IEOR 265 – Project Report
Forecasting March Madness

Thibault Corneloup, William Sele, Benjamin Tran Dinh

May 7, 2015

Abstract

This report presents our team's methodology and results in predicting basketball matches outcomes from historical data. We implemented different regression techniques to forecast the probabilities of winning for the 68 teams participating in the annual NCAA College Basketball tournament, also known as March Madness. The context of the project is a competition held by the platform for predictive modeling Kaggle.
The report presents the construction of our features, and the two models that we submitted for the competition. We then present and analyze their performance during the modeling phase, and the actual competition.

1 Introduction

The National Collegiate Athletic Association (NCAA) Men's Division I Basketball Tournament is a single-elimination tournament played each spring in the United States, currently featuring 68 college basketball teams, to determine the national champion of the major college basketball teams. Played mostly during March, it is known informally as March Madness or the Big Dance, and has become one of the most famous annual sporting events in the United States.

Although experts and thousands of fans intend to predict the results of the tournament, the odds of forecasting a perfect bracket are astronomical. In this context, Kaggle launches a machine learning contest. Based on data provided by Kaggle but also on any other external data, the goal is to predict the outcome of the men's NCAA basketball tournament. In a first round, participants will build and test their models against the previous four tournaments. Then, the real competition begins: each contestant forecasts the 2015 results, by submitting a "win or loss" likelihood for every possible matchup.

The first part of the project was to identify a set of features that are relevant in determining the outcome of a game. The available data allows us to compute the ratio of games won, game played at home/away, whether or not the previous game has been won (team moral), for example. More precise statistics are available such as the ratio of 3 pointers scored, number of overtime periods, etc. Identifying the most relevant data was one big part of our work.

Then we built different models in order to try to fit historical data. Linear models seem to be the most popular in sports results prediction, but several variations are available for exploration: logistic regression, L2 regularization,

elastic net, etc. We will go over the several engineering choices that we made in order to build our models. We built two different models that differ in their choice of indicators, formulation and the related optimization problems. We will compare their design and motivate their construction choices. In a last section we discuss their results during the actual competition.

2 Domain Knowledge

2.1 Home and Away Games

In Basketball, the teams perform highly differently when they play in their stadium than when they travel to play away games. Hosts team won on average two third of their games during the season 2014-2015, as shown in Table 1:

Category	Record	Percent
Away Teams	1959-3973	33.02%
Home Teams	3973-1959	66.98%

Table 1: Win-Loss Records for NCAA 2014-2015 [2]

Thus, it comforted our feeling that the differentiation between home and away games was relevant. It is also important to note that for the March Madness, all games are performed in a neutral stadium.

2.2 Strength of Schedule

During the regular season games, each team will play twice against each team within its conference, once at home and once away. However, the teams can also schedule non-conference games. Teams can choose (or at least negotiate) their non conference opponents during the regular season which accounts for about a third of the regular season. That is why taking into account the strength of schedule is of great importance when trying to assess the strength of a given team. We will see later in the computation of the RPI factor that the selection committee not only takes into account your opponents results, but also your opponents' opponents. This induces a complex strategical planing for the teams: there is a trade-off between scheduling games against high-level teams, thus taking the risk to lose most of their games, and scheduling games against worse teams, thus decreasing their strength of schedule. The best plan usually used is to schedule teams in a higher-ranked conference, against which they stand a chance. However, many teams want to play against these teams, making it complicated to schedule such games as we would like [8].

2.3 Offensive/Defensive capabilities

A widespread belief, supported by experts, has it that defensive teams perform better during the final tournament: "Defense wins championships". Coming up with a measure of defensiveness of a team might be a useful feature to introduce. However, no strong study has been conducted to determine whether this statement is verified in NCAA Basketball, and if it is, remains to question of how to quantify this effect [1].

2.4 Ability to Win Close Games

During the regular season, the teams levels are very spread out, some teams being much better than others. However, during the March Madness, only the best 68 teams have been selected, such that it is more likely to have close games. During the regular season, only 25% of the games are won by five points or less. We decided to use this metric to decide which game are close games, and which are not. Our intuition was that a team able to win close games will be able to win against its major opponents, and has the mental and physical strength to win tight games.

3 Choice and Construction of Input Indicators

Kaggle provides almost two decades of historical data. Use of any kind of data (provided or not by Kaggle, public or not) is allowed in the competition. In annex, we provide a comprehensive description of the data provided by Kaggle. We chose to only use this data and no external sources.

Furthermore, Kaggle also provides the seed ranking of each team at the beginning of the tournament. This is a ranking decided by a committee that determines in which order games are played, and is a strong predictor of the probability to win. The committee usually gives the smallest seeds to the best teams, such that these teams will not play against each other during the first rounds. However we chose not to use this manually generated data to focus on predicting game outcomes with factual data.

3.1 Getting started with the computation of the RPI (Rating Percentage Index)

One way to compare two teams would be to come up with a scoring system. The difference between two team scores then has to be translated into a probability of winning. This first approach of the data only takes into consideration which team won and which team lost each game. The idea is to create a ratio that allow us to score the teams, based upon a team's wins and losses and its strength of schedule. The main scoring system is the Rating Percentage Index. It is computed as described by Equation (1):

$$RPI = 0.25WP + 0.50OWP + 0.25OOWP \tag{1}$$

where
WP is the Team Winning Percentage
OWP is the Opponents' Winning Percentage
$OOWP$ is the Opponents' Opponents' Winning Percentage

To account for Home and Away bias, a Home win is worth 0.6 while a road win is worth 1.4. Conversely, a Home loss is worth 1.4 while a road loss is worth 0.6. Moreover to obtain an accurate estimate of the opponent's strength, all games played against the main team are not taken into account while computing the OWP (the main team being the team for which the RPI is being computed).

This rating is particularly important as it is one of the official indicators used by the NCAA Selection Committee to assign team seeds. We tried to build on this index to create our own strength indicator.

3.2 Construction of our own Strength Indicator

Considering the margin of victory

The first weakness of the RPI that comes to mind is the fact that it does not take into consideration the points difference. Indeed, a victory by 25 points will be "rewarded" as much as a close victory by 2 points. The idea is that a team that wins its games by wide margins is generally thought to be better than another team that wins its games by narrow margins.

Differentiating the games by period of the year

Willing to incorporate relevant data in order to improve the predictions, we figured that we might take into consideration the trend of victory of teams. Indeed, a team can make progresses along the season, and the last games of the season can be very important for the team to be confident for the final tournament. Our intuition was that the last part of the season would be more significant to explain the March Madness games. At this point, we had one indicator for home games and one for away games. To include this into our model, we separated the games by date into three equals parts (beginning of the season, middle and end) and created thus six indicators (three for home games, three for away games)[11].

One possible objection to this procedure is the bias vs. variance trade-off. By splitting the season this way, the indicators become more noisy. However, experimental results showed that splitting the season was beneficial.

Computation of our Strength Indicator

We built a strength indicator ranging between 0 and 1, associated with a given team, season, period (third of a season) and location (Home or Away). It is computed as described in Equation (2).

$$SI_{k,s,p,l}(T) = \frac{\frac{1}{\overline{m}_{s,l}} \sum_{i \in W_{T,s,p,l}} m_i \widetilde{SI}_{k,s,p,\tilde{l}}(O_i)}{\frac{1}{\overline{m}_{s,l}} \sum_{i \in W_{T,s,p,l}} m_i \widetilde{SI}_{k,s,p,\tilde{l}}(O_i) + \frac{1}{\overline{m}_{s,\tilde{l}}} \sum_{i \in L_{T,s,p,l}} m_i \left[1 - \widetilde{SI}_k(O_i)\right]} \tag{2}$$

where:

$$\widetilde{SI}_{k,s,p,\tilde{l}}(O_i) = \begin{cases} 0.1 + 0.8\, SI_{k-1,s,p,\tilde{l}}(O_i) & \text{if } k \geq 2 \\ 1 & \text{if } k = 1 \end{cases} \tag{3}$$

and:
$k \in [|1, 4|]$ is the number of times the indicator has been iterated,

$s \in [|2008, 2015|]$ is the selected season,
$p \in [|1, 3|]$ is the "period" of the season such that

$$p = \begin{cases} 1 \implies \text{day number} \in [|1, 44|] \\ 2 \implies \text{day number} \in [|45, 88|] \\ 3 \implies \text{day number} \in [|89, 132|] \end{cases} \tag{4}$$

l is either Home or Away,
$\tilde{l}$ is Away if l is Home, Home if l is Away,
T is the selected team,
$\overline{m}_{s,l} \in \mathbb{R}^+$ is the average victory margin for season s when the winner is at location l,
$W_{T,s,p,l}$ is the set of games won by team T in season s, period p, when in location l,
$L_{T,s,p,l}$ is the set of games lost by team T in season s, period p, when in location l,
$m_i \in \mathbb{R}^+$ is the victory margin of game i,
O_i is the team T's opponent in game i,

The indicator is therefore 0 if the team has lost every game in this period and location, 1 if the team has won every game in this period and location. For our indicator to take the strength of schedule properly into account, we decided to iterate until k=4. Statistics of this indicators are shown in Tables 2 and 3.

Indicator	$SI_{Home,p=1}$	$SI_{Home,p=2}$	$SI_{Home,p=3}$
Mean	0.541221	0.558367	0.510134
St. Dev.	0.379954	0.350477	0.328578
Min	0.000000	0.000000	0.000000
Max	1.000000	1.000000	1.000000

Table 2: Strength Indicator statistics for Home games in regular seasons from 2008 to 2014.

Indicator	$SI_{Away,p=1}$	$SI_{Away,p=2}$	$SI_{Away,p=3}$
Mean	0.450771	0.426752	0.452901
St. Dev.	0.384395	0.346769	0.336125
Min	0.000000	0.000000	0.000000
Max	1.000000	1.000000	1.000000

Table 3: Strength Indicator statistics for Away games in regular seasons from 2008 to 2014.

3.3 Construction of a Defense Indicator and a Close Game Performance Indicator

Defense Indicator

Our domain knowledge lead us to believe that defense was a particularly decisive feature in order to predict the outcome of a game. That is why we

decided to create a Defense Indicator (DI) distinct from the Strength Indicator. It is computed as shown by Equation (5).

$$DI_{s,l} = \frac{1}{\bar{c}_{s,l}} \frac{\sum_{i \in G_{T,s,l}} c_i \, SI'_{s,\tilde{l}}(O_i)}{Card(G_{T,s,l})} \tag{5}$$

where:

$$SI'_{s,\tilde{l}}(O_i) = 0.8 + 0.4 \, SI_{k=4,s,\tilde{l}}(O_i) \tag{6}$$

and
$s \in [|2008, 2015|]$ is the selected season,
l is either Home or Away,
$\tilde{l}$ is Away if l is Home, Home if l is Away,
T is the selected team,
$\bar{c}_{s,l} \in \mathbb{R}^+$ is the average conceded points per game during season s at location l by all participating teams,
$G_{T,s,l}$ is the set of games in which team T took part with location status l.
$c_i \in \mathbb{R}^+$ is the number of points conceded by team T during game i.

This gives an indicator centered approximately around one, adjusted for strength of schedule and differentiated for Home an Away games. The lower the Defense Indicator, the better the defense of the team. More statistics for this indicator are shown in Table 4

Indicator	DI_{Home}	DI_{Away}
Mean	1.026384	0.970065
St. Dev.	0.113477	0.103992
Min	0.712178	0.694459
Max	1.563642	1.406166

Table 4: Defense Indicator statistics for games in regular seasons from 2008 to 2014

Close Game Performance Indicator

Between 2008 and 2014, NCAA Basketball games were won by an average margin of 11 points. Only 25% of the games were won by 5 points or less. Since the ability to win close games seemed critical for a knockout tournament such as March Madness, we decided to build a Close Game Indicator (CGI) computed as described in Equation (7).

$$CGI_s(T) = \frac{Card(CGW_{T,s})}{Card(CGW_{T,s}) + Card(CGL_{T,s})} \tag{7}$$

where:
$s \in [|2008, 2015|]$ is the selected season,

T is the selected team,
$CGW_{T,s}$ is the set of games won during season s by team T by 5 points or less,
$CGL_{T,s}$ is the set of games lost during season s by team T by 5 points or less.

This indicator is simply the proportion of close games (victory margin 5 points or less) won by team T during season s. It ranges from 0 to 1 with a mean close to 0.5 as shown in Table 5.

Indicator	Mean	St. Dev.	Min	Max
CGI	0.498428	0.193637	0	1

Table 5: Close Game Indicator statistics for games in regular season from 2008 to 2014

3.4 Making sense of the rest of the data

This year, Kaggle provided detailed match statistics, such as field goals attempted, field goals made, three pointers made, defensive rebounds... This additional data may contain some predictive information that we could use in our models. In a similar fashion as for the strength indicator, we could imagine creating one strength indicator for each additional column that we have access to. The intuition is that the information "Team A is stronger at *scoring three pointers* than team B" may partially explain victory of team A over team B.

We make again the distinction between Home and Away games and define a strength indicator for each feature X similar to equation (2), by replacing the following:
$\overline{m}_{s,l} \in \mathbb{R}^+$ is the average margin for feature X for season s when the team with greater X is at location l,
$W_{T,s,p,l}$ is the set of games where team T has greater X in season s, period p, when in location l,
$L_{T,s,p,l}$ is the set of games where team T has fewer X in season s, period p, when in location l,
$m_i \in \mathbb{R}^+$ is the margin for feature X in game i,
O_i is the team T's opponent in game i,

Optionally we have the engineering choice to omit the period p when computing those indicators and accept to have fewer, more robust indicators, which also makes the computation less expensive.

4 Competition scoring system and objective function framework for modeling

Each participant will be evaluated according to LogLoss function described in (8). The participant with the lowest LogLoss overall score would win the competition [3].

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)\right] \tag{8}$$

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 teams 2 wins,
$log()$ is the natural (base e) logarithm.

The first aspect to notice is that we are not simply asked to predict a winner in a 0 or 1 manner. Here, predicting 0.92 instead of 0.65 can make for a significant variation in the score, even if both numbers could be interpreted as a victory prediction for team 1. Therefore, while a Support Vector Machine would have been efficient in case of a win/lose prediction, it cannot be applied here.

4.1 First Approach

A first way to approach the problem is to consider a linear model to fit the point spread for each match of past March Madness competitions. We suppose the true model is of the form:

$$\Delta Y_i = \beta^T \Delta X_i + \epsilon_i \tag{9}$$

where:
Y_i is the point difference for match i,
ΔX_i is a vector of differences between indicators of teams A and B involved in match i,
ϵ_i is normal, centered noise.

We can therefore find the $\hat{\beta}$ by performing a linear regression over the games of the previous seasons.
There remains some engineering choices as to whether we perform some regularization or not. Experimental results show more consistent performance when we apply a L2 regularization when doing the regression. The optimization problem becomes:

$$\min_{\hat{\beta}} \|\Delta Y - \hat{\beta}\Delta X\|_2 + \lambda\|\hat{\beta}\|_2 \tag{10}$$

The value of λ can be found by cross-validation.

The remaining problem is to find a way to translate point spread predictions into a probability of winning. As we could imagine, the distribution of point spreads is normal and centered around 0. The standard deviation is 10. Looking at the distribution of odds provided by bookmakers suggests that odds are distributed uniformly on $[0;1]$. Our goal is therefore to find a function $f : \mathbb{R} \to [0;1]$ that would transform our ΔY normal distribution to uniformly distributed points in $[0;1]$. The logistic function, or sigmoid function, is a good candidate:

$$S(y) = \frac{1}{1+e^{-\mu y}} \tag{11}$$

where μ is inversely proportional to our standard deviation in the original point spread distribution. In order to minimize our score according to the LogLoss function, we can treat μ as a tuning parameter and choose the value that minimizes our LogLoss error on the previous seasons' data.
We arrived at the following coefficients:

- $\lambda = 0.50$
- $\mu = 0.11$

It is interesting to note that our predictions distribution ends up not being uniform over $[0; 1]$ for the cross validation to reach its minimal error. We notice that the sigmoid translation function clusters most of our data around 0.5, making the prediction fairly conservative (as opposed to making bold assumptions that one team will win with probability 0.99, for instance).The component of luck involved in winning a match of basketball, that makes the data fairly noisy, explains this phenomenon: matches where we predict the wrong outcome will tend to push our μ to decrease, thus making our predictions more dense around 0.5.

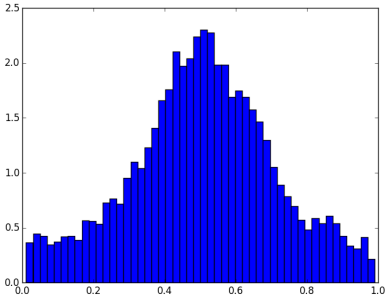

Figure 1: Probabilities prediction distribution

Indicator	$\hat{\beta}$ at Home	$\hat{\beta}$ Away
Score difference	19.17	21.72
Field goals made	5.83	-6.00
Field goals attempted	3.90	4.44
Three pointers made	-2.28	9.35
Three pointers attempted	0.86	-3.66
Free throws made	10.82	2.64
Free throws attempted	-3.69	0.60
Defensive rebounds	2.55	3.89
Assists	-4.89	3.33
Steals	2.54	1.06
Blocks	2.63	-2.40

Table 6: Regression coefficients for Approach 1

In Table 6 are displayed indicators that we used for this first approach, as well

as the corresponding regression coefficients (predicting point difference). The coefficients show a big predictive power for the Score difference-based strength indicator, as well as the free throws made. It is unclear whether the Home/Away distinction is justified in this context. For example, we have some indicators with regressors of opposite signs whether takes Home or Away performances into account.

This approach has the advantage of taking into account a Y vector that contains more precise data than only 0 and 1. However, we can only take into account the LogLoss scoring function with one tuning parameter, when we could use this structure at an earlier point in our regression. This method is exposed in our second approach.

4.2 Second Approach

For this approach we are going to use only the indicators we built, and use the LogLoss scoring function as our optimizing objective function. The Input matrix X will then have the columns shown in Figure 2:

$$\begin{pmatrix} \Delta SI_{H,1} & \Delta SI_{H,2} & \Delta SI_{H,3} & \Delta SI_{A,1} & \Delta SI_{A,2} & \Delta SI_{A,3} & \Delta DI_H & \Delta DI_A & \Delta CGI \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Figure 2: Input Matrix X for the second approach

where $\Delta Indicator = Indicator(Winner_i) - Indicator(Loser_i)$, except for $\Delta DI = DI(Loser_i) - DI(Winner_i)$ since DI is the only indicator for which lower is better. The statistics for these inputs are given by Table 7. Then the

Input	Mean	St. Dev.	Min	Max
$\Delta SI_{H,1}$	0.1002	0.2811	-0.9622	1.0000
$\Delta SI_{H,2}$	0.0504	0.2194	-0.8019	0.8766
$\Delta SI_{H,3}$	0.0096	0.2824	-0.9191	0.9706
$\Delta SI_{A,1}$	0.1112	0.3462	-1.0000	1.0000
$\Delta SI_{A,2}$	0.0854	0.3171	-1.0000	1.0000
$\Delta SI_{A,3}$	0.0406	0.2688	-0.7066	0.9407
ΔDI_H	0.0406	0.1192	-0.2528	0.4235
ΔDI_A	0.0423	0.1188	-0.4043	0.4320
ΔCGI	0.0180	0.2887	-1.0000	1.0000

Table 7: Input statistics for the second approach

measured y_i is always 1, and by forcing the intercept to 0.5, we insure that our model is symmetrical and can handle losing probabilities (< 0.5).
To sum up, the indicators are computed for each team and regular season. During training, the model tries to fit the βs from the (regular season, team) indicators to their respective tournament outcomes. In other words, to predict the outcome of season s tournament, the model will use only indicators from season s. However, the βs will be the same across seasons. The program to

optimize is described in Equations (12) to (15):

$$\min_{\hat{\beta}\in\mathbb{R}^9} -\frac{1}{n}\sum_{i=1}^{n} log(\hat{y}_i) \tag{12}$$

$$\text{subject to: } \hat{y}_i = max\left\{10^{-6}, 0.5 + \sum_{j=1}^{9}\hat{\beta}_j X_i^j\right\} \qquad \forall i \in [|1,n|] \tag{13}$$

$$0.5 + \sum_{j=1}^{9}\hat{\beta}_j X_i^j \geq 0 \qquad \forall i \in [|1,n|] \tag{14}$$

$$0.5 + \sum_{j=1}^{9}\hat{\beta}_j X_i^j \leq 1 \qquad \forall i \in [|1,n|] \tag{15}$$

We then had to correct our model for collinearity (the different ΔSI will likely have a positive covariance), so we added a penalty for the L_2 norm of $\hat{\beta}$. Moreover, to try and rule out noisy inputs, we added a penalty for the L_1 norm of $\hat{\beta}$, forming an *elastic net*. The resulting objective function is described by Equation (16).

$$\min_{\hat{\beta}\in\mathbb{R}^9} -\frac{1}{n}\sum_{i=1}^{n} log(\hat{y}_i) + \lambda\|\hat{\beta}\|_2 + \mu\|\hat{\beta}\|_1 \tag{16}$$

From this program, we obtained a $\hat{\beta}$ associated with ΔCGI close to 0 (10^{-5}). As a result, we ruled out this indicator from our model and ran a similar program without the L_1 norm penalty for the remaining indicators. The values obtained for the 8 $\hat{\beta}_j$ are displayed in Table 8.

Test set	2011	2012	2013	2014		Training on $[\|2008, 2014\|]$
Intercept (fixed)	0.5	0.5	0.5	0.5		0.5
$\Delta SI_{H,1}$	0.4626	0.5210	0.4714	0.4477		0.4645
$\Delta SI_{H,2}$	0.2440	0.3297	0.2254	0.2857		0.2588
$\Delta SI_{H,3}$	0.0568	0.0466	0.0160	0.0653		0.0265
$\Delta SI_{A,1}$	0.2169	0.2451	0.2394	0.2165		0.2286
$\Delta SI_{A,2}$	0.1877	0.1874	0.1826	0.1867		0.1980
$\Delta SI_{A,3}$	0.2365	0.2246	0.2892	0.1869		0.2265
ΔDI_H	0.2693	0.1812	0.2277	0.1878	Total	0.2196
ΔDI_H	0.2912	0.2061	0.3228	0.2660	CV Error	0.2613
CV Error	0.5563	0.6172	0.5767	0.5131	0.5658	–

Table 8: Values for the $\hat{\beta}$s associated the different indicators. When the test year is specified, the training was performed on $[|2008, 2014|]\backslash\{\text{Test year}\}$. On these training sets, the optimal lambda was $\lambda = 0.064$. We used the same λ for the final training on $[|2008, 2014|]$.

One interesting value is the $\hat{\beta}$ found in the final training for $\Delta SI_{H,3}$. This very low estimate contradicts our first intuition that "in shape" teams are more likely to perform well in the upcoming tournament. This result seems to be

supported by data and sports analysts. According to Nate Silver [10], teams that over-perform at the end of the season, thus granted a higher seed, tend to perform poorly during March Madness. Perhaps the certainty of being selected as one of the 68 finalists (or to be eliminated) makes the teams show less efforts in the last games of the regular season. This counter-intuitive result was also confirmed by Kvam and Sokol (2006) [6] who found that most of the variability of a team's success in a tournament could be explained by the regular season games played before February.

We performed a *cross validation* by simulating the prediction of the outcome of a whole tournament using only information about the associated regular season and other seasons tournaments (i.e. excluding data from this particular tournament from the training set). This means that our CV Error for 2014 simulates exactly what score we would have obtained last year, had we participated in the Kaggle competition (indeed, only data available before the submission deadline was used). The winner in 2014 scored at 0.52951 [4], while our virtual score was 0.5131 (Table 8) meaning that our model *would have* won the 2014 competition. Unfortunately, our excitement was to quickly vanish after the beginning of the 2015 tournament.

5 Results and Comments

Despite a higher cross validation error, our first approach turned out to be our best estimate. With a final score of 0.57194, it did much better than our second model which scored an ugly 0.72737. However, while last year's best score was around 0.53, this year's best score reached 0.439 pushing our predictions quite low in the leader-board (we ended up #272 out of 341 teams).

The first approach scored a reasonable 0.57 with its fairly conservative predictions. Our main and most predictive indicator being the same as the second approach (Score difference-based strength indicator), most of the predictions went in the same direction. However, the first approach revealed to take less risks than the second one.

The third game was a big upset for our 2^{nd} model which had boldly predicted a 99.5% chance of winning for the favorite (#3 Iowa St v #14 UAB). The University of Alabama at Birmingham having decided otherwise, we were assigned a score of 5.3 for this game while the winner of the competition, who had predicted a 95.2 % chance of winning for the favorite, was assigned a score of 3.0 (see Table 9). A difference of more than 2 units, when most differences fall around 0.3, is hard to make up for.

The 2^{nd} approach had one other major setback when its 100% prediction for the favorite of (#1 Villanova v #8 NC State) turned out wrong. We immediately thought we had taken too much risk. However, when comparing risks taken by our model and by the winner, we realized that the winner actually took more risk than we did, but performed much better on the associated games, as shown by Table 10.

Team 1	Team 2	2^{nd} approach prediction	Kaggle winner prediction	2^{nd} approach score	Kaggle winner score
#1 Villanova (W)	#16 Lafayette	0.9685	1.0000	0.0320	0.0000
#8 NC State (W)	#9 LSU	0.1402	0.6478	1.9647	0.4342
#3 Iowa St	#14 UAB (W)	0.9950	0.9518	5.2983	3.0318
#4 Georgetown(W)	#13 E Washington	0.6094	0.8565	0.4953	0.1549
#5 Utah (W)	#12 SF Austin	0.5241	0.8396	0.6461	0.1749
#6 SMU	#11 UCLA (W)	0.5560	0.5627	0.8120	0.8271
#1 Kentucky (W)	#16 Hampton	1.0000	1.0000	0.0000	0.0000
		Average score after 7 games:		1.32121	0.66042

Table 9: Performance of our 2*nd* approach compared to Kaggle winner for the first 7 games of the 2015 March Madness Tournament.

	2^{nd} approach	Kaggle winner
Number of risky predictions (>0.99 or <0.01)	5	6
Number of incorrect risky predictions	2	0

Table 10: Risk taking of our 2^{nd} approach compared to Kaggle winner on all 64 games of the 2015 March Madness Tournament.

Then why did we not perform as well as we expected? A first answer could be the lack of human expertise. When observing the methodologies of efficient models such as Lopez and Matthews [7] or Silver [9], we realized that they all relied heavily on *handmade*, or at least *black box* expert ratings. Combining several ratings also turned out to be very helpful. This principle is confirmed by the good score (0.489) obtained by the median submission of the Kaggle Competition [5], as well as experimentally in other domains[13]. Our decision not to rely on any other data than facts (not even seeds) explains a lot of our poor performance. Moreover, it seems ([9] and [7]) that distance from a team's home campus to the tournament site is a relevant information that we did not take into account. Another factor is the absence of major players for the tournament. This kind of data is much harder to collect and requires an expertise that we did not have. Finally, luck was not on our side. Had our 2 incorrect risky predictions (see Table 10) turned out right, we estimate our 2^{nd} approach would have obtained a score close to 0.52 (although all teams would have benefited from this hypothesis, our riskier than average predictions would have given us an advantage over other teams).

References

[1] Atkinson Adrian, *Defense Wins Championships? No, Offense And Balance Do*, `https://www.teamrankings.com/blog/ncaa-basketball/defense-wins-championships-no-offense-and-balance-do-stat-geek-idol`, Posted in March 2012.

[2] Covers.com, *NCAA College Basketball Trends*, `http://www.covers.com/pageLoader/pageLoader.aspx?page=/data/ncb/trends/league/`

season.html, Accessed in February 2015.

[3] Kaggle.com, *Evaluation*, https://www.kaggle.com/c/march-machine-learning-mania-2015/details/evaluation, Accessed in February 2015.

[4] Kaggle.com, *Private Leaderboard - 2014 March Machine Learning Mania*, https://www.kaggle.com/c/march-machine-learning-mania/leaderboard, Accessed in February 2015.

[5] Kaggle.com, *Private Leaderboard - March Machine Learning Mania 2015*, https://www.kaggle.com/c/march-machine-learning-mania-2015/leaderboard, Accessed in February 2015.

[6] Kvam, P. and Sokol, J. S., *A Logistic Regression/Markov Chain Model for NCAA Basketball*, Naval Research Logistics (NrL) 2006 53:788–803.

[7] Lopez M. J. and Matthews G. J., *Building an NCAA men's basketball predictive model and quantifying its success*, DE GRUYTER, J. Quant. Anal. Sports 2015; 11(1): 5–12.

[8] NCAA.org, *What is the best way for a team to improve its RPI?*, https://www.ncaa.org/sites/default/files/FAQ_for_MLAX_RPI.pdf, Updated in March 2008.

[9] Silver N., *How FiveThirtyEight's March Madness Bracket Works*, http://fivethirtyeight.com/features/march-madness-predictions-2015-methodology/, Posted on March 15, 2015.

[10] Silver N., *In N.C.A.A. Tournament, Overachievers Often Disappoint*, http://fivethirtyeight.com/features/in-n-c-a-a-tournament-overachievers-often-disappoint/ Posted on March 11, 2011.

[11] Tan P. and Harrison J., *An Iterative Strength Rating Based Model for the Prediction NCAA Basketball Games*, http://courses.cs.washington.edu/courses/cse140/13wi/projects/jarrison-report.pdf, Accessed in February 2015.

[12] VegasInsider.com, *College Basketball Odds*, http://www.vegasinsider.com/college-basketball/, Accessed in March 2015.

[13] Webb and Zheng, *Multistrategy ensemble learning: reducing error by combining ensemble learning techniques*, Knowledge and Data Engineering, IEEE Transactions on , vol.16, no.8, pp.980,991, Aug. 2004

6 Appendix: Available Data

teams.csv
This file identifies the different college teams present in the dataset. Each team has a 4 digit id number.

seasons.csv

This file identifies the different seasons included in the historical data, along with certain season-level properties.

"season" - indicates the year in which the tournament was played

"dayzero" - tells you the date corresponding to daynum=0 during that season. All game dates have been aligned upon a common scale so that the championship game of the final tournament is on daynum=154. Working backward, the national semifinals are always on daynum=152, the "play-in" games are on days 134/135, Selection Sunday is on day 132, and so on. All game data includes the day number in order to make it easier to perform date calculations. If you really want to know the exact date a game was played on, you can combine the game's "daynum" with the season's "dayzero". For instance, since day zero during the 2011-2012 season was 10/31/2011, if we know that the earliest regular season games that year were played on daynum=7, they were therefore played on 11/07/2011.

"regionW/X/Y/Z" - by convention, the four regions in the final tournament are always named W, X, Y, and Z. Whichever region's name comes first alphabetically, that region will be Region W. And whichever Region plays against Region W in the national semifinals, that will be Region X. For the other two regions, whichever region's name comes first alphabetically, that region will be Region Y, and the other will be Region Z. This allows us to identify the regions and brackets in a standardized way in other files. For instance, during the 2012 tournament, the four regions were East, Midwest, South, and West. Being the first alphabetically, East becomes W. Since the East regional champion (Ohio State) played against the Midwest regional champion (Kansas) in the national semifinals, that makes Midwest be region X. For the other two (South and West), since South comes first alphabetically, that makes South Y and therefore West is Z. So for this season, the W/X/Y/Z are East,Midwest,South,West.

regular_season_compact_results.csv

This file identifies the game-by-game results for 30 seasons of historical data, from 1985 to 2014. Each year, it includes all games played from daynum 0 through 132 (which by definition is "Selection Sunday", the day that tournament pairings are announced). Each row in the file represents a single game played.

"season" - this is the year of the associated entry in seasons.csv (the year in which the final tournament occurs)

"daynum" - this integer always ranges from 0 to 132, and tells you what day the game was played on. It represents an offset from the "dayzero" date in the "seasons.csv" file. For example, the first game in the file was daynum=20. Combined with the fact from the "season.csv" file that day zero was 10/29/1984, that means the first game was played 20 days later, or 11/18/1984. There are no teams that ever played more than one game on a given date, so you can use this fact if you need a unique key. In order to accomplish this uniqueness, we had to adjust one game's date. In March 2008, the SEC postseason tournament had to reschedule one game (Georgia-Kentucky) to a subsequent day, so Georgia had to actually play two games on the same day. In order to enforce this uniqueness, we moved the game date for the Georgia-Kentucky game back to its original date.

"wteam" - this identifies the id number of the team that won the game, as

listed in the "teams.csv" file. No matter whether the game was won by the home team or visiting team, "wteam" always identifies the winning team.
"wscore" - this identifies the number of points scored by the winning team.
"lteam" - this identifies the id number of the team that lost the game.
"lscore" - this identifies the number of points scored by the losing team.
"numot" - this indicates the number of overtime periods in the game, an integer 0 or higher.
"wloc" - this identifies the "location" of the winning team. If the winning team was the home team, this value will be "H". If the winning team was the visiting team, this value will be "A". If it was played on a neutral court, then this value will be "N". Sometimes it is unclear whether the site should be considered neutral, since it is near one team's home court, or even on their court during a tournament, but for this determination we have simply used the Kenneth Massey data in its current state, where the "@" sign is either listed with the winning team, the losing team, or neither team.

regular_season_detailed_results.csv
This file is a more detailed set of game results, covering seasons 2003-2014. This includes team-level total statistics for each game (total field goals attempted, offensive rebounds, etc.) The column names should be self-explanatory to basketball fans (as above, "w" or "l" refers to the winning or losing team):
wfgm - field goals made
wfga - field goals attempted
wfgm3 - three pointers made
wfga3 - three pointers attempted
wftm - free throws made
wfta - free throws attempted
wor - offensive rebounds
wdr - defensive rebounds
wast - assists
wto - turnovers
wstl - steals
wblk - blocks
wpf - personal fouls

tourney_compact_results.csv
This file identifies the game-by-game NCAA tournament results for all seasons of historical data. The data is formatted exactly like the regular_season_compact_results.csv data. Note that these games also include the play-in games (which always occurred on day 134/135) for those years that had play-in games.

tourney_detailed_results.csv
This file contains the more detailed results for tournament games from 2003 onward.

tourney_seeds.csv
This file identifies the seeds for all teams in each NCAA tournament, for all seasons of historical data. Thus, there are between 64-68 rows for each year, depending on the bracket structure.
"season" - the year

"seed" - this is a 3/4-character identifier of the seed, where the first character is either W, X, Y, or Z (identifying the region the team was in) and the next two digits (either 01, 02, ..., 15, or 16) tells you the seed within the region. For play-in teams, there is a fourth character (a or b) to further distinguish the seeds, since teams that face each other in the play-in games will have the same first three characters. For example, the first record in the file is seed W01, which means we are looking at the number 1 seed in the W region (which we can see from the "seasons.csv" file was the East region). This seed is also referenced in the "tourney_slots.csv" file that tells us which bracket slots face which other bracket slots in which rounds.
"team" - this identifies the id number of the team, as specified in the teams.csv file

tourney_slots.csv

This file identifies the mechanism by which teams are paired against each other, depending upon their seeds. Because of the existence of play-in games for particular seed numbers, the pairings have small differences from year to year. If there were N teams in the tournament during a particular year, there were N-1 teams eliminated (leaving one champion) and therefore N-1 games played, as well as N-1 slots in the tournament bracket, and thus there will be N-1 records in this file for that season.
"season" - the year
"slot" - this uniquely identifies one of the tournament games. For play-in games, it is a three-character string identifying the seed fulfilled by the winning team, such as W16 or Z13. For regular tournament games, it is a four-character string, where the first two characters tell you which round the game is (R1, R2, R3, R4, R5, or R6) and the second two characters tell you the expected seed of the favored team. Thus the first row is R1W1, identifying the Round 1 game played in the W bracket, where the favored team is the 1 seed. As a further example, the R2W1 slot indicates the Round 2 game that would have the 1 seed from the W bracket, assuming that all favored teams have won up to that point. The slot names are different for the final two rounds, where R5WX identifies the national semifinal game between the winners of regions W and X, and R5YZ identifies the national semifinal game between the winners of regions Y and Z, and R6CH identifies the championship game. The "slot" value is used in other columns in order to represent the advancement and pairings of winners of previous games.
"strongseed" - this indicates the expected stronger-seeded team that plays in this game. For Round 1 games, a team seed is identified in this column (as listed in the "seed" column in the tourney_seeds.csv file), whereas for subsequent games, a slot is identified in this column. In the first record of this file (slot R1W1), we see that seed W01 is the "strongseed", which during the 1985 tournament would have been Georgetown. Whereas for games from Round 2 or later, rather than a team seed, we will see a "slot" referenced in this column. So in the 33rd record of this file (slot R2W1), it tells us that the winners of slots R1W1 and R1W8 will face each other in Round 2. Of course, in the last few games of the tournament - the national semifinals and finals - it's not really meaningful to talk about a "strong seed" or "weak seed", but those games are represented in the same format for the sake of uniformity.
"weakseed" - this indicates the expected weaker-seeded team that plays in

this game, assuming all favored teams have won so far. For Round 1 games, a team seed is identified in this column (as listed in the "seed" column in the tourney_seeds.csv file), whereas for subsequent games, a slot is identified in this column.