Cryptocurrency Prediction using Social Media Sentiment & Volume

SPRING 2019 | DATA ENGINEERING & PLATFORMS FOR ANALYTICS

GROUP 5

Outline

- Executive Summary
- Business Use Case
- Tech Stack – Data and Tools
- Design Considerations
- Enhanced Entity Relationship Model
- Dimensional Model

Executive Summary

- Background:
  - The cryptocurrency market is very nascent and volatile. This makes the asset class very risky but also potentially lucrative.
  - Understanding the impact of social media on price direction can provide a cryptocurrency trader a better trading strategy.

Objective: To collect market information and social media engagement data for the major cryptocurrencies and tokens to create a platform for short-term trading strategy.

Business Use Case

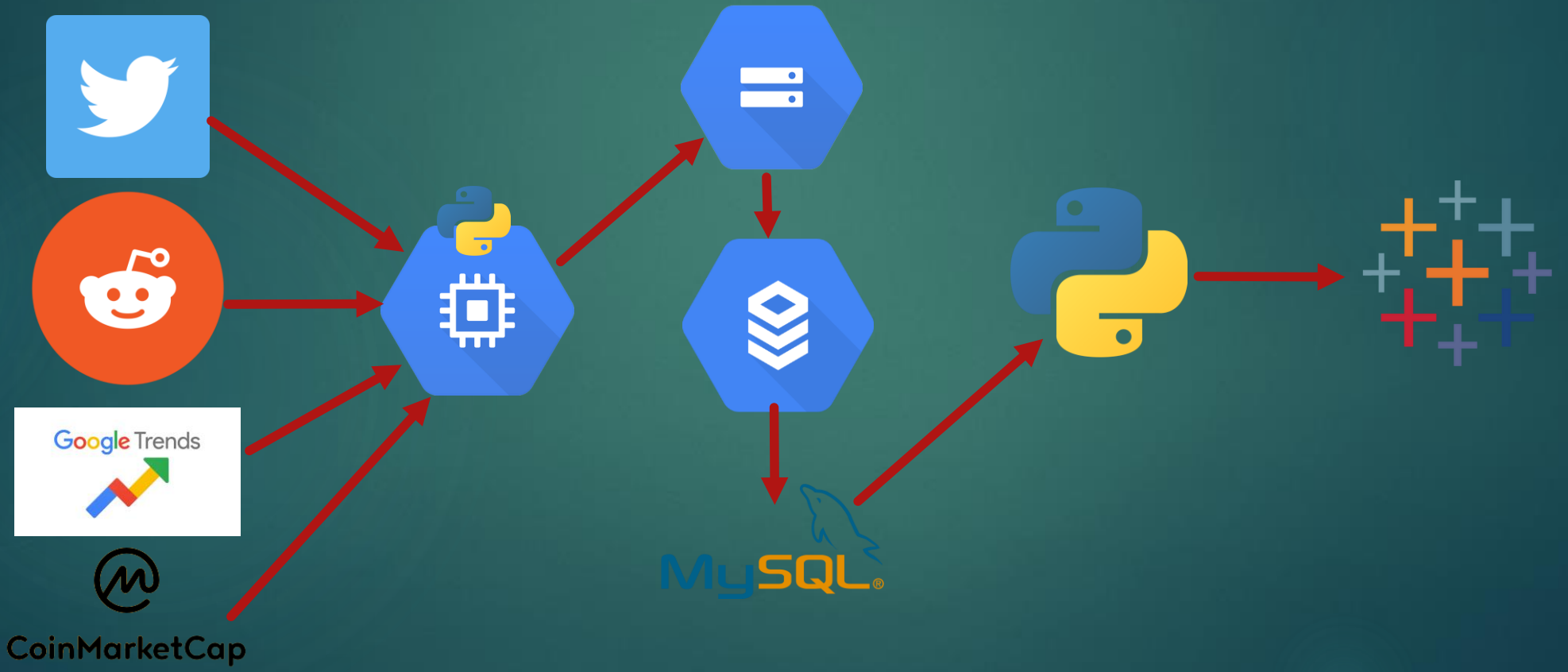
A price prediction model based on metrics and sentiments from multiple online resources and incorporate it into larger system that automatically and intelligently manages a cryptocurrency portfolio.

Users:

- Day-traders can use our database to build complex trading models based on social media sentiment and volume
- Long-term investors can find established/safer assets to diversify their portfolio
- Industry experts can use the dataset to analyze the publics knowledge and interest in their specific cryptocurrency/blockchain technology

Data & Tools

Data sources → ETL → Storage& database → Analysis → Reporting



Database Design Considerations

- Application Type
  - Online Analytical Processing for Business Intelligence and end users (OLAP)
- Data Format and Size
  - Cryptocurrency Financial Data
- Data Maintenance and Support
  - Open Source Project
- Hardware and Software
  - Google Cloud compute and virtual machines, and Google Cloud buckets for storage

- Elements relevant for pricing model analysis will be stored
- ACID properties and use cases were considered when developing the constraints, datatypes and ranges of the database

Data Extraction

praw/psaw API

Reddit Table

Reddit posts from /r/Cryptocurrency from 2017-2019

```
In [5]:
import praw
from psaw import PushshiftAPI

In [6]:
#Connect to reddit API
reddit = praw.Reddit(client_id='GjfUHQE8AYnXLg', client_secret='PfbhtsXJGAAUNiEyHPRGPuFJ0ro', user_agent='DEPA_Pr
api = PushshiftAPI()

In [ ]:
#Extract all posts to /r/CryptoCurrency from 2017-2019

start_epoch=int(dt.datetime(2017, 1, 1).timestamp())
end_epoch = int(dt.datetime(2019, 1, 1).timestamp())

reddit_data = pd.DataFrame(api.search_submissions(after=start_epoch,
                           before=end_epoch,
                           subreddit='CryptoCurrency',
                           filter=['author', 'title', 'subreddit', 'num_comments', 'created', 'score']))
```

	author	created_utc	num_comments	score	subreddit	title	date_id
0	robertbint	2019-01-01	0	1	CryptoCurrency	Buy or Sell Bitcoins online - A Few Pointers a...	3652
1	h214289	2019-01-01	0	1	CryptoCurrency	Dec 31 2018 Important Crypto News	3652
2	coinmarshal	2019-01-01	5	1	CryptoCurrency	I am sharing Crypto with my WhatsApp buddies. ...	3652
3	h1121900	2019-01-01	0	1	CryptoCurrency	Dec 31 2018 Important Crypto News	3652
4	instasmarter	2019-01-01	13	1	CryptoCurrency	Best places to spend Bitcoin	3652
5	h1121900	2019-01-01	0	1	CryptoCurrency	Dec 31 2018 Important Crypto News	3652
6	h2022395	2019-01-01	0	1	CryptoCurrency	Dec 31 2018 Important Crypto News	3652

Scrapped all reddit posts to /r/cryptocurrency for our timeframe

beautifulsoup

Twitter Table

```
In [125]:
import re
import csv
import requests
from bs4 import BeautifulSoup
from IPython.display import HTML

In [182]:
import pandas as pd
tweets_df = pd.DataFrame({'Date': [], 'No. of Tweets': [], 'Coin': []})

for i in range(0,len(df_names['symbol'])-1):
    #len(coin_name['symbol'])-1

    coin = df_names['symbol'][i]
    url = 'https://bitinfocharts.com/comparison/tweets-'+coin.lower()+'.html'
    headers = {'User-Agent': 'Chrome/54.0.2840.90'}
    response = requests.get(url, headers=headers)
    html = response.text

    from bs4 import BeautifulSoup
    soup = BeautifulSoup(html, 'html.parser')

    x = soup.find_all('script')

    data_1 = re.findall(r'({\"new\sDate.*\"})', str(x))
    data_1 = str(data_1)


    if data_1 == '[]':
        continue
    data_2 = data_1.split("],(")
    data_2[0] = data_2[0][3:]
    data_2[len(data_2) - 1] = data_2[len(data_2) - 1][:-4]
    data_pd = pd.DataFrame(data_2)
    data_clean = data_pd[0].str.split(",", expand = True)
    data_clean = data_clean.iloc[:, 0:2]
    data_clean.columns = ['Date', 'No. of Tweets']
    data_clean["Date"] = data_clean["Date"].str.slice(10,20)
    data_clean["Coin"] = coin.upper()
    tweets_df = tweets_df.append(data_clean, ignore_index = True)
```

	Date	No. of Tweets	coin_id	date_id
0	2018-01-08	7	0	3294
1	2018-01-09	20	0	3295
2	2018-01-10	21	0	3296
3	2018-01-11	4	0	3297
4	2018-01-12	6	0	3298
5	2018-01-13	14	0	3299
6	2018-01-14	3	0	3300
7	2018-01-15	18	0	3301
8	2018-01-16	1	0	3302

Scrapped # of tweets per currency. Granularity = day

cryptory API

Google Trends table

```
In [ ]:
#Pulling google trends data from cryptory

i=1
google_data_list = []
for name in list(df_names['name']):
    if(i>0 and i<101):
        i=i+1
        kw_list = []
        kw_list.append(name)
        try:
            data = my_cryptory.get_google_trends(kw_list)
            google_data_list.append(data)
        except:
            continue;

In [ ]:
trend_df1 = pd.concat(google_data_list)

trend_df2 = pd.DataFrame(df1.pivot_table(index = 'date').unstack()).reset_index()
```

	date	trend	coin_id
0	2017-01-01	14.939345	0
1	2017-01-02	14.939345	0
2	2017-01-03	15.686312	0
3	2017-01-04	24.276435	0
4	2017-01-05	22.035534	0
5	2017-01-06	17.553730	0
6	2017-01-07	13.071927	0
7	2017-01-08	10.457541	0
8	2017-01-09	8.216640	0
9	2017-01-10	18.300697	0

Scrapped google trend data for every currency for our timeframe of interest

Data Storage

STEP 1 Compute Engine VM

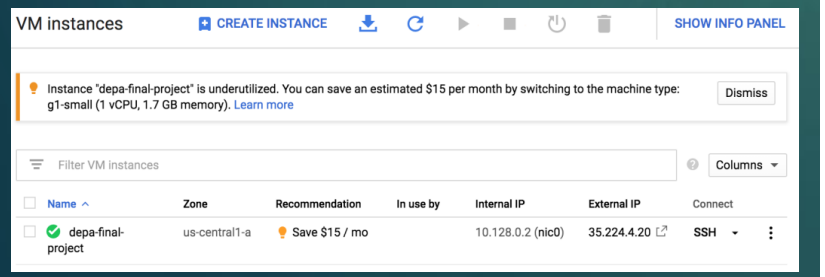
- Run all webscraping scripts in python 3
- Transform and normalize tables
- Push data to buckets

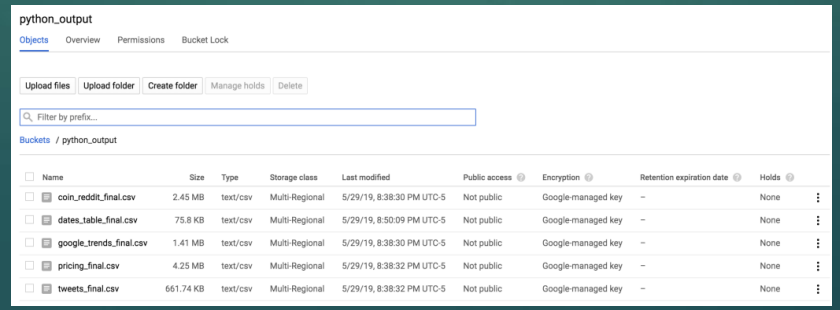
STEP 2 Storage Buckets

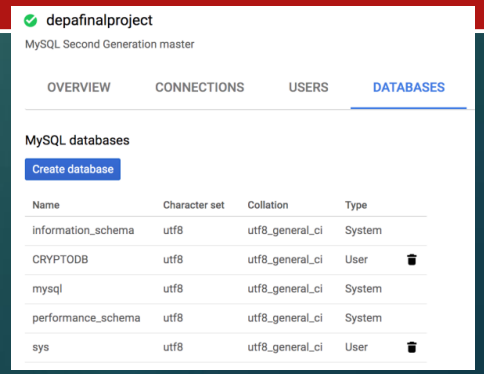
- Store data as .csv files
- Store raw data files for potential future use-cases

STEP 3 CloudSQL Instance

- Cryptodb Database created through MySQL connection
- Run DDL scripts
- Import DML/Data from storage bucket csv's



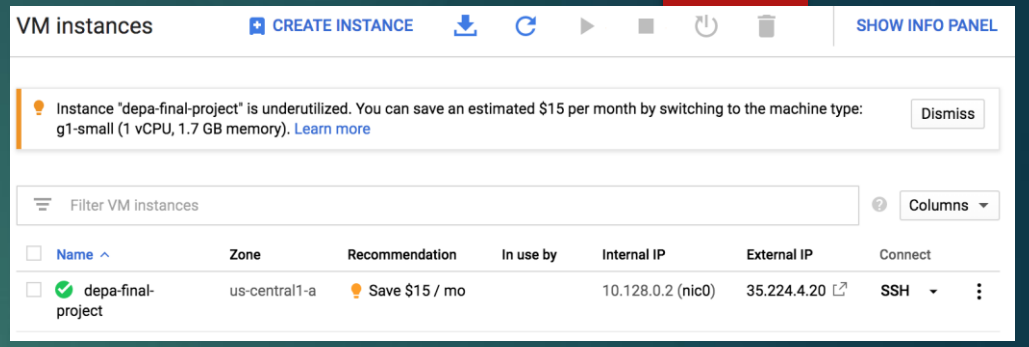




Data Storage

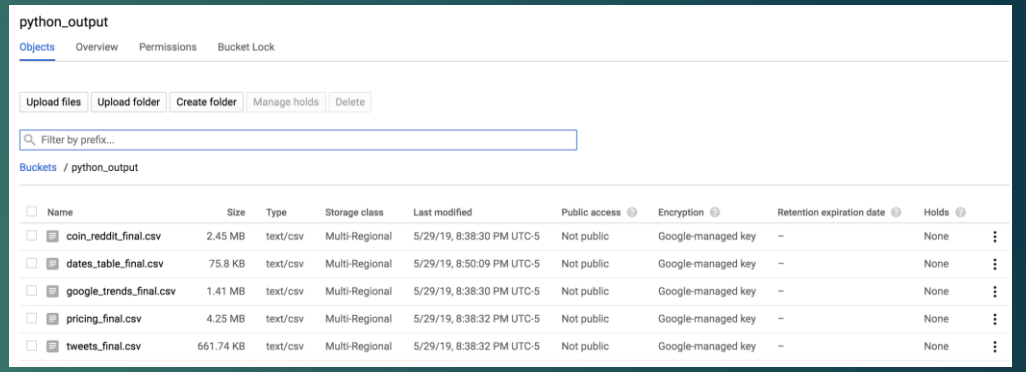
Compute Engine VM

- Run all webscraping scripts in python 3
- Transform and normalize tables
- Push data to buckets



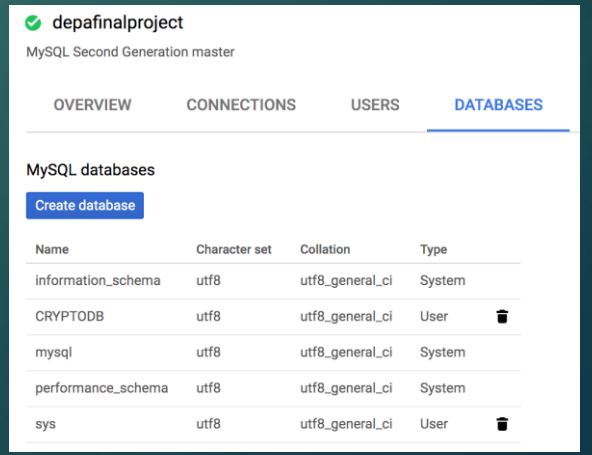
Storage Buckets

- Store data as .csv files
- Store raw data files for potential future use-cases



CloudSQL Instance

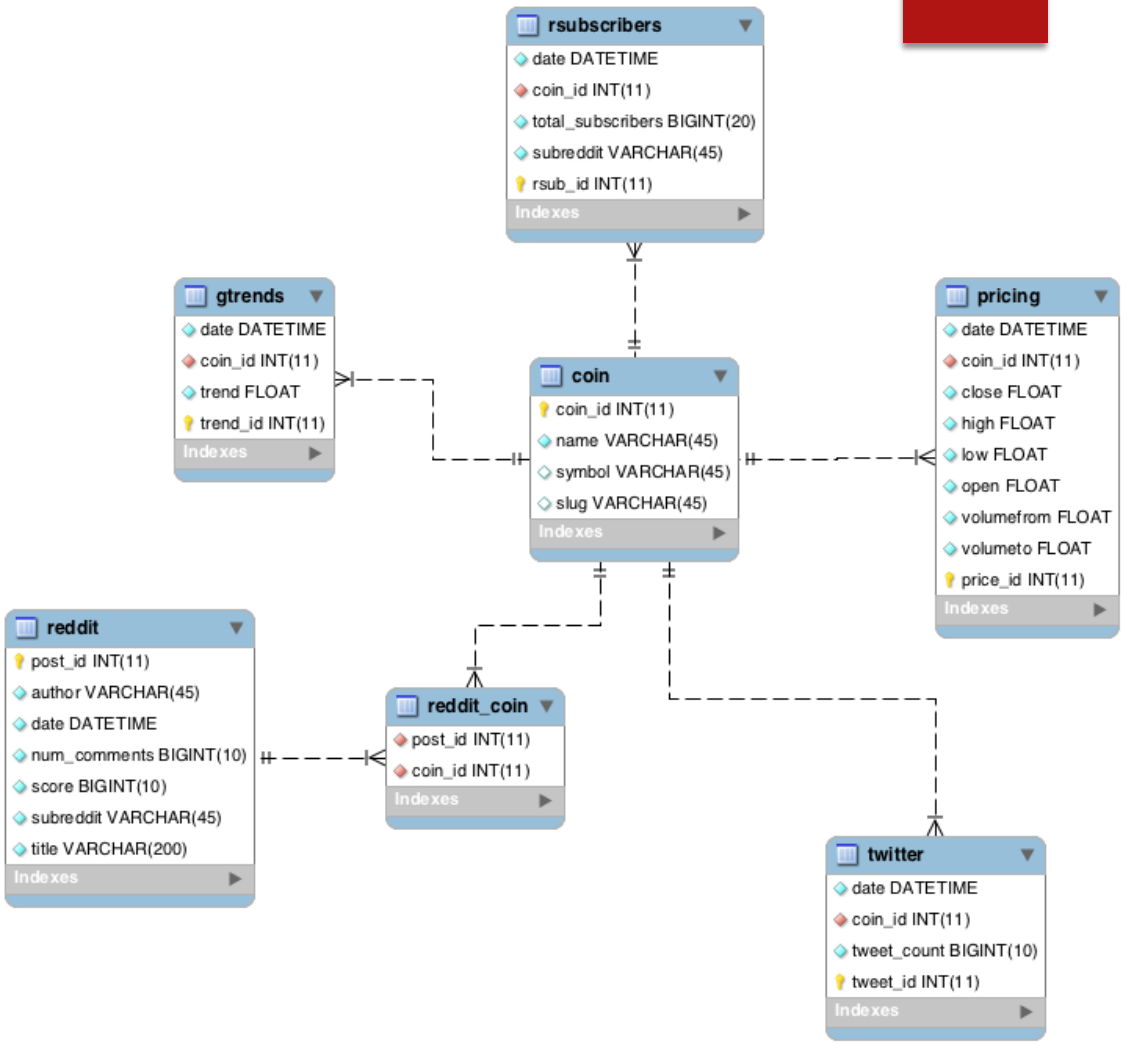
- Cryptodb Database created through MySQL connection
- Run DDL scripts
- Import DML/Data from storage bucket csv's



Enhanced Entity Relationship Diagram

- Entities:
  - reddit: /r/cryptocurrency sub post data
  - reddit_subs: subscriber count for >150 coin subreddits
  - Pricing: cryptocurrency market OHLCV data
  - Twitter: tweet count by coin mentioned
  - gtrends: Google trend data by coin
  - reddit_Coin: join-table linking reddit post data to coin ID
  - date : date information

- Relationship & Cardinality
- Datatypes



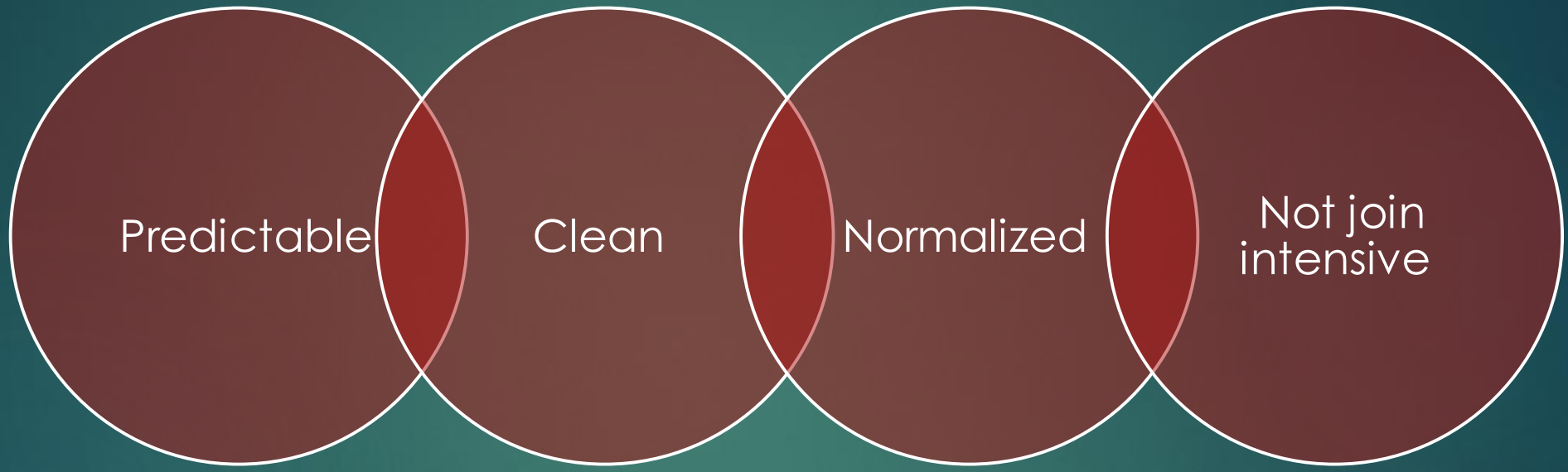
Dimensional Model (Snowflake)

TBD

Tableau

MongoDB

Our data is:



Thus, for our particular database, we decided not use MongoDB.

Inappropriate Model	Degraded Performance	Inappropriate Language	Not ACID
• MongoDB cannot model or store relationships if no complexity exists • Our datasets are sourced through webscraping methods via APIs	• Use speed plummets as the amount of data increases and the number of join keys grows	• "almost SQL" languages are inappropriate for SQL "joins" and similar functionality	• Non-relational databases like MongoDB do not follow ACID properties

Neo4j

For our particular database, we also decided not to use Neo4j for the following reasons

Inappropriate database

- Graph databases are inappropriate for traditional relational data.
- Difficult to perform mass analytics queries across all the relationships and records.

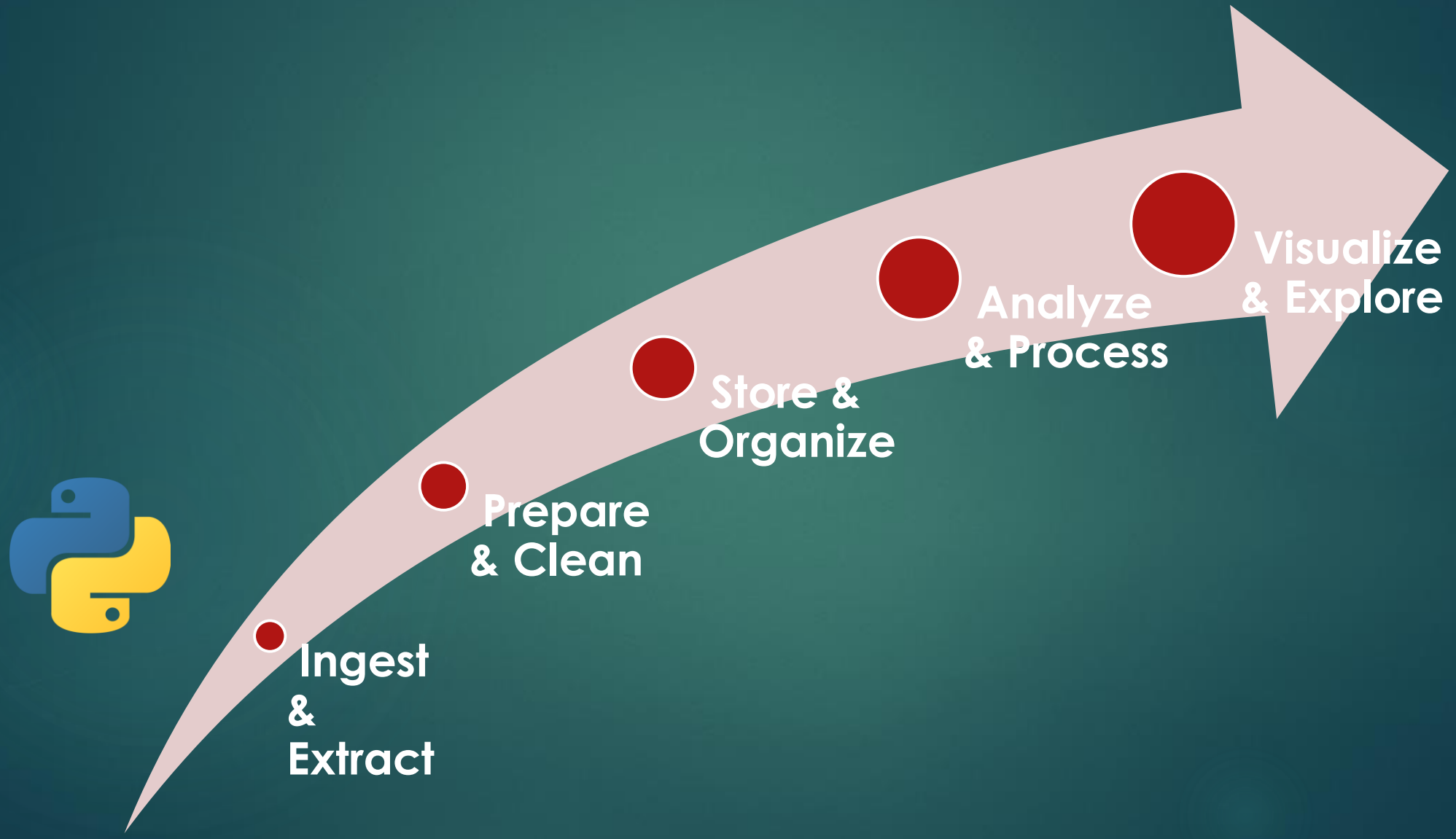
Not efficient for high volumes of transactions

- Graph databases are not as useful for operational use cases since they are not good at handling large queries that span the entire database.

Not an independent master data management solution

- A graph database is just a data store and doesn't give you a business-facing user interface to query or manage relationships.
- It will not provide advanced match and survivorship functionality or data quality capabilities.

Summary of Data Analysis Pipeline



Future Recommendations

Conclusion