The main problems I found about this dataset are the many unbalanced classes. There are 28 classes where some classes have over 600 observations while several classes have 1 observation.

The basic idea of my strategy is to split the data into partitions based on the number of observations, run a machine learning algorithm on each partition, use the results to determine which class the new observation belongs to. I will now go into more detail for each of the parts.

1. Split the data into partitions.

I was thinking about splitting the data by number of observations. I would take the largest class, then add classes until the newest class is less than 10% of the new total. That class would start the next partition and we would repeat the process.

Next I wanted to use Synthetic Minority Oversampling Technique (SMOTE) to balance each partition.

1. Run a machine learning algorithm on each partition

Next I wanted to use a random forest algorithm and/or multinomial logistic regression on each partition. I am still looking for a way to measure the model’s certainty of the results. I want a probability that the model believes that the observation belongs to a certain class. I am thinking potentially about a Bayesian random forest so I can use a model odds ratio or something.

I was also thinking about using an ensemble between a multinomial logistic regression and a random forest. This ensemble will maximize the accuracy of the results of each partition.

1. Using the results to determine which class the new observation belongs to

To determine which class the observation belongs to, we would weight the probability that each partition level model generates. These weights would give more power to the larger, post SMOTE datasets to reflect the higher level of confidence where there are more observations.

We would then pick the class with the highest weighted probability.

This procedure is fairly complicated so I am looking to find ways to make the model simpler.