**NCAA Bracket Predictor**

# Sprint 2

Team Members:

Kevin Brosam, Nate Lang, John Hattas, Alex Berkhout, Matt Petter

**User Stories:**

Nate is looking for a way to compare multiple tournaments at a time and find a way to predict the winner based on previous tournaments. He wants to be able to select several indicators and have the program find the ideal combination for the best possible bracket. He also wants to be able to use advanced statistics such as SRS or RPI.

Alex is interested in seeing how different indicators affect the predicted bracket. He wants to be able to select many different combinations of indicators and weights to vary the output bracket, so he can compare different possibilities. He would also like to be able to compare these outputted brackets against past seasons to get a general idea of how accurate it might be. Most importantly, he needs many different options for statistics that he would be able to use as indicators.

Charles is a fan of NBA basketball, but is not very familiar with college basketball. He understands how brackets work, so all he needs to know is how each team in a matchup compares to each other. The website gives him all the important metrics for evaluating how good each team is, which Charles really appreciates. Since he is familiar with how basketball works, he also wants to see each team’s strength of schedule, their record, and how well they did in previous years of the tournament, which the website also provides.

John wants to test combinations of statistics to see how well they would have performed during the 2014-2015 season. He enters in the season and which indicators he wants, then runs the software to figure out which combination of indicators yielded the best results, and how many points/games correct he achieved for that season with those indicators.

John figured out which indicators worked well for the 2014-2015 season, now he wants to try it out with different weights for the indicators. John can put in the weights for each indicator to get an even better prediction than before.

**Task Cards:**

Get Kaggle data and adapt code to use it

* Data from Kaggle’s yearly NCAA ML Competition
* <https://www.kaggle.com/c/mens-machine-learning-competition-2018>
* We need Kaggle data because our last dataset included data from tournament, which is unreasonable since you don’t have this data to predict the tournament.

Create SRS, SOS, RPI, stat columns

* We want to add these statistics to our data.
* SRS is Simple Rating System
* RPI is Ratings Percentage Index
* SOS is Strength of Schedule

Refine code and optimize for better run time

* Simplify code for readability
* Add more comments
* Move things into functions to make actual simulation shorter and more clear

Implement a basic machine learning algorithm

* This will determine the best combination of indicators and weights
* We \*kind of\* did this by testing every combination of indicators, but not weights.
* Need to do more in next sprint

Run basic machine learning algorithm for multiple years

* Same as above, need to do more next time

Calculate and add extra stat columns to our dataset

* These include

Have multiple working indicators

* Be able to use multiple indicators with each other

Have a weighting system for indicators

* Be able to assign different weights for the indicators.
* We realized we need to normalize the indicators so one does not dominate the other. This is our first plan of action for next sprint.

Brain storm about “intangible stats”

* Unique stats that we might create to help us predict

**Sprint Backlog:**

|  |  |  |
| --- | --- | --- |
| Task | Priority [1-10 (1 being lowest)] | Completed(Y/N) |
| Create extra stats column | 7 | Y |
| Weighting system for indicators | 5 | Y |
| Basic Machine Learning algorithm | 4 | N not really |
| Multiple working indicators | 10 | Y |
| Create SRS, SOS, RPI methods | 7 | Y (Partly) |
| Optimize code | 7 | Y |
| Import Kaggle Data, adapt code | 10 | Y |
| Run against multiple years | 3 | Y |
| Brain storm about “intangible stats” | 4 | N |

**Product Backlog:**

|  |  |  |
| --- | --- | --- |
| Task | Priority [1-10 (1 being lowest)] | Completed(Y/N) |
| Develop an algorithm that predicts previous tournaments results | 1 | N |
| Integrate more advanced statistics | 7 | Y |
| Create picture of the bracket with appropriate teams | 7 | N |
| Collect Data | 10 | Y |
| Have a basic working model | 10 | Y |
| Back test for better prediction results | 4 | N |
| Potentially display through HTML | 1 | N |
| Update for 2018 tournament | 3 | Y |
| Compare different basic algorithms to find the easiest while not losing accuracy | 8 | N |
| Display data in charts and tables, potentially using R | 5 | N |
| Add location as one of the indicator | 2 (if reasonably possible) | N |
| Create User Interface | 4 | N |

## Sprint Retrospective

Overall this Sprint was a success and we made good progress on the project. It may not look like we did because the backlog has uncompleted tasks but the main parts of the project we needed to accomplish were completed. Spring Break was during this sprint so it went by in an instant. We still got a ton of work done both over Spring Break and after. The main portion of the progress this sprint was converting the Kaggle data to our project. Essentially, we are now using a whole new data set which so far has been more accurate. The biggest leap with this new data set is that there was a large portion of missing data depending on if a team went winless or undefeated. This stemmed from the fact that there were no winning or winning opponent stats, so we had to devise a method to produce these stats accurately. The data was also formatted very differently so we had to adapt our code to be able to interact with it successfully.

On the agile side of things, we were more consistent then Sprint 1 on our daily scrums. They were not “daily” but they occurred more often then they did for Sprint 1. We created a bot in our discord chat to remind us to do the daily sprint. We would just talk about what we have done and what we are going to do to accomplish our tasks. We paired programmed more and we adapted to what we needed to do for the project. When we switched data sets we were able to reuse most of our code which saved us time and we spent a fair amount of time this Sprint optimizing it. One of the biggest inhibitions to our project is the amount of time each back test takes, and by splitting up the code into chunks and trying to reduce some of the computational time we are trying to limit the amount of time each back test takes.

The results of this Sprint are technically the same as last, we do not have a perfect bracket but, we are getting closer to getting an algorithm which could theoretically work. The new data set helps because our last data set had improper data in it which artificially gave us better results. Going forward as a group we can make our daily scrums daily, paired program again, spend time refining the code to make it quicker to run and simply spend time back testing it for better results. We also hope to create a more visual product going into the next to sprints. Our code is all backend right now and we want to port it to HTML or some other form of visualizing the bracket. We are optimistic for the next sprint because this years NCAA bracket will be complete.