Write Up

Our ML model will take in information about the performance difference of two teams (Team 1 and Team 2) as input, and then output the result that ‘1’ indicates Team 1 wins and ‘0’ indicates Team 2 wins.

To create our representation for each team, we concatenated all the features’ average values of the two teams into two 42 dimensional vector. Since ML models normally take a single input, we represented each matchup as the difference between the two team vectors (Team 1 vector - Team 2 vector). This is one way of representing the matchup. And ‘Win?’ is the dependent variable that we need to predict. So this becomes a classification problem for the model to solve.

Besides the features that Merkle provides, we wanted to create other important features that might improve the model. For variable ‘Home’, 1 means team1 has Home court advantage, 0 means two teams played in a neutral court; ‘F4’ means whether the team was the final four in last 10 years (1 means yes); ‘Top_player’ means whether the team has star players(depends on a performance formula uPER)(1 means yes); ‘C10’ means the top 10 conferences (1 means yes).

After settling the dataset, we chose the model and the feature that we need to use. We used gradient boosting in our final model considering the highest test and training accuracy, which are higher than those gained from logistic regression, SVM, decision tree, random forest etc. In our model, significant variables are selected and tested utilizing clf.feature_importances_. It is relatively straightforward to retrieve importance for each variable and sixteen significant variables are included based on their importance,which are greater than 0.02. For final model,

Significant variables:

['2P%', '3P%', 'FT%', 'AST', 'BLK',"TOV", 'PTS', 'Opp 2P', 'Opp FT%', 'Opp DRB', 'Opp STL', 'Opp PTS', 'WinDiff', 'C10', 'Home']

All original features:

['FG', 'FGA', 'FG%', '2P', '2PA', '2P%', '3P', '3PA', '3P%', 'FT', 'FTA',

'FT%', 'ORB', 'DRB', 'TRB', 'AST', 'STL', 'BLK', 'TOV', 'PF', 'PTS',

'Opp FG', 'Opp FGA', 'Opp FG%', 'Opp 2P', 'Opp 2PA', 'Opp 2P%',

'Opp 3P', 'Opp 3PA', 'Opp 3P%', 'Opp FT', 'Opp FTA', 'Opp FT%',

'Opp ORB', 'Opp DRB', 'Opp TRB', 'Opp AST', 'Opp STL', 'Opp BLK',

'Opp TOV', 'Opp PF', 'Opp PTS', 'Win_diff', 'F4', 'C10', 'Top_Player',

'Home']

Importance for all features:

array([0.01814962, 0.01068142, 0.01520834, 0.0135407 , 0.0117333 ,
0.0210774 , 0.00768222, 0.00767986, 0.02813623, 0.01828434,

0.01325861, 0.02132167, 0.01116907, 0.00535462, 0.01107209,
0.03307589, 0.01001035, 0.04749707, 0.02999653, 0.00261078,
0.03206095, 0.01896407, 0.00646495, 0.01349147, 0.03091819,
0.01129695, 0.01503533, 0.01450233, 0.00823115, 0.00778952,
0.0020223 , 0.01521704, 0.03225639, 0.00218068, 0.03277973,
0.01693353, 0.014648 , 0.02305358, 0.01792583, 0.0100475 ,
0.01011063, 0.01978424, 0.17399848, 0.0010899 , 0.05872485,
0.00625163, 0.0666807])

We treated every matchup the same. After selecting the gradient boosting as our final model, we created a program that can perform the algorithm for each matchup in 2018. We got the result for the first round, second round and so on. In the end, our predicted champion is 'Villanova'

Head-up:

the original file, 2018TeamStats Final.csv was changed on the second day, and has errors. So we use the first version that Merkle provided. However, in the first version, the outputs of 'Win?' are upside-down. Thus, we did some transformation on it and made it as same as 2017's.

For the extra four features, we didn't use extra dataset. We just applied those features based on our own knowledge and research.

More details could be found in the codes.

Reference:

1. Top 11 conference:
http://bleacherreport.com/articles/2749184-ranking-the-top-college-basketball-conferences-in-2017-18