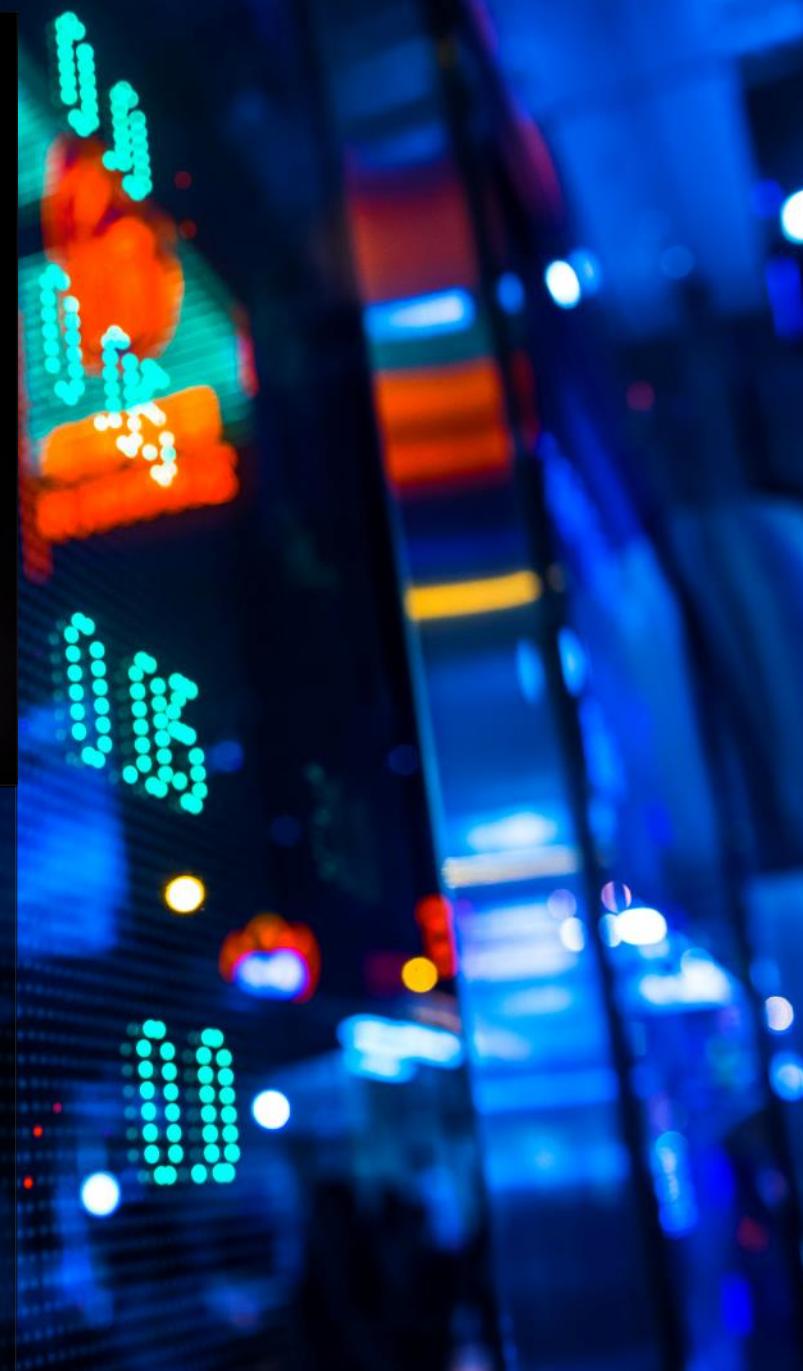




# MTGPREDICTBYIM AGE

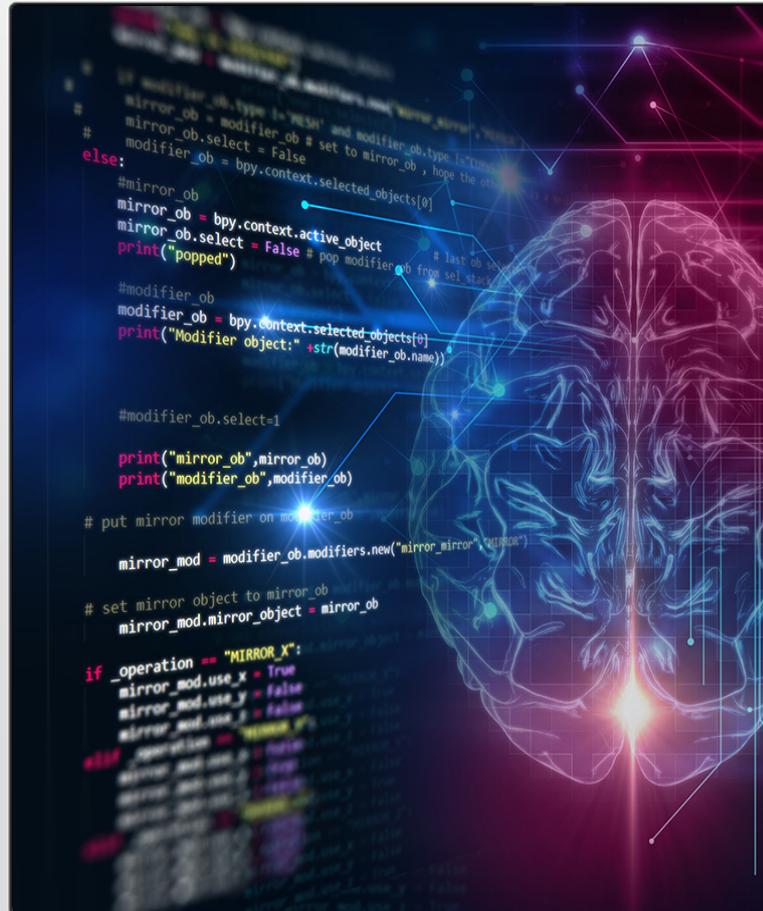
CST383

DAVID KIM, ERICKA KOYAMA, JAMES DESELMS



# INTRODUCTION

- CAN WE PREDICT THE TYPE OF A CARD BASED ON THE COVER ART IMAGE WITH AN ACCURACY RATE HIGHER THAN RANDOM CHANCE, 20%?

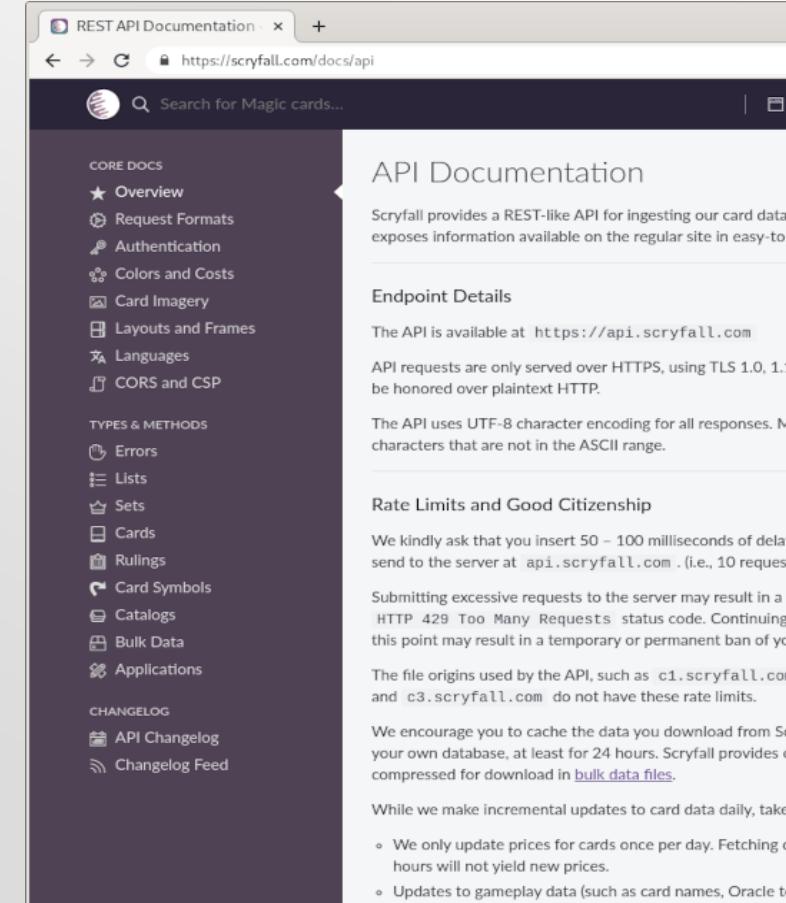




**SELECTION**

# SELECTION OF DATA: SCRYFALL API

HEAD



The screenshot shows a web browser window titled "REST API Documentation" with the URL <https://scryfall.com/docs/api>. The page has a dark purple header bar with a search bar containing "Search for Magic cards...". Below the header is a sidebar with a dark purple background and white text, listing various API endpoints and methods. The main content area to the right of the sidebar contains several sections: "API Documentation" (describing the REST-like API for ingesting card data), "Endpoint Details" (mentioning the API endpoint at <https://api.scryfall.com> and HTTPS requirements), "Rate Limits and Good Citizenship" (warning against excessive requests and encouraging caching), and a "CHangelog" section.

REST API Documentation

https://scryfall.com/docs/api

Search for Magic cards...

CORE DOCS

- ★ Overview
- ⚙ Request Formats
- 🔑 Authentication
- 🎨 Colors and Costs
- 🖼 Card Imagery
- .ImageLayouts and Frames
- 🌐 Languages
- cors and CSP

TYPES & METHODS

- ⌚ Errors
- 📄 Lists
- 📦 Sets
- 🎴 Cards
- 📜 Rulings
- 🎴 Card Symbols
- 💻 Catalogs
- 📄 Bulk Data
- ⚙ Applications

CHANGELOG

- 📄 API Changelog
- RSS Changelog Feed

API Documentation

Scryfall provides a REST-like API for ingesting our card data. This exposes information available on the regular site in easy-to-use JSON format.

Endpoint Details

The API is available at <https://api.scryfall.com>. API requests are only served over HTTPS, using TLS 1.0, 1.1, and 1.2. The API uses UTF-8 character encoding for all responses. No characters that are not in the ASCII range.

Rate Limits and Good Citizenship

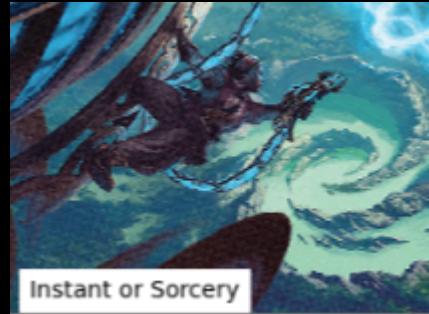
We kindly ask that you insert 50 – 100 milliseconds of delay between requests to the server at [api.scryfall.com](https://api.scryfall.com). (i.e., 10 requests per second). Submitting excessive requests to the server may result in a temporary or permanent ban of your account. The file origins used by the API, such as `c1.scryfall.com` and `c3.scryfall.com`, do not have these rate limits.

We encourage you to cache the data you download from Scryfall and store it in your own database, at least for 24 hours. Scryfall provides compressed files for download in [bulk data files](#).

While we make incremental updates to card data daily, take note of the following:

- We only update prices for cards once per day. Fetching card data every hour will not yield new prices.
- Updates to gameplay data (such as card names, Oracle texts, and so on) are frequent. If you only need gameplay information, download the compressed files.

# Selection of Data - Types of Cards



Enchantment

Land

Instant or  
Sorcery

Artifact

C

# Selection of Data - Data Munging and Feature Engineering

```
#Load both of those cards into a dataframe.
df = pd.DataFrame(cards.data()).append(more_cards.data())
#Split the image_uris into three columns.
df = pd.concat([df.drop(['image_uris'], axis=1), df['image_uris'].apply(pd.Series)], axis=1)
#Drop unneeded columns.
df = df.replace(regex=r'^Creature(.*)', value='Creature')
df = df.drop(['id', 'object', 'oracle_id', 'multiverse_ids', 'mtgo_id', 'mtgo_foil_id', 'tcgplayer_id', 'cardmarket_id', 'lang',
'released_at', 'uri', 'scryfall_uri', 'layout', 'highres_image', 'image_status', 'keywords', 'legalities', 'games',
'reserved', 'foil', 'nonfoil', 'oversized', 'promo', 'reprint', 'variation', 'set_type', 'set_uri', 'set_search_uri',
'scryfall_set_uri', 'rulings_uri', 'prints_search_uri', 'collector_number', 'digital', 'rarity', 'flavor_text', 'set',
'card_back_id', 'artist_ids', 'illustration_id', 'border_color', 'frame', 'full_art', 'textless', 'booster', 'set_name',
'story_spotlight', 'edhrec_rank', 'prices', 'related_uris', 'purchase_uris', 'power', 'toughness', 'border_crop',
'produced_mana', 'all_parts', 'watermark', 'promo_types', 'small', 'normal', 'large', 'png'], axis=1)
#Use regex to manipulate typelines so they contain no subtypes or supertypes.
df = df.replace(regex=r'^Artifact(.*)', value='Artifact')
df = df.replace(regex=r'^Enchantment(.*)', value='Enchantment')
#Combine instant and sorcery into one type.
df = df.replace(regex=r'^Sorcery(.*)', value='Instant or Sorcery')
df = df.replace(regex=r'^Instant(.*)', value='Instant or Sorcery')
#Change all Basic Lands into just Land.
df = df.replace(regex=r'^Basic(.*)', value='Land')
# Try to get even number of each card type
df = df.groupby('type_line').head(no_cards_per_type)
#Turn colors into a concatenated string (e.g., WUBRG) instead of a list (e.g., [W,U,B,R,G]).
df['colors'] = df['colors'].apply(lambda x: 'C' if not x else ''.join(x))
#print some info about data
df.info()
```