Cryptocurrency Prediction using Social Media Indicators

SPRING 2019 | DATA ENGINEERING PLATFORMS FOR ANALYTICS

GROUP 5 – AKARSH SAHU, NAZIH KALO, ANGELA TENG, SOWMYA NALLAPANENI

Outline

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

Executive Summary

Project Background/Hypothesis

- The cryptocurrency market is very **nascent and volatile**. This makes the asset class very risky but also potentially lucrative.
- Behavioral sciences and related scientific literature provide evidence that there is a **close relationship between social media and price fluctuations of cryptocurrencies**.
- **Hypothesis:** Social media indicators are strongly correlated with prices and can be a good prediction indicator for short-term investments in cryptocurrencies

Project Objective

- **Automated Data Integration** Connect Data Sources, Ingestion, and Consolidation
- **Efficient and Scalable Data Storage:** Normalized EER Model, OLAP Dimensional Model, and Cloud SQL Integration
- **Generate real-time insights on cryptocurrency price:** Leveraging Data Visualization, Statistical Analyses, and Forecasting.

Conclusion/Future Recommendations

- **Strong Correlation** b/w Cryptocurrency price and Social media indicators. But Correlation doesn't imply causation!
- Build a regression model based on metrics available to **test the hypothesis** further.
- **Leverage NLP and Sentiment Analysis** to generate sentiments from text sources.
- Build a **real-time predictive model** utilizing all the metrics , sentiment, and demographics.

Business Use Case

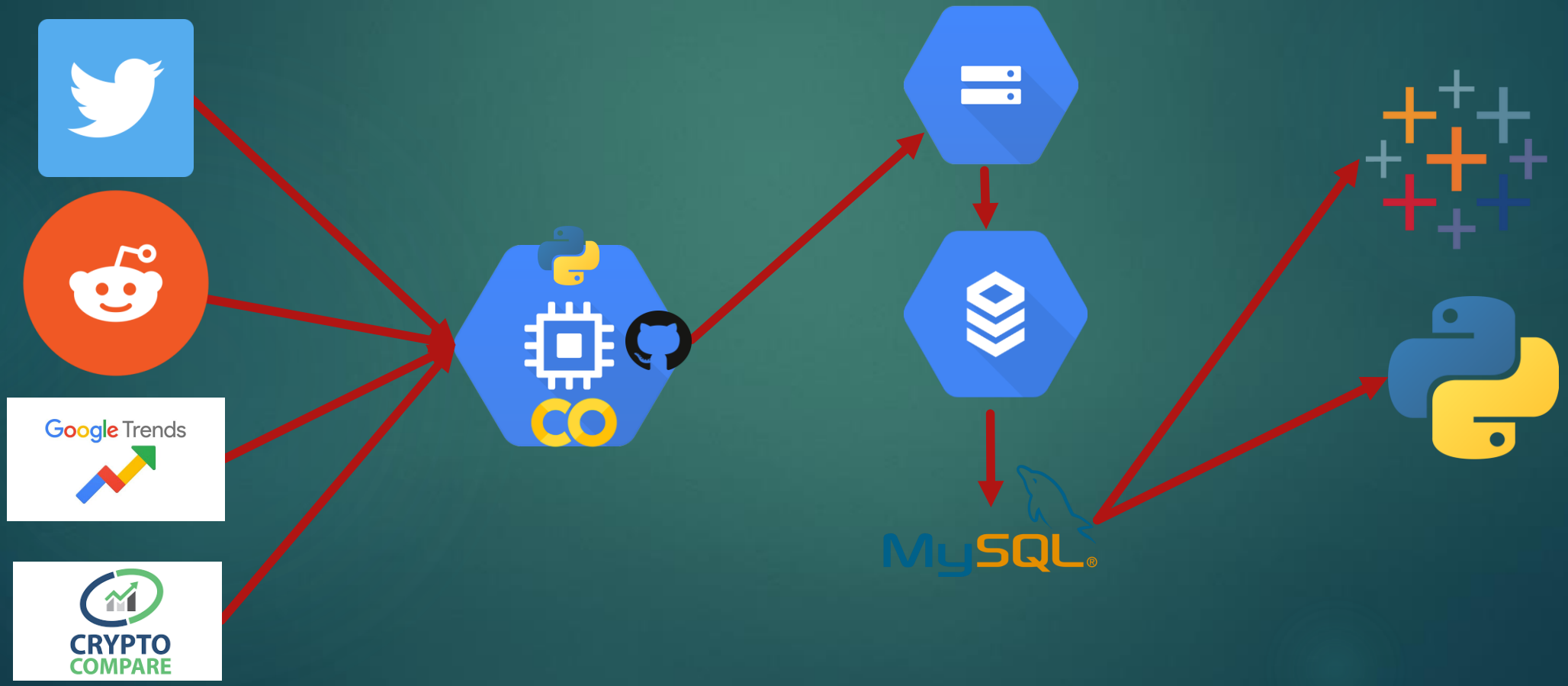
- **Day-traders** can use our database to build complex trading models based on social media sentiment and volume

- **Long-term investors** can find established/safer assets to diversify their portfolio

- **Industry experts** can use the dataset to analyze the public's knowledge and interest in their specific cryptocurrency/blockchain technology

Data & Tools

Data sources | Ingestion | Prepare | Storage& database | Analysis/Visualization



Data Source

Data source
www.bitinfocharts.com

Description
Number of daily tweets for each cryptocurrency

Time Period
2014-Today

Number of Records
48,598

Data source
www.reddit.com

Description
1. All posts to /r/cryptocurrency subreddit
2. Total daily subscribers to all cryptocurrency subreddits

Time Period
2014-Today

Number of Records
1. 442,038
2. 121,138

Data source
https://trends.google.com

Description
Google trend data

Time Period
2014-Today

Number of Records
120,975

Data source
www.cryptocompare.com

Description
OHLCV for all cryptocurrencies

Time Period
2009-Today

Number of Records
82,637

Ingestion

Beautifulsoup & Requests API
Scraped # of tweets per currency

Praw & Psaw APIs
Scraped all posts from/r/cryptocurrency

Cryptory API
Scraped google trend data per currency

Requests API
Scraped pricing data for every currency

Twitter Table

```
import re
import csv
import requests
from bs4 import BeautifulSoup
from IPython.display import HTML
```

```
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\xDate.*\}]}', 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][:-6]
    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)
```

Reddit Table

Reddit posts from /r/Cryptocurrency from 2017-2019

```
import praw
from psaw import PushshiftAPI
```

```
#Connect to reddit API
reddit = praw.Reddit(client_id='GjfUHQE8AYnXLg', client_secret='PfbhtsXJGAAUNiDy8PRGPuFJ0ro', user_agent='DEPA_Pr
api = PushshiftAPI()
```

```
#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']))
```

Reddit Subscribers

```
# load package
from cryptory import Cryptory

# initialise object
# pull data from start of 2017 to present day
my_cryptory = Cryptory(from_date = "2017-01-01",to_date="2019-01-01", ascending=True)
```

```
#Subreddits based on symbol

empty_df = pd.DataFrame(columns = ['date','total_subscribers','subreddit'])
subreddit_list = []
for name in list(df_names['symbol']):
    try:
        subred = my_cryptory.extract_reddit_metrics(subreddit = name, metric = "total-subscribers", col_label
        if (type(subred) == ValueError):
            subreddit_list.append(empty_df)
        else:
            subreddit_list.append(subred)
    except:
        subreddit_list.append(empty_df)
```

Google Trends table

```
#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;
```

```
trend_df1 = pd.concat(google_data_list)

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

Pricing Table

```
import requests
import datetime
import pandas as pd
```

```
def daily_price_historical(symbol, comparison_symbol, all_data=True, limit=1, aggregate=1, exchange=''):
    url = 'https://min-api.cryptocompare.com/data/histoday?fsym={}&tsym={}&limit={}&aggregate={}'\
            .format(symbol.upper(), comparison_symbol.upper(), limit, aggregate)
    if exchange:
        url += '&e={}'.format(exchange)
    if all_data:
        url += '&allData=true'
    page = requests.get(url)
    data = page.json()['Data']
    df = pd.DataFrame(data)
    df['timestamp'] = [datetime.datetime.fromtimestamp(d) for d in df.time]
    return df

pricing_list = []

for symbol in list(df_names['symbol']):
# initialise scraper with time interval
    try:
        df = daily_price_historical(symbol, 'USD')
        df['name'] = symbol
        pricing_list.append(df)
    except:
        df2 = pd.DataFrame(columns = df.columns)
        pricing_list.append(df2)
```

Prepare/Clean

Cleaning Tweets & Merging with coin ids

```
#Convert Date into datetime object
tweets_df['Date'] = tweets_df['Date'].astype('datetime64')


#Remove the null rows
tweets_df = tweets_df[tweets_df['No. of Tweets'] != 'null']


#Convert No. of Tweets into int object
tweets_df['No. of Tweets'] = tweets_df['No. of Tweets'].astype('int')


#Adding coin_id column
tweets_df = tweets_df.merge(df_names[['symbol', 'coin_id']], how = 'left', left_on = 'Coin', right_on = 'symbol')

#Drop redundant name columns
tweets_df = tweets_df.drop(['Coin', 'symbol'], axis = 1)


#Change to tweet_count
tweets_df.rename(columns = {'No. of Tweets':'tweet_count'}, inplace = True)
#Set date & coinid as index
tweets_df.set_index(['Date', 'coin_id'], inplace = 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

Creating Reddit-Coin Join Table

```
# # Create reddit_coins Join Table
# #### Matches the reddit posts in reddit_data table to the coins in coins table


#Zip the names into tuple of (name, slug, symbol)
zipped_names = list(zip(df_names['name'],df_names['slug'],df_names['symbol']))

#Create a search list - seperating the three terms with OR operator (|)
search_list = []
for (name, slug, symbol) in zipped_names:
    listt = [name, slug, symbol]
    pat = '|'.join(listt)
    search_list.append(pat)


#Add boolean/dummy columns
import re
dummy_df = pd.DataFrame(dict((name, reddit_data.title.str.contains(name, re.IGNORECASE))
                             for name in search_list))


#Convert dummy columns into rows with post_id as the index
i, j = np.where(dummy_df)

coins_mentioned_series = pd.Series(dict(zip(zip(i, j), dummy_df.columns[j])))


#Create final join table between reddit and coins table. Rename columns of dataframe.
coin_reddit_join = pd.DataFrame(coins_mentioned_series).reset_index()
coin_reddit_join.columns = ['post_id', 'coin_id', 'coin_name']
coin_reddit_join = coin_reddit_join.drop('coin_name', axis = 1)


coin_reddit_join2 = coin_reddit_join.set_index(['post_id', 'coin_id'],)

coin_reddit_join2
```

post_id	author	date	num_comments	score	subreddit	title
0	Mahaprajapati	2019-05-31 23:56:44	0	1	CryptoCurrency	Swap (XWP) added to CNPool.cc today
1	webnowcompany	2019-05-31 23:55:15	0	1	CryptoCurrency	Millionaire Alexander Amado Johnson and his wi...
2	sebikun	2019-05-31 23:48:29	4	29	CryptoCurrency	How is OmiseGOing? By Kasima Tharnpipitchai, D...
3	albuquerquetulio	2019-05-31 23:47:14	0	1	CryptoCurrency	Gemini is down??
4	cryptotradinglife	2019-05-31 23:46:36	0	1	CryptoCurrency	Analyst: Regardless of Newest Bitcoin Pullback...

Merging coin ids with Trends

```
#Concatenate list of dataframes
trend_df1 = pd.concat(google_data_list)

#Pivot the dataframe
trend_df2 = pd.DataFrame(trend_df1.pivot_table(index = 'date').unstack()).reset_index()

#Adding coin_id column
trend_df3 = trend_df2.merge(df_names[['name', 'coin_id']], how = 'left', left_on = 'level_0',

#Dropping columns
trend_df3 = trend_df3.drop(['level_0', 'name'], axis = 1)

#Rename columns
trend_df3.columns = ['date', 'trend', 'coin_id']

trend_df3.set_index(['date', 'coin_id'], inplace = True)
```

		trend
date	coin_id	
2015-01-01	0	4.145594
2015-01-02	0	20.267349
2015-01-03	0	27.176672
2015-01-04	0	24.873564
2015-01-05	0	26.255429

Cleaned/Parsed HTTP request & merged with coin ids

```
#Concat list of dataframes
pricing_df = pd.concat(pricing_list)


#Merge with coin_id
pricing_df2 = pricing_df.merge(df_names[['symbol', 'coin_id']], how = 'left', left_on = 'n
pricing_df3 = pricing_df2.drop(['name', 'symbol', 'time'], axis = 1)


#Make it a datetime object
pricing_df3['timestamp'] = pd.to_datetime(pricing_df3['timestamp'])


#Rename timestamp to date
pricing_df3.rename(columns = {'timestamp':'date'}, inplace = True)


pricing_df3.set_index(['date', 'coin_id'], inplace = True)
```

		close	high	low	open	volumefrom	volumeto
date	coin_id						
2017-08-11	0	0.06687	0.4444	0.06687	0.06950	39437.24	2994.43
2017-08-12	0	0.16250	0.1830	0.06687	0.06687	6232.81	765.38
2017-08-13	0	0.19000	0.3880	0.12500	0.16250	10382.08	1885.11
2017-08-14	0	0.32000	0.3300	0.13100	0.19000	7936.82	1788.88
2017-08-15	0	1.00000	1.8000	0.32000	0.32000	3467.80	2905.58

```
#Concatenate DF's of name-based subreddits
reddit_subscribers_name = pd.concat(subreddit_list2)

#Add the date_id column
reddit_subscribers_name1 = reddit_subscribers_name.merge(dates_df, how = 'left', left_on = 'date', right_on =

#Add the coin_id column
reddit_subscribers_name2 = reddit_subscribers_name1.merge(df_names[['name', 'coin_id']], how = 'left', left_c

#Drop the name column
reddit_subscribers_name3 = reddit_subscribers_name2.drop('name', axis = 1)

# CONCATENATE BOTH SUBSCRIBERS TABLES INTO ONE

reddit_subscribers_final = pd.concat([reddit_subscribers3, reddit_subscribers_name3])

#Drop Redundant date columns
reddit_subscribers_final2 = reddit_subscribers_final.drop('date', axis = 1)
```

		total_subscribers	subreddit
date	coin_id		
2015-01-01	2	882	AION
2015-01-02	2	884	AION
2015-01-03	2	885	AION
2015-01-04	2	887	AION
2015-01-05	2	896	AION

Storage

Compute Engine VM

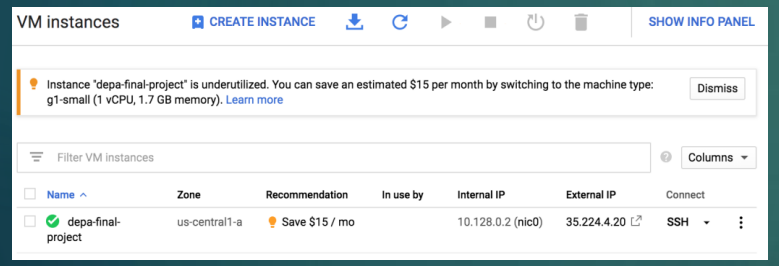
- Run all python web scraping and transformation scripts in a VM instance

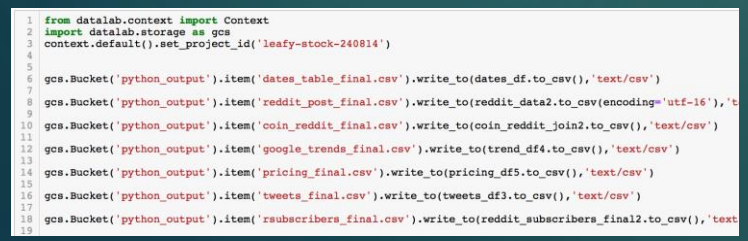
Storage Buckets

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

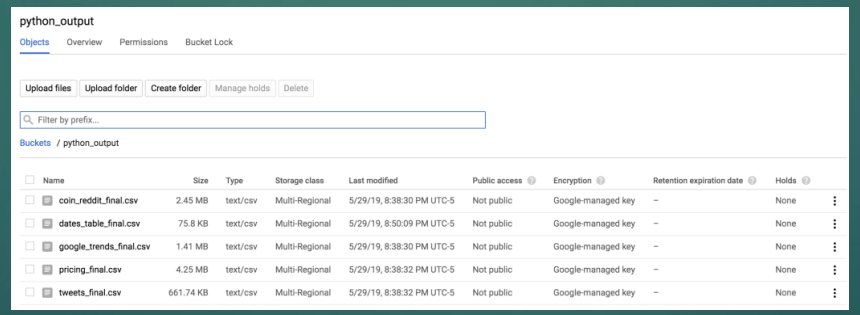
CloudSQL Instance

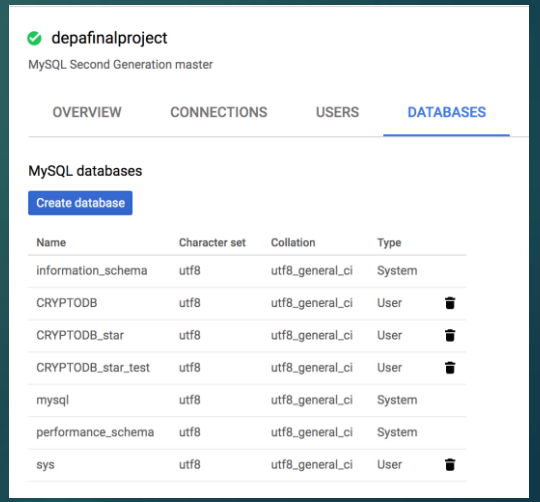
- Create DB through MySQL connection using DDL scripts
- Import Data from storage bucket csv's
- Second DDL/DML for star schema





```
from datalab.context import Context
import datalab.storage as gcs
context.default().set_project_id('leafy-stock-240814')

gcs.Bucket('python_output').item('dates_table_final.csv').write_to(dates_df.to_csv(),'text/csv')
gcs.Bucket('python_output').item('reddit_post_final.csv').write_to(reddit_data2.to_csv(encoding='utf-16'),'t
gcs.Bucket('python_output').item('coin_reddit_final.csv').write_to(coin_reddit_join2.to_csv(),'text/csv')
gcs.Bucket('python_output').item('google_trends_final.csv').write_to(trend_df4.to_csv(),'text/csv')
gcs.Bucket('python_output').item('pricing_final.csv').write_to(pricing_df5.to_csv(),'text/csv')
gcs.Bucket('python_output').item('tweets_final.csv').write_to(tweets_df3.to_csv(),'text/csv')
gcs.Bucket('python_output').item('rsubscribers_final.csv').write_to(reddit_subscribers_final2.to_csv(),'text
```

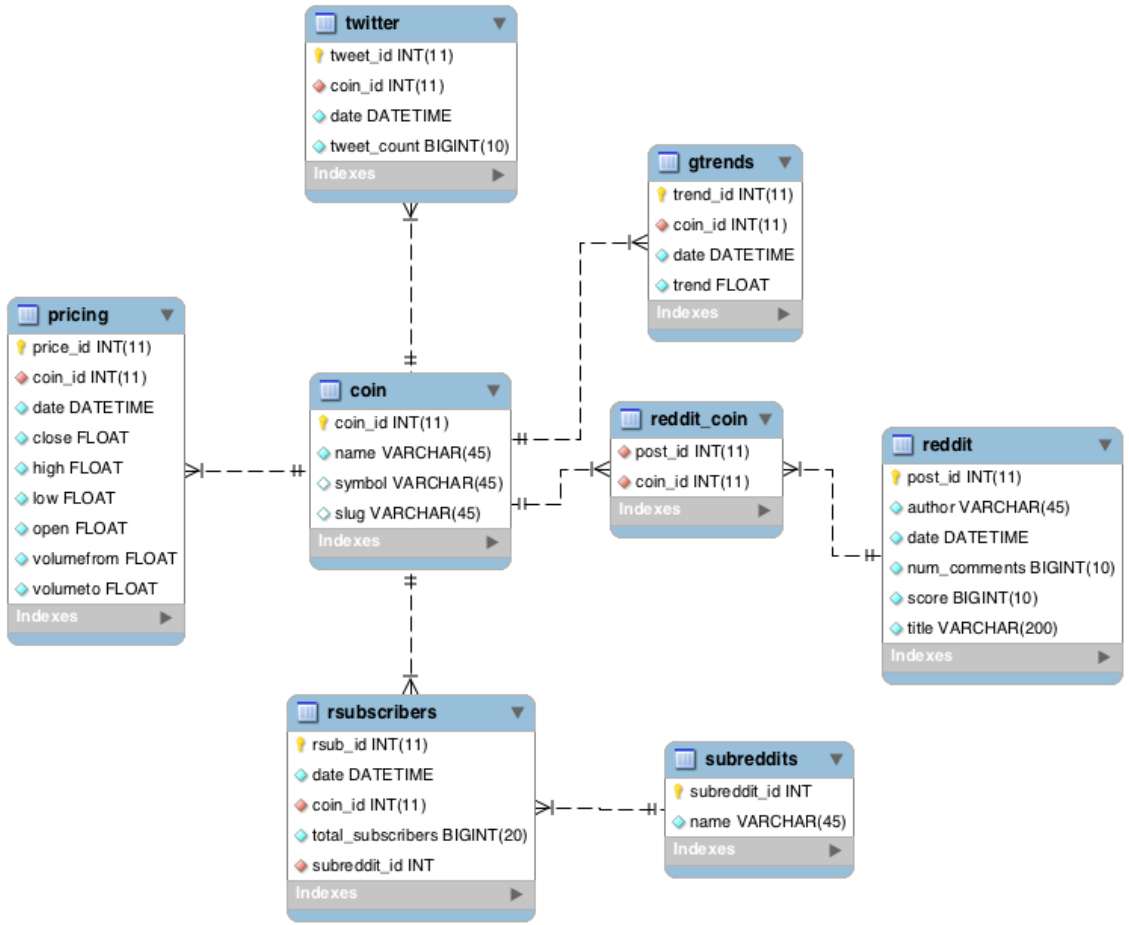




Enhanced Entity Relationship Diagram

- Entities:
  - **reddit:** /r/cryptocurrency subreddit post data
  - **rsubscribers:** subscriber count for over 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 data
  - **subreddits**: name of subreddit and subreddit id
  - **coin:** coin information containing coin names

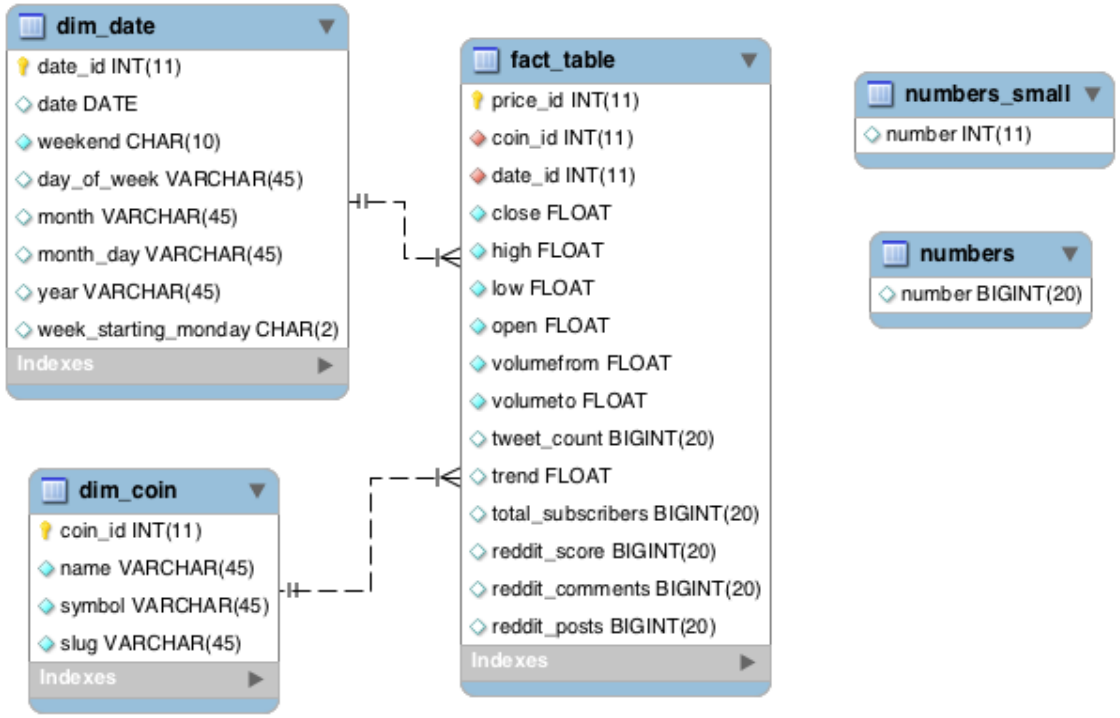
- **Normalized Data:**
  - No functional dependencies
  - No transitive dependencies



Dimensional Model (Star Schema)

- 4 Step Dimensional Design Process
  - Business Process
    - Trader/User-oriented
  - Grain
    - Currency-date level detail
  - Dimensions
    - Determined by grain statement
  - Facts
    - Social media aggregates

- Balancing user requirements with data realities



NoSQL and Graph Databases

mongoDB

Inappropriate Model

- MongoDB cannot model or store relationships if no complexity exists
- Our datasets are sourced through webscraping methods via APIs

Degraded Performance

- Use speed plummets as the amount of data increases and the number of join keys grows

Inappropriate Language

- "almost SQL" languages are inappropriate for SQL "joins" and similar functionality

Not ACID

- Non-relational databases like MongoDB do not follow ACID properties

neo4j

Inappropriate database

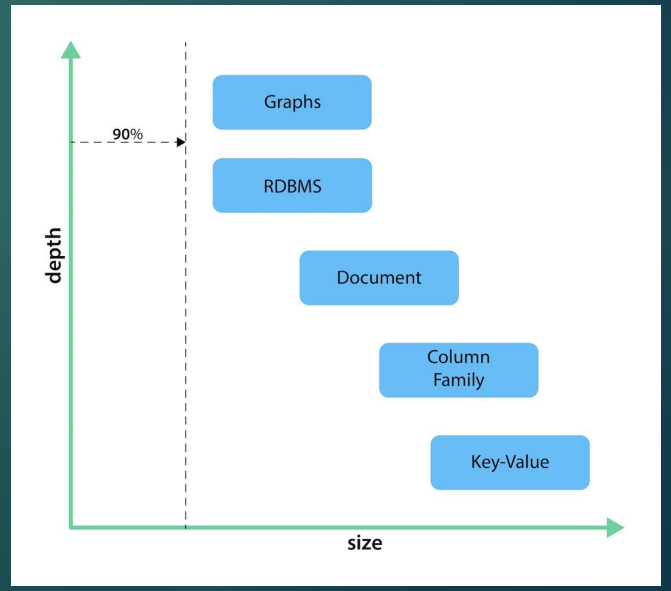
- Graph databases are inappropriate for traditional relational data.
- Difficult to perform mass analytics queries across all the relationships and records.

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

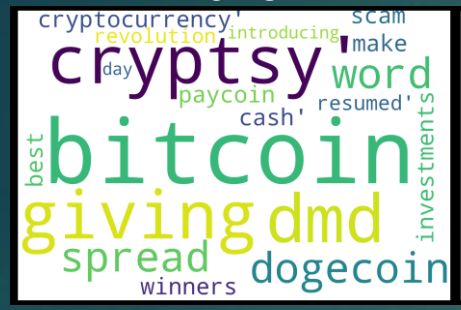
- A graph database is just a data store and doesn't give you a business-facing user interface to query or manage relationships.
- It will not provide advanced match and survivorship functionality or data quality capabilities.



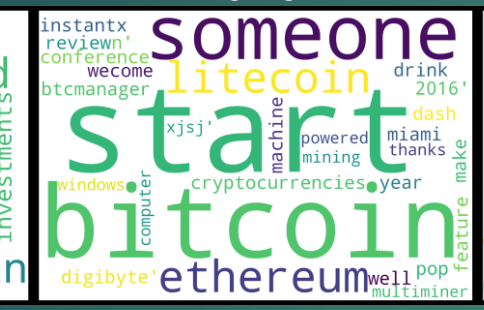
Analysis/Visualization

Basic Reddit Word Analysis

2015



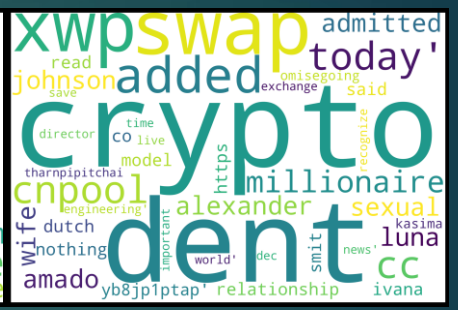
2016

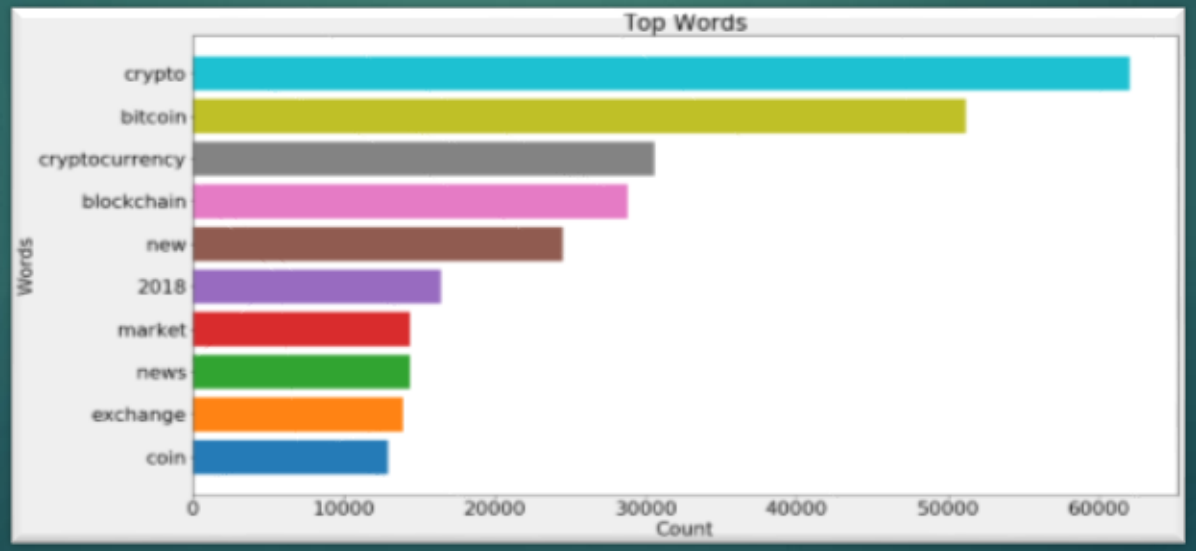


2017

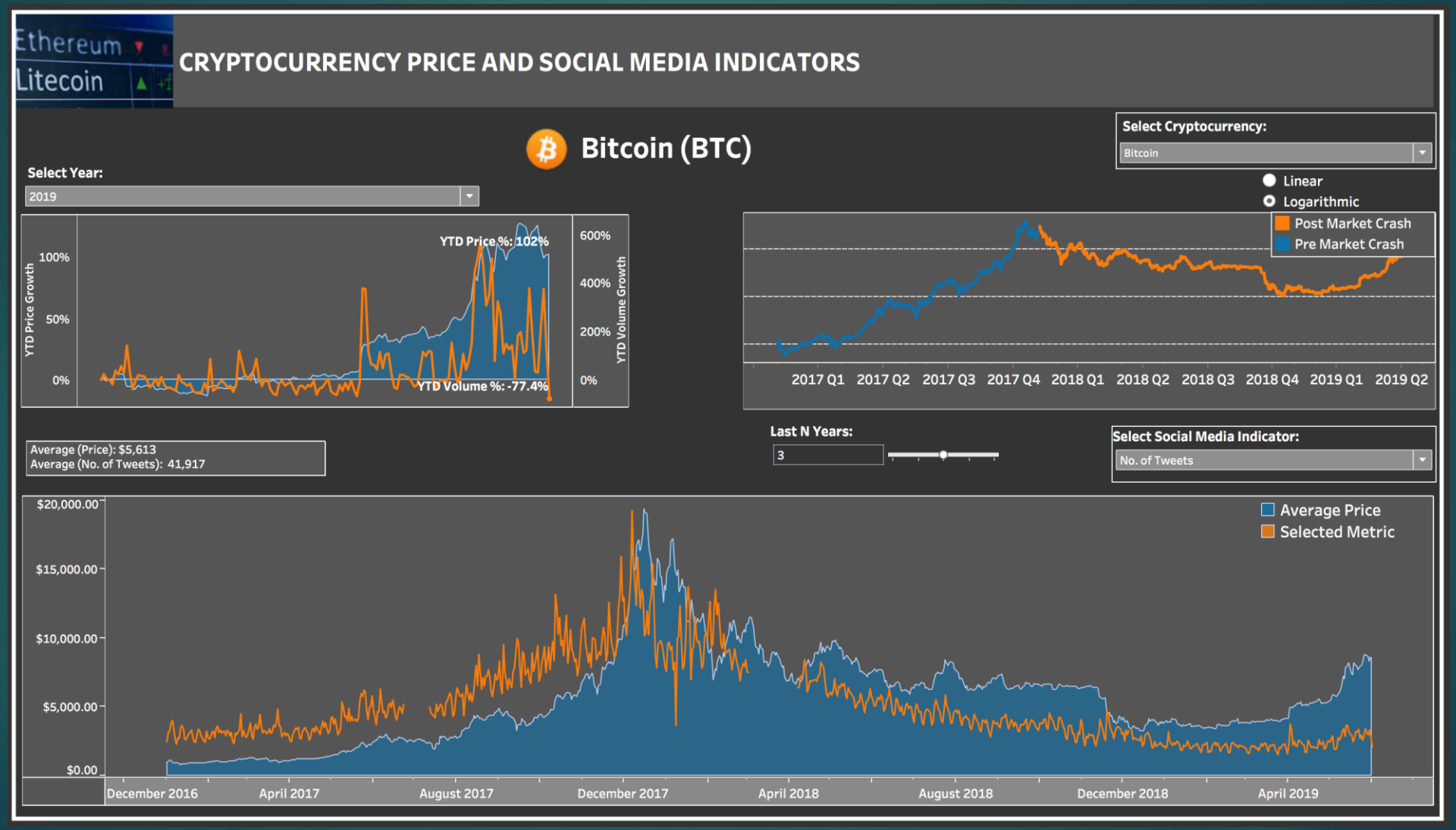
2018

2019



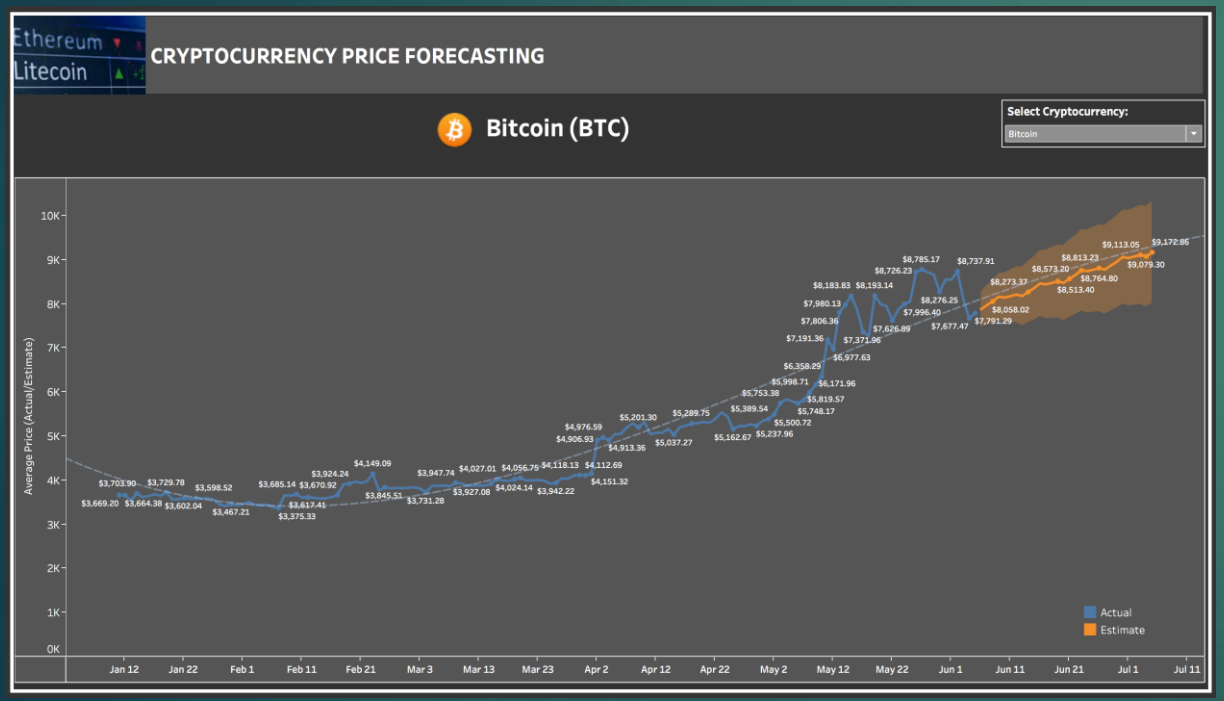


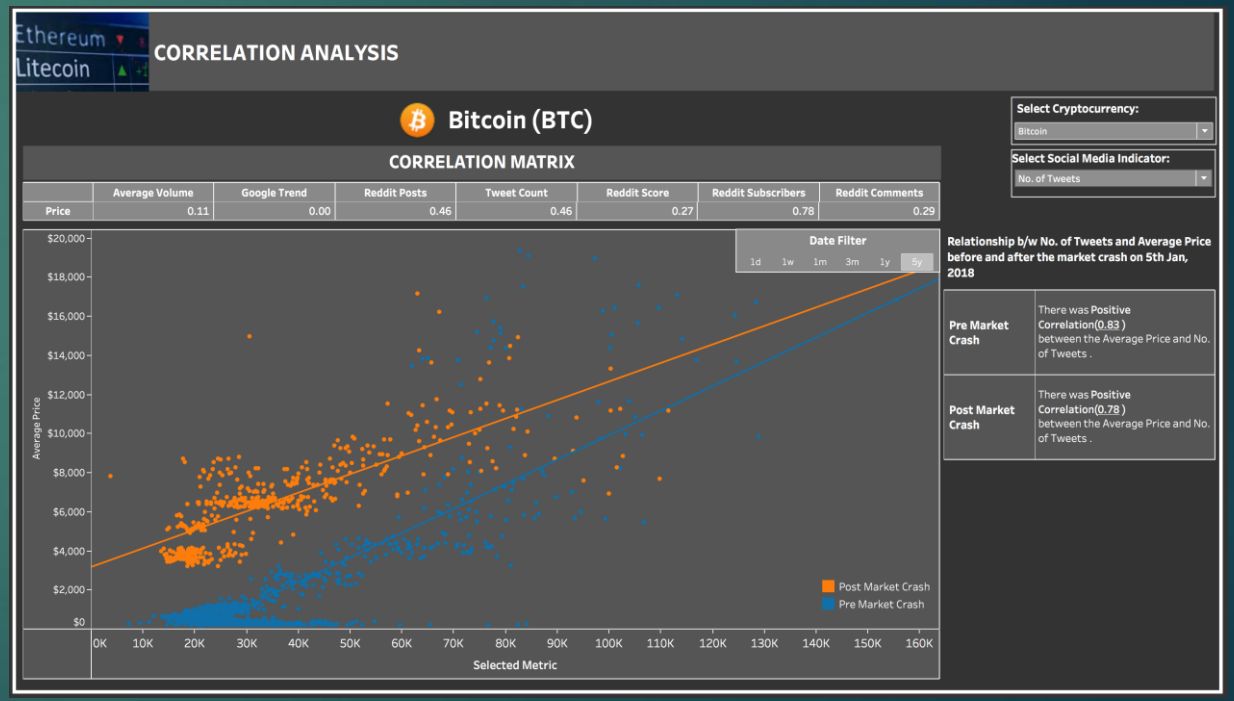
Analysis/Visualization



Analysis/Visualization

[Interactive Demo: Tableau Public Website]





Future Work & Improvements

- Acquire user demographic data
- Include additional market data
  - Market cap, circulating supply, spread, etc.
- Develop scheduler for real-time data ingestion
- Build predictive model
  - Using NLP on twitter and reddit data
- Revisit dimensional design process and level of granularity

Thank you!

ANY QUESTIONS?

Database Design Considerations

Requirements → Conceptual → Physical model

Application Type

Online Analytical Processing for Business Intelligence and end users (OLAP)

Data Format and Size

Cryptocurrency Financial Data

Business Requirements

Optimal user experience

Modeling Steps

- Elements relevant for pricing model analysis will be stored

- ACID properties used to set constraints, datatypes and ranges of the database

- technical, performance, and process requirements

Hardware and Software

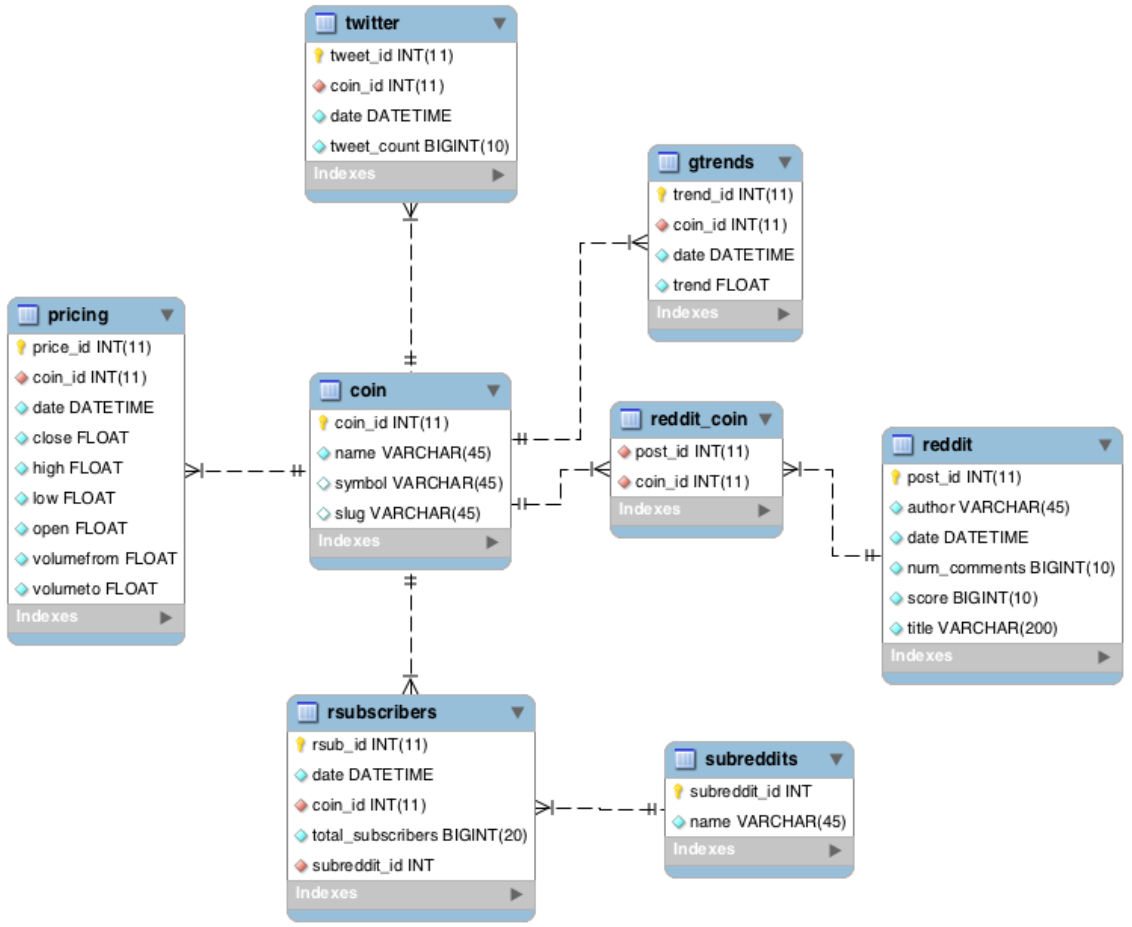
Google Cloud compute and virtual machines, and Google Cloud buckets for storage

- Financial constraints, usability constraints

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
- Normalization Achieved:
  - No functional dependencies
  - No transitive dependencies



Graph Databases/ Neo4j

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

Inappropriate database

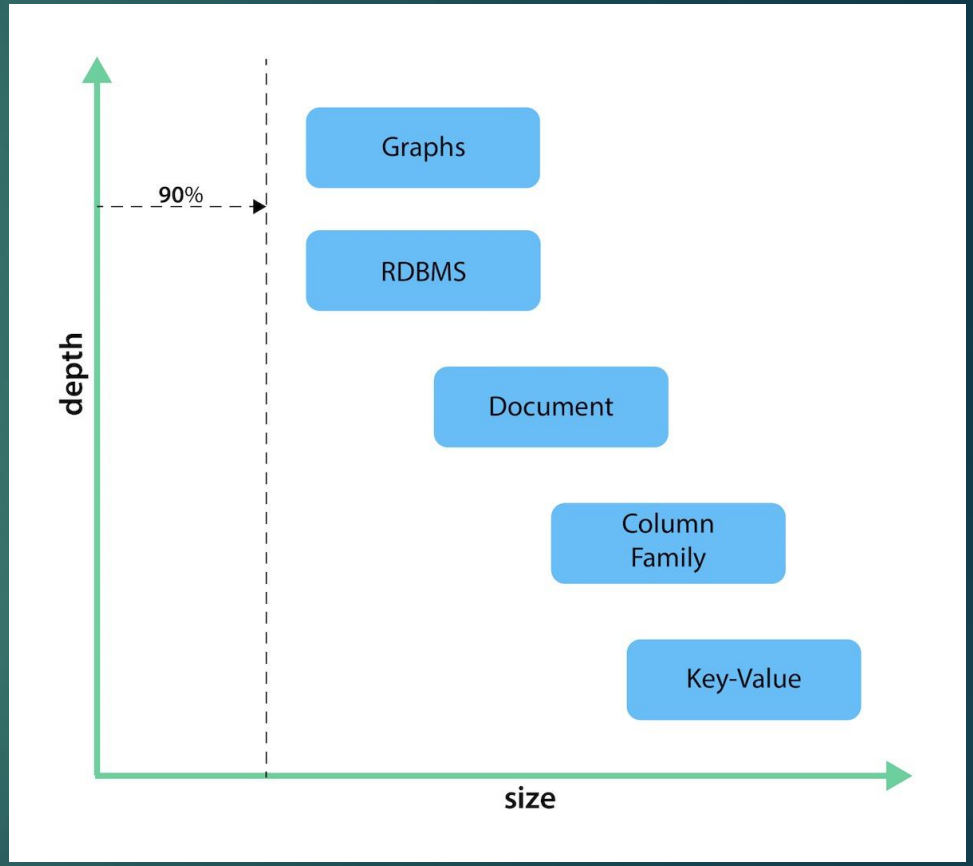
- Graph databases are inappropriate for traditional relational data.
- Difficult to perform mass analytics queries across all the relationships and records.

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.



APPENDIX

Summary of Data Analysis Pipeline

Conclusion

- Through this project we:
  - Created a database that

Ingestion

Beautifulsoup API

Scraped # of tweets per currency

Praw & Psaw APIs

Scraped all posts from/r/cryptocurrency

Cryptory API

Scraped google trend data per currency

Cryptocompare api

Scraped pricing data for every currency

Twitter Table

```
import re
import csv
import requests
from bs4 import BeautifulSoup
from IPython.display import HTML
```

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

for i in range(0,len(df_names['symbol'])-1):
    #testcoin_name='symbol'][-2

    coin = df_names['symbol'][i]
    url = 'https://bitinfocharts.com/comparison/tweets-'+coin.lower()+'.html'
    headers = {'User-Agent': 'Chrome/54.0.2840.98'}
    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

Reddit Table

Reddit posts from /r/Cryptocurrency from 2017-2019

```
import praw
from psaw import PushshiftAPI
```

```
#Connect to reddit API
reddit = praw.Reddit(client_id='GjfUHQE8AYnXLg', client_secret='PfbhtsXJGAAUNiEyHPHGPuFJUro', user_agent='DEPA_Pr
api = PushshiftAPI()
```

```
#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']))
```

Reddit Subscribers

```
# load package
from cryptory import Cryptory

# initialise object
# pull data from start of 2017 to present day
my_cryptory = Cryptory(from_date = "2017-01-01",to_date="2019-01-01", ascending=True)
```

```
#Subreddits based on symbol

empty_df = pd.DataFrame(columns = ['date','total_subscribers','subreddit'])
subreddit_list = []
for name in list(df_names['symbol']):
    try:
        subred = my_cryptory.extract_reddit_metrics(subreddit = name, metric = "total-subscribers", col_label
        if (type(subred) == ValueError):
            subreddit_list.append(empty_df)
        else:
            subreddit_list.append(subred)
    except:
        subreddit_list.append(empty_df)
```

	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

Google Trends table

```
#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;
```

```
trend_df1 = pd.concat(go
trend_df2 = pd.DataFrame
```

		trend
date	coin_id	
2015-01-01	0	4.145594
2015-01-02	0	20.267349
2015-01-03	0	27.176672
2015-01-04	0	24.873564
2015-01-05	0	26.255429

Pricing Table

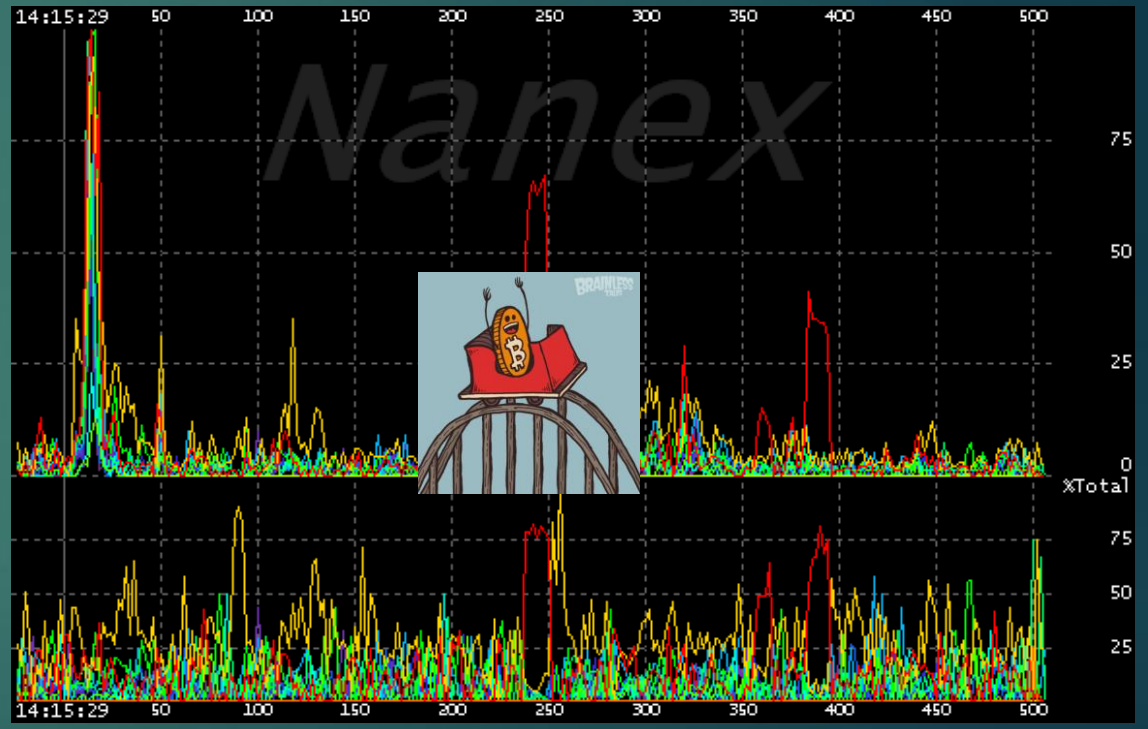
```
import requests
import datetime
import pandas as pd
```

```
def daily_price_historical(symbol, comparison_symbol, all_data=True, limit=1, aggregate=1, exchange=''):
    url = 'https://min-api.cryptocompare.com/data/histoday?fsym={}&tsym={}&limit={}&aggregate={}'\
            .format(symbol.upper(), comparison_symbol.upper(), limit, aggregate)
    if exchange:
        url += '&e={}'.format(exchange)
    if all_data:
        url += '&allData=true'
    page = requests.get(url)
    data = page.json()['Data']
    df = pd.DataFrame(data)
    df['timestamp'] = [datetime.datetime.fromtimestamp(d) for d in df.time]
    return df

pricing_list = []

for symbol in list(df_names['symbol']):
# initialise scraper with time interval
    try:
        df = daily_price_historical(symbol, 'USD')
        df['name'] = symbol
        pricing_list.append(df)
    except:
        df2 = pd.DataFrame(columns = df.columns)
        pricing_list.append(df2)
```

		close	high	low	open	volumefrom	volumeto
date	coin_id						
2017-08-11	0	0.06687	0.4444	0.06687	0.06950	39437.24	2994.43
2017-08-12	0	0.16250	0.1830	0.06687	0.06687	6232.81	765.38
2017-08-13	0	0.19000	0.3880	0.12500	0.16250	10382.08	1885.11
2017-08-14	0	0.32000	0.3300	0.13100	0.19000	7936.82	1788.88
2017-08-15	0	1.00000	1.8000	0.32000	0.32000	3467.80	2905.58

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.

Executive Summary

Project Background

- The cryptocurrency market is very volatile.
- Understanding the impact of social media on price direction can provide a cryptocurrency trader a better trading strategy.

Objective

- **Automated Data Integration** Connect Data Sources, Ingestion, and Consolidation
- **Efficient and Scalable Data Storage:** Normalized EER Model,OLAP Dimensional Model,and Cloud SQL Integration
- **Generate real-time insights on cryptocurrency price:** Leveraging Data Visualization, Statistical Analyses, and Forecasting.

Future Recommendations

- Leverage NLP and Sentiment Analysis
- Build a real-time predictive model utilizing all the metrcs , sentiment, and demography.
- Build a regression model based on metrics available to test the hypothesis further.

Prepare/Clean

Cleaning Tweets & Merging with coin_id

Creating Reddit-Coin Join Table

Cleaning Reddit Subscribers

```
#Convert Date into datetime object
tweets_df['Date'] = tweets_df['Date'].astype('datetime64')


#Remove the null rows
tweets_df = tweets_df[tweets_df['No. of Tweets'] != 'null']


#Convert No. of Tweets into int object
tweets_df['No. of Tweets'] = tweets_df['No. of Tweets'].astype('int')


#Adding coin_id column
tweets_df = tweets_df.merge(df_names[['symbol', 'coin_id']], how = 'left', left_on = 'Coin', right_on = 'symb
#Drop redundant name columns
tweets_df = tweets_df.drop(['Coin', 'symbol'], axis = 1)


#Change to tweet_count
tweets_df.rename(columns = {'No. of Tweets':'tweet_count'}, inplace = True)

#Set date & coinid as index
tweets_df.set_index(['Date', 'coin_id'], inplace = True)
```

```
# # Create reddit_coins Join Table
# #### Matches the reddit posts in reddit_data table to the coins in coins table


#Zip the names into tuple of (name, slug, symbol)
zipped_names = list(zip(df_names['name'],df_names['slug'],df_names['symbol']))

#Create a search list - seperating the three terms with OR operator (|)
search_list = []
for (name, slug, symbol) in zipped_names:
    listt = [name, slug, symbol]
    pat = '|'.join(listt)
    search_list.append(pat)


#Add boolean/dummy columns
import re
dummy_df = pd.DataFrame(dict((name, reddit_data.title.str.contains(name, re.IGNORECASE))
                             for name in search_list))

#Convert dummy columns into rows with post_id as the index
i, j = np.where(dummy_df)

coins_mentioned_series = pd.Series(dict(zip(zip(i, j), dummy_df.columns[j])))


#Create final join table between reddit and coins table. Rename columns of dataframe.
coin_reddit_join = pd.DataFrame(coins_mentioned_series).reset_index()
coin_reddit_join.columns = ['post_id', 'coin_id', 'coin_name']
coin_reddit_join = coin_reddit_join.drop('coin_name', axis = 1)


coin_reddit_join2 = coin_reddit_join.set_index(['post_id', 'coin_id'],)

coin_reddit_join2
```

post_id	coin_id
1	77
2	66
4	11
	77
5	72
7	11
	95
8	77
11	77
16	4
	5
	11
	35
	36
	66
	77

```
#Concat list of dataframes
pricing_df = pd.concat(pricing_list)

#Merge with coin_id
```

```
#Concatenat
reddit_subs

#Add the da
reddit_subs

#Add the co
reddit_subs

#Drop the n
reddit_subs

# CONCATENA

reddit_subs

#Drop Redun
reddit_subs
```

```
#Concatenate list of dataframes
trend_df1 = pd.concat(google_data_list)

#Pivot the dataframe
trend_df2 = pd.DataFrame(trend_df1.pivot_table(index = 'date').unstack()).reset_index()

#Adding coin_id column
trend_df3 = trend_df2.merge(df_names[['name', 'coin_id']], how = 'left', left_on = 'level_0',

#Dropping columns
trend_df3 = trend_df3.drop(['level_0', 'name'], axis = 1)

#Rename columns
trend_df3.columns = ['date', 'trend', 'coin_id']

trend_df3.set_index(['date', 'coin_id'], inplace = True)
```

```
ol', 'coin_id']], how = 'left',
l', 'time'], axis = 1)

ing_df3['timestamp'])

e'}, inplace = True)

lace = True)
```

	author	date	num_comments	score	subreddit	title
post_id						
0	Mahaprajapati	2019-05-31 23:56:44	0	1	CryptoCurrency	Swap (XWP) added to CNPool.cc today
1	webnowcompany	2019-05-31 23:55:15	0	1	CryptoCurrency	Millionaire Alexander Amado Johnson and his wi...
2	sebikun	2019-05-31 23:48:29	4	29	CryptoCurrency	How is OmiseGOing? By Kasima Tharnpipitchai, D...
3	albuquerquetulio	2019-05-31 23:47:14	0	1	CryptoCurrency	Gemini is down??
4	cryptotradinglife	2019-05-31 23:46:36	0	1	CryptoCurrency	Analyst: Regardless of Newest Bitcoin Pullback...

		total_subscribers	subreddit
date	coin_id		
2015-01-01	2	882	AION
2015-01-02	2	884	AION
2015-01-03	2	885	AION
2015-01-04	2	887	AION
2015-01-05	2	896	AION

Data Extraction and Web Scraping

Reddit

- Praw/psaw API
- Scraped all reddit posts from/r/cryptocurrency for our timeframe of 2 years

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='GjfUHQR8AYnXLg', 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

Twitter

- Beautifulsoup API
- Scraped # of tweets per currency. Granularity at the day level

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')

    s = soup.find_all('script')

    data_1 = re.findall(r'{\"new\dDate.*\}]}', str(s))
    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

Google Trends

- Cryptory API
- Scraped google trend data for every currency in our timeframe of interest

Google Trends table

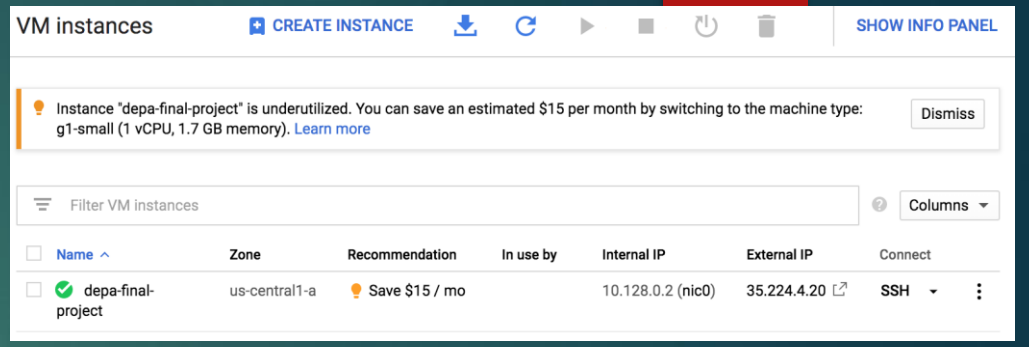
```
In [ ]:
#Pulling google trends data from cryptory

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

Data Storage

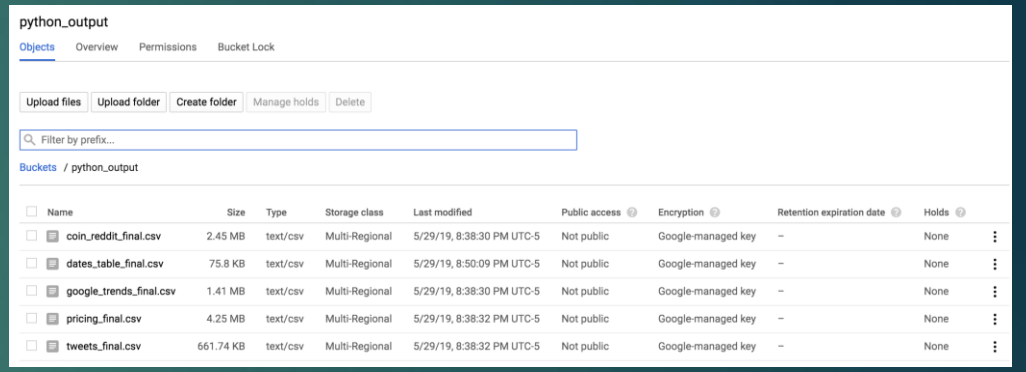
Compute Engine VM

- 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