**Contact Information:**

Isaac Hess

Jacob Jashinsky

Alec Otterson

**Data Source:**

Kaggle.com: Historical NCAA data

**Introduction:**

We chose to use the data provided through Kaggle for a machine learning competition to predict the bracket of the 2018 NCAA basketball national championship. This problem is interesting because there are a lot of variables that go into sports competitions, including a lot of statistical data and random variables such as the human aspect. The challenge of putting an entire bracket together is particularly challenging, as things can change even from game to game. There is a large repository of data that can lend to helping predict likely outcomes, though, and it has become a popular pastime for people and groups to try their hands at predicting the likely final bracket.

We wanted to apply our machine learning knowledge to have a computer use all of the available data to come up with a bracket of its own, hopefully a fairly accurate one.

**Data Preparation:**

We were able to download our needed data from Kaggle.com easily enough, in csv form. We had to do some preprocessing of the data in order to get it into a format that we could use. Since the data is made up of two teams per game, we were uncertain how to go about trying to isolate a particular team and use their data against another team that was not a part of that game. We decided to use the data for a particular team from their last game and then take the data of the team they are playing and subtract one from the other. This gave us a dataset with a comparison of each team and whether that first team beat the second team or not. In order to achieve this, we had to split the original dataset by each team, determine which number game in the dataset the game was for each team, and then pull the data for each team and merge it back into a final dataset for the current game. Our final data has 20 columns and 79,553 rows.

**Mining:**

We started off by running the data through a bunch of algorithms to see which ones would perform the best. We tried decision trees, neural networks, bagging classifier with kneighbors, bagging classifier with neural networks, ada boost, gradient boosting, random forest, and support vector machines. Out of these, the neural network with 100 hidden layers, logistic activation, adam solver, learning rate of 0.001, max iterations of 400, momentum of 0.9, and early stopping of max 10 iterations with no change.

* + Describe the process you used to mine the data, or learn patterns from it. What algorithms did you try, why did you try them? What parameters did you use and why?
  + Make sure to discuss different things you tried along the way, even if they resulted in dead ends.
  + Highlight challenges you faced and how you overcame them.

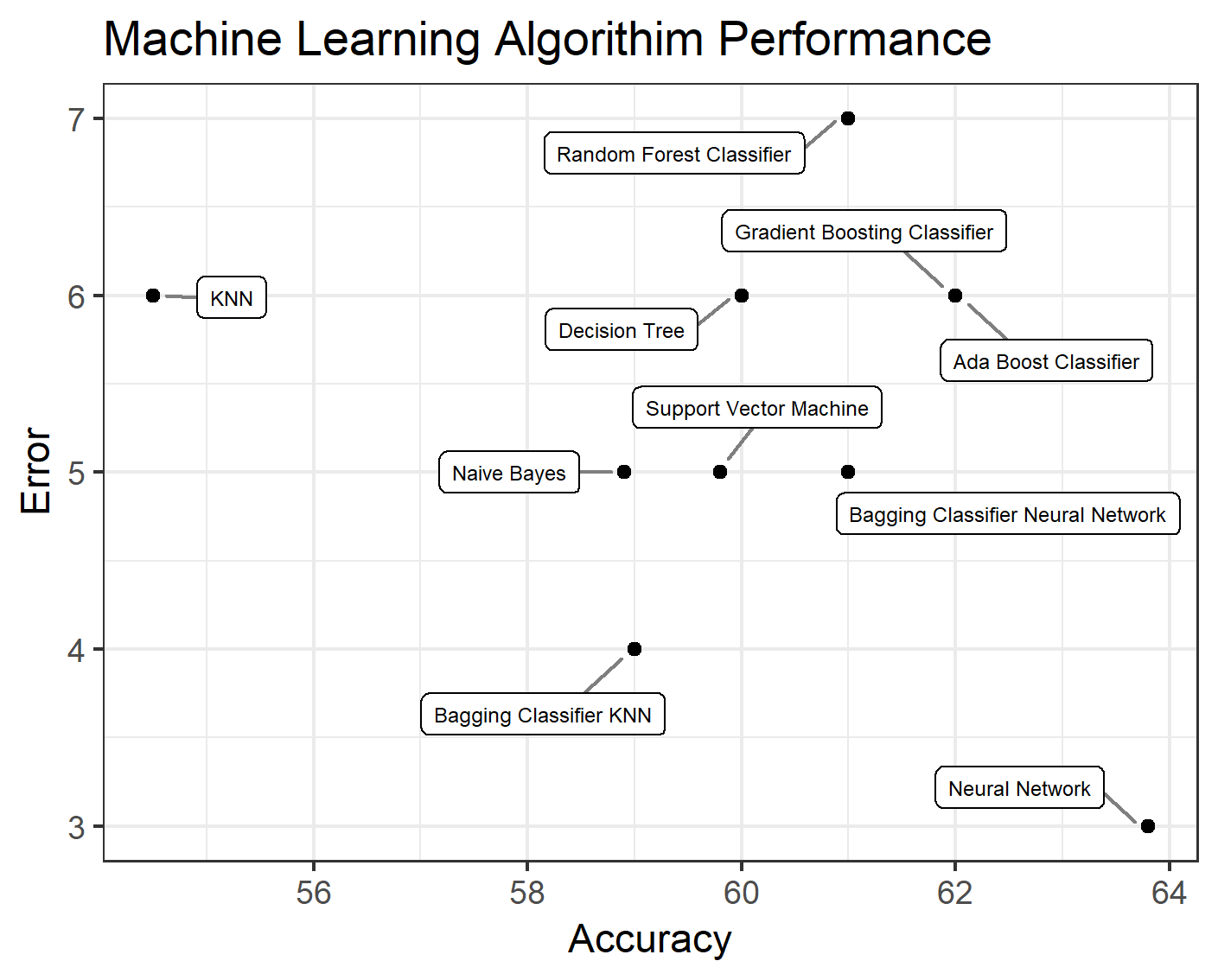
**Results:**

A sample of the original data with 2 rows for each of the columns.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Season | DayNum | | WTeamID | | WScore | | LTeamID | LScore | WLoc | NumOT | WFGM | WFGA |
| 2003 | 134 | | 1421 | | 92 | | 1411 | 84 | N | 1 | 32 | 69 |
| 2003 | 136 | | 1112 | | 80 | | 1436 | 51 | N | 0 | 31 | 66 |
| WFGM3 | WFGA3 | | WFTM | | WFTA | | WOR | WDR | WAst | WTO | WStl | WBlk |
| 11 | 29 | | 17 | | 26 | | 14 | 30 | 17 | 12 | 5 | 3 |
| 7 | 23 | | 11 | | 14 | | 11 | 36 | 22 | 16 | 10 | 7 |
| WPF | LFGM | | LFGA | | LFGM3 | | LFGA3 | LFTM | LFTA | LOR | LDR | LAst |
| 22 | 29 | | 67 | | 12 | | 31 | 14 | 31 | 17 | 28 | 16 |
| 8 | 20 | | 64 | | 4 | | 16 | 7 | 7 | 8 | 26 | 12 |
| LTO | LStl | LBlk | | LPF | |
| 15 | 5 | 0 | | 22 | |
| 17 | 10 | 3 | | 15 | |

A sample of the final data with 2 rows for each of the columns.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| team1\_win | | team1\_id | team2\_id | diff\_game\_order | | diff\_score | diff\_num\_ot | diff\_fgm | diff\_fga | diff\_fgm3 |
| 1 | | 1122 | 1404 | 27 | | 4 | 0 | -3 | -1 | -4 |
| 1 | | 1336 | 1321 | -4 | | -14 | 0 | -11 | -7 | 1 |
| diff\_fga3 | | diff\_ftm | diff\_fta | diff\_or | | diff\_dr | diff\_ast | diff\_to | diff\_stl | diff\_blk |
| -5 | | 14 | 14 | -1 | | 1 | -9 | -1 | 1 | -2 |
| -2 | | 7 | 10 | 3 | | -7 | -3 | 2 | 4 | -2 |
| diff\_pf | diff\_won\_previous\_game | | | |
| -1 | 0 | | | |
| -8 | 0 | | | |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Error** | **True positive** | **True negative** | **False negative** | **False positive** |
| Decision Tree | 60 | 6 | 61.5 | 63.7 | 38.5 | 36.3 |
| Neural Network | 63.8 | 3 | 65 | 62.6 | 35 | 37.4 |
| Bagging KNN | 59 | 4 | 61 | 60.9 | 39 | 39.1 |
| Ada Boost Classifier | 62 | 6 | 63.5 | 62.7 | 36.5 | 37.3 |
| Gradient Boosting Classifier | 62 | 6 | 64 | 63.5 | 36 | 36.5 |
| Random Forest Classifier | 61 | 7 | 62.9 | 62.9 | 37.1 | 37.1 |
| Support Vector Machine | 59.8 | 5 | 60 | 59 | 40 | 41 |
| Bagging Neural Network | 61 | 5 | 57.1 | 66.6 | 42.9 | 33.4 |
| Naive Bayes | 58.9 | 5 | 58.2 | 59 | 41.8 | 41 |
| KNN | 54.5 | 6 | 54.5 | 55 | 45.5 | 45 |

(insert chart of different algorithms used on data)

(insert chart of different options used with final algorithm on data)

(insert graph of iterations of final algorithm with final options on data)

* + Present the results that you obtained from the work done in the previous sections.
  + Include graphs and charts to support your findings. (Don't forget to include proper titles, axis labels, etc. for all graphs)

**Conclusions:**

* + Describe why your results could be of value to a business or stakeholder in your area. What would they know or what could they do differently as a result of your work.
  + Explain why your results constitute something "interesting."
  + Don't forget to discuss potential limitations or ethical issues.

L**essons Learned:**

* + What did you learn from this project?
  + What would you do differently if you could start this project again?