



2019 NCAA March Madness Prediction Contest

John Ahlfield

Thinkful Supervised Learning Capstone

<https://www.kaggle.com/c/mens-machine-learning-competition-2019/overview>





Objective

- Create the best prediction for the 2019 NCAA Men's Basketball Tournament from historical data provided by the NCAA
- Make probabilistic predictions for every possible matchup
 - 68 total teams: 2278 total matchup possibilities





Evaluation

- LogLoss:

Submissions are scored on the log loss:

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

where

- n is the number of games played
- \hat{y}_i is the predicted probability of team 1 beating team 2
- y_i is 1 if team 1 wins, 0 if team 2 wins
- $\log()$ is the natural (base e) logarithm





Data

- Historical game data from 2003-2018 regular seasons
- Large, robust dataset
 - 82041 games
 - No null values
- Indexed by individual game

df.head()																					
	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT	WFGM	WFGA	...	LFGA3	LFTM	LFTA	LOR	LDR	LAst	LTO	LStl	LBlk	LPF
0	2003	10	1104	68	1328	62	N	0	27	58	...	10	16	22	10	22	8	18	9	2	20
1	2003	10	1272	70	1393	63	N	0	26	62	...	24	9	20	20	25	7	12	8	6	16
2	2003	11	1266	73	1437	61	N	0	24	58	...	26	14	23	31	22	9	12	2	5	23
3	2003	11	1296	56	1457	50	N	0	18	38	...	22	8	15	17	20	9	19	4	3	23
4	2003	11	1400	77	1208	71	N	0	30	61	...	16	17	27	21	15	12	10	7	1	14

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 82041 entries, 0 to 82040
Data columns (total 34 columns):
Season      82041 non-null int64
DayNum      82041 non-null int64
WTeamID     82041 non-null int64
WScore      82041 non-null int64
LTeamID     82041 non-null int64
LScore      82041 non-null int64
WLoc        82041 non-null object
NumOT       82041 non-null int64
WFGM        82041 non-null int64
WFGA        82041 non-null int64
WFGM3       82041 non-null int64
WFGA3       82041 non-null int64
WFTM        82041 non-null int64
WFTA        82041 non-null int64
WOR         82041 non-null int64
WDR         82041 non-null int64
WAst        82041 non-null int64
WTO         82041 non-null int64
WStl        82041 non-null int64
WBlk        82041 non-null int64
WPF         82041 non-null int64
LFGM        82041 non-null int64
LFGA        82041 non-null int64
LFGM3       82041 non-null int64
LFGA3       82041 non-null int64
LFTM        82041 non-null int64
LFTA        82041 non-null int64
LOR         82041 non-null int64
LDR         82041 non-null int64
LAst        82041 non-null int64
LTO         82041 non-null int64
LStl        82041 non-null int64
LBlk        82041 non-null int64
LPF         82041 non-null int64
dtypes: int64(33), object(1)
memory usage: 21.3+ MB
```




Feature Creation

- Utilize net values rather than totals
 - More important to know how one team directly compares to its opponent

```
training_set.head()
```

	net_fgm	net_fga	net_fgm3	net_fga3	net_ftm	net_fta	net_or	net_dr	net_tr	net_ast	net_to	net_stl	net_blk	net_pf	win
0	5	5	1	4	-5	-4	4	2	6	5	5	-2	-1	2	1
1	2	-5	2	-4	1	-1	-5	3	-2	9	1	-4	-2	2	1
2	2	-15	5	-8	3	6	-14	4	-10	6	-2	3	-3	2	1
3	0	-11	-3	-13	9	16	-11	-1	-12	2	-7	10	-1	-5	1
4	6	-1	0	-2	-6	-14	-4	7	3	0	4	-3	3	6	1





Correcting Imbalance

- Only have data from the winning team's perspective

```
inverse_df = ~training_set  
inverse_df['win'] = 0  
inverse_df.head()
```

	net_fgm	net_fga	net_fgm3	net_fga3	net_ftm	net_fta	net_or	net_dr	net_tr	net_ast	net_to	net_stl	net_blk	net_pf	win
0	-5	-5	-1	-4	5	4	-4	-2	-6	-5	-5	2	1	-2	0
1	-2	5	-2	4	-1	1	5	-3	2	-9	-1	4	2	-2	0
2	-2	15	-5	8	-3	-6	14	-4	10	-6	2	-3	3	-2	0
3	0	11	3	13	-9	-16	11	1	12	-2	7	-10	1	5	0
4	-6	1	0	2	6	14	4	-7	-3	0	-4	3	-3	-6	0

- Append to previous dataframe to create the final training set





Creating the Test Set

- To predict this year’s tournament, only use data from this season
 - Yearly performance highly variable due to roster turnover in collegiate sports
- For each team, find their average (per game) net in each stat

	net_fgm	net_fga	net_fgm3	net_fga3	net_ftm	net_fta	net_or	net_dr	net_tr	net_ast	net_to	net_stl	net_blk	net_pf
teamid														
1101	0.740741	2.185185	-0.148148	0.222222	-3.481481	-4.777778	-0.888889	-1.814815	-2.703704	1.851852	-1.074074	0.629630	0.703704	3.370370
1102	-2.482759	1.655172	-1.344828	1.068966	0.275862	-0.206897	0.655172	-3.344828	-2.689655	-0.689655	-1.206897	0.862069	-0.965517	-0.586207
1103	-1.451613	1.193548	1.290323	4.032258	-3.290323	-3.870968	0.258065	-1.935484	-1.677419	-0.419355	0.838710	-0.419355	-1.258065	2.645161
1104	1.117647	-3.352941	-0.558824	-2.058824	0.705882	1.323529	-1.558824	1.764706	0.205882	0.911765	0.970588	-0.176471	1.705882	-0.970588
1105	-6.709677	-4.741935	-1.483871	0.193548	-1.032258	-0.516129	-0.258065	-2.322581	-2.580645	-3.870968	5.193548	-3.677419	-3.290323	-0.741935





Test Set by Matchup

- For each pair of teams in the tournament (68 teams), find the difference in their average net stats
 - This is the final test set that will be used for prediction

	team1	team2	net_fgm	net_fga	net_fgm3	net_fga3	net_ftm	net_fta	net_or	net_dr	net_tr	net_ast	net_to	net_stl	net_blk
0	1101	1113	-0.743130	2.249701	-0.857826	-0.261649	-7.997611	-10.519713	0.433692	-1.782557	-1.348865	2.819594	3.151732	-0.563919	0.639188
1	1101	1120	-1.821759	-1.814815	-1.835648	-3.621528	-6.731481	-7.246528	-2.763889	-3.283565	-6.047454	0.226852	1.925926	-0.651620	-1.421296
2	1101	1124	-1.968937	2.507766	1.174432	5.222222	-4.513740	-5.584229	-1.792115	-6.363202	-8.155317	1.948626	-2.783751	0.726404	0.768220
3	1101	1125	-1.228956	4.912458	-4.693603	-9.626263	-3.420875	-4.656566	0.444444	-5.117845	-4.673401	-2.663300	-1.922559	1.872054	0.946128
4	1101	1133	-0.353009	2.153935	0.726852	4.159722	-3.887731	-5.965278	-1.420139	-3.221065	-4.641204	1.726852	-0.886574	0.473380	-0.640046





Testing!



- Bernoulli Naïve Bayes
- Linear Regression
 - OLS, Ridge, Lasso, ElasticNet
- K-Nearest Neighbors
 - Classifier vs. regressor, weighted vs. unweighted
- Random Forest
 - Classifier vs. regressor, singular tree vs. forest, vary feature number and tree depth
- Support Vector Machines
 - Classifier vs. regressor
- Gradient Boosting
 - Classifier vs. regressor, vary # of iterations and tree depth





BNB, Linear Regression Results

- BNB performs terribly
 - LogLoss evaluation heavily penalizes classifiers
- Linear models all performed similarly
- Reference score: 0.69314
 - 0.5 for all entries
- Winning score: 0.41477

Model	Score
Bernoulli Naïve Bayes	10.41658
Ordinary Least Squares	0.55828
Ridge Regression	0.55813
Lasso Regression	0.55814
ElasticNet Regression	0.55814





K-Nearest Neighbors Results



KNN-C, unweighted	Score
K = 1	9.86836
K = 5	9.32010
K = 10	9.32010
K = 20	9.32010

KNN-R, weighted	Score
K = 1	9.86836
K = 5	4.11915
K = 10	2.57424
K = 20	1.58433

KNN-R, unweighted	Score
K = 1	9.86836
K = 5	4.11535
K = 10	2.57020
K = 20	1.58347
K = 50	1.11713
K = 100	1.08257
K = 250	1.05742
K = 500	1.04405
K = 1000	0.59362
K = 2500	0.57171
K = 5000	0.56129
K = 10000	0.55682
K = 25000	0.56637
K = 50000	0.58977

KNN-C, weighted	Score
K = 1	9.86836
K = 5	9.32010
K = 10	8.22363
K = 20	9.32010

- Weighted superior for classifier
 - Inferior for regressor...why?





Decision Tree Results

DT-C (features, depth)	Score
1, 5	14.80259
1, 10	9.32010
1, 25	8.22361
5, 5	10.41660
5, 10	10.96480
5, 25	8.22364

DT-R (features, depth)	Score
1, 5	0.70718
1, 10	0.75809
1, 25	9.86833
5, 5	0.76421
5, 10	3.50208
5, 25	8.77186

- Fewer features superior
- Regression models deteriorate as tree depth increases





Random Forest Results

RF-C (features, depth)	Score
1, 5	9.86834
1, 10	10.96481
1, 25	9.32008
5, 5	10.41657
5, 10	9.86834
5, 25	9.86834

RF-R (features, depth)	Score
1, 5	0.61797
1, 10	0.59822
1, 25	2.55253
5, 5	0.61765
5, 10	0.74119
5, 25	5.20347

- Forest classifier provides mixed results compared to single tree
- Forest regressor is superior to single tree





Support Vector Machine Results

Support Vector Machine	Score
Classifier	10.41658
Regressor	2.67314

- Highly inefficient for this dataset
 - Training set ~160k points
 - Regressor took more than an hour!





Gradient Boosting Results

GB-C (depth, iterations)	Score
2, 100	10.41658
2, 500	9.86834
3, 100	9.86834
3, 500	9.86834

GB-R (depth, iterations)	Score
2, 100	1.06083
2, 500	1.05566
3, 100	1.07425
3, 500	1.09745

- Poor performance regardless of parametrization





Summary

- Each model produces relatively similar overall brackets
 - Competition scoring does not propagate errors
- Strongly overvalues teams with weaker schedules
- Improve by adding in team rankings

BNB	Score
1 st round	24
2 nd round	10
Sweet 16	5
Elite 8	1
Final 4	0
NCG	0

DT-C (1, 25)	Score
1 st round	26
2 nd round	9
Sweet 16	5
Elite 8	0
Final 4	0
NCG	0

KNN-R (10k)	Score
1 st round	23
2 nd round	10
Sweet 16	5
Elite 8	1
Final 4	0
NCG	0

