Machine Learning with Magic

Thomas Barnett and Samyak Jain

Overview:

Magic: the Gathering is one of most complex trading card games, with many different strategies and archetypes. The game consists of two players who take alternating turns until one of them wins. Decks contain many different card types, such as sorceries, enchantments, and creatures. Additionally, cards can have any of five different colors, including white, blue, black, red, and green. Just like any other strategy game, it is important to recognize one's role in the game. There is usually one player on the offensive, and one of the defensive; recognizing your role is key to making correct game decisions.

Magic decks fall under three primary archetypes: aggressive, control, and midrange decks. Aggressive decks want to play offensively from the beginning, and win as quickly as possible. Control decks play very defensively, winning in the long game. Midrange decks switch between the two positions as the game progresses. Our project takes the contents of a Magic deck as its input, and outputs which archetype the deck falls under. This machine learner would help an AI playing Magic know what its role should be by understanding which deck archetype it's playing.

Process:

We began our project by collecting the data by first downloading 500 decklists from MTGTop8.com, a website that archives decklists from recent tournaments. Then we wrote a script that would compile all of these decklists into one file, while also cross-referencing the data with Scryfall.com to get the values we needed for specific attributes. We did this with the help of the scrython wrapper, which allowed easy communication between our python script and the Scryfall website. Finally, we converted this data into two CSV files, so that we could easily import it into Weka. One of the CSV files contained the attribute values for the deck on each line, while the other had the frequency of each card for each deck. At this point, we were ready to start testing on Weka.

We decided to use Average cost of a card, Number of Creatures, Number of Lands, Number of Instants, Number of Sorceries, Number of Enchantments, Number of Artifacts, Number of Planeswalkers, Amount of Red, Amount of Blue, Amount of Black, Amount of White, Amount of Green. The Weka visualization of our attributes is shown below in Figure 1.

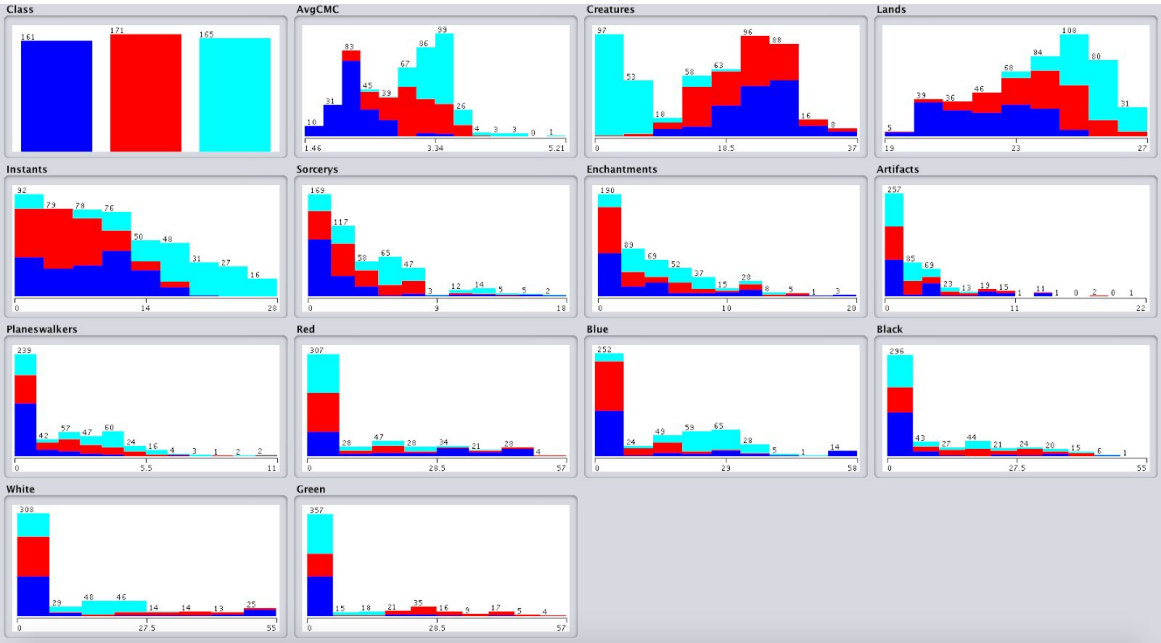

Figure 1: Attribute Data Set

Results:

The first algorithm we experimented with was C4.5, getting a promising initial accuracy of 91.3%, especially when compared to our ZeroR accuracy of 34.4%. We then used a pruned version, which improved our accuracy to 91.5% by reducing overfitting (tree shown in figure 2). The final step we took was using AdaBoost on our decision tree, giving us a final accuracy of 92.35.

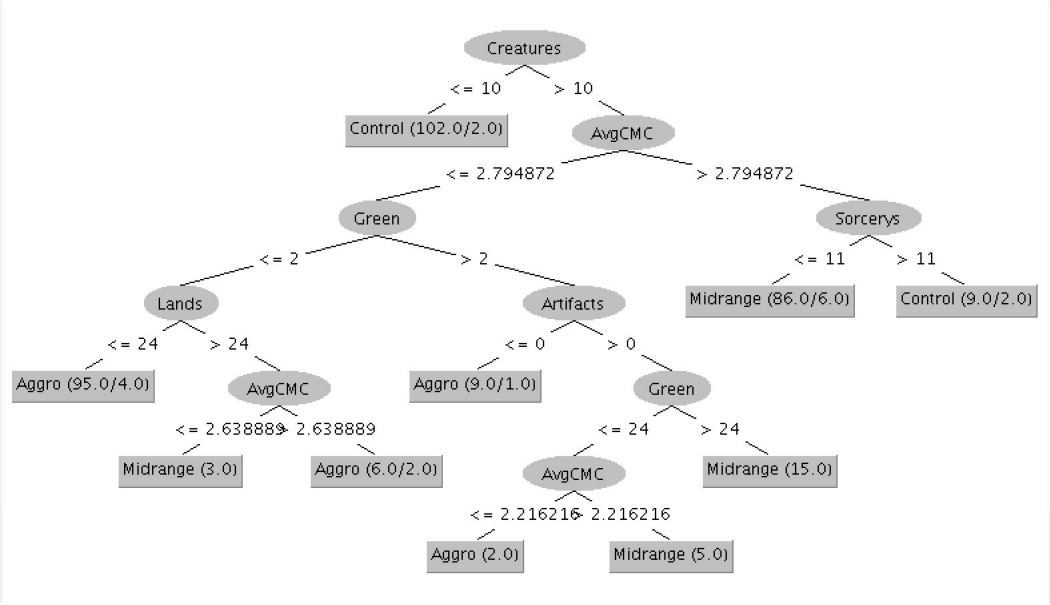

Figure 2: C4.5 Pruned Decision Tree

The next algorithm we tried was K Nearest Neighbors. We used Weka's CrossValidate tool to determine the optimal Nearest Neighbor algorithm used five Nearest Neighbors, which predicted the deck archetype with 90.9% accuracy. We also tried using distance weighting in our Nearest Neighbor algorithm, which weights neighbors closer the example higher than neighbors further away. This increased our accuracy up to 91.1% by reducing the effect of non-representative neighbors.

The Naive Bayes (bag of words) algorithm was the third machine learner we tested. We switched to using the frequency of each individual card in a given deck as our input for bag of words, similar to using a vector of word frequencies in a normal bag of words model. We tweaked some parameters to get higher accuracy. For example, we used the kernel estimator because our card frequencies were non-binary (you can have between 0 - 4 of any card). This algorithm yielded our highest accuracy yet, getting 93.2% accuracy with 10-fold cross validation.

If we were to continue this project, we would further develop the ability for an AI to use our machine learning when playing games. For example, taking in a select number of cards rather than a full decklist to try and determine the archetype.