Using Decision Trees to Predict the Price of Magic the Gathering Singles

Philippe Demontigny PHDEMONTIGNY@GMAIL.COM
University of Waterloo, 350 University Ave W, Waterloo, ON, N2L 6P4

Abstract

In this paper, we investigate the problem of predicting the price of Magic the Gathering (MTG) cards based on individual card attributes. This is the first project of its kind to tackle the complex economy of MTG singles using machine learning. The goal was to determine whether decision trees can be used to accurately classify cards given the current selection of cards in publication. We discuss the performance of three of the most promising models that we tested, each of which uses a random forest classifier with up to five possible labels. These labels represent subsets of MTG cards that have been grouped based on their average price across all internet retailers. We also discuss possible extensions of our work and different avenues of study that could yield interesting results with respect to MTG card classification.

1. Introduction

1.1. Background

Magic the Gathering (MTG) is one of the most widely played trading card games available today. It has been running successfully for over 20 years, and has tournaments that attract players from all over the world. However, MTG is also one of the more expensive hobbies, with tournament-winning decks often costing hundreds of dollars. Furthermore, new sets of cards are released three or more times per year, which forces players to constantly invest in new cards to remain competitive. For casual players, this poses a particular challenge because the strongest cards in each set are often outside their price range. It is therefore beneficial for these players to get ahead of the market, by investing early in cards that will reach a higher cost after they are released. Doing so will generate the most value for their dollar, and open the door for favorable trades when these cards are in high demand.

There are currently many resources available that help players understand the economics of Magic the Gathering cards. Websites such as `quietspeculation.com` are devoted entirely to producing articles about current trends in the MTG card market. There are also sites such as `mtgstocks.com` that act as stock exchanges in-miniature for the world of MTG singles, reporting daily changes in the market for individual cards. Professional players are also known to weigh in on why they believe certain cards will be more valuable than others. Craig Wescoe is one such player who releases a new "Financial Predictions" article for each set (Wescoe, 2014). Cited here is his most recent post for the set "Khans of Tarkir", which was released in September 2014.

To the best of our knowledge, no one has built an algorithm to predict the price of individual MTG cards yet. However, some work has been done investigating the price drivers of cards in general (Thompson, 2012). Other research has been conducted using linear regression to predict the success of entire decks of cards (R. Hau, 2012). Our focus is unique in that it specifically looks at the viability of the cards themselves, as opposed to studying how they interact with each other.

1.2. Our Project

Our goal with this project is to determine whether decision trees can be used to predict the price of MTG cards using *only* the information contained on the card itself. It is generally accepted that the price of a card is determined by its performance in major tournaments, and this can be easily seen by looking at the changes in price of tournament-level cards over time. In fact, the correlation between tournament performance and card price is so strong, that predicting price based

Project report for CS886: Applied Machine Learning. University of Waterloo, Fall 2014.

on tournament performance should not a challenging problem. A more interesting one is predicting a card's performance in tournaments *before* it even becomes available. Solving this problem means using only the information on the card itself, as no other information would be available before release. This challenge models the situation in Craig Wescoe's article (Wescoe, 2014), where he tries to guess what the tournament winners will be in upcoming sets using his own experience as a guide. With this project, we study whether a machine learning algorithm can find trends in MTG cards that escape even the pros, and use these trends to predict the selling price of these cards with similar accuracy.

1.3. Assumptions

The first and most critical assumption in this project is that card price is correlated with card strength. This would imply that price is a good metric to use for measuring the effectiveness of a particular card in tournaments. This is a fair assumption to make because the MTG card market operates under the familiar principles of supply and demand. Thus good cards will tend to have higher prices due to increased demand, and vice-versa. A weaker assumption that we will make is that card prices remain static after 3 months of availability. This is of course not true in general, as every so often unexpected cards end up in top-performing decklists or new winning combinations appear with the release of new sets. However, we believe that these situations are rare enough that they can be ignored for the purposes of this study. Our choice of 3 months as the timeframe is not itself significant; it is a heuristic for the amount of time it takes for new cards to settle into the metagame.

One final assumption is that cards are priced in increments of 0.25 USD. Thus for all of the cards we consider in this study, we will round the price to the nearest 0.25 USD. This is a reflection of how retailers choose to price their cards. For example, with common cards, stores will often sell the ones that nobody uses for around 0.25 USD, and the more popular ones will be priced between 0.5 and 1.00 USD. While it may be more convenient at times to use average card price as a metric, we are trying to predict costs in the real-world card market, which operates under a similar pricing scheme as described above.

2. Description of Available Data

Currently, there are over 24,000 Magic the Gathering cards in print. Fortunately for us, a complete list of the attributes for each card can be found at `mtgjson.com`, where their database is stored in JSON format and is up to date with the most recent set. After downloading the dataset and removing all cards that did not conform to the standard MTG format (such as cards in the “Vanguard” series), we were left with data on 24,470 cards. However, the data required more pruning before it could be useful. First, we decided to consider only a subset of the total card sets that have been released. Specifically, we ignored all sets that are considered “special” because they do not match the usual block construction. The full list of special sets can be found at (tcgplayer, 2014). Out of the remaining sets, we also chose to ignore all cards from Alpha, Beta, Unlimited, Revised, Arabian Nights, Antiquities, and Legends. We did this because each of these sets contains a high number of cards with hyper-inflated prices, meaning that the cards are expensive simply because they are old and therefore hard to find. Most cards in the Alpha, Beta, Revised, and Unlimited sets fall into this category, because very few copies of each card were produced. Other examples include “Library of Alexandria” from Arabian Nights, “Candelabra of Tawnos” from Antiquities, and “Karakas” from Legends, which each sell for well over $100.

In order to make our results as valid as possible, we also needed to remove all *exact reprints* and *functional reprints* from each set. An exact reprint of a card is another card with the same name and the same attributes that comes from a different set. A functional reprint is an exact reprint in all regards except that the name is different (Curse, 2014b). Leaving these values in the dataset would affect our measurements of accuracy, because they would be treated as the same card by the decision tree even though they may have different prices. To remedy this, we kept only the most recent reprints of these cards in our dataset. In doing so, we hoped to minimize the amount of price-inflation arising from the age of the card. After all pruning processes were completed, we were left with only 12,746 cards, almost half of what we started with.

For finding the price of cards, a number of possible sources are available. For example, at `magictraders.com`, they keep lists of the current market prices for MTG cards that can be downloaded in useful CSV formats (League, 2013). However, there are a number of problems with this dataset. The first and most unfortunate issue is that the data has not been updated in over a year, so many of the more recent sets are not represented. Furthermore, the data was acquired by aggregating the prices from sales made on ebay, and the number of averaged transactions per card varied from only a single sale for many cards, to over 20 sales for others. Thus this dataset is not appropriate for this project, as it does not accurately reflect the average price being offered by internet vendors. A better dataset are the MTG Price Guides that are published by tcgplayer.com (tcgplayer, 2014). These lists are updated in real-time with aggregated prices from almost all major online retailers, making it the ideal dataset for our study. We used the median prices across all lists for the 12,746 cards in our dataset, which we accessed on November 19, 2014.

3. Feature Construction

Before we could decide upon our feature set, we first had to identify which attributes of Magic the Gathering cards are the most relevant for determining their overall strength. In the MTGJSON dataset, there are many attributes that do not directly affect gameplay, such as the artist's name, the flavor text, and the set number. After removing all features of this type, we were left with the following collection of attributes: layout, manacost, cmc, colors, types, subtypes, rarity, text, power, toughness, and loyalty. We discuss how we translate each of these attributes into features in the sections below:

3.1. Layout

Most MTG cards have the same design layout, which we refer to as the "normal" layout. However, some cards have more interesting formats, such as split cards, double-sided cards, and leveling cards. In our dataset, this attribute determines which of these layouts a card uses, and is a categorical variable with 10 different categories. We translate this variable into 10 distinct boolean features, one for each type of layout. For example, the feature "layout: normal" has a value of 1 if a card has the standard layout and a value of 0 if it does not.

3.2. Manacost

All MTG cards have a "manacost", which is determined by a single number and/or a collection of mana symbols. There are a total of five different symbols corresponding to five different card colors. If a card's manacost has multiple symbols of different colors, then it is considered a "multicolored" card, and associates with other cards that share any of its colors. We use the manacost attribute to create a total of 8 features for each card. These features are referred to as: "color = White", "color = Blue", "color = Black", "color = Red", "color = Green", "num colors", "num mana symbols", and "cost contains X". The first five features are boolean variables that test whether a card associates with a given color. The "num colors" variable is an integer-valued feature that has range $[0, 5]$ and counts the number of colors that each card associates with (with no colors being a possibility). The "num mana symbols" feature counts the number of mana symbols in the manacost of a card. Note that this number is different from the number of colors, as there can be multiple mana symbols of the same color. Lastly, the "cost contains X" feature is a boolean feature that determines whether there is an "X" in the manacost of a card. This is an attribute that appears in many cards that can scale in power as the game progresses.

3.3. CMC

"CMC" stands for "Converted Mana Cost," which is the amount of mana required to use a card in Magic the Gathering. Mana is the currency of MTG, and comes in five different colors (plus colorless). Cards will often require certain colored mana before they can be used, which is denoted by the collection of mana symbols in the manacost (described above). The CMC is a non-negative integer that counts the total amount of mana required to use a card, regardless of color. In our model, we use this value as a single integer-valued feature.

3.4. Colors

The colors attribute is similar to the manacost attribute in that it lists the colors that a card associates with. There are five possible colors, and thus 32 possible combinations of colors (including colorless and "all five colors"). Thus in our model, we split this attribute into 32 separate boolean features, where each card has a value of 1 for the feature that describes its color combination, and a value of 0 for all of the others.

3.5. Types

A card's type determines how that card will be used during the course of a game. There are various card types, including Instant, Sorcery, Artifact, Creature, Enchantment, Land, and Planeswalker. In our model, we treat the type as a categorical variable, and divide it into a series of boolean features. Each card has a value of 1 for the feature that describes its type, and a value of 0 for all of the others.

3.6. Subtypes

Many cards in Magic the Gathering will also have a secondary type that describes it in more detail. These subtypes include values such as Arcane, Aura, Equipment, Human, Elf, Bear, etc.. However, there is a danger in using all of the subtypes in our classifier, as many of them are set-specific. This means that the subtypes are never found outside of the set they originally appeared in. If we want to be able to predict the value of future cards, we should limit ourselves to only those subtypes that appear in multiple sets, which implies that they may appear again in the future. Furthermore, to avoid over-fitting, we want to exclude subtypes that appear too infrequently. Taking both of these points into account, we cut our list down from 252 subtypes to 80, and added one boolean feature for each (include "N/A," for cards without a subtype). Note that in this case, cards may have multiple subtypes, and therefore multiple boolean features with a value of 1.

3.7. Rarity

The rarity of a card is related to the number of copies of that card that have been printed in total. Typically, Magic the Gathering cards are distributed in packs of 15. Of those 15 cards, one is always rare, three are uncommon, and 10-11 are common. Occasionally, the rare card will be replaced with a mythic rare card, which occurs in roughly 1 out of every 8 packs. Some cards fall outside of these traditional rarity categories, are are denoted in our dataset as “special.” To make rarity into a feature, we mapped the values described above to the integers [-1,3], where special cards are mapped to -1, and the rest are mapped in order of increasing rarity.

3.8. Power/Toughness

Some cards in MTG have a pair of inter-valued attributes called the power and the toughness that determine how strong the card is during one of the many stages of the game. It is important to note that these two values do not directly correlate to the overall strength of a card, as higher numbers often come with major drawbacks, while lower numbers are often be paired with powerful abilities. As these attributes are integers, having one feature for each is sufficient for our model. When a card does not have a power or a toughness score, then we set each of these features to -1.

3.9. Loyalty

A relatively new type of card in Magic the Gathering is the “planeswalker,” which has an attribute called loyalty. Loyalty is represented by a positive integer, so we include it as a single feature in our model. Once again, if a given card does not have a loyalty score, we set this feature to -1.

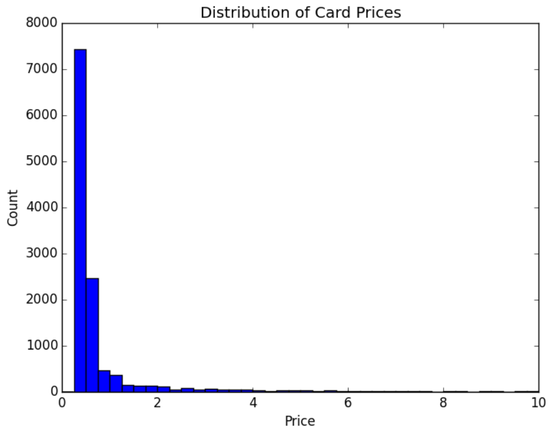
3.10. Text

Interpreting the textbox of a card is the most complicated aspect of this project. The textbox of a Magic the Gathering card provides the details of how that card functions in a game, and thus contains most of the relevant information for determining a card’s strength. Fortunately for us, the writers for MTG cards use a very specific language to describe game state and interactions. For example, a player’s deck is always referred to as a “library,” using a card is referred to as “casting” it, etc.. Thus a natural candidate for interpreting the textbox is by using a bag of words. However, the danger with bag of words is that it will often produce too many features, making over-fitting a real concern. Thus for our project, we chose to use only a subset of the words from the entire collection. This subset was selected in part due to the frequency with which each word appears in a card’s textbox, and also from various web sources, including (Wikipedia, 2014) and (Curse, 2014a). We also used a stop word list taken from (Lextek) to filter some of the less descriptive words. Altogether, we created 195 features, each of which represents a single word or phrase in a MTG card textbox.

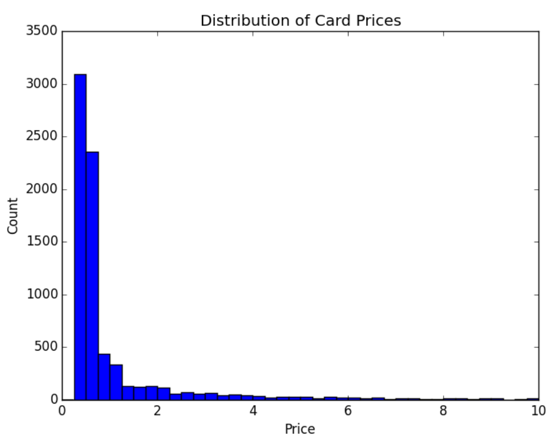
The textbox was also used to generate some integer-valued features based on effects that scaled in power depending on the card. For example, there are cards capable of dealing damage, gaining life, and losing life, and the amount of damage/life depends on the card. Thus we created three new features that have non-negative integer values based on these amounts. If a card does not deal damage or effect life totals, then we set the value of these features to 0. A similar effect comes in the form of modifications to the power and toughness of a card. These changes are denoted by a string of the form “+X/+Y”, where X is the power and Y is the toughness. To represent these effects, we created four more features, namely “power bonus,” “toughness bonus,” “power reduction,” and “toughness reduction.” The first two features count positive changes to the power/toughness, while the latter two count negative changes. In all cases, if there are multiple modifications being made, the largest one is recorded.

4. Creating Our Model

For our machine learning model, we chose to use a random forest decision tree classifier to predict the price. Over the course of this project, we have found that using a decision tree classifier is more effective than using a decision tree regressor and trying and predict the price directly. This is because the data we have shows a very strong skew towards the 0.25 USD price, as shown in the following figure.



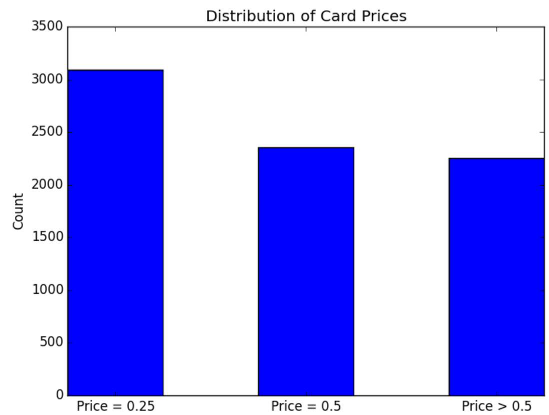
Of the nearly 13000 data points in our set, over 7,500 of them come with a price tag of 0.25 USD. This is not entirely surprising considering the widespread availability of common cards in Magic the Gathering. However, because of this extreme bias, early test results were placing too much emphasis on the 0.25 USD category, to the point where guessing 0.25 for every card achieved the highest accuracy. To address this problem, we first removed all cards in the common rarity slot from our dataset, since almost all cards in this category were worth 0.25 USD anyway. This left us with only 7,701 cards to train on, but the distribution starts to look a bit nicer, as the following figure shows.



At this point, the difficult question was how to divide these cards into distinct classes so as to achieve a balance between the strength of our model and the significance of our results. Our goal is for the algorithm to be able to divide cards into strength tiers, similar to how professionals rank individual cards (Scott-Vargas, 2014). However, the distribution of the data makes it difficult to establish meaningful tiers. For example, consider our first classification scheme shown below, which divides our data in the most uniform way possible, which requires three tiers. For the remainder of this paper, we will refer to each class in our models as a "tier".

- Tier 1: Price = 0.25 USD
- Tier 2: Price = 0.5 USD
- Tier 3: Price > 0.5 USD

This division is desirable because the distribution of cards into tiers is as equal as possible. In other words, this division minimizes the negative effect of having a skewed distribution, as the following figure shows.



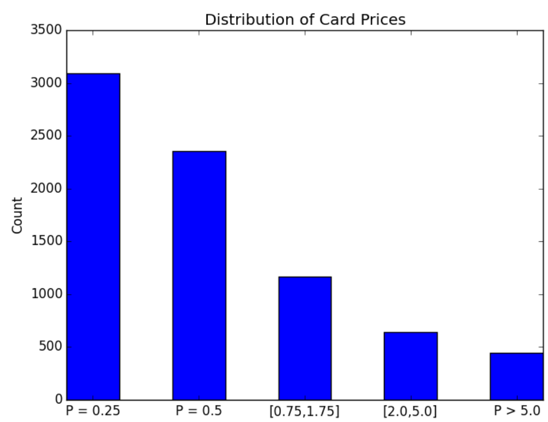
However, the downside of using this division is that the usefulness of our output is compromised. This model can only tell us whether the card is worth more or less than 0.5 USD. For common and uncommon cards, this may be a useful cutoff point, as the vast majority of cards in this rarity category are worth only 0.25 cents. However, rare cards that are worth only 0.5 cents are rather insignificant, so much so that they are commonly referred to as "bulk rares" (Andres, 2013) because they are worth more than a typical common card only because they are harder to find. So for cards in the rare or mythic rare category, this algorithm can only tell us whether or not a card will be a "bulk rare," which is not particularly exciting. To make our output more interesting, then, we will also consider the following tier division:

- Tier 1: Price = 0.25 USD
- Tier 2: Price = 0.5 USD
- Tier 3: 0.75 ≤ Price ≤ 1.75 USD
- Tier 4: 2.0 ≤ Price ≤ 5.0 USD
- Tier 5: Price > 5.0 USD

Here we see that the tiers carry more information than in the previous model. With this classification scheme, we have a 5-tiered ranking system for cards that is actually meaningful. Based on the author's knowledge of the game, we make the following claims about the cards found in each tier:

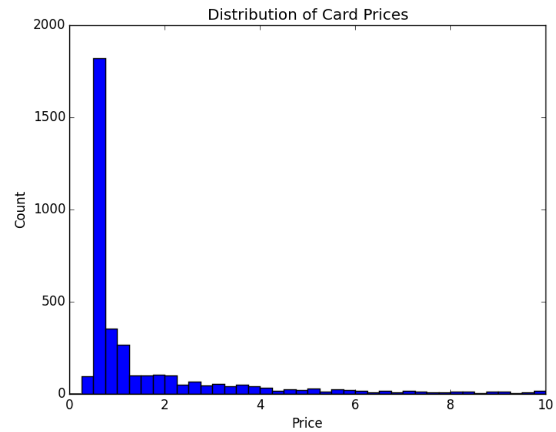
- Tier 1: Mostly commons and uncommons that don't see much play
- Tier 2: Good commons and uncommons, as well as weak rares
- Tier 3: Very good uncommons, and situational rares
- Tier 4: Strong rare cards and good mythic rare cards
- Tier 5: Top-tier rare and mythic rare cards that see regular tournament play

Having a nice ranking system such as this comes at a price, however, as we no longer have a balanced division of cards as we did in the previous design, as shown in the following figure.

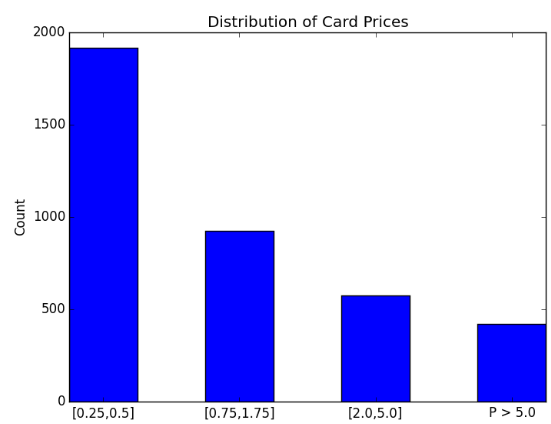


Here we see a decay in the size of the classes that is rather significant, but still much less pronounced than the original distribution. We consider this division to be the absolute maximum in terms of the number of tiers we can have while maintaining strong representation in each. Notice that every tier in this classification scheme contains at least 400 cards. If a set were to have less than that, then we believe that the skew would again become a serious problem. Thus we will not pursue any models that use more than five classes in this project.

One final model we consider is one that includes only rare and mythic rare cards. This leaves us with a dataset of only 3837 cards, but we consider this to be an acceptable number because it is still greater than 10 times the number of features in our model. When both the common and uncommon cards are removed, the distribution looks like the following:



Notice here that the majority of rare cards are not 0.25 USD as previously observed. Instead, we find that the most frequent price is 0.5 USD. Thus for this model, a natural division of cards to consider is one that uses the same 5-tier system discussed above, but with the 0.25 USD class and the 0.5 USD class combined. Doing so gives us the following distribution, which is very similar to the previous one, with the same advantages and drawbacks.



4.1. Implementation Details

For our decision tree algorithm, we used an implementation from the Python package scikit-learn. Their RandomForestClassifier class uses the CART algorithm with pruning to construct each tree. In order to test the effectiveness of our algorithm, we ran the random forest classifier using 80% of the data as our training set, and the remaining 20% as our testing set. We used 10 trees per forest, each with $\sqrt{p}$ features taken from the total set of p features. For splitting, we used the entropy measure for information gain at each level. Lastly, to ensure that the particular choice of cards for our test set did not impact our classification, we averaged the results from $\sqrt{n}$ random forests, each with its own randomly selected test set, where n is the total number of cards in our dataset.

4.2. Results

For each tier in our models, we were interested in the following effectiveness metrics:

1. The *Accuracy* for the entire algorithm, which is the total proportion of correctly classified cards.

2. The *Precision* for each tier T, which is the probability that our algorithm will be correct when it guesses that a card belongs to tier T.

3. The *Sensitivity* for each tier T, which is probability that our algorithm chooses correctly when given a card that belongs to tier T.

In the next sections, we discuss the results we observed from each of our models.

4.3. No-Commons: 3 Tiers

Unsurprisingly, when using only 3 classes in our model, we achieved the best performance. The total accuracy for the model was 67.5%, and the metrics for the individual tiers are summarized in the table below.

	Precision	Sensitivity
Price = 0.25	78.08%	91.44%
Price = 0.5	55.21%	53.44%
Price > 0.5	61.70%	49.25%

Notice that in all cases above, the algorithm's precision was over 50%. This is far better than random guessing, and means that our algorithm's output is relatively trustworthy. However, the precision never passed the 80% threshold, which implies that there is still a lot of noise in our random forests. By comparison, the sensitivity for 0.25 cent cards is very high; over 90%. This tells us that our algorithm is very good at picking out 0.25 cent cards from a list. Unfortunately, the other sensitivity results are much less promising.

4.4. No-Commons: 5 Tiers

When we increase the number of classes to five, the effectiveness of our model is greatly reduced. While the total accuracy only drops to 60.9%, the other metrics are where the real damage is felt.

	Precision	Sensitivity
Price = 0.25	75.02%	92.59%
Price = 0.5	52.15%	62.79%
$0.75 \leq$ Price ≤ 1.75	28.97%	16.03%
$2.0 \leq$ Price ≤ 5.0	32.24%	12.01%
Price > 0.5	51.21%	19.07%

Here we see that detecting cards with price 0.25 or 0.5 USD is still the main strength of our model, with sensitivity measures topping those of the previous model in both tiers. However, the sensitivity to tier 3 and higher cards is very low, worse even than random guessing (which would be correct 20% of the time). This is slightly made up for by precision that's better than random guessing for all classes, with the precision for tier 5 cards exceeding 50%. This is arguably a good trait for our model to have, as being precise about which cards are worth over 5 USD is good from a financial perspective.

4.5. Only-Rares: 4 Tiers

Lastly, when only rare or mythic rare cards are considered, only a marginal difference in precision and sensitivity is observed, accompanied by a drop in overall accuracy to 51.6%.

	Precision	Sensitivity
Price ≤ 0.5	57.87%	85.34%
$0.75 \leq$ Price ≤ 1.75	29.24%	17.63%
$2.0 \leq$ Price ≤ 5.0	32.47%	13.86%
Price > 5.0	50.36%	24.03%

The only metric worth noting is the sensitivity for cards with price greater than 5 USD, which in this model does slightly better than random. However, this increase is to be expected because the proportion of cards in that tier is higher than in previous models due to the lack of inexpensive common and uncommon cards. Overall, however, using only rare cards did not make our predictions any better than before.

5. Conclusions

Based on our analysis of this project, it is likely that even after 20 years of Magic the Gathering, there are still not enough cards in circulation to accurately train a decision tree classifier to predict the price of MTG singles. This is in large part due to the overwhelming number of cards at the low-end of the price spectrum, which over-saturates the branches of the decision tree with paths that lead to 0.25 USD. However, while it is tempting at this point to conclude that there is not enough that differentiates expensive cards from cheap ones in terms of their attributes, we believe that it is too soon to make this argument. It is still likely that the skew in the distribution of prices is obscuring combinations of features that uniquely belong to more expensive cards. This implies that there are many possibilities for extending this work, and we will list a few of our most promising ideas here.

1. One possible project is to analyze the features from this paper to determine which ones led to the highest information gain. In doing so, one could prune the list of features enough so that using smaller, more balanced sets of cards will not lead to terrible over-fitting.

2. Another tactic is to use a decision tree algorithm such as ID3 instead of the CART algorithm implemented in scikit-learn. This is possible when studying MTG cards because all features can be interpreted as categorical ones. Thus using ID3 or similar algorithms would avoid the need to create separate binary features for all categorical variables.

3. To try an achieve better results, one could also consider using a k-nearest neighbors approach for predicting the price. While this method won't help to explain why certain cards become clustered together, it may achieve better performance than the random forest implementation used in this project.

4. Lastly, one question that is still open is whether there are certain combinations of features that uniquely belong to cards in a certain price range. Instead of trying to predict the price, then, one could instead study the features more closely, in hopes of finding commonalities between cards of certain prices.

References

Chas Andres. Bulk rares, 2013. URL http://www.starcitygames.com/article/25681_Bulk-Rares.html.

Inc. Curse. Category:keyword actions — mtg salvation wiki, 2014a. URL http://mtgsalvation.gamepedia.com/Category:Keyword_actions. [Online; accessed October 23, 2014].

Inc. Curse. Functional reprint — mtg salvation wiki, 2014b. URL http://mtgsalvation.gamepedia.com/Functional_reprint. [Online; accessed October 28, 2014].

Magic Online Trading League. Complete magic card price guides, 2013. URL http://classic.magictraders.com/pricelists/. [Online; accessed: October 11, 2014].

Lextek. Stop word list 1 — onix text retrieval toolkit, api reference. URL http://www.lextek.com/manuals/onix/stopwords1.html. [Online; accessed October 23, 2014].

H. Tran R. Hau, E. Plotkin. Magic: The gathering deck performance prediction. Technical report, Department of Computer Science, Stanford University, 2012.

Luis Scott-Vargas. Khans of tarkir limited set review abzan, 2014. URL http://www.channelfireball.com/articles/khans-of-tarkir-limited-set-review-abzan/. [Online; accessed: October 16, 2014].

tcgplayer. Printable real-time price guides by set, 2014. URL http://magic.tcgplayer.com/magic_price_guides.asp. [Online; accessed: October 19, 2014].

M.J. Thompson. Flogging the data: Market forces and game design, 2012. URL http://www.mtgsalvation.com/articles/15124-flogging-the-data-determinants-of... [Online; accessed: October 11, 2014].

Craig Wescoe. Financial predictions for khans of tarkir, 2014. URL http://magic.tcgplayer.com/db/article.asp?ID=12072. [Online; accessed: October 11, 2014].

Wikipedia. List of magic: The gathering keywords — wikipedia, 2014. URL en.wikipedia.org/wiki/List_of_Magic:_The_Gathering_keywords. [Online; accessed October 23, 2014].