,y)
- Original values for the weights are random

Neural Network

Feature Combination	Accuracy
T1score+T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2score+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.67167**
T1Eff+T1poss+T1OE+T1DE+T1efg+T1top+T1torp+T1ftr+T2Eff+T2poss+T2OE+T2DE+T2efg+T2top+T2torp+T2ftr	**0.716418**

Results

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



NCAA 2016 Tournament Bracket Prediction

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										

Any Questions?