
March Machine Learning Mania 2016

Team Member

Xin Ming | Kaize Hou | Chenxi Zhang

Content

- Problem
- Background
- Approach
- Data
- Result
- Evaluation
- Conclusion
- Reference

Problem

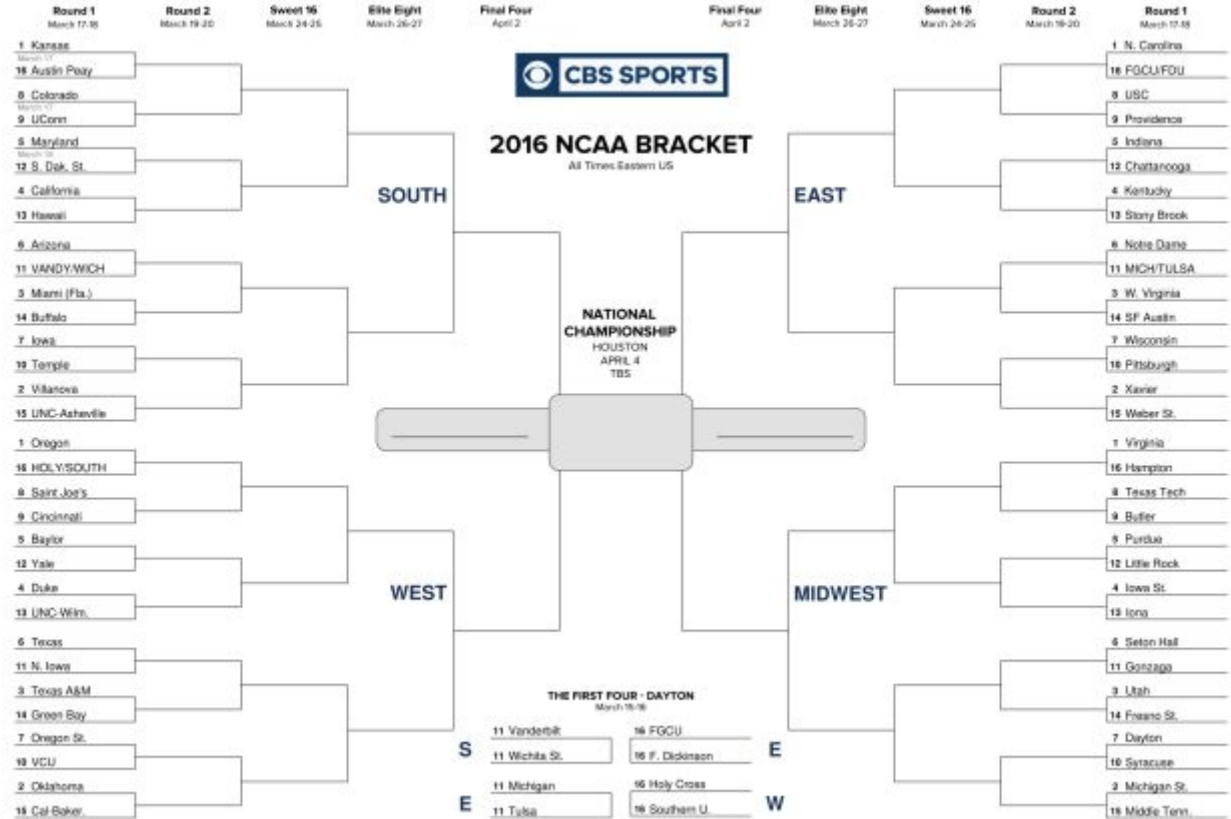
Predict the 2016 NCAA
Basketball Tournament

Input: two team

Output:

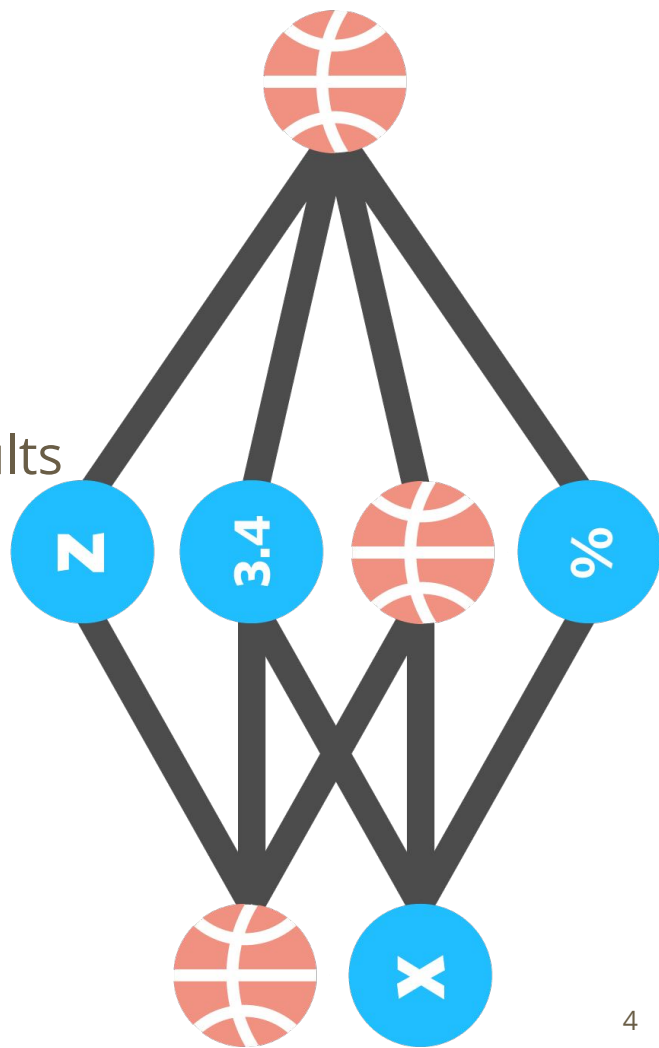
The probability that

Team 1 beat Team 2

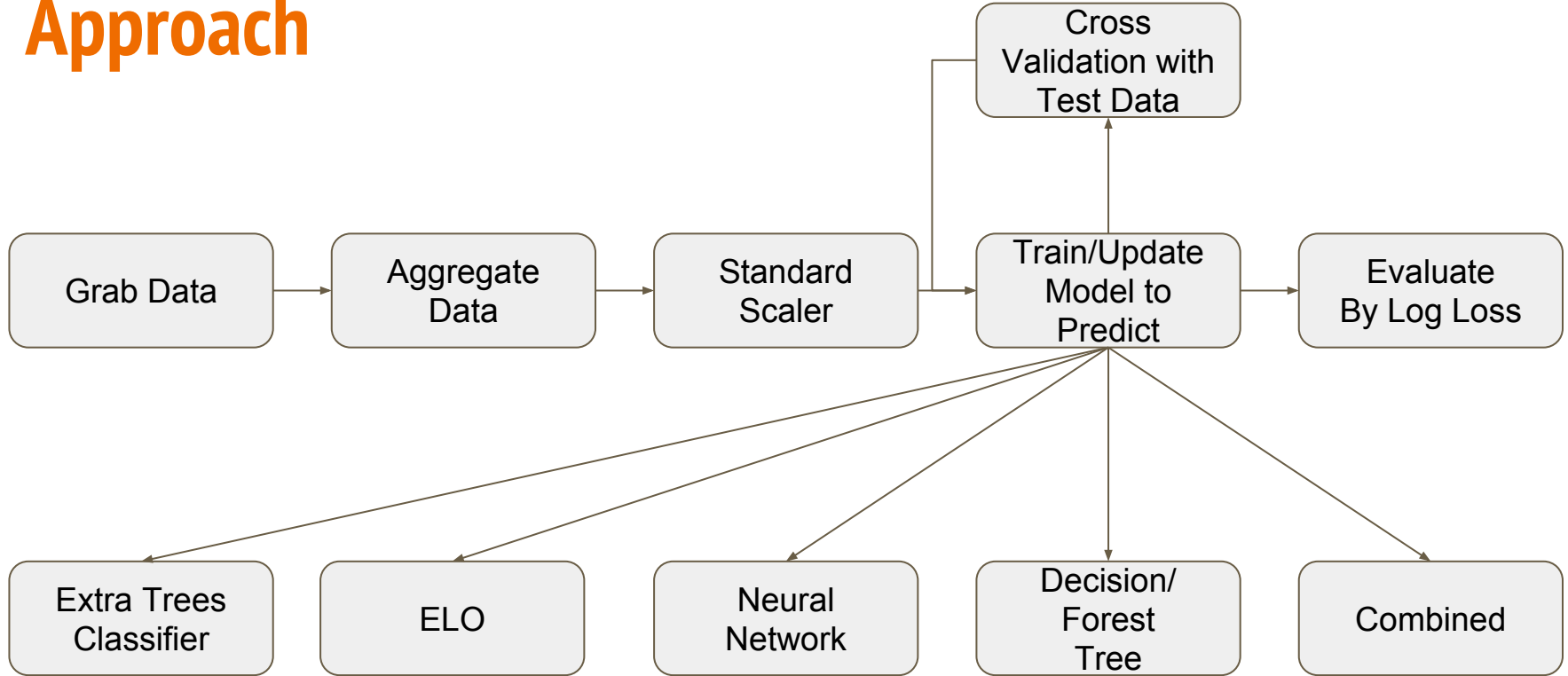


Background

- Kaggle Competition
- Predict 2016 NCAA ☐TOURNAMENT
- Data from 1986 - 2016 seasons NCAA results



Approach





Data

Table Name	Desc.	The no. of Features	The no. of historical data
Tourney Compact Results	all seasons of historical data	8	1984
Tourney Seeds	seeds for all teams	3	2083
Regular Season Compact Results	31 seasons of historical data	8	145290
...

Evaluation and Result

#	Approach	Log loss	# on leader board
1	ELO rating system + Extra Trees Classifier + GPIndividual	0.589	286
2	Extra Trees Classifier v3	0.608	364
3	Extra Trees Classifier v2	0.632	399
4	Extra Trees Classifier v1	0.796	481
5	ELO rating system + GPIndividual	0.865	490
6	GPIndividual	0.993	563
7	GPIndividual	0.996	564

-	MLMKZ	0.589194
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been		
287	—  no one ‡	0.589194
365	— Jeremiah ‡	0.608481
-	MLMKZ	0.608559
-	MLMKZ	0.865466
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been		
491	—  SecondPlan	0.865466
-	MLMKZ	0.993600
Post-Deadline Entry If you would have submitted this entry during the competition, you would have been		
564	— Cherry's Daddy	0.996282

Ranks on Leaderboard Approach

1

2

5

6

gp1.txt : gp2.txt

gp2.txt : gp3.txt : gp4.txt

/home/xinerd/Desktop/compare code/ Browse...

def GPIndividual1(data):
 predictions = (np.sinh((((np.sinh(data["te
 ((np.cos(((np.round(data["t
 ((np.floor(np.minimum((((
 np.minimum(((np.abs(data["
 ((np.minimum((-1.0), (dat
 np.minimum((((data["team2v
 np.sin(np.sinh((((((-((-((-
 (((((np.sin(data["team1Lma
 np.maximum(np.minimum((c
 np.minimum(np.maximum((c
 ((((-np.abs(np.abs(((data[
 ((3.0 == np.maximum(np.rc
 np.minimum(np.ceil(((data
 ((-(((data["team2Wmin"] >=
 np.floor(np.cos(((data["tea
 np.ceil(((2.674680 <= np.at
 ((np.sinh(np.sin(data["tear
 (((data["team2"] <= (((da
 np.floor(np.cos((data["team
 np.minimum((((np.minimum(
 (((0.591941 <= (data["team2
 (data["team1losses"] * ((da
 np.maximum((((np.minimum(
 np.round((np.round(((data["
 ((np.abs(np.sinh(np.abs(dat
 np.minimum(np.cos(data["t
 np.maximum(np.round(((2.2
 ((data["team2wins"] - np.si
 ((math.tanh((- (1.619430)))

/home/xinerd/Desktop/compare code/ Browse...

def GPIndividual1(data):
 predictions = (np.sinh((((np.sinh(data["te
 ((np.cos(((np.round(data["t
 ((np.floor(np.minimum((((
 np.minimum(((np.abs(data["
 ((np.minimum((-1.0), (dat
 np.minimum((((data["team2v
 np.sin(np.sinh((((((-((-((-
 (((((np.sin(data["team1Lma
 np.maximum(np.minimum((d
 np.minimum(np.maximum((d
 ((((-np.abs(np.abs(((data[
 ((3.0 == np.maximum(np.rc
 np.minimum(np.ceil(((data
 ((-(((data["team2Wmin"] >=
 np.floor(np.cos(((data["tea
 np.ceil(((2.675680 <= np.al
 ((np.sinh(np.sin(data["tear
 (((data["team2"] <= (((da
 np.floor(np.cos((data["tear
 np.minimum((((np.minimum(
 (((0.602941 <= (data["team2
 (data["team1losses"] * ((da
 np.maximum((((np.minimum(
 np.round((np.round(((data["
 ((np.abs(np.sinh(np.abs(dat
 np.minimum(np.cos(data["t
 np.maximum(np.round(((2.2
 ((data["team2wins"] - np.si
 ((math.tanh((- (1.630430)))

/home/xinerd/Desktop/compare code/ Browse...

def GPIndividual1(data):
 predictions = (np.sinh((((np.sinh(data["te
 ((np.cos(((np.round(data["t
 ((np.floor(np.minimum((((
 np.minimum(((np.abs(data["
 ((np.minimum((-1.0), (dat
 np.minimum((((data["team2v
 np.sin(np.sinh((((((-((-((-
 (((((np.sin(data["team1Lma
 np.maximum(np.minimum((d
 np.minimum(np.maximum((d
 ((((-np.abs(np.abs(((data[
 ((3.0 == np.maximum(np.rc
 np.minimum(np.ceil(((data
 ((-(((data["team2Wmin"] >=
 np.floor(np.cos(((data["tea
 np.ceil(((2.675680 <= np.al
 ((np.sinh(np.sin(data["tear
 (((data["team2"] <= (((da
 np.floor(np.cos((data["tear
 np.minimum((((np.minimum(
 (((0.602941 <= (data["team2
 (data["team1losses"] * ((da
 np.maximum((((np.minimum(
 np.round((np.round(((data["
 ((np.abs(np.sinh(np.abs(dat
 np.minimum(np.cos(data["t
 np.maximum(np.round(((2.2
 ((data["team2wins"] - np.si
 ((math.tanh((- (1.630430)))

GP Compare

Max_depth : The maximum depth of the tree.

Max_depth v1: 50

Max_depth v2: 47

Max_depth v3: 40

Extra Trees Classifier Compare

Conclusion

Model: **FBA**

Other Features:

- Player Profile
- Judge
- Peripheral Odds (Gambling)

Further Work

- Random Forest Classifier
- Neural Network

Conclusion

Model: **FBA** (Failure Based Algorithm)

Other Features:

- Player Profile
- Judge
- Peripheral Odds (Gambling)

Further Work

- Random Forest Classifier
- Neural Network

Reference

1. [Kaggle March Machine Learning Mania 2016](#)
2. Lopez, Michael J. and Gregory J. Matthews. 2015. Building an NCAA men's basketball predictive model and quantifying its success. Journal of Quantitative Analysis in Sports. 11(1): 5-12. Retrieved 28 Apr. 2016, from doi: 10.1515/jqas-2014-0058
3. [Genetic Programming](#)
4. [Elo rating system](#)

Python Library

Used Python Library

- **Sklearn**
- Numpi
- Scipi
- Pandas
- subprocess
- Lasagne
- Theano
- Seaborn

Thank You.