**NCAA Bracket Predictor**

# Sprint 3

Team Members:

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

**User Stories:**

Alex wants to be able to easily access the predictor and run it. A website that has the predictor hosted would be the best option. He wants to be able to select the year and what indicator/indicators will be used. He also wants it to show the matchups and each winner and loser. He also wants to be able to easily run multiple predictions.

Kevin wants to be able to see which indicators a stronger impact on which team wins than others. He prefers visual data that shows who wins but also wants to see some of the math behind the winners. A statistical approach to finding the winner would be best.

Nathan wants to know which indicator he is able to select from and the meaning of each indicator. He then wants a drop down menu to select the year. Furthermore, Nathan wants a visual representation of the winning team at the end and have the losing team crossed out.

**Task Cards:**

-Have Django code display a bracket depending on selected indicator

-Create graphs to give a visual model

-Develop a statistical method for variable removal

-Have Django code able to select which year to use

-Develop a machine learning approach to find best possible bracket

-Have Django show what % is right

-Have a list of possible indicators in the Django code

-Have the Django code display the bracket and which teams it selected

-Host website

**Sprint Backlog:**

|  |  |  |
| --- | --- | --- |
| Task | Priority [1-10 (1 being lowest)] | Completed(Y/N) |
| Have Django code display a bracket depending on selected indicator | 6 | Y |
| Create graphs to give visual model | 9 | Y |
| Develop a statistical method for variable removal | 4 | Y |
| Have Django code able to select which year to use | 8 | Y |
| Have Django display what % of bracket is correct | 4 | N |
| Host website | 10 | Y |
| Have Django code display what possible indicators it can use | 7 | Y |
| Have Django code display the bracket and which teams it picked | 7 | Y |
| Develop a machine learning approach to find best possible bracket | 7 | 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 | Y |
| Collect Data | 10 | Y |
| Have a basic working model | 10 | Y |
| Back test for better prediction results | 4 | N |
| Potentially display through HTML | 1 | Y |
| 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 | 5 | Y |
| Add location as one of the indicator | 2 (if reasonably possible) | N |
| Create User Interface | 4 | Y |

## Sprint Retrospective

The primary focus of this sprint was on creating the website for the product and using a statistical method for determining which variables are important. We succeeded in completing both of those and in the next sprint we will focus on refining the variables and updating the website with more functionality. One of the challenges we ran into this sprint was finding a website hosting service that was free and that was compatible with Django. It took us a fair amount of time to merge our locally hosted Django project onto the website and because of this we did not get as much done as we could have. The statistical analysis portion of the project went well, and we will continue to use and refine our approach to develop a better algorithm to determine our bracket.

On the agile side of things, we were having daily sprints about every two days which for this project is probably enough. We did paired programming both on the Django code and on the statistical code. I think we could do better at adding comments to the code, which is something that is tedious but helps the whole team understand the code better. In this sprint we did not have many test cases simply due to the nature of what we were working on. We did not have any set outputs for the code and as such there was not much that we could test against.