**Using Multinomial Naïve Bayes to Predict the Outcome of a *Slay The Spire* Run**

**Kent Torell**

**Introduction:**

*Slay The Spire* is a deck-building, rogue-like, turn-based role playing game that has been gaining popularity since it was first released as an early access game in November of 2017. The game is fairly difficult, with only about 15-20% of all runs achieving victory, and, while a large portion of the success chance comes from tight gameplay, an even larger portion comes down to luck and deck composition. This project focuses on predicting the outcome of the final fight of each run (which at the inception of the project was the Act 3 boss) based solely on the composition of the deck going into the fight.

**Data:**

With permission from MegaCrit, the developers of *Slay The Spire,* I was able to obtain run data for approximately 188,000 runs. From these runs I only included runs that reached the boss of Act 3. This gave me about 30,000 runs with which to train and test a model.

While there is no actually limit to the number of cards that can go into the deck most runs usually have between 20 and 40 cards in them. The data that was provided for this project had runs consisting of card counts between 1 and 279 with an average deck size of 29.9 cards \*\*\*ADD BOX AND WHISKER HERE?\*\*\*. The features used for this project were the counts of cards in any given run. The number of cards implemented into the game when data was collected was 283; each card has an upgraded version as well, bringing the final count to 566 features.

In order to break the data into training and validation sets a random selection was chosen from the overall data. A simple random sample was taken with 80% of the data going towards training and 40% going towards validation. A random sample was acceptable because each run was completely independent of all other runs. This gave a training matrix of 23,534x566 and a validation set of 5,884 runs.

**Model:**

Each run is a sparse set of counts that produce a binary outcome (either a win or a loss). Because of this, the model chosen to predict the outcomes was a Multinomial Naïve Bayes (MNB). Naïve Bayes models are based on Bayes’ theorem which states that:

p(a|**b**) = p(a)p(**b**|a)/p(**b**)

The numerator is equivalent to p(a, **b**), whichcan also be written as:

p(b1­, b2, … bn, a) = p(b1|b2, b3, …, bn, a) p(b2|b3, b4, …, bn, a)… p(bn-1|bn, a) p(bn|a) p(a)

The naïve portion of Naïve Bayes refers to the assumption that all the features are independent of one another. Under this assumption the following equation holds:

p(bi|bi+1, …, bn, a) = p(bi, a)

which implies that

p(a|**b**) ∝ p(a)

For the purposes of predicting outcomes in this model the following equation holds:

Outcome = argmaxvictory/loss[p(victory)]