March Machine Learning Mania 2016

Xin Ming
Chenxi Zhang
Zekai Hou
University of Massachusetts Lowell

***Abstract*— Our goal here is to predict the 2016 NCAA Basketball Tournament. The data given includes their past compact results and the tourney slots. Our method takes in those data as inputs and find the probabilities that one team beat another in every match of the 2016 tournament.**

Keywords - NCAA; kaggle; machine learning;

I. Introduction

Prediction towards various kinds of games is always one of the most popular topic nowadays. No matter in the discuss between sports fans or the experts forecast in lottery industry. For fans, being accurate in prediction is either a way to show off that you are an expert of the field, or commonly, they are eager to win the prize pool in lottery. So for the lottery industry, an accurate prediction will help them to earn innumerable profits from the result. As a result, they are willing to pay to find a good model, and hire actuaries with high salary.

Since 1939, the best colleges and universities across the United States have participated in a yearly tournament called the NCAA Mens Basketball Championship. This basketball tournament has become one of the most popular and famous sporting tournaments in the United States. Each year, millions of people fill out a bracket to predict the outcome of the popular mens college basketball tournament that tips off in March.

Kaggle, a website that organizes free analytic and modeling contests, released a contest named "March Machine Learning Mania 2016". Till now, there are 598 teams in this contest. The prize is $25 thousand. Although the contest is ended, we can still make submissions to see our score. Our best is No.286 on the leader board, a little far from the leader due to our limited knowledge.

Obviously, it is extremely hard to find a model that can perfectly fit the prediction (otherwise we are all millionaires). Finding a perfect model may be impossible, but we can trying to get close to the truth. Different method will lead to different accuracy, so it is important to find out which one would be the best. Our final approach is ELO rating system+Extra Trees Classifier+GPIndividual, from which we gain the score mentioned above. We also listed all the method we tried, almost half of them are inefficient since they are below the "All 0.5 Benchmark".

A simple way to improve the model is trying to combine the method weve tried. Find out advantages in all method and build up a new model with them. We did that and luckily it seemed work. In the approach part we will illustrate our method in detail.

II. Background

A well-known approach is the Las Vegas point spread. It is a number provides the predicted difference in total points scored between the visiting and the home team: a spread of -5.5, that means the home team is expected to win by 5.5 points. To win a wager placed on a 5.5 point favorite, one would need that squad to win by six points or more. Meanwhile, a bet on the underdog at that same point spread would win either if the underdog loses by 5 points or fewer, thereby covering the spread, or if the underdog wins. In one word, the point spread accounts for all pregame factors which might determine the games outcome, including relative team strength, injuries, and location.

Kenneth Deakins and his team tried two techniques-Support Vector Machine (SVM) and K-Nearest Neighbors (KNN)- ended up working best for attempting to predict March Madness tournament results using regular season statistics. their individual game prediction accuracy for KNN was 69.68% of games predicted correctly, and SVM had an overall individual game prediction of 70.09%. KNN is a non-parametric method used for classification and regression. SVMs are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

As for the method to predict, there are so many. For example, extra trees classifier, ELO, neural network, decision tree and so on. Even linear regression is a feasible way. We also read what other competitors do to solve the task. However most of which they posted are really inaccurate, some of the prediction they made are even below the All 0.5 Benchmark, which means just set every value of possibility to 0.5. So we decide to combine their work and try to give out our achievement.

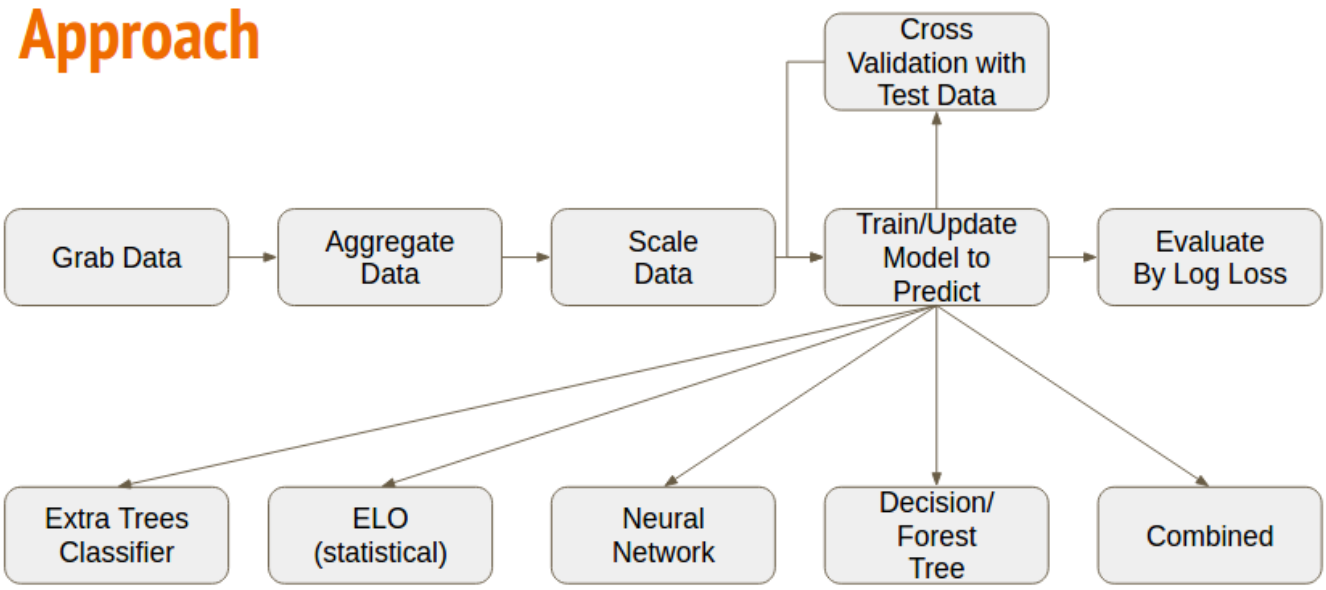


Figure 1

III. Approach

The general approach works like what showed in figure 1. Firstly we load the data by using pandas, then we also use pandas to merge the data to what we need, after we put all data together, we use StandardScaler to scale the data into a suitable range. Then we try different model like extra trees classifier, in order to find the best parameters for the model, we use the training data and labeled result to train the model, then adjust the parameters in order to get a good log loss value. In this scenario, we change the max depth of the tree to smaller value so as to get a better result. The maximum depth of the tree stands for the maximum level that a tree expanded, If it is none, then nodes are expanded until all leaves are pure or until all leaves contain less than min samples split samples. By updating it to smaller value, we observed that the log loss becomes smaller too which means it yield a better result. From rank 2 to rank 4 in Figure 2 shows the result corresponding to value of max depth from 50, 47, 40.

#	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

Figure 2

Then we tried to use extra trees classifier with ELO and Genetic programming together. The Elo rating system is a method for calculating the relative skill levels of players in competitor-versus-competitor games such as chess, also in other games like football, basketball etc. although elo is a statistical method to evaluate a player, the result of a team, which is a scores, can contribute to a new feature and affect the final result by using other ML model. Assume we have ranks for two teams, RA and RB, so the expectation probability that team A defeated team B is :

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}.$$

If a team get scores SA in the real game (win = 1, draw = 0.5, lose = 0) is different with the expectation probability, then the rank of this team should be adjusted by using following function.

$$R'_A = R_A + K(S_A - E_A).$$

K is normally set to 16 or 32 in master game, if team A (score is 1613) has a draw with the team B (score is 1573), then EA is

$$\frac{1}{1 + 10^{(1573-1613)/400}}$$

approximately equals to 0.5573, and the new rank of A becomes to 1613+32(0.5 0.5573) = 1611.166 This could be a useful feature to evaluate the teams, higher rank has more chance to defeat lower rank team. Then finally we utilize a genetic programming model trained by Reza from Kaggle, it will gave us the initial predictions about probability of team matches, then we still use extra trees classifier to calculate the final result.

IV. Data set

The Data that we had in this competition for prediction was provided by Kaggle, the Dataset was organized into 9 csv file:

- RegularSeasonCompactResults.csv: This file contains the Compact information(win score, lose score.etc) of the NCAA regular season from 1985 to 2016.
- RegularSeasonDetailedResults.csv: This file contains the Detailed information(field goals, free throws.etc) of the NCAA regular season from 1985 to 2016.
- TourneyCompactResults.csv: This file contains the Compact information(win score, lose score.etc) of the NCAA March-Madness from 1985 to 2015.
- TourneyDetailedResults.csv: This file contains the Detailed information(field goals, free throws.etc) of the NCAA March-Madness from 1985 to 2015.
- Team.csv: This file contains the name and the ID of each team such as: Team id:1102 Team name:Air Force.
- Seasons.csv:This file contains the basic information of every season such as the Region
- TourneySlots.csv: This file contains the slot information of every March-Masness
- TourneySeeds.csv: This file contains the list of seed team of March-Madness from 1985 to 2016.
- SampleSubmisson.csv: This file contains the sample format for our final submission.

Majorly we used the RegularSeasonCompactResults, TourneyCompactResults and TourneySeeds.

V. Evaluation

Figure 3

The features that we use majorly from following csv files, which including TourneyCompactResults.csv, TourneySeeds.csv, RegularSeasonCompactResults.csv.

We imported libraries showed below

```
import numpy as np
import pandas as pd
import math
from sklearn.metrics import log_loss
from sklearn.preprocessing import StandardScaler
import csv
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn import ensemble
```

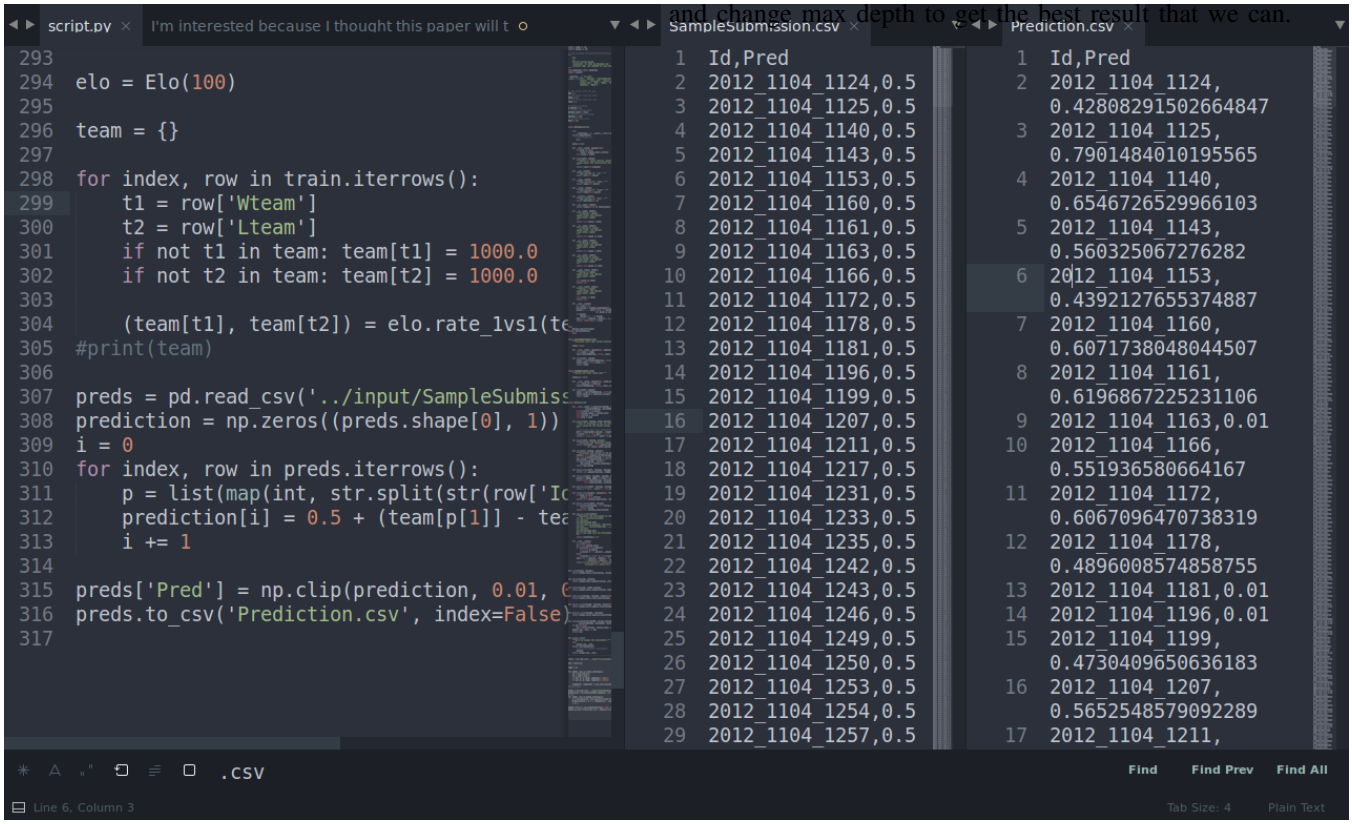
And load data by using

```
tourneyresults = pd.read_csv('TourneyCompactResult
tourneyseeds = pd.read_csv('TourneySeeds.csv')
regularseasoncompactresults = \
pd.read_csv('RegularSeasonCompactResults.csv')
sample = pd.read_csv('SampleSubmission.csv')
```

We used Elo rating system from open source code in Github (Heungsub Lee) and utilized GP(genetic programming model from Reza), you can find resource in reference page. Lasted we define our extra trees classifier model equals

```
ExtraTreesClassifier(n_estimators=5000,max_feature
criterion= 'entropy',min_samples_split= 1,
max_depth= 40, min_samples_leaf= 1, n_jobs = -1)
```

and change max depth to get the best result that we can.



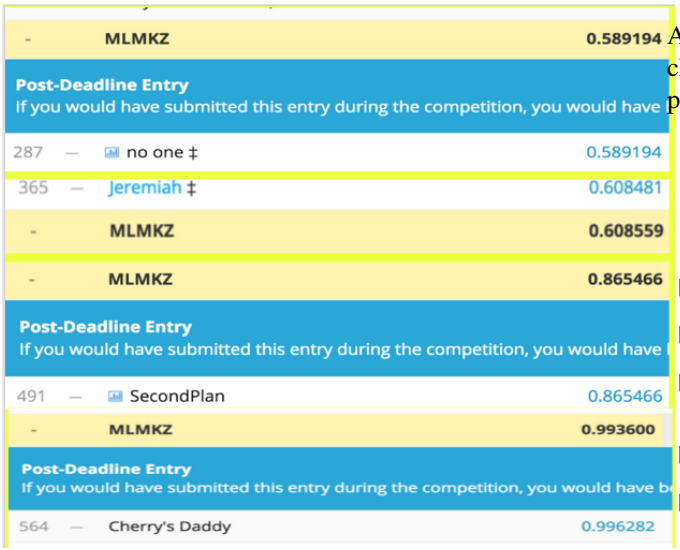


Figure 5

VI. Conclusion

Firstly, this is a difficult problem, we have to consider large number of features and evaluate the weight of each one in order to find the best parameterized model to generate best result. However, even we trained a good model by using machine learning technique, it could still yield a bad result since the real match can not be predict confirmly, any unexpected reason - like what we showed in our presentation - could affect the final result and generate a different result. After participating to this competition, we realized that the data we used was very limited, we only used the data from compact result and generate them into some new features such as the average winning score median winning score and things like that, for the further optimization we need to take the detailed results into consideration, besides the data showed in Kaggle, the player profile leveled data also play a significant place to the match, moreover, there are also some high level classification algorithm(Neural Network, SVM.etc), we should be able to try those model in the future as well.

VII. Team roles

As a small team, we have three people and work together every time to research the existing approaches and read related papers for broaden our vision towards to this project. We also learned python together and trouble-shoot different problem we came across when trying utilize different codes and finding the usage of many python libraries we listed in both presentation and evaluation, one thing we found is really power and useful is a library called sklearn. Although we didnt win the first place, we had a pleasure time to learn things as a team.

Acknowledgment

This project was inspired by Professor Kate Saenko. Appreciate that we can participate to Machine Learning class and has this chance to get real practice by doing this project.

References

[1] Gemalto/safenet. 2016. Building a Trusted Foundation for the Internet of Things.
[2] Kenneth Deakins, March 2012, Predicting The NCAA Tournament Using Machine Learning Methods, teamrankings
[3] Yuanhao (Stanley) Yang, May 2015, Predicting Regular Season Results of NBA Teams Based on Regression Analysis of Common Basketball Statistics, University of California at Berkeley
[4] Jared Forsyth, Andrew Wilde, A Machine Learning Approach to March Madness, Winter 2014, Brigham Young University
[5] Carson K. Leung*, Kyle W. Joseph, September 2014, Sports Data Mining: Predicting Results for the College Football Games, Procedia Computer Science
[6] Kaggle March Machine Learning Mania 2016
[7] Lopez, Michael J. and Gregory J. Matthews. 2015. Building an NCAA mens 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
[8] Genetic Programming
[9] Elo rating system
[10] Scikit-learn