CSIT 6910 Independent Project

NCAA basketball tournament prediction

Jordy Domingos - jordydomingos@gmail.com
Supervisor : Dr David Rossiter

Content Table

1. Introduction

a. Context and motivation

As a Master's student that is not going to continue his studies toward a Phd degree, I needed to find an area in which I would like to work in the close future. Recently, I took a Data Mining course given by Dr Lei Chen and I found that area really interesting. After further thinking, I decided that I would put most of my energy into learning more concepts and techniques related to the Big Data field.
I managed to secure an internship at Airbus as a Big Data Engineer starting in september 2015 and while waiting patiently for the time to come, I decided to work on an small project related to the Big Data field and the work that I will have to do during my internship.
To find my project, I decided to go on kaggle.com, a website that hosts data mining competitions. I took inspiration from a contest where the goal was to predict the outcome of the games played in the NCAA basketball tournament[1] (http://www.kaggle.com/c/march-machine-learning-mania-2015) .

b. Project description

The goal of the project was to predict the 2015 NCAA Basketball tournament winner by predicting the outcome of all possible game between the qualified teams.
The NCAA Men's Division I Basketball Championship is a single-elimination tournament played each spring in the United States, currently featuring 68 college basketball teams, to determine the national championship of the major college basketball teams. The tournament, is organized by the National Collegiate Athletic Association (NCAA). 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.

To fully understand the project, a basic knowledge of basketball statistics is recommended. You can find a glossary on http://www.basketballstatmanager.com/stat-list.aspx.

c. Technical environment

I used **R studio and the R programming language**[2] for all the code that I wrote for this project.
Both of them were new to me.

[1] http://en.wikipedia.org/wiki/NCAA_Men%27s_Division_I_Basketball_Championship
[2] http://www.r-project.org/

2. Data set description

For this project I used 2 main source of data.
The majority of data that I had came from kaggle's contest.
The contest provided lots of data input but I am not going to describe them all. I will just talk about the ones that I found useful.

The full file list & information can be found at :
https://www.kaggle.com/c/march-machine-learning-mania-2015/data.

a. Kaggle's contest files

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 (starting day of the season & region names).

regular_season_detailed_results.csv
This file identifies the game-by-game detailed results from 2003 to 2015.
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):
The game statistics for each teams were:

fgm - field goals made	fga - field goals attempted	fgm3 - three pointers made	fga3 - three pointers attempted
ftm - free throws made	fta - free throws attempted	or - offensive rebounds	dr - defensive rebounds
ast - assists	to - turnovers	stl - steals	blk - blocks
pf - personal fouls			

Figure 1, Description of regular_season_detailed_results.csv column's name

tourney_detailed_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_detailed_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.

conference_affiliation_1996_2015.csv
This file lists all the teams and their conference affiliation from 1996 to 2015.

b. Files from external sources

To add more information in my data set, I took files hosted on https://rpiarchive.ncaa.org/default.aspx. This website stores information regarding previous NCAA championship and conference championship statistics from the past years. I decided to take information from 2009 to 2015. For each season the files that I took were :

Conference rankings.pdf
The file contains conference level information for the all the teams.

Conference name	Division 1 win/lost games summary	NON - Division 1 win/lost games summary	Division 1 winning percentage
Average opponent success rate	strength of schedule	opponents average strength of schedule	road success percentage
road RPI	Normal RPI		

Figure 2, Description of information contained in conference ranking.pdf

Nitty-Gritty.pdf
The file contains the most important aspects or practical details of a season for each team

RPI rank	Average opponent RPI rank	Average opponent RPI	Division 1 win/loss summary
Non-conf RPI rank	Non-conf RPI	Conf Record	Road game win/loss summary

strength of schedule	Non-conf strength of schedule	opponents average strength of schedule	Non-conf opponents average strength of schedule
win/loss summary against 1 - 50 top teams based on RPI	win/loss summary against 51 - 100 top teams based on RPI	win/loss summary against 101 - 200 top teams based on RPI	win/loss summary against top 100 teams based on RPI

Figure 3, Description of information contained in Nitty-Gritty.pdf

Team Ranking (all games).pdf

The file contains a resume of the performance of each team involved in a division 1 championship.
The statistics for each teams were:

Division 1 win/lost games summary	NON - Division 1 win/lost games summary	Division 1 winning percentage	strength of schedule
opponents average strength of schedule	road success percentage	road RPI	

Figure 4, Description of information contained in Team Ranking(all games).pdf

As you can see these files contains lots of redundancies.
Also, you may have noticed that not all files are in CSV format.
In the next section, I will describe the process used to create my structured data from these data.

3. Data pre-processing

In order to have a consistent data set to and make the knowledge discovery possible, I needed to transform the raw data that I had into a structured data set. The file taken from Kaggle were already structured. The only problem was that to get the full details of a team strength and weakness, lots of processing operation were needed. Indeed, I needed to compute a year resume of the team from the data that I had.

All the other files were unstructured which means that I had to take the information that I needed from them and then make sure that they respected the format of the other files.

a. Processing on structured files

The information that I had for each team were at a game level. In order to know the long term team's performance, I used all these information to create a season resume for each team. This resume contains an average of all games statistics of a team.
This allowed me to have fast and easy way to compare 2 teams.

b. Processing on unstructured files

My goal was to extract some information from the unstructured file and add them in the file previously created that contained the teams' resume. The information taken from the files are highlighted in green.

Conference rankings.pdf

Conference name	Division 1 win/lost games summary	NON - Division 1 win/lost games summary	Division 1 winning percentage
Average opponent success rate	strength of schedule	opponents average strength of schedule	road success percentage
road RPI	Normal RPI		

Figure 5, Description of information contained in conference ranking.pdf with extracted information highlighted

Nitty-Gritty.pdf

RPI rank	Average opponent RPI rank	Average opponent RPI	Division 1 win/loss summary
Non-conf RPI rank	Non-conf RPI	Conference win/lost games summary	Road game win/loss summary
strength of schedule	Non-conf strength of schedule	opponents average strength of schedule	Non-conf opponents average strength of schedule
win/loss summary against 1 - 50 top teams based on RPI	win/loss summary against 5 1 - 100 top teams based on RPI	win/loss summary against 101 - 200 top teams based on RPI	win/loss summary against top 100 teams based on RPI

Figure 6, Description of information contained in conference Nitty-Gritty.pdf with extracted information highlighted

Team Ranking (all games).pdf

Division 1 win/lost games summary	NON - Division 1 win/lost games summary	Division 1 winning percentage	strength of schedule
opponents average strength of schedule	road success percentage	road RPI	RPI

Figure 7, Description of information contained in conference Team Ranking(all games).pdf with extracted information highlighted

i. Spelling issues

The first problem that I face with the data from the external source was that on the web, a team can have its name spelled differently according to the website preference. You can use the nickname of the team, some abbreviation (eg.. "st" instead of "saint") for example.
For that I built a file that contains most of the way of spelling a team coupled with the corresponding team ID.

team_spelling.csv
This file links most of the team spelling possibilities with their id in the file teams.csv.

ii. Handling pdf format

The second problem was that the information were in pdf format. To convert the pdf information to csv I used **Tabula**[3], which allowed me to select part of the pdf file and export it to csv.
From my opinion, this is a good tool when you have a small amount of data, or when all the data that you need have the same pattern on each page. In my case, the pattern were similar but I had a lot of pages so it took me a lot of time to gather the information.

iii. Structuring the files

When the information were gathered, they were no formatting at all. Moreover, some data needed to be dismiss. I created a program to format all the data and add them to the structured data.

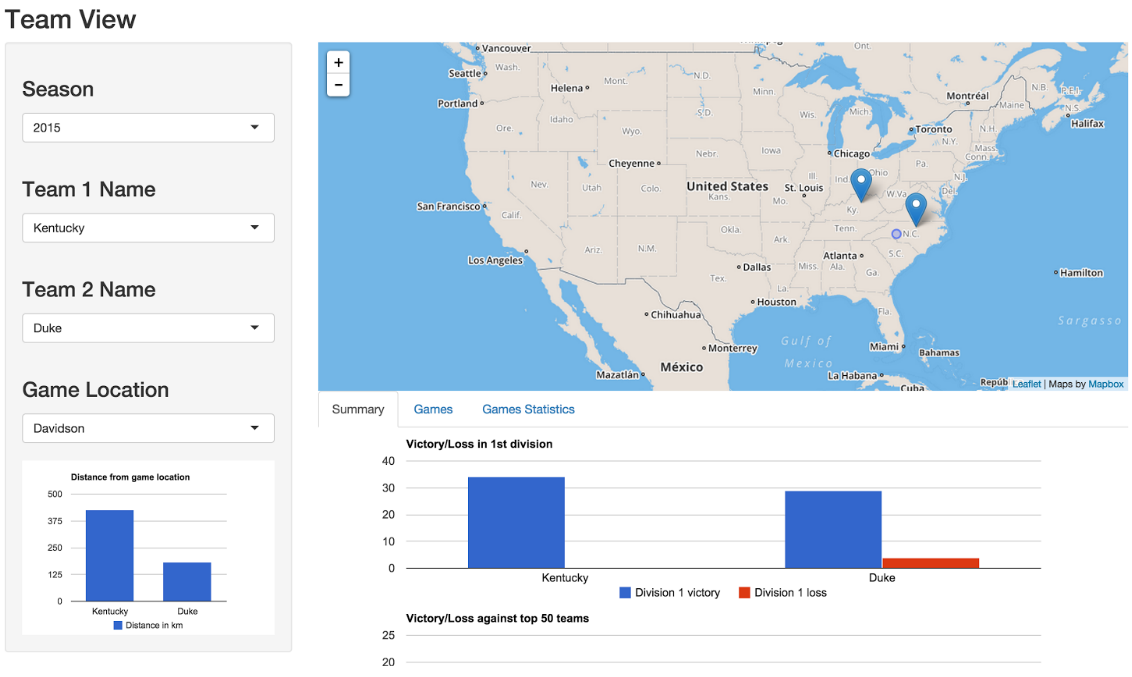
[3] http://tabula.technology/

4. Data visualization

As I said in the introduction, to predict the winner, I needed to have a way to compare 2 teams. To do so, I decided to build a web based interactive GUI that will allow me to navigate through all the data that I have for 2 given teams.
To build this GUI I used an R package named **shiny**[4] which is a web application framework for R.
The power of that package comes from the fact that it converts R code, to HTML, CSS & JavaScript**.** It also create a web service that allow you to visualize your app on your web browser.

a. GUI description

The left panel of the GUI allow you to select a season from 2010 to 2015, select two team and a game location. The bar plot of the left panel shows the distance in kilometers between the game location at the two teams university. You can also visualise the distance using the map where the two teams and the game place are plot. As an example I made a comparison between Kentucky and Duke of the season 2015.



Team View

Season: 2015

Team 1 Name: Kentucky

Team 2 Name: Duke

Game Location: Davidson

Figure 8, Team comparator main view

In the main view you have 3 tabs that I am going to describe now.

[4] http://shiny.rstudio.com/

i. Summary

This tab panel contains a brief season summary of the teams.

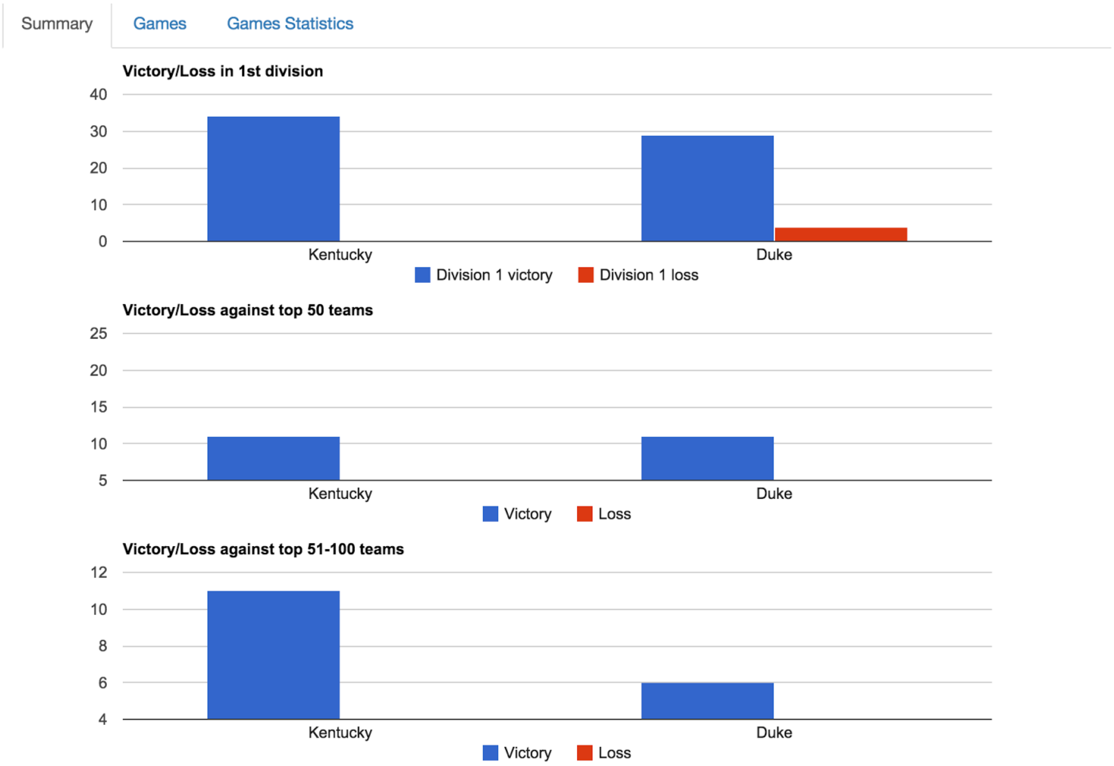


Figure 9, Team comparator summary view (Part 1)

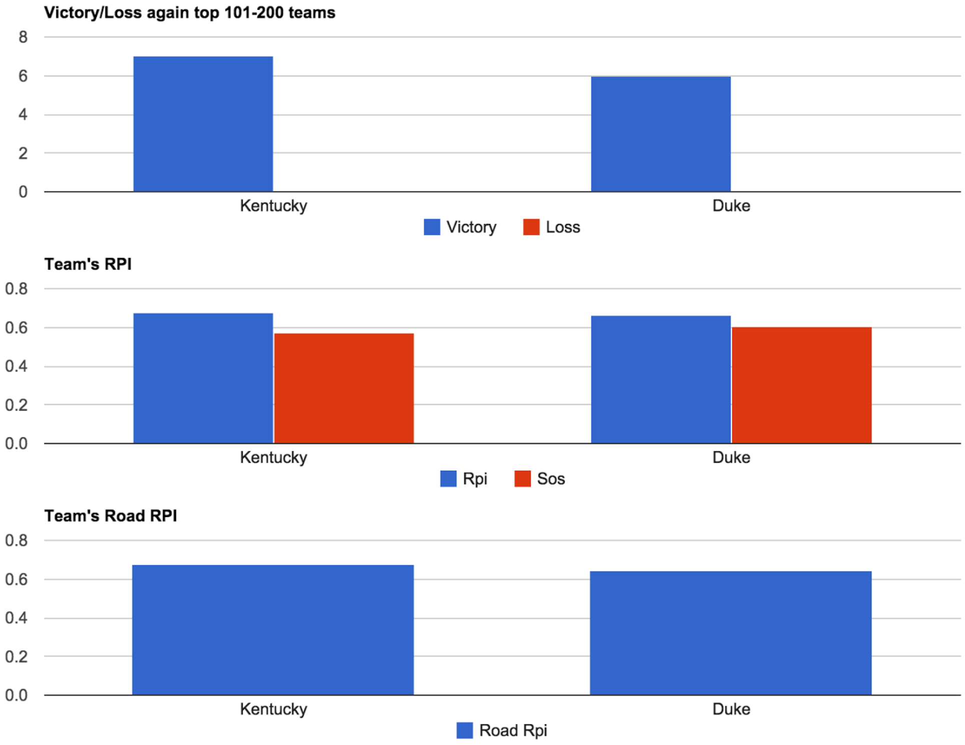
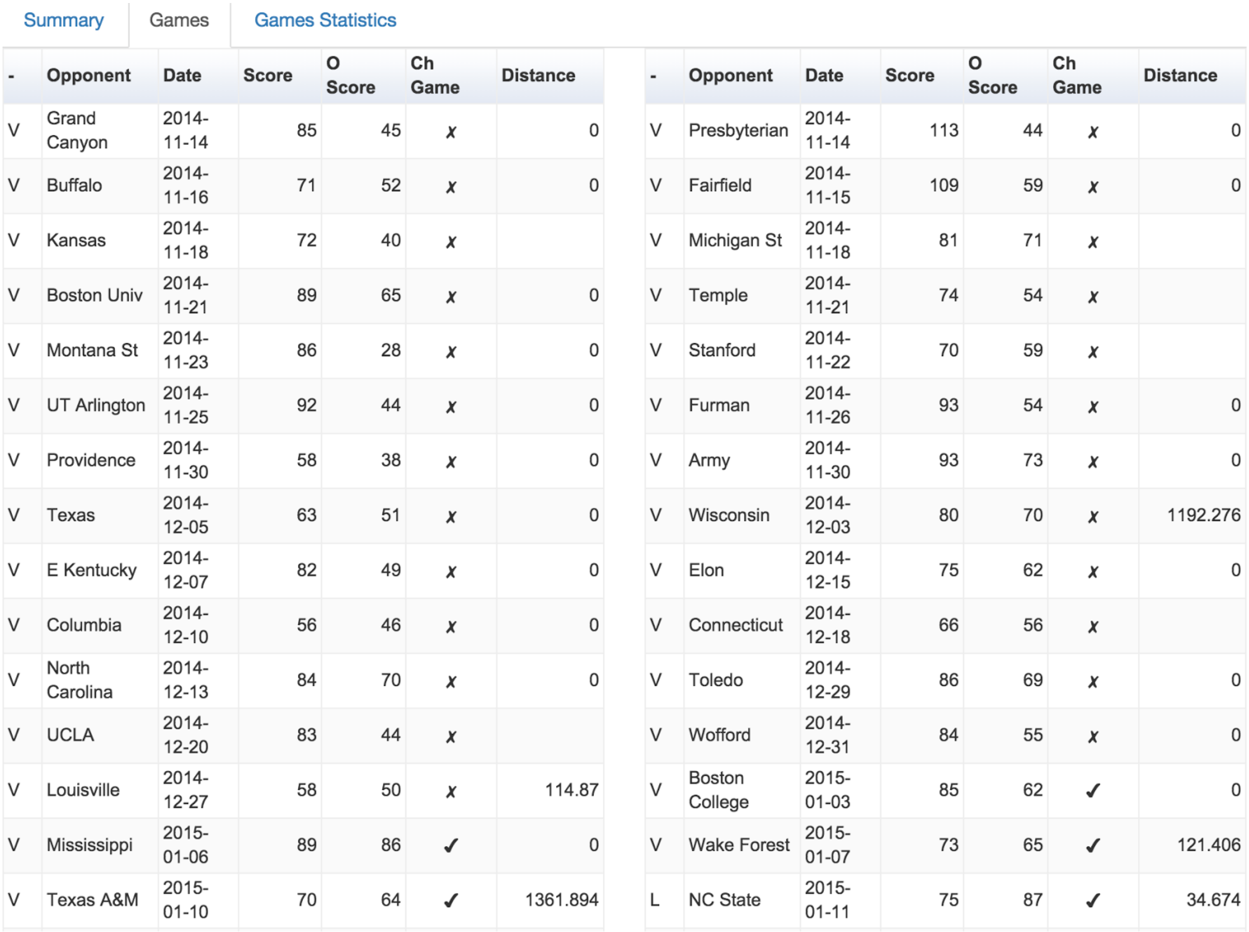


Figure 10, Team comparator summary view (Part 2)

ii. Games

This tab panel contains basic information regarding all games played by a team. (Only the first ones are shown here)



-	Opponent	Date	Score	O Score	Ch Game	Distance
V	Grand Canyon	2014-11-14	85	45	✗	0
V	Buffalo	2014-11-16	71	52	✗	0
V	Kansas	2014-11-18	72	40	✗	
V	Boston Univ	2014-11-21	89	65	✗	0
V	Montana St	2014-11-23	86	28	✗	0
V	UT Arlington	2014-11-25	92	44	✗	0
V	Providence	2014-11-30	58	38	✗	0
V	Texas	2014-12-05	63	51	✗	0
V	E Kentucky	2014-12-07	82	49	✗	0
V	Columbia	2014-12-10	56	46	✗	0
V	North Carolina	2014-12-13	84	70	✗	0
V	UCLA	2014-12-20	83	44	✗	
V	Louisville	2014-12-27	58	50	✗	114.87
V	Mississippi	2015-01-06	89	86	✓	0
V	Texas A&M	2015-01-10	70	64	✓	1361.894

-	Opponent	Date	Score	O Score	Ch Game	Distance
V	Presbyterian	2014-11-14	113	44	✗	0
V	Fairfield	2014-11-15	109	59	✗	0
V	Michigan St	2014-11-18	81	71	✗	
V	Temple	2014-11-21	74	54	✗	
V	Stanford	2014-11-22	70	59	✗	
V	Furman	2014-11-26	93	54	✗	0
V	Army	2014-11-30	93	73	✗	0
V	Wisconsin	2014-12-03	80	70	✗	1192.276
V	Elon	2014-12-15	75	62	✗	0
V	Connecticut	2014-12-18	66	56	✗	
V	Toledo	2014-12-29	86	69	✗	0
V	Wofford	2014-12-31	84	55	✗	0
V	Boston College	2015-01-03	85	62	✓	0
V	Wake Forest	2015-01-07	73	65	✓	121.406
L	NC State	2015-01-11	75	87	✓	34.674

Figure 11, Team comparator games view

iii. Games Statistics

This tab panel contains the average of all statistics present in the data set.

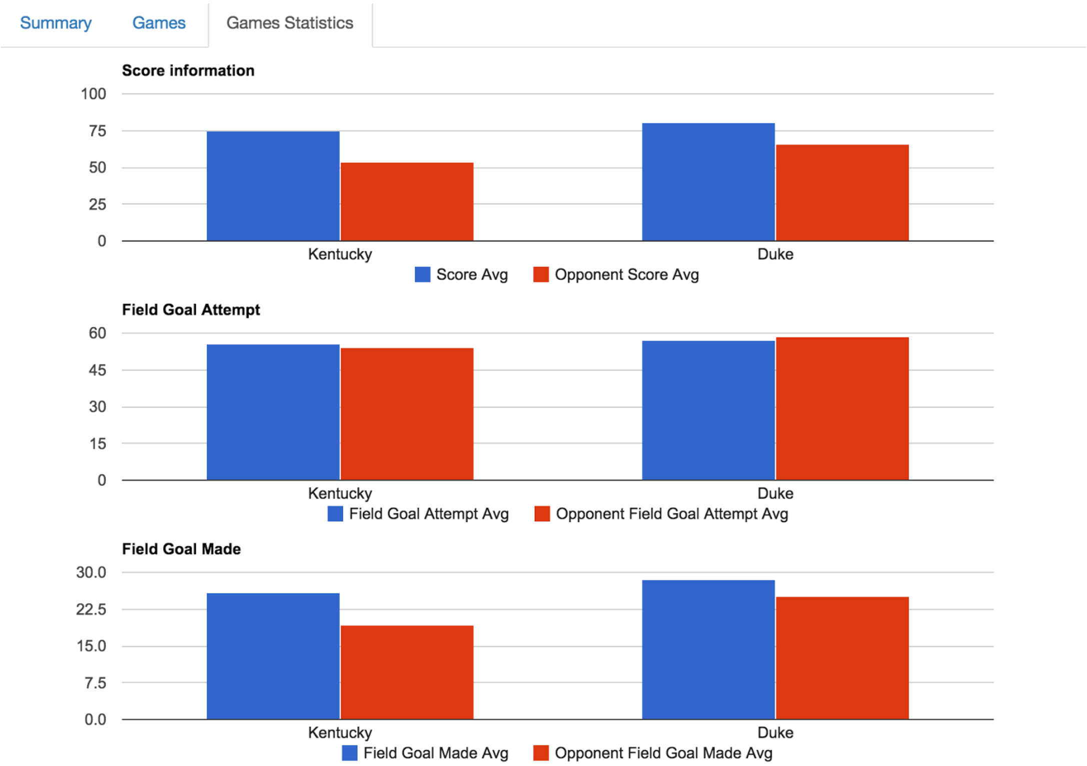


Figure 12, Team comparator games statistics view (Part 1)

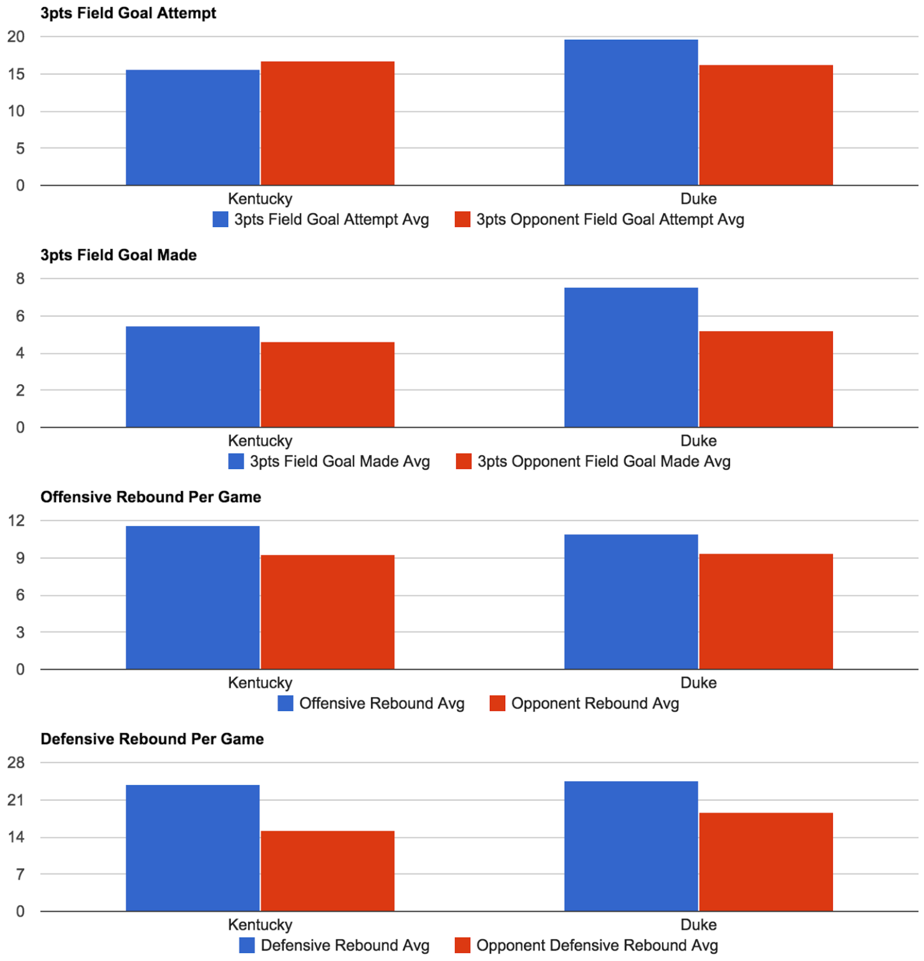


Figure 13, Team comparator games statistics view (Part 2)

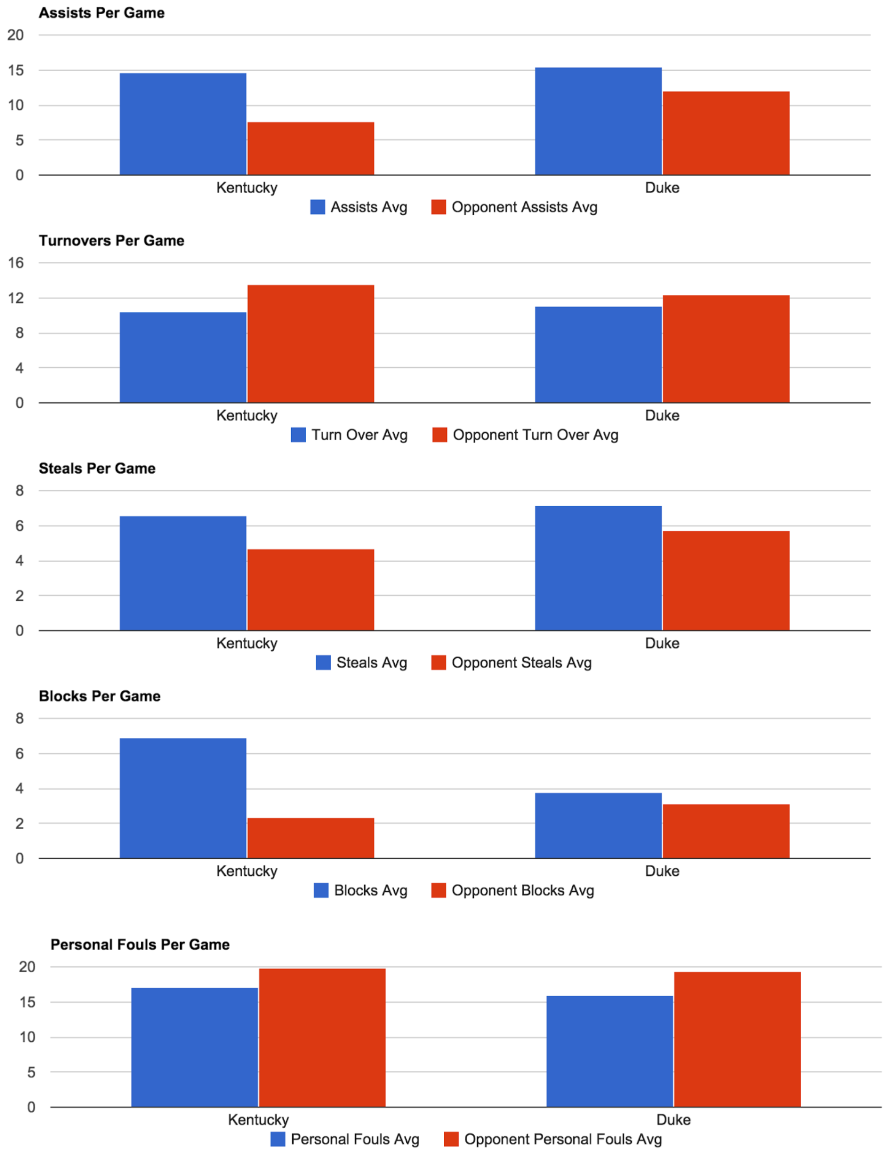


Figure 14, Team comparator games statistics view (Part 3)

5. Dimensionality Reduction

Below is the list of feature that I create for a particular team.
I created 40 features for each teams so the total number of feature would be 81 (2 (number of teams involved in a game) * 40 (number of features) + 1 (game_id)).

Team Game Information

team_id The id of the current team	**div_win** The percentage of games won against division's 1 teams	**march_win** The percentage of games won against division's 1 teams in March	**seed** The seed assigned by the NCAA committee to the team	**sos** Strength of schedule
road_success The percentage of games won away.	**road_rpi** The road Rating Percentage Index	**rpi** The Rating Percentage Index	**o_success** The average percentage of games won by the opponents	**win_1_50** The percentage of games won against top 50 rpi teams
win_51_100 The percentage of games won against team ranked from 51 to 100 based on RPI	**win_101_200** The percentage of games won against team ranked from 101 to 200 based on RPI	**score** Team's average score	**o_score** Opponent team's average score against the team	

Figure 15, Team game information table

Game Statistics

fgm Field goal made	**fga** Field goal attempt	**fgm3** 3pts made	**fga3** 3pts attempt
ftm Free throw made	**fta** Free throw attempt	**or** Offensive rebound	**dr** Defensive rebound
ast Assists	**to** Turnover	**stl** Steal	**blk** Block
pf Personal fouls			

Figure 16, Game statistics information table

o_fgm Opponents average field goal made	**o_fga** Opponents average Field goal attempt	**o_fgm3** Opponents average 3pts made	**o_fga3** Opponents average 3pts attempt
o_ftm Opponents average Free throw made	**o_fta** Opponents average Free throw attempt	**o_or** Opponents average Offensive rebound	**o_dr** Opponents average Defensive rebound
o_ast Opponents average Assists	**o_to** Opponents average Turnover	**o_stl** Opponents average Steal	**o_blk** Opponents average Block
o_pf Opponents average Personal fouls			

Figure 17, Opponent team game statistics information table

The problem is that when the dimensionality is too high the processing time increase.
I computed the differences between the same variable in the two teams. This helped me reducing the number by almost half (81 to 41).

6.Model Building

For this part I built a data set that contained all conference and championship games starting from 2011 to 2014.
I used random forests utilizing conditional inference trees as base learners as classifier with 10-cross-validation.

a. Feature selection

To reduce even more the number of features I used a wrapper feature selection method.
Indeed, features selection helps to :

- improved model interpretability,
- shorter training times,
- enhanced generalisation by reducing overfitting.

Here are the definitions of the variable importance measures. The first measure is computed from permuting OOB data: For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences. If the standard deviation of the differences is equal to 0 for a variable, the division is not done (but the average is almost always equal to 0 in that case).
The second measure is the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index. For regression, it is measured by residual sum of squares.

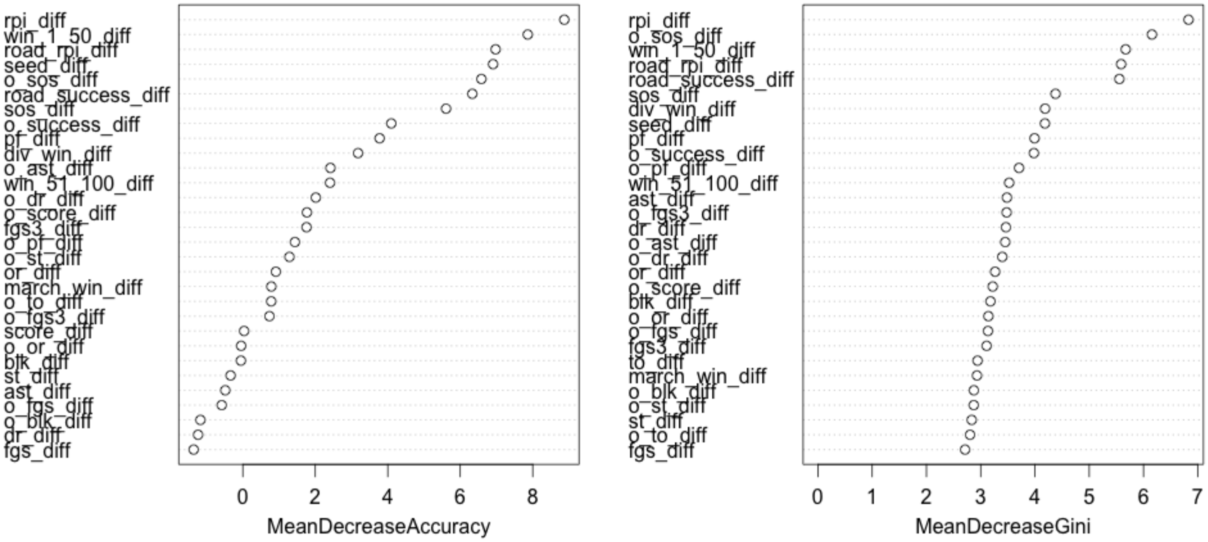


Figure 18, an example of importance plot

I used the mean random forest importance measurement with k-fold validation and decided to keep only the feature that has an importance higher than 1.

seed_diff	4.834569886
rpi_diff	4.825974902
o_sos_diff	4.64209834
win_1_50_diff	4.479213369
road_rpi_diff	4.018153216
sos_diff	3.098175027
o_success_diff	3.074533696
road_success_diff	3.035722491
div_win_diff	2.700464131
pf_diff	2.098696798
blk_diff	2.045316104
o_fgs3_diff	1.86514727
win_51_100_diff	1.49855353
o_pf_diff	1.421167232
or_diff	1.310316155
o_st_diff	1.178643273
march_win_diff	0.61122382
o_ast_diff	0.578897207
o_score_diff	0.467375427
score_diff	0.461285623
o_fgs_diff	0.416509667
to_diff	0.232672976

st_diff	0.153982306
dr_diff	0.137961507
o_dr_diff	-0.093836977
ast_diff	-0.148056242
fgs3_diff	-0.281097716
o_to_diff	-0.302378505
o_blk_diff	-0.325062374
fgs_diff	-0.496643607
o_or_diff	-0.598168918

Figure 19, List of features with selected features highlighted

b. Parameters optimization

i. Number of randomly pre selected variables

Starting with the default value of mtry, I used the tuneRF function of the randomForest package that search for the optimal value (with respect to Out-of-Bag error estimate) of mtry for randomForest. Therefore I was able to find the best value for each model to build.

ii. Number of trees & depth of the trees

The number of trees is a difficult value to set. I chose to have 1000 trees created to keep the learning phase short and maximize the model accuracy.
Regarding the depth of trees, I decided to keep them unstopped and unpruned.

c. Validation Results

To validate my model, I computed the ROC AUC (area under the curve) score for all the fold generated by my 10-fold cross validation algorithm. Here are the value for each fold.

0.763392857
0.891826923
0.59375
0.563636364
0.746031746
0.648809524
0.693181818
0.785714286
0.527777778
0.732142857

Figure 20 ,ROC AUC values for the 10 fold cross validation

The mean ROC AUC was 0.70.

7. Results

The final step of the project was to test my model on the 2015 tournament games,
You can see above the results obtained with my classifier for each game as well as a tournament simulation where I try to predict the winner.

a. Prediction result

For the 2015 tournament, my predictor had an **ROC AUC of 0.69** using a binary classification and a **0.60 log loss** between the prediction and the results using random trees probabilities.

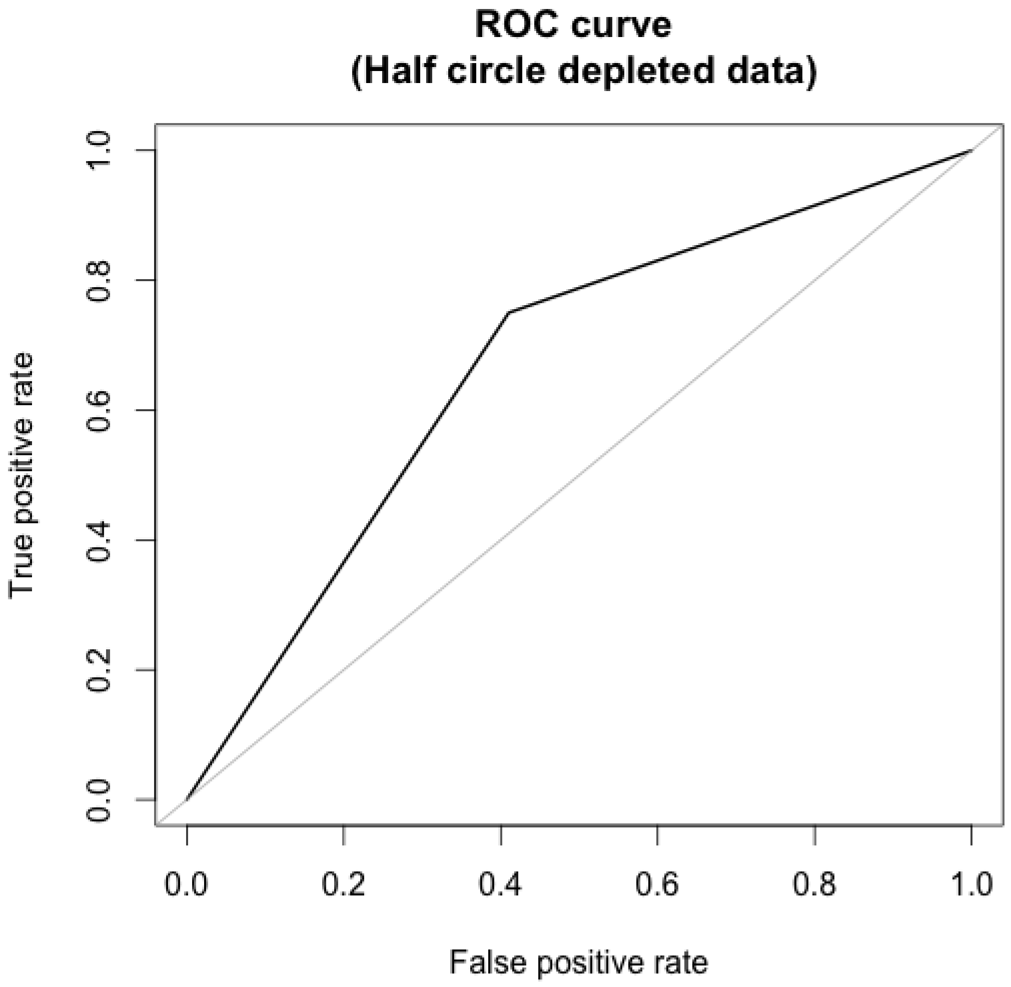


Figure 21, ROC curve

There exists multiple ways to improve these numbers and from my opinion some features regarding the players or the distance traveled by the teams could be helpful. Also, finding a formula that could describe the attack and defense potential of a team by taking into account the game details average that I had may help.

b. Tournament simulation

I tried to run a simulation using my prediction of all possible games in the tournament to see if I could find the potential winner. You can find my bracket below :

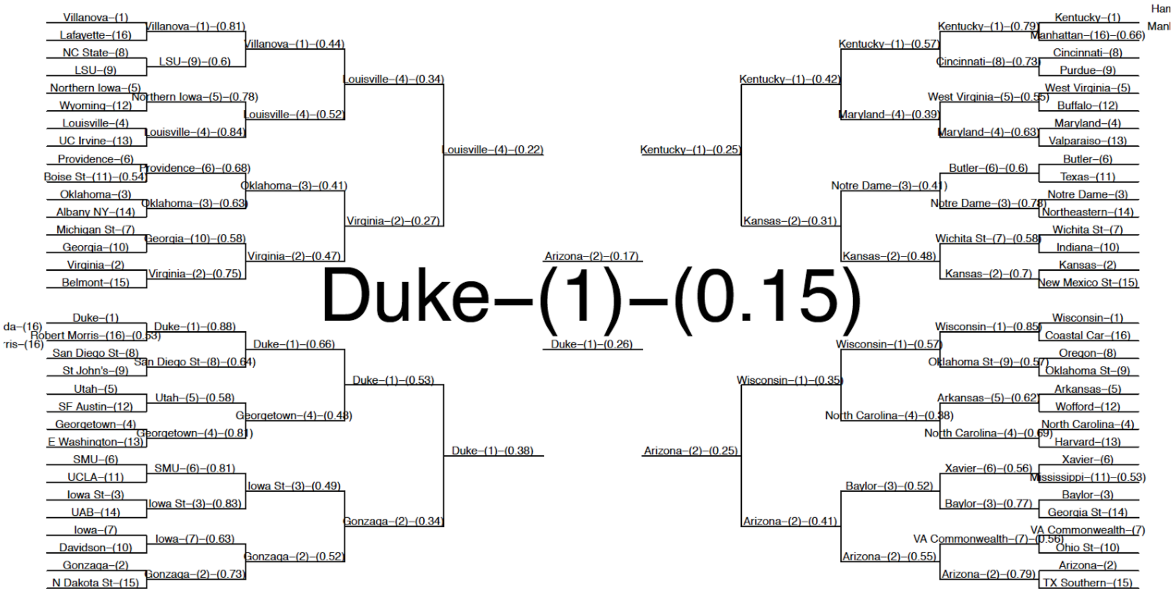


Figure 22, NCAA national tournament bracket prediction

8. Appendix

a. Minutes of the 1st meeting

Date :

Monday 2 March

Time :

2:00 pm

Place :

Room 3512

Attending :

Jordy Domingos
Dr. David Rossiter

Recorder :

Jordy Domingos

Approval of minutes

Previous minute approved.

Discussion Items

Detailed description of the project.

Choice of the first objectifs.

Targets :

- build web based interactive GUI system for text display of data
- investigate google chart api

Meeting adjournment and next meeting

The meeting was adjourned at 2:00 PM. The next meeting will be held in 3 weeks.

b. Minutes of the 2nd meeting

Date :

Monday 23 March

Time :

2:00 pm

Place :

Room 3512

Attending :

Jordy Domingos
Dr. David Rossiter

Recorder :

Jordy Domingos

Approval of minutes :

Previous minute approved.

Discussion Items :

Description of the work done (data pre-processing & investigation on google chart API)

Targets for the next meeting :

- build web based interactive GUI system

Meeting adjournment and next meeting :

The meeting was adjourned at 2:00 PM. The next meeting will be held in April.

c. Minutes of the 3rd meeting

Date :

Friday 10 April

Time :

12 midday

Place :

Room 3512

Attending :

Jordy Domingos
Dr. David Rossiter

Recorder :

Jordy Domingos

Approval of minutes ;

Previous minute approved.

Discussion Items ;

Description of the web GUI & data processing added

Targets for the next meeting :

- Finish the prediction phase

Meeting adjournment and next meeting ;

The meeting was adjourned at 12:30 PM. The next meeting will be held in 2 weeks.

d. Minutes of the 4th meeting

Date :

Friday 24 April

Time :

3:30 pm

Place :

Room 3512

Attending :

Jordy Domingos
Professor David Rossiter

Recorder :

Jordy Domingos

Approval of minutes :

Previous minute approved.

Discussion Items :

Description of the feature selection and final results.

Meeting adjournment and next meeting :

The meeting was adjourned at 4:00 PM. This was the last meeting..