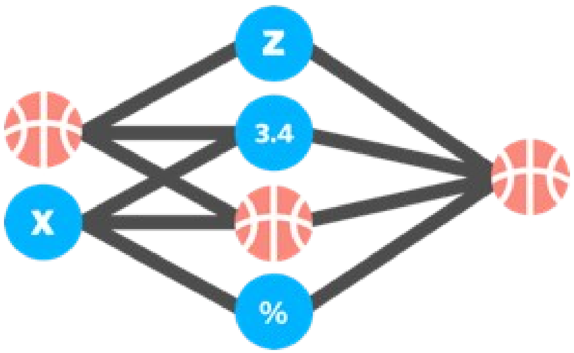


Can March Madness be Tamed with Math?

An exploration of constructing a probabilistic NCAA tournament bracket.

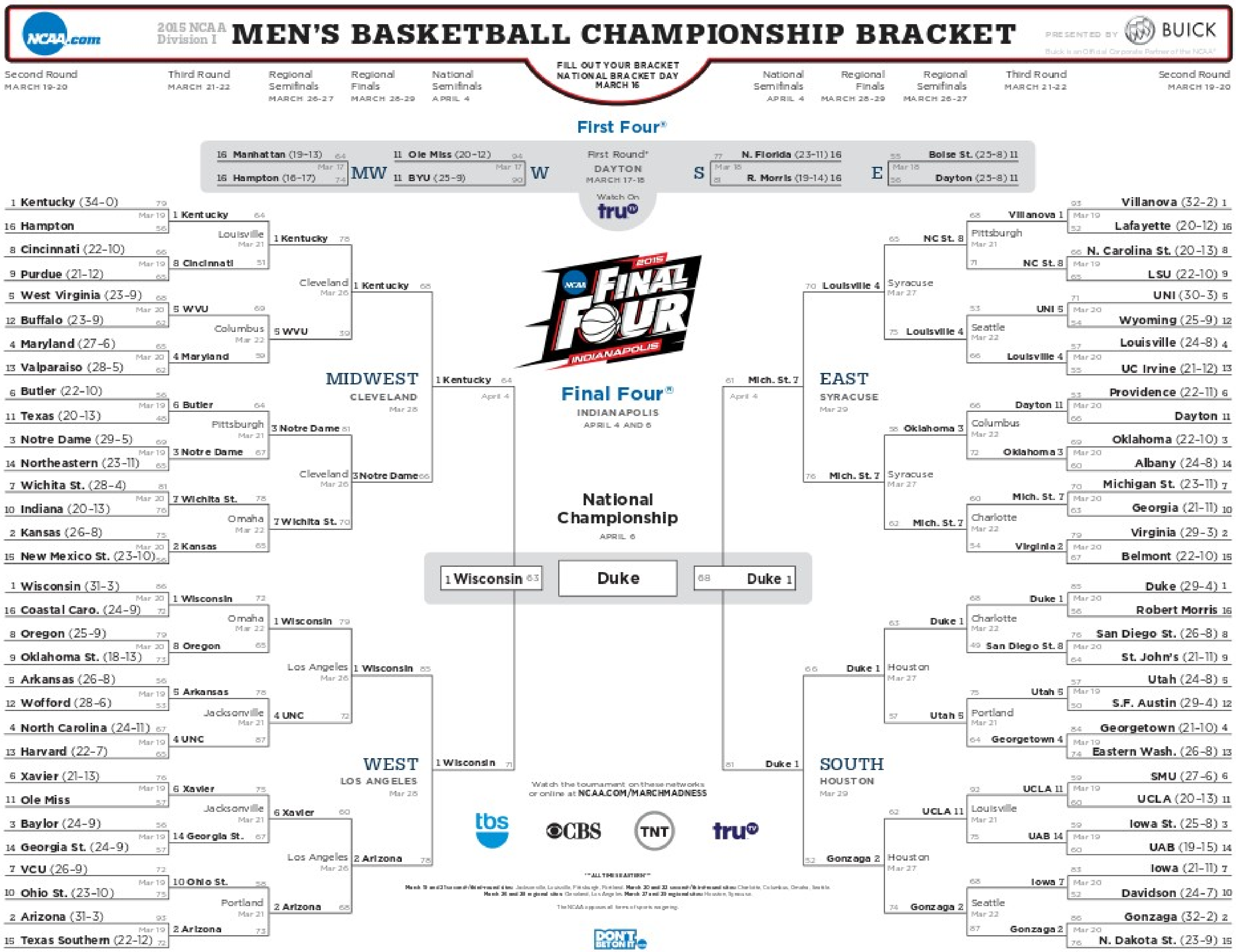
baltimore

Baltimore-python
4/15/2015
Daniel O'Neill

Overview

- Background on Competition
- Benchmarks and Early Models
- Refining the Models
- Tracking the Competition
- Results
- Take-aways/Future Work/Recommendations
- Questions

NCAA College Tournament

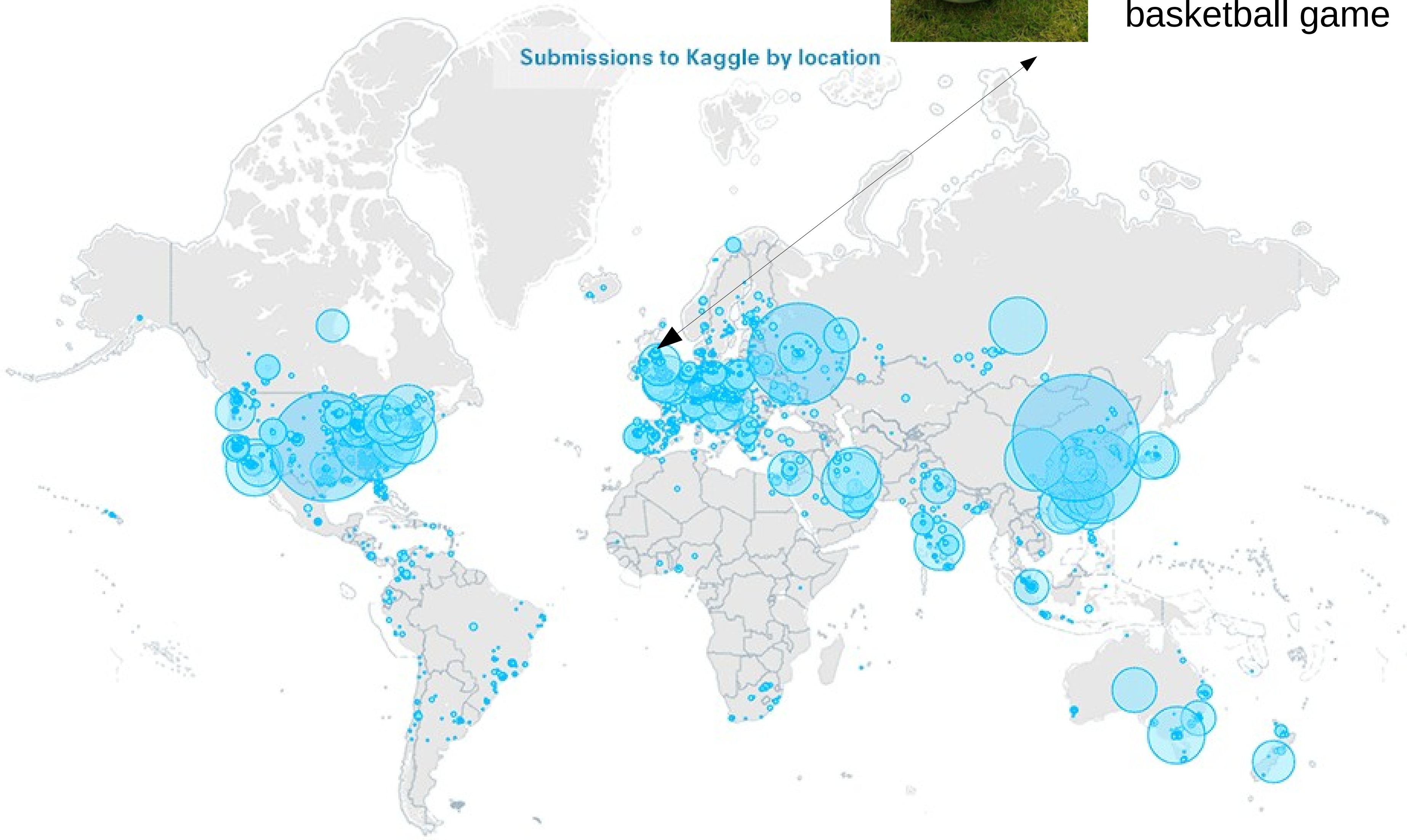


Machine Learning Mania 2015

kaggle

- www.kaggle.com
- Founded in 2010
- Over 200,000 kagglers
- Devoted to a community of data scientists and learners solving data science problems
- Competition and Learning

Bluefool:
Kaggle Master that had never seen a basketball game



Rules of the Game

- Provide Probabilities of Team A beating Team B for every possible combination (68*67/2=2278)
- Scored on Games that Actually Happen (63)
- All games weighted equally
- Lowest LogLoss wins!
- Correctly predict a game with 80% :0.22
- Incorrectly predict a game with 80%:1.61

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

Data Provided

- Stage 1 (Feb 2-Selection Sunday):
  - Team Name File (Mapped team name to number)
  - Regular Season / Tourney results (1985-2014)
  - Regular Season / Tourney detailed results (2003-2014)
  - Tourney Seeds
  - Tourney Slot
  - Evaluated on 2010-2014 tournaments
- Stage 2 (Submissions due Day before Tourney):
  - 2015 Versions of data
  - Follow leader board live
- Additional Data: Massey Ordinals, Vegas Spreads, www.kenpom.com

My Philosophy on the Competition

- There is an element of luck
  - Sample size is very small for each prediction (1 game) True probabilities are never evaluated
  - Probability of winning better than ESPN tournament
- Submission does not have to be great
  - Has to be reasonable
  - Want it to be slightly different than a standard approach to separate from other competitors
- Wanted to learn something

Overview

- Background on Competition
- Model Building
- Tracking the Competition
- Results
- Take-aways/Future Work/Recommendations
- Questions

Pandas

Python Data Analysis Library

- Pandas is convenient for both live exploratory analysis and for incorporating into scripts
- Uses DataFrame as primary object
- Comparable to R
- Allows for quick selection and sub-selection of data
- Column names add to readability
- Easy to aggregate

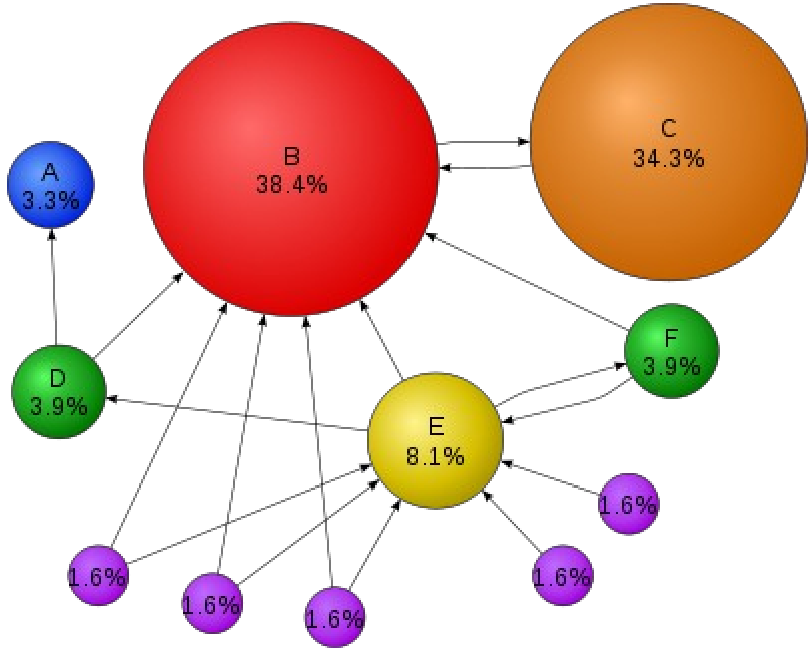
- reg_season=pd.read_csv('regular_season_compact_results.csv')
- train=reg_season[(reg_season.season>2000)&(reg_season.season<2010)]
- train['win_pct']=1.*train['wins']/(1.*train['wins']+1.*train['losses'])
- combined=pd.merge(results,seeds,how='inner',left_on=['wteam','season'],right_on=['team','season'])
- Describe()
- Head(), tail()
- Sorting

Initial Benchmarks

- Data was loaded into a Pandas dataframe
- Submission file was created using Python itertools to find every combination of two teams
- Lower number team placed first and then the probability of Team A beating Team B would be calculated
- Checked with a 0.5 submission for all games
- Moved to a linear model based on seeds

PageRank

- Weights items based on their relative importance
- Originally used in links for weighting webpages and the probability of a user moving to that page
- Incorporates a level of randomness to ensure that the model does not spiral away from convergence



$$M = (1-p)\cdot A + \ p \cdot B \qquad B = \frac{1}{n}\cdot\begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$M^k z$ converges to the vector v*

Applying it to Basketball

- All teams are a node in the graph
- Each game is an edge
- The losing team points to the winning team
  - Later iterations of my model used score differential to weight the edges
  - The score differential was adjusted to reflect home or away teams
  - Took the log of the difference to adjust for some scaling problems

Adjusting the Randomness

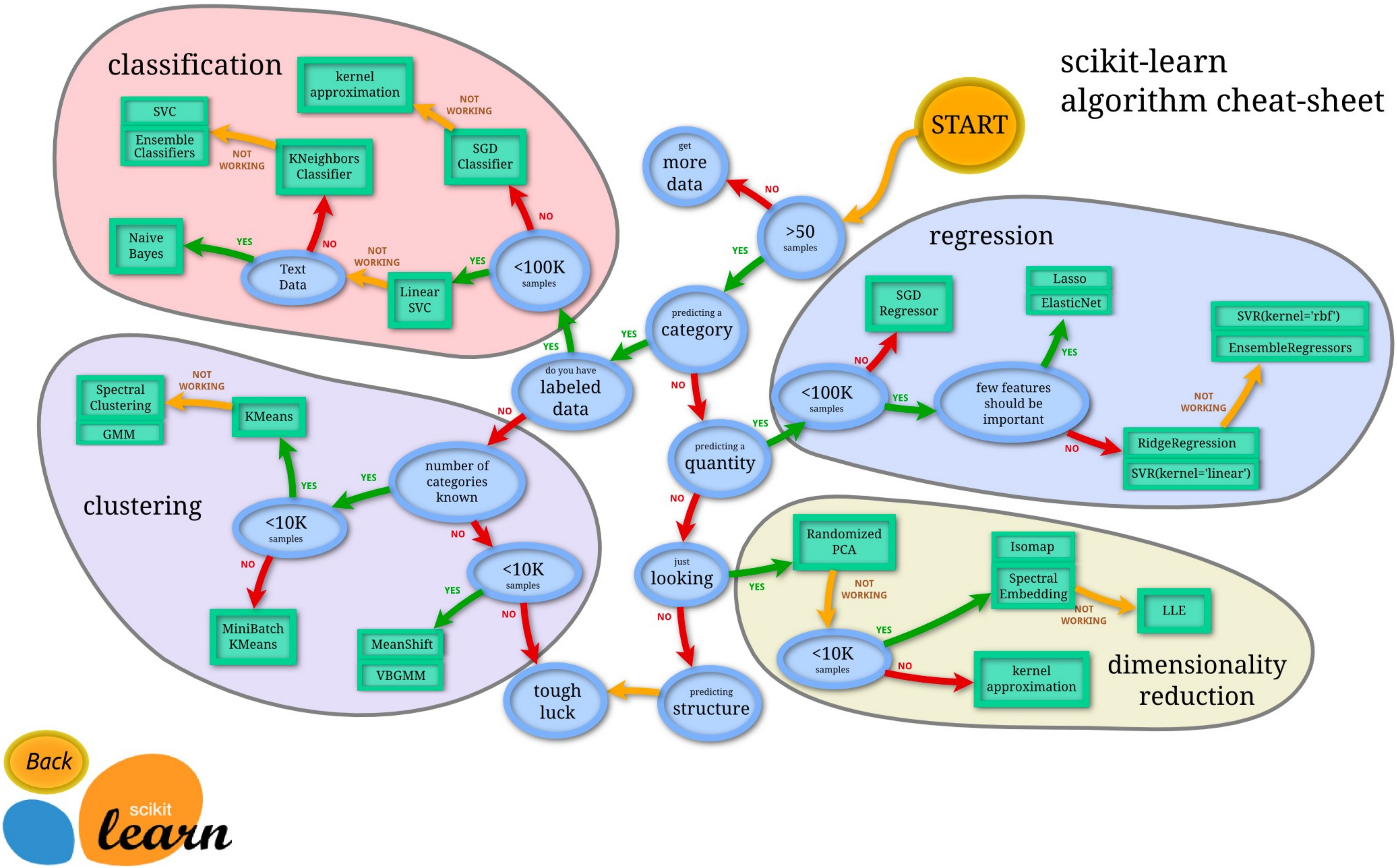
- Weight of the matrix that all teams have equal importance was given more probability in increments until a LogLoss on the test data was achieved
- Ended up close to only using roughly 10% of the weighted matrix. This demonstrated that all teams are fairly equal
- Rated Miami high because it beat Duke by a large differential. Did not make tournament.

95.00%	5.00%
Duke	Duke
Kentucky	Notre Dame
Wisconsin	NC State
Virginia	Virginia
Gonzaga	North Carolina
North Carolina	Kentucky
Villanova	Miami FL
Arizona	Georgetown
Kansas	Villanova
VA Commonwealth	Kansas
Ohio St	Wisconsin
Baylor	Butler
Wichita St	Oklahoma
Notre Dame	Maryland
Northern Iowa	Providence
BYU	BYU
Arkansas	Baylor
Maryland	Syracuse
Cincinnati	Iowa St
Utah	West Virginia

SciKit Learn

Machine Learning Library built on numpy scipy and matplotlib

- Extensive Machine Learning library for Python
- Tools for Data Splitting
  - cross_validation train_test_split, Kfold, StratifiedKfold
- Data Preprocessing
  - PCA, Data Scaling, BoxCox
- Classification-Nearest neighbors, randomforest,GBM
- Regression-Linear, Logistic, Ridge Lasso
- Model Selection: GridSearch, RandomGridSearch
- Well documented
- All models follow similar structure of fit and predict
  - Ensembling advantages



http://peekaboo-vision.blogspot.com/2013/01/machine-learning-cheat-sheet-for-scikit.html

Constructing Models

- All data provided was given in terms of the winning team
- Binary Classification
- All data was doubled and inversed
  - Seed difference was multiplied by -1
  - All team attributes were flipped

Primarily Used Linear and Logistic Regression

- Created Several Models Based on a Linear or Logistic Framework
- Averaged them to reduce extremes
- No regularization was used throughout


1 Villanova
16 Lafayette
8 NC State
9 LSU
5 Northern Iowa
12 Wyoming
4 Louisville
13 UC Irvine
6 Providence .51
11 Boise St .49
3 Oklahoma
14 Albany NY
7 Michigan St
10 Georgia
2 Virginia
15 Belmont
1 Villanova .68
8 NC State .32
5 Northern Iowa .46
4 Louisville .54
6 Providence .35
3 Oklahoma .65
7 Michigan St .06
2 Virginia .94
1 Villanova .61
4 Louisville .39
3 Oklahoma .32
2 Virginia .68
1 Villanova .38
2 Virginia .62
2 Virginia .59
1 Duke .99
16 North Florida .01
8 San Diego St
9 St John's
5 Utah
12 SF Austin
4 Georgetown
13 E Washington
6 SMU
11 UCLA
3 Iowa St
14 UAB
7 Iowa
10 Davidson
2 Gonzaga
15 N Dakota St
1 Duke .59
8 San Diego St .41
5 Utah .54
4 Georgetown .46
11 UCLA .56
14 UAB .44
7 Iowa .54
2 Gonzaga .46
1 Duke .65
5 Utah .35
11 UCLA .29
7 Iowa .71
1 Duke .65
7 Iowa .35
1 Duke .41
2 Virginia .41
Kentucky
1 Kentucky .59
1 Kentucky .99
16 Manhattan .01
8 Cincinnati
9 Purdue
5 West Virginia
12 Buffalo
4 Maryland
13 Valparaiso
6 Butler
11 Texas
3 Notre Dame
14 Northeastern
7 Wichita St
10 Indiana
2 Kansas
15 New Mexico St
1 Kentucky .89
8 Cincinnati .11
5 West Virginia .44
4 Maryland .56
6 Butler .52
3 Notre Dame .48
7 Wichita St .45
2 Kansas .55
1 Kentucky .85
4 Maryland .15
6 Butler .45
2 Kansas .55
1 Kentucky .74
2 Kansas .26
1 Kentucky .67
1 Wisconsin
16 Coastal Car
8 Oregon
9 Oklahoma St
5 Arkansas
12 Wofford
4 North Carolina
13 Harvard
6 Xavier
11 Mississippi
3 Baylor
14 Georgia St
7 VA Commonwealt
10 Ohio St
2 Arizona
15 TX Southern
1 Wisconsin .78
8 Oregon .22
5 Arkansas .43
4 North Carolina .57
6 Xavier .45
14 Georgia St .55
10 Ohio St .35
2 Arizona .65
1 Wisconsin .61
4 North Carolina .39
14 Georgia St .24
2 Arizona .76
1 Wisconsin .57
2 Arizona .43
1 Wisconsin .33

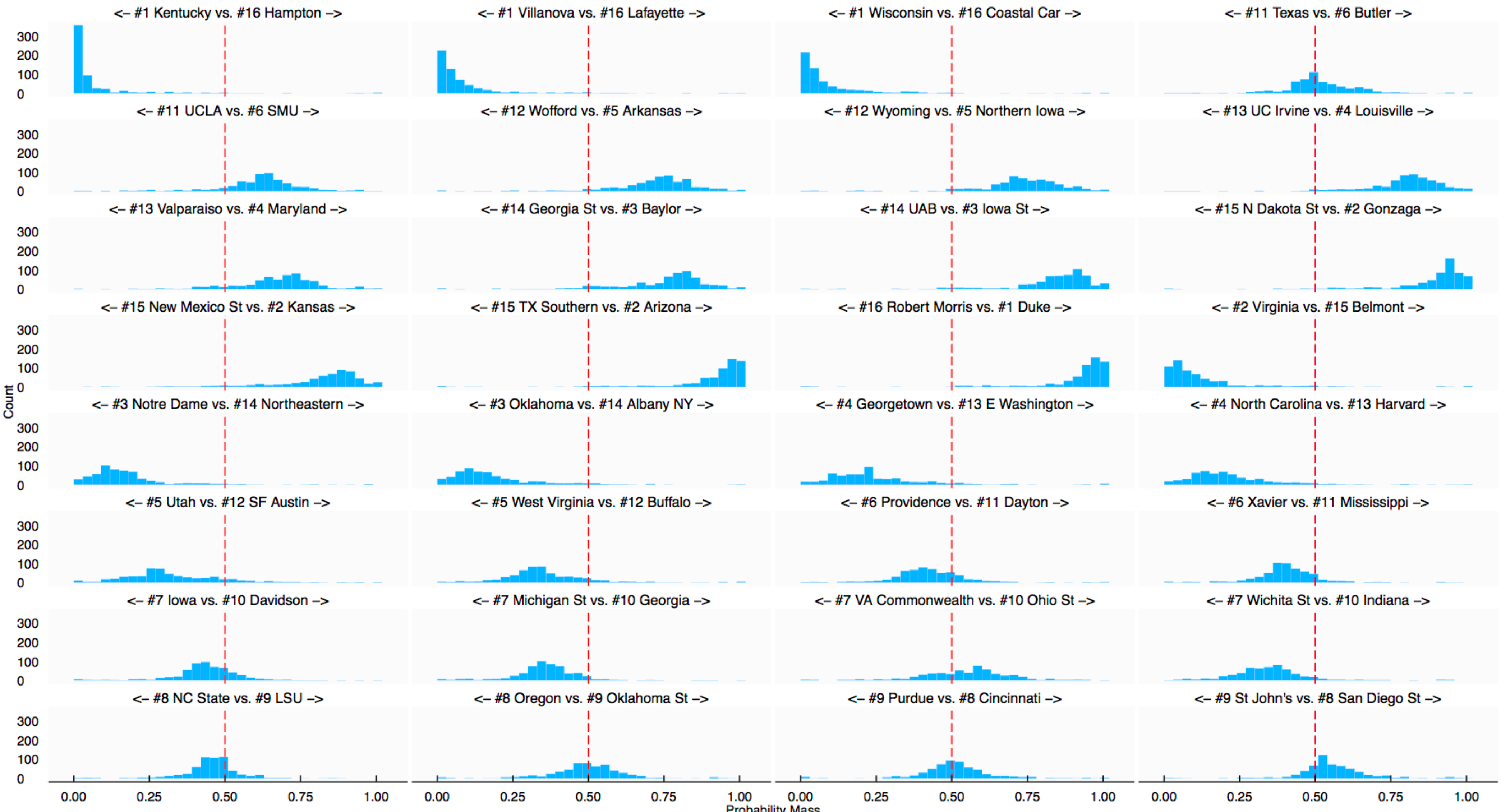
Overview

- Background on Competition
- Benchmarks and Early Models
- Model Building
- Tracking the Competition
- Results
- Take-aways/Future Work/Recommendations
- Questions

More Madness!

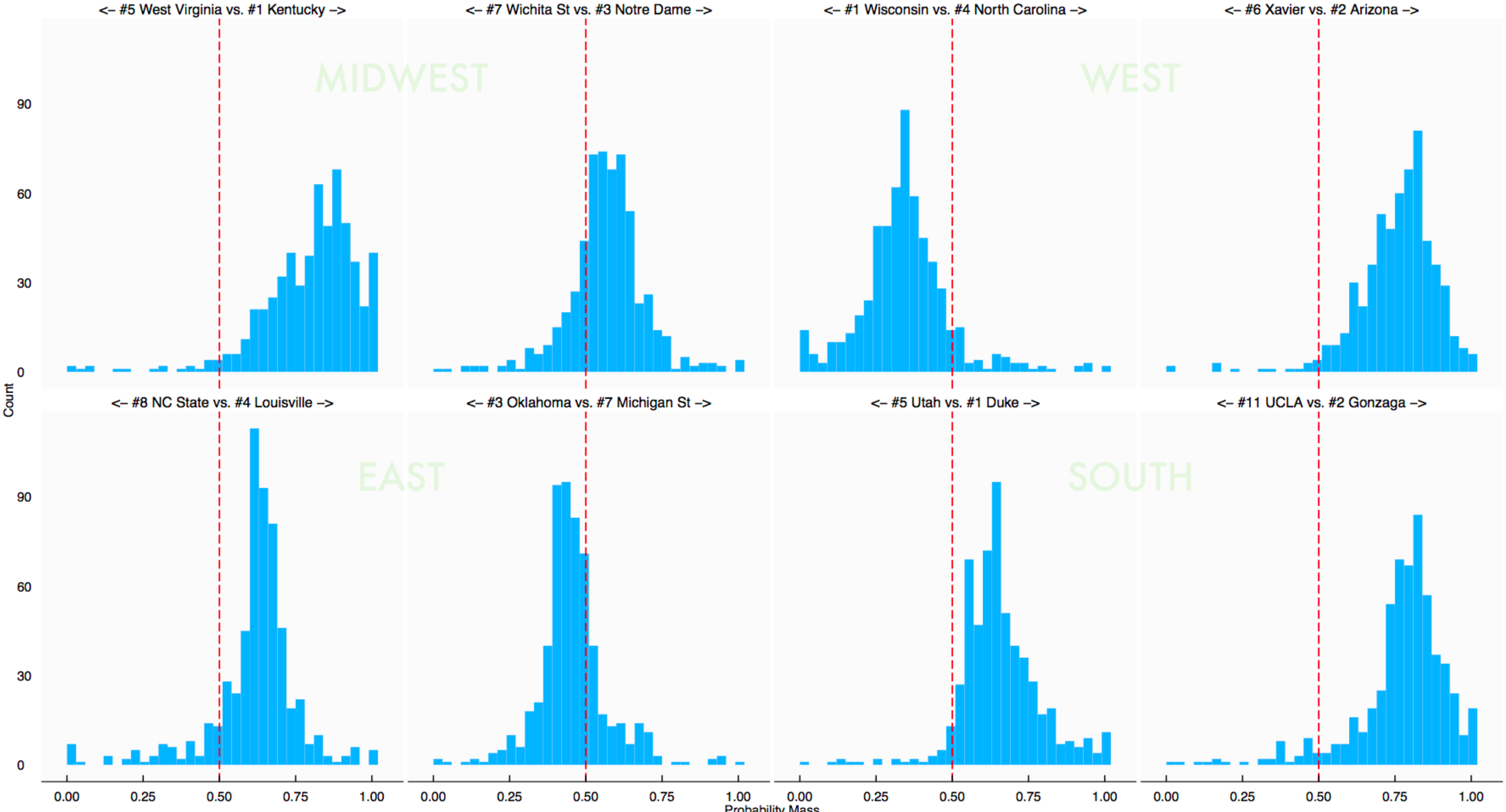
- NetProphet
  - http://netprophetblog.blogspot.com/
- Released his submission with random noise incorporated
- BlueFool and others incorporated his model
- One contestant copied his submission exactly and then gave Kentucky a 100% chance of victory in all games

March Machine Learning Mania (Round of 64 Predictions)



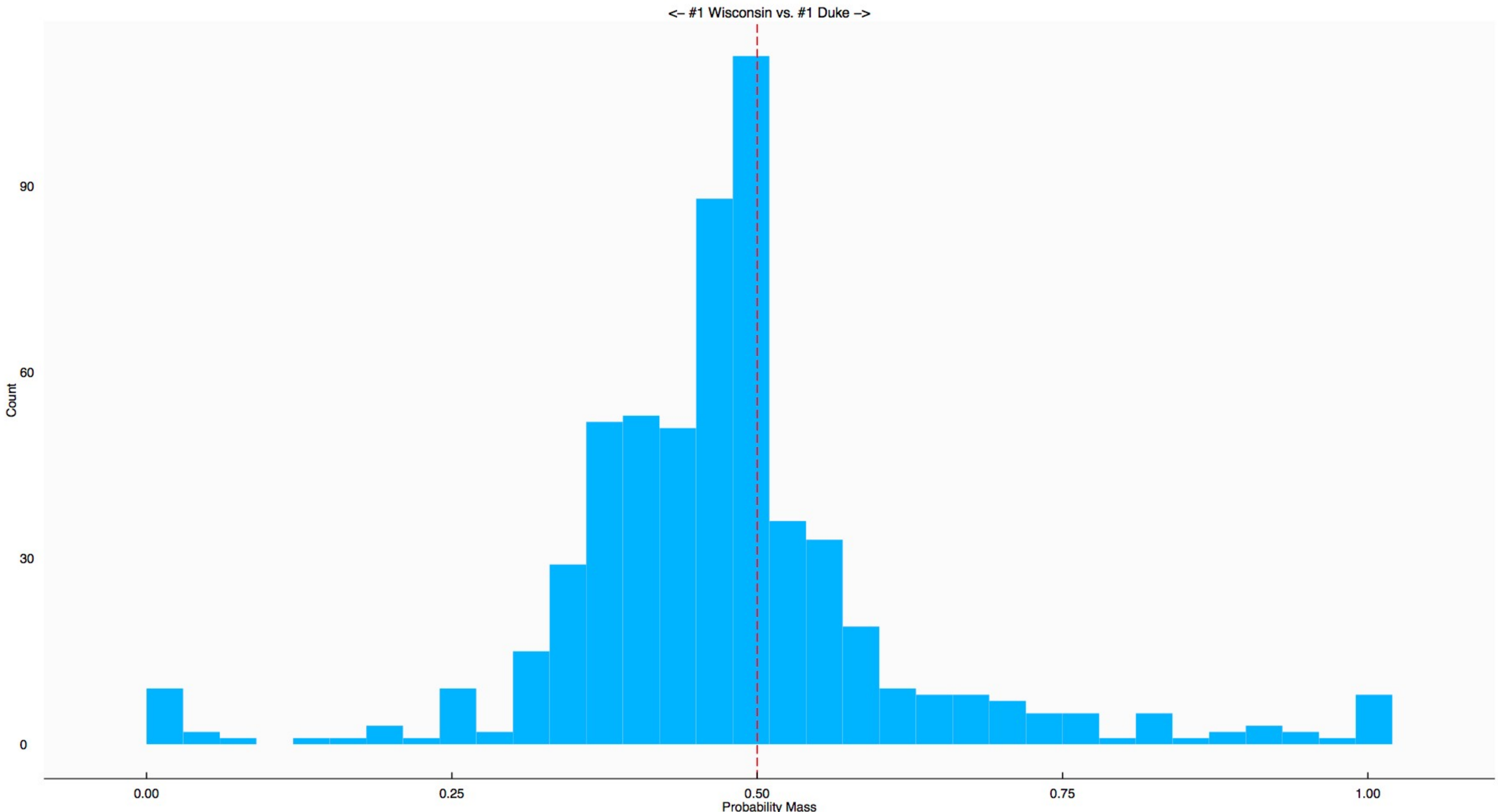
www.kaggle.com/c/march-machine-learning-mania-2015

March Machine Learning Mania (Round of 16 Predictions)



www.kaggle.com/c/march-machine-learning-mania-2015

March Machine Learning Mania (Championship Prediction)



www.kaggle.com/c/march-machine-learning-mania-2015

Overview

- Background on Competition
- Benchmarks and Early Models
- Model Building
- Tracking the Competition
- Results
- Take-aways/Future Work/Recommendations
- Questions

Results

- After Day 2 I was 18th out of 341. Feeling good
- At the end I was 216th. Glad I don't bet
- The winner had some odd predictions
  - 100% chance that #14 Georgia St would beat #3 GA
- BlueFool only trained on tournament data and was able to come in 8th

Overview

- Background on Competition
- Benchmarks and Early Models
- Model Building
- Tracking the Competition
- Results
- Take-aways/Future Work/Recommendations
- Questions

Take Aways

- There is a level of gamesmanship with such a small sample size
- Favorites do win but all teams are playing at a high level
- Last year's winners ran simulations of how many times there model would win and found almost the entire board had a shot

Future Models

- Incorporate Coach data
- Perform more feature analysis and possible feature engineering
- Weight the later part of the season
- Focus more on blending models
- Incorporate more ranking systems

Favorite Resources on Data Science

- Applied Predictive Modeling- Max Kuhn, Kjell Johnson
- The LION Way: Learning plus Intelligent Optimization- Mauro Brunato, Roberto Battiti
- Podcasts: Talking Machines, O'Reilly Data Show
- Coursera
- edX

Questions?