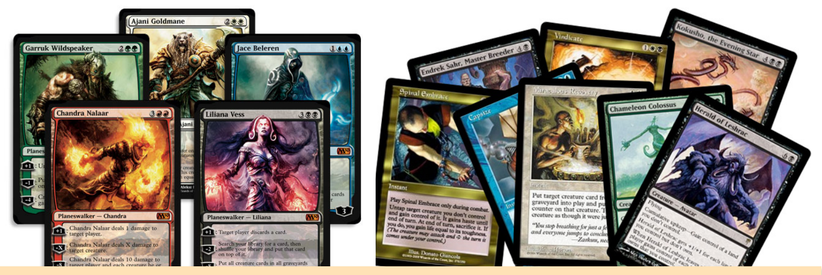
Project by: Tanguy Racinet and Dzenan Softic
Contact: tanracinet at gmail.com | softic.dzenan at gmail.com

MAGIC The Gathering® Deck Builder Toolkit



What is Magic?

- Magic is the oldest and the most popular trading card game ever
- It contains ~ 12.000 different cards
- Each player uses decks containing 60 to 75 cards
- Numerous strategies and card combinations.

Problem

- Deck building requires a certain skill level
- Major issue for new and inexperienced players
- Cards inside deck need to have synergy in order to perform well
- It is hard to find synergy among diverse number of cards

Our goal

- Provide a deck-builder toolkit to help players
- Give insight on winning strategies in the current metagame
- What cards are relevant given the user's current cards
- Which cards should user add to get good synergy

Methodology

Classification: Naïve Bayes

- We work with decks as collections of cards identified by their names
- Every deck is labeled with an "Archetype"
- We choose Naïve Bayes to classify collection of cards to certain "Archetype"
- We use predicted "Archetype" to narrow down dimensionality of data in order to identify synergies specific to user cards

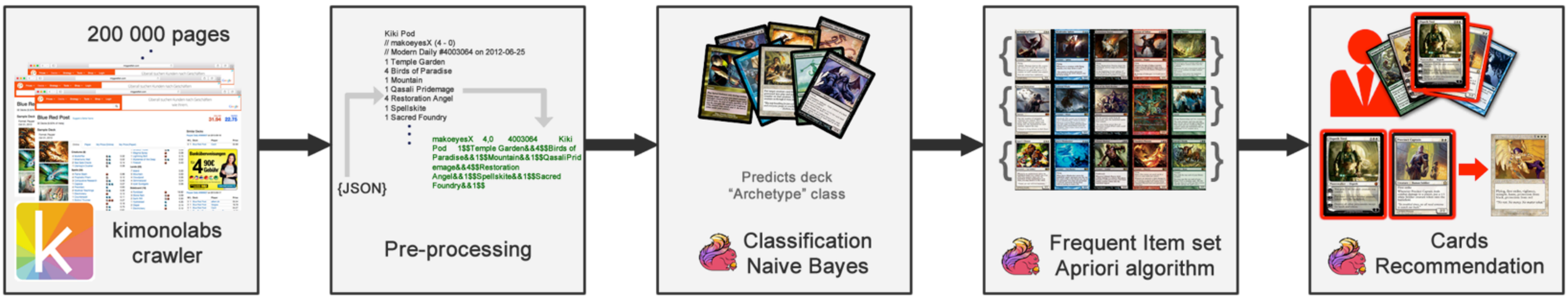
Frequent Item Set mining

- We need to identify good synergies between cards in order to give recommendations
- Identifying synergies in Magic is very difficult due to cards complexity
- We seek to find frequent card combinations from training dataset in order to represent synergies
- We use Apriori algorithm to create combinations of 4 cards that are most played together
- We use item set confidence and winning-loosing ratio to score particular item set

Recommendation system

- We take users cards as input and first define "Archetype" in which he should play
- We recommend missing cards in user's collection to make best performing synergies
- We give top combinations based on user's pool of card
- Recommendation system is of great value for deck customization based on proven winning synergies

Deck Builder Toolkit System



- We use Kimonolabs service to crawl more than 200 000 pages
- Crawler produces JSON files containing: player name, archetype and download link for decks containing a list of cards, an event reference, a player's score and format

- We use the provided JSON to download more than 200 000 deck lists
- We preprocess the downloaded files taking just the relevant information: player name, event reference, score, list of cards, format and archetype
- We finally merge the preprocessed data into one dataset in a format that can be used as an input training set

- We use Naïve Bayes classifier to classify the user's collection of cards to a certain "Archetype"
- We implement Naïve Bayes in a distributed fashion due to the size of our dataset
- We use Apache Flink as implementation platform

- We use Apriori algorithm to mine frequent cards combinations and to score its synergy performance
- We make four iterations to produce combinations of four cards
- For each frequent combination we provide confidence percentage and winning percentage
- We use Apache Flink as implementation platform

- We use user collection of cards as input
- We predict in which "Archetype" player should play
- We map user cards to our frequent item sets mined database
- We output cards that user should consider to add to personal collection in order to have good performing synergies
- We rate combinations so user can get cards with bigger chance to win

Experiments

Step 1. Input for the Classification

User has just 4 cards and want to build a deck:

- User does not know in which "Archetype" can play

{Remand, Blood Moon, Vedalken Schackles, Snapcaster Mage}

Step 2. Classification output

With just 4 cards as an input (deck should have 60-75 cards), Classifier is able to predict correct "Archetype" for user input:

"Archetype" => "UR Delver"

Step 3. Generating frequent item set

Generated frequent item set for user's archetype "UR Delver"

Card #1	Card #2	Card #3	Card #4	Support %	Winning ratio
Burst Lightning	Spellstutter Sprite	Steam Vents	Vendilion Clique	0.67	0.79
Delver of Secrets	Electrolyze	Faerie Conclave	Misty Rainforest	0.62	0.78
Electrolyze	Faerie Conclave	Mana Leak	Mountain	0.62	0.78

• • •

Step 4. Mapping user cards to generated frequent set

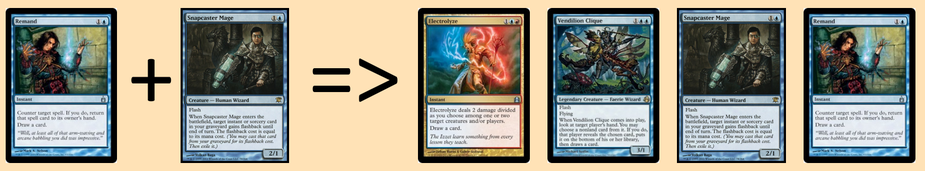
{Remand, **Blood Moon**, Vedalken Schackles, **Snapcaster Mage**}

{**Blood Moon**, Delver of Secrets, Lightning Bolt, Misty Rainforest}
{Burst Lightning, Delver of Secrets, Lightning Bolt, Mana Leak}
{**Blood Moon**, Electrolyze, Misty Rainforest, **Snapcaster Mage**}
{**Blood Moon**, Electrolyze, Scalding Tarn, Spell Snare}

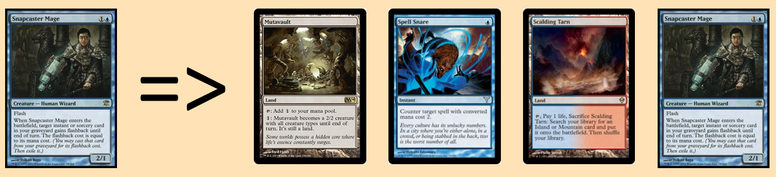
• • •

Step 5. Recommend top combinations for user cards

Suggest which cards user can add to deck to perform good. Rank is combination of support and winning ratio.



Rank: 0.66



Rank: 0.74

Results

- We crawled more than 200K pages from mtggoldfish.com , which contains deck lists from online daily events. We obtained ~ 25K different decks, each containing 60-75 cards. We focused on specific format "Modern" instead of taking 200K decks in order to provide meaningful data for players.
- Naïve Bayes classification accuracy is71% with 17.5K decks for training and 7.5K decks for test.
- With Apriori algorithm, we choose to make frequent item sets with four cards. This give us more insight on card synergies instead of giving just a subset of an "Archetype".
- Manual analysis of the recommendations suggest that cards are coherent and have value to a player.

Discussion

- Classification accuracy of 71% is not satisfying, and should be improved. The problem rely in small vocabulary (~1900 distinct card names), with large number of classes (more than 100 different "Archetypes")
- In order to provide reliable service for players, data needs to be updated frequently and mine new frequent item sets, since meta game is evolving
- Recommendations system provides the most played ones, thus can be of low interest for experienced players
- Future work would explore new ways to improve classification and overall performance of system. Algorithms other than Apriori should be explored and tested
- To make system more valuable it should mine frequent item set on more than one "Archetype"

Git Repository: https://github.com/dzeno/deck-builder-toolkit