'Meme' Stock Outlier Detection

MSDS 436 Final Project

Arthur Swanson
Husein Adenwala

Executive Summary

Introduction: Meme stocks have become very popular among retail investors that comment on social media platforms during the COVID-19 pandemic. Meme stocks are created when a company's shares start trending with individual investors on social media platforms and quickly skyrocket in price.

With a large amount of data being produced on social media platforms, it is difficult to identify the meme stock before or when the price takes off. Unsupervised algorithms aided by stock data can aid in identifying metrics of a meme stock.

Results: We provide evidence that these stocks display common stylized facts for the dynamics of price, trading volume, and social media activity. Understanding these properties helps investment analysts in their decisions.

Conclusion: Stock price was inversely correlated (mildly) to positive sentiment on Twitter. K-means clustering shows that Tesla was the most distinct cluster and could potentially suggest that we can detect meme stock activity using K-means clustering but more data over a longer period is needed.

Research Objective

Identify stocks

Identify meme stocks using unsupervised clustering

Find insights

Find insights from twitter sentiment analysis and stock price data

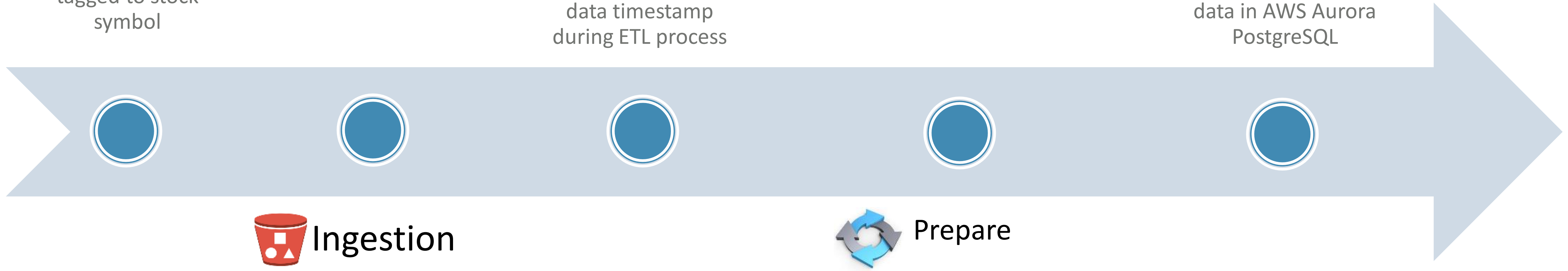
Identify trends

Help investment analysts identify trends in stock attribution data to make better investment decisions.

Sourcing the Data

Data Source	Process
Twitter API	Hourly batch scripts to get data using the Twitter development (free tier) API for a group of stocks limited by API count limitation. Stock symbols were used as key words to search for the tweets; however, this did capture many irrelevant tweets that were cleaned up during sentiment analysis.
Yahoo Finance API	Due to limits in number of requests on Y Finance API, we ran on-demand scripts on an EC2 instance to get Stock price attribution data. This step was done after sentiment analysis and data transformation as the Y Finance data is curated and did not require any pre-processing.

Data Pipeline Overview



Twitter API

Using a Twitter developer account, Python API to get tweets tagged to stock symbol

Ingestion

Load tweet data in S3 bucket with batch scripts

Y Finance API

Get min, max, average, volume, open, close, ticker, data timestamp during ETL process

Prepare

Sentiment analysis on tweets.
Merge Y Finance and Twitter data on stock symbol and date stamp

Database

Amazon RDS

Store the transformed data in AWS Aurora PostgreSQL

Data Ingestion, Analysis and Preparation

- Hourly python batch scripts to collect Twitter data using API.
- Total of 83,000 tweets were collected with limited twitter development access.
- The tweets with date and timestamps were ingested in Amazon S3 in csv format.
- Used NLP pipeline in an Ec2 instance with python to filter irrelevant tweets, perform text preprocessing, tokenization and perform sentiment analysis using a Bert based transformer model.
- Used Yahoo Finance API to collect stock attribution data for the stocks and date and timestamps on post processed Twitter data.
- Yahoo Finance and Twitter sentiment data was loaded to AWS aurora (PostgreSQL).

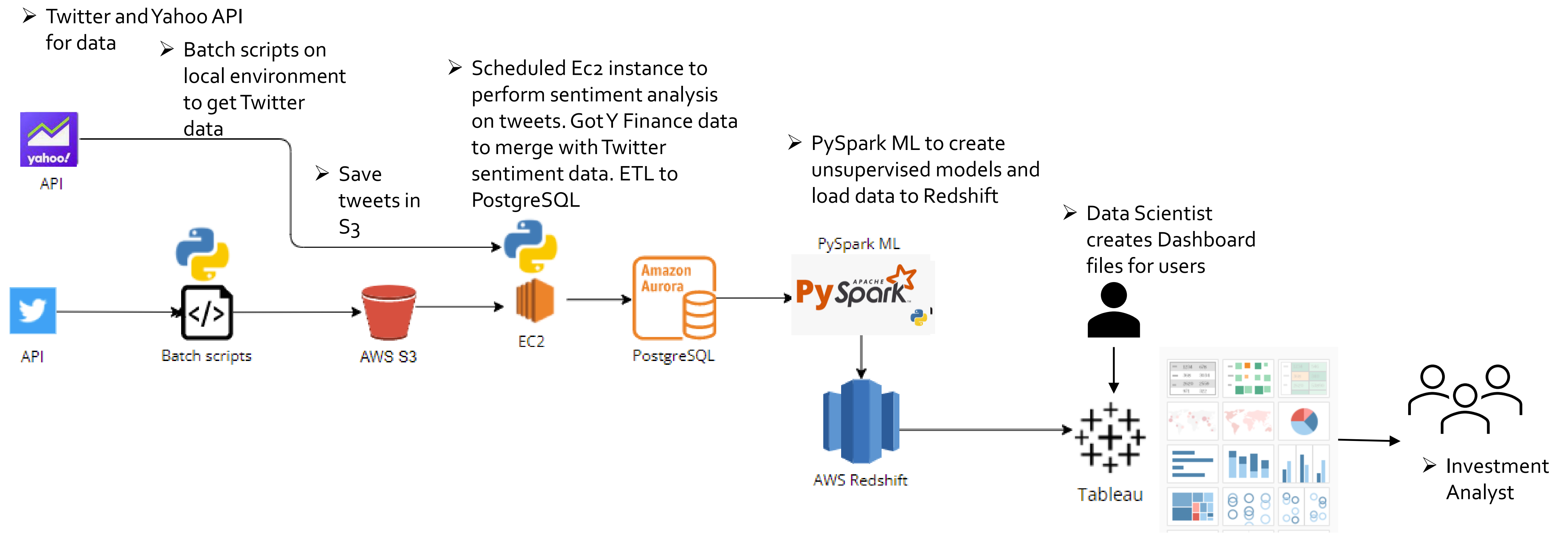
Data Governance

Only collected tweets on days where stock market was open for trading.

Collected tweets at a random sample and did not save user information.

Methodology

Overview of the System Design



➢ Twitter and Yahoo API for data
➢ Batch scripts on local environment to get Twitter data
➢ Save tweets in S3
➢ Scheduled Ec2 instance to perform sentiment analysis on tweets. Got Y Finance data to merge with Twitter sentiment data. ETL to PostgreSQL
➢ PySpark ML to create unsupervised models and load data to Redshift
➢ Data Scientist creates Dashboard files for users
API
API
Batch scripts
AWS S3
EC2
Amazon Aurora
PostgreSQL
PySpark ML
AWS Redshift
Tableau
➢ Investment Analyst

Methodology and Tools used

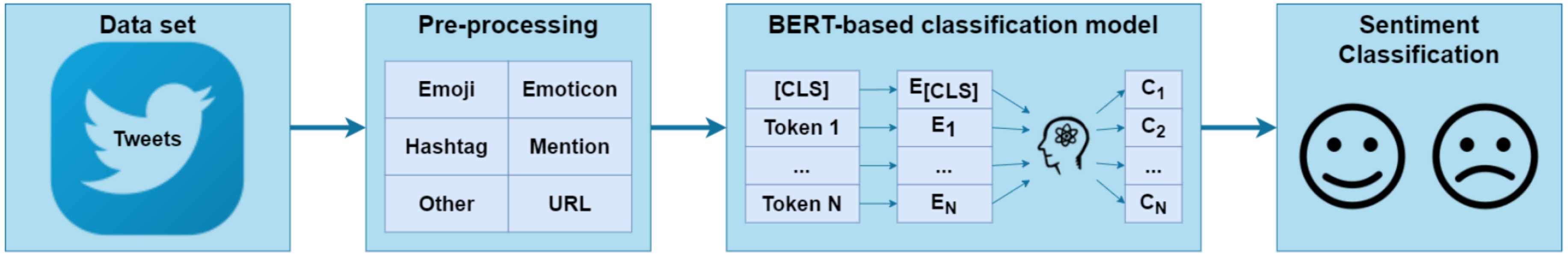
- Python to collect data from API
- Automation/Scheduling Scripts and Batch Processing
- Boto3 - Interface with Amazon APIs
- AWS S3 – Storage
- Ec2 instance with Cron scheduling python- ETL (Data Transformation, NLP and getting Y Finance data)
- AWS Aurora - PostgreSQL to store the processed data
- PySpark ML – K-Means clustering model (DBSCAN did not provide discernable clusters)
- The transformed Yahoo and Twitter sentiment data with K-means clustering was pushed to AWS redshift
- AWS Redshift - Data Warehouse and Insight Result Storage for Deployment to Tableau
- Created Tableau TWB files that provide clustering results and dashboard for the end users

NLP Sentiment Analysis

- Python NLTK library for cleaning text: Removed punctuation and special characters, lower cased, removed URLs and converted emoji to text.

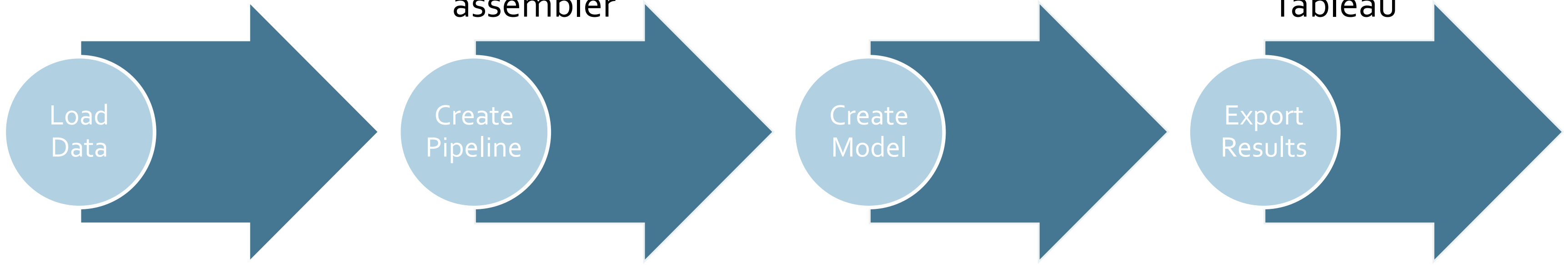
- Used FLAIR NLP framework in python to tokenize and encode the text to get and predict the sentiment of the tweet (FLAIR uses a BERT based transformer model)

- Tweets were classified into positive and negative sentiments



Unsupervised Models

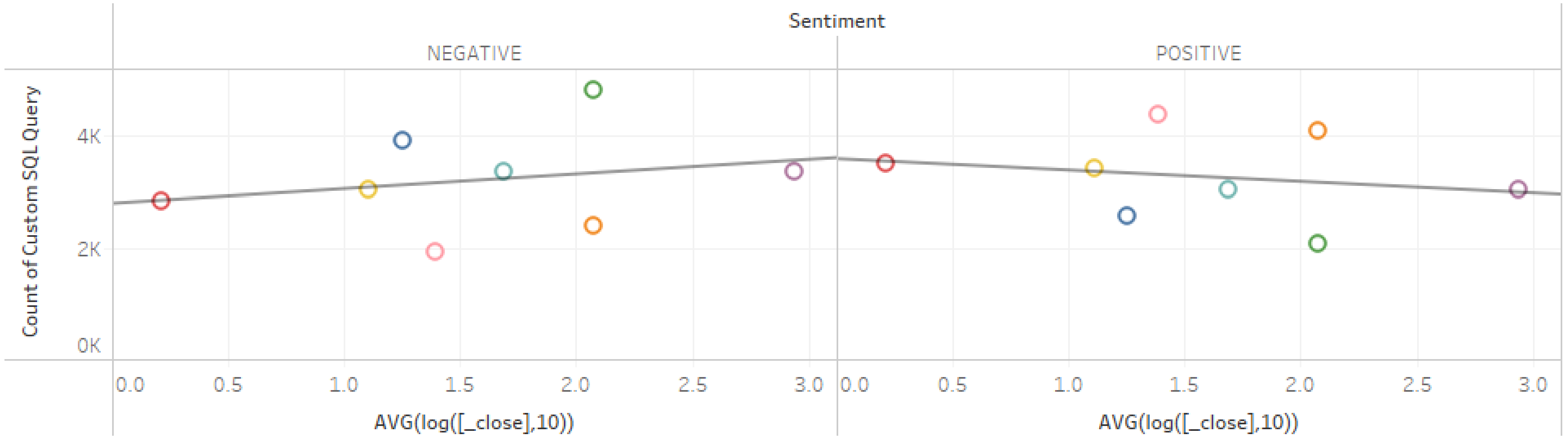
- ➢ Loaded data from AWS Aurora using sqlalchemy
- ➢ Imported Pipeline stack string indexer, standard scalar and vector assembler
- ➢ Created the model and evaluated the Silhouette score for the K-mean
- ➢ Exported the model output merge with the loaded data to Tableau



Results: Price – Sentiment Trend

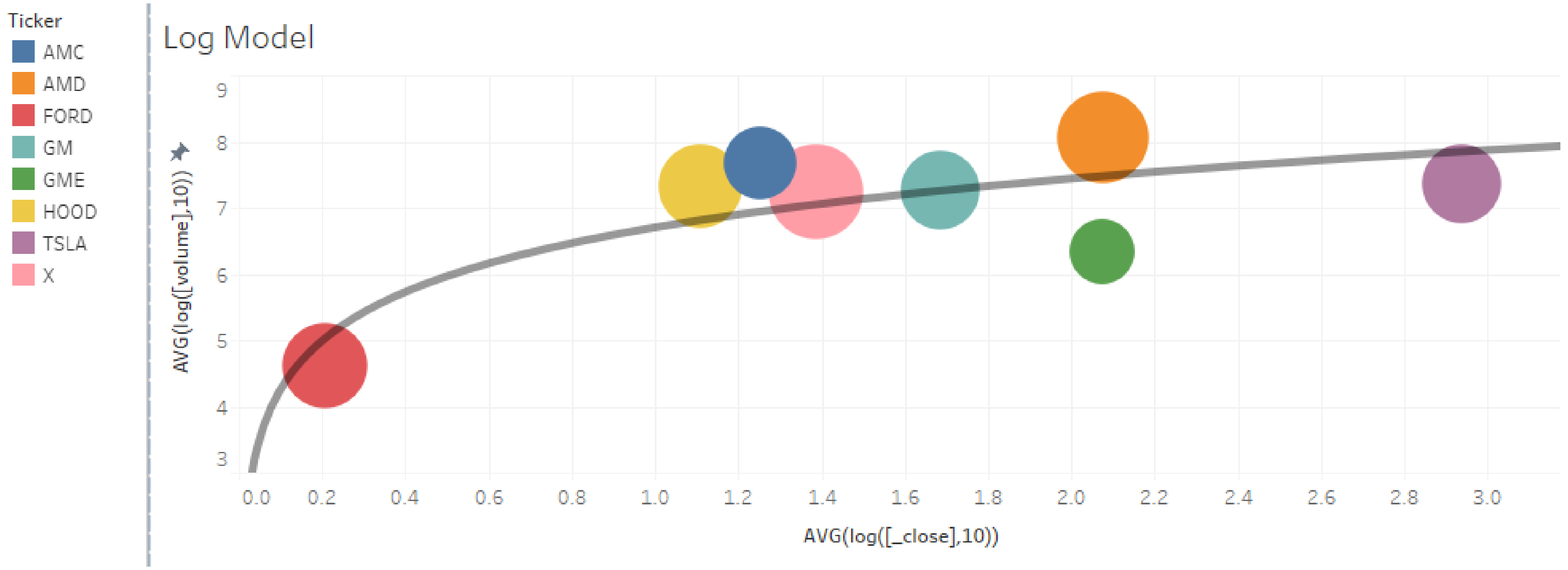
- Inverse (modest) relation between twitter sentiment and price
- Need more data over longer period to assess trend

Overall Trend



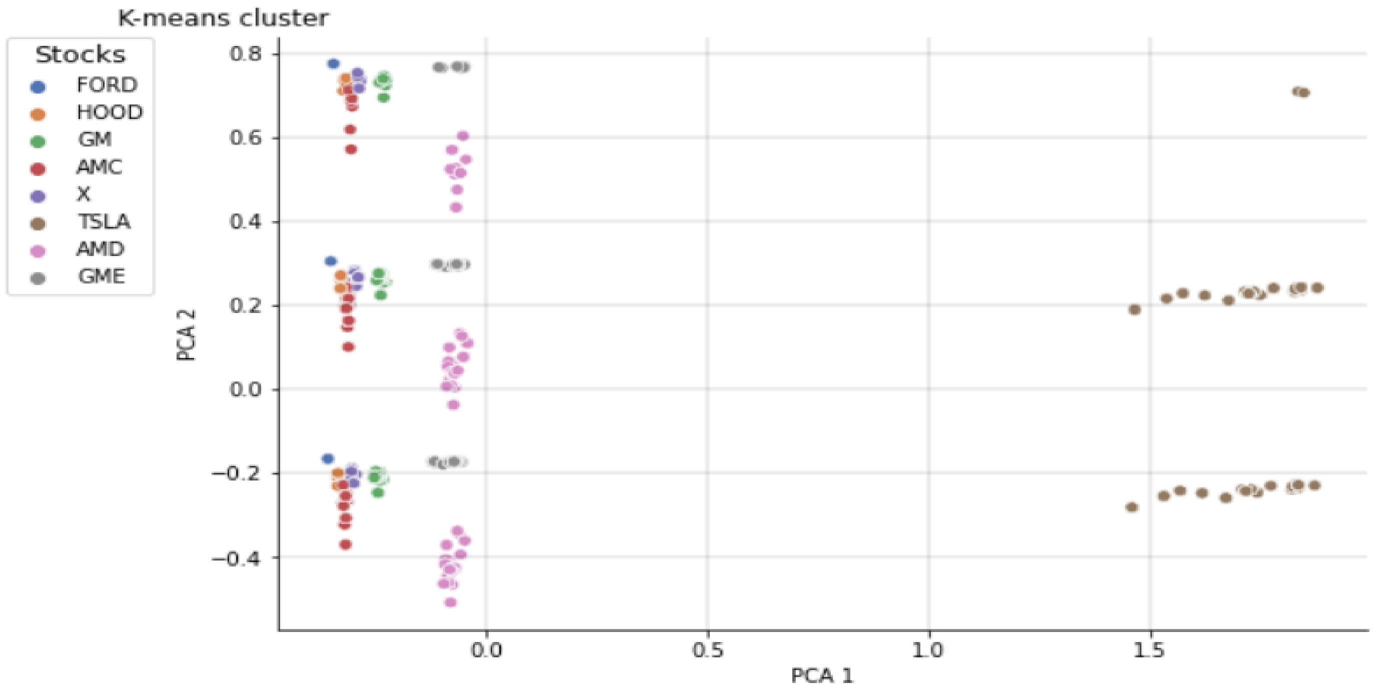
Results: Price - Volume Trend

- Price and Volume on logarithmic trend.
- TSLA and FORD on extreme end.



Results: K-Means Cluster

- TSLA can be identified as a distinct cluster
- However, FORD and TSLA were 'meme' stock and the cluster only detected TSLA



Conclusion

Recommendations

Using paid subscription, we can backlog tweets over longer period.

Get a better stock API for more resolute data over 5-minute windows.

Storing tweet and pricing in a NoSQL database like MongoDB would allow for flexibly in changing the schema as we gather more knowledge about the data requirements.

Lessons Learned

Amazon EMR was not utilized and would have been a valuable feature to add.

Stream processing would be ideal application but could not be implemented.

Using all the processes in AWS would be useful for larger application.

References

Das A (2021, Feb 10). K Means Clustering using PySpark on Big Data. Towards Data Science. Retrieved March 10 2022, from, https://towardsdatascience.com/k-means-clustering-using-pyspark-on-big-data-6214beacdc8b.

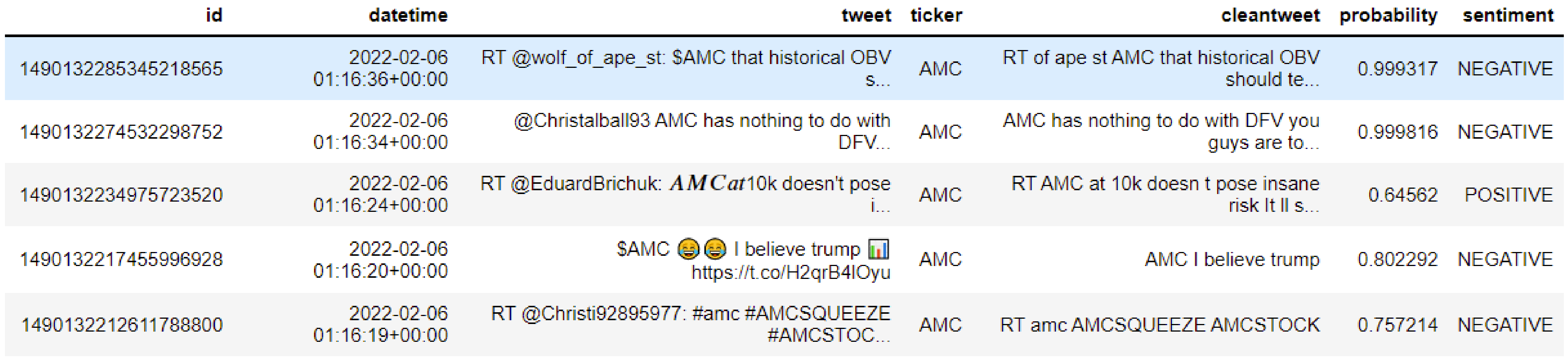
Jain S (2020, July 26). An Implementation of DBSCAN on PySpark. Towards Data Science. Retrieved March 10 2022 from, https://towardsdatascience.com/an-efficient-implementation-of-dbscan-on-pyspark-3e2be646f57d.

Magajna T (2018, Dec 24) Text Classification with State of the Art NLP Library — Flair. Towards Data Science. Retrieved March 10 20202, form https://towardsdatascience.com/text-classification-with-state-of-the-art-nlp-library-flair-b541d7add21f.

Galarnyk M (2017, Aug 2). T (2018, Dec 24) Text Classification with State of the Art NLP Library — Flair. Towards Data Science. Retrieved March 10 20202 form, https://towardsdatascience.com/setting-up-and-using-jupyter-notebooks-on-aws-61a9648db6c5.

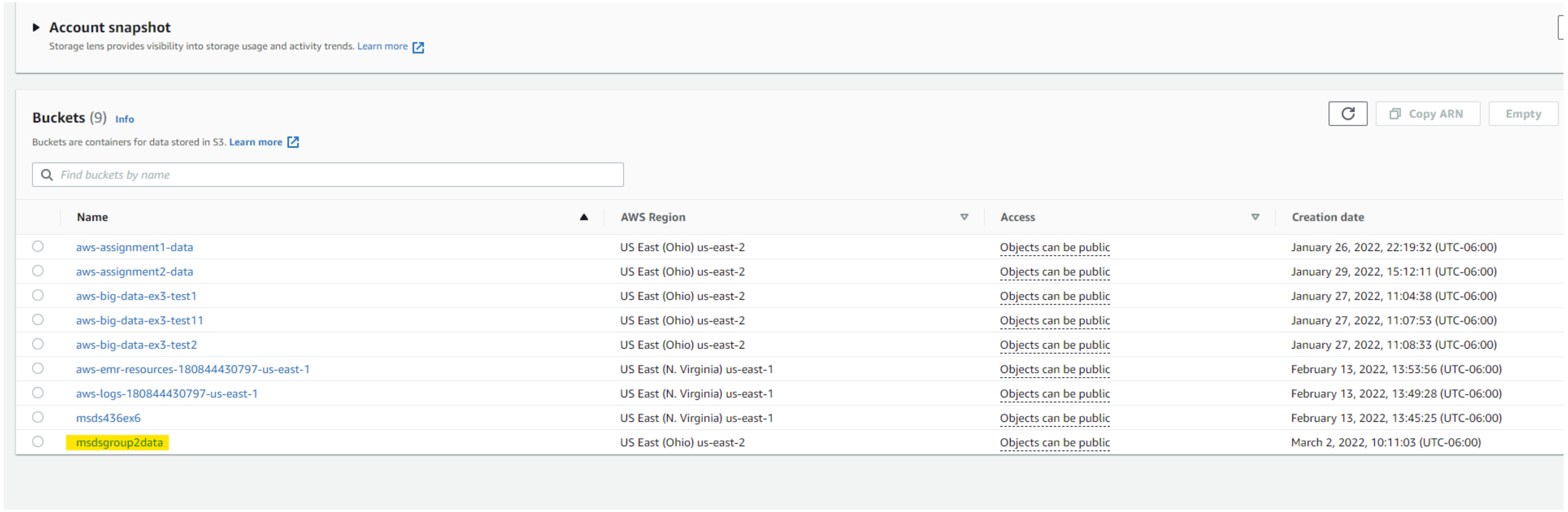
Appendix

NLP Sentiment example



id	datetime	tweet	ticker	cleantweet	probability	sentiment
1490132285345218565	2022-02-06 01:16:36+00:00	RT @wolf_of_ape_st: $AMC that historical OBV s...	AMC	RT of ape st AMC that historical OBV should te...	0.999317	NEGATIVE
1490132274532298752	2022-02-06 01:16:34+00:00	@Christalball93 AMC has nothing to do with DFV...	AMC	AMC has nothing to do with DFV you guys are to...	0.999816	NEGATIVE
1490132234975723520	2022-02-06 01:16:24+00:00	RT @EduardBrichuk: $AMCat$10k doesn't pose i...	AMC	RT AMC at 10k doesn t pose insane risk It ll s...	0.64562	POSITIVE
1490132217455996928	2022-02-06 01:16:20+00:00	$AMC 😂😂 I believe trump 📊 https://t.co/H2qrB4lOyu	AMC	AMC I believe trump	0.802292	NEGATIVE
1490132212611788800	2022-02-06 01:16:19+00:00	RT @Christi92895977: #amc #AMCSQUEEZE #AMCSTOC...	AMC	RT amc AMCSQUEEZE AMCSTOCK	0.757214	NEGATIVE

S3 Bucket example



▸ **Account snapshot**

Storage lens provides visibility into storage usage and activity trends. Learn more

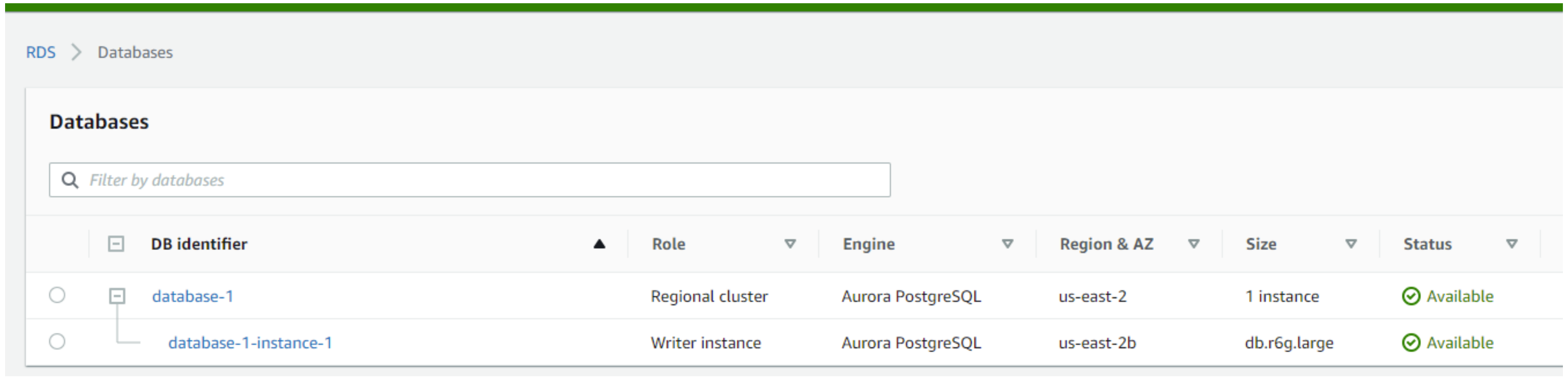
Buckets (9) Info

Buckets are containers for data stored in S3. Learn more

Find buckets by name

Name	AWS Region	Access	Creation date
aws-assignment1-data	US East (Ohio) us-east-2	Objects can be public	January 26, 2022, 22:19:32 (UTC-06:00)
aws-assignment2-data	US East (Ohio) us-east-2	Objects can be public	January 29, 2022, 15:12:11 (UTC-06:00)
aws-big-data-ex3-test1	US East (Ohio) us-east-2	Objects can be public	January 27, 2022, 11:04:38 (UTC-06:00)
aws-big-data-ex3-test11	US East (Ohio) us-east-2	Objects can be public	January 27, 2022, 11:07:53 (UTC-06:00)
aws-big-data-ex3-test2	US East (Ohio) us-east-2	Objects can be public	January 27, 2022, 11:08:33 (UTC-06:00)
aws-emr-resources-180844430797-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	February 13, 2022, 13:53:56 (UTC-06:00)
aws-logs-180844430797-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	February 13, 2022, 13:49:28 (UTC-06:00)
msds436ex6	US East (N. Virginia) us-east-1	Objects can be public	February 13, 2022, 13:45:25 (UTC-06:00)
msdsgroup2data	US East (Ohio) us-east-2	Objects can be public	March 2, 2022, 10:11:03 (UTC-06:00)

Amazon Aurora PostgreSQL instance

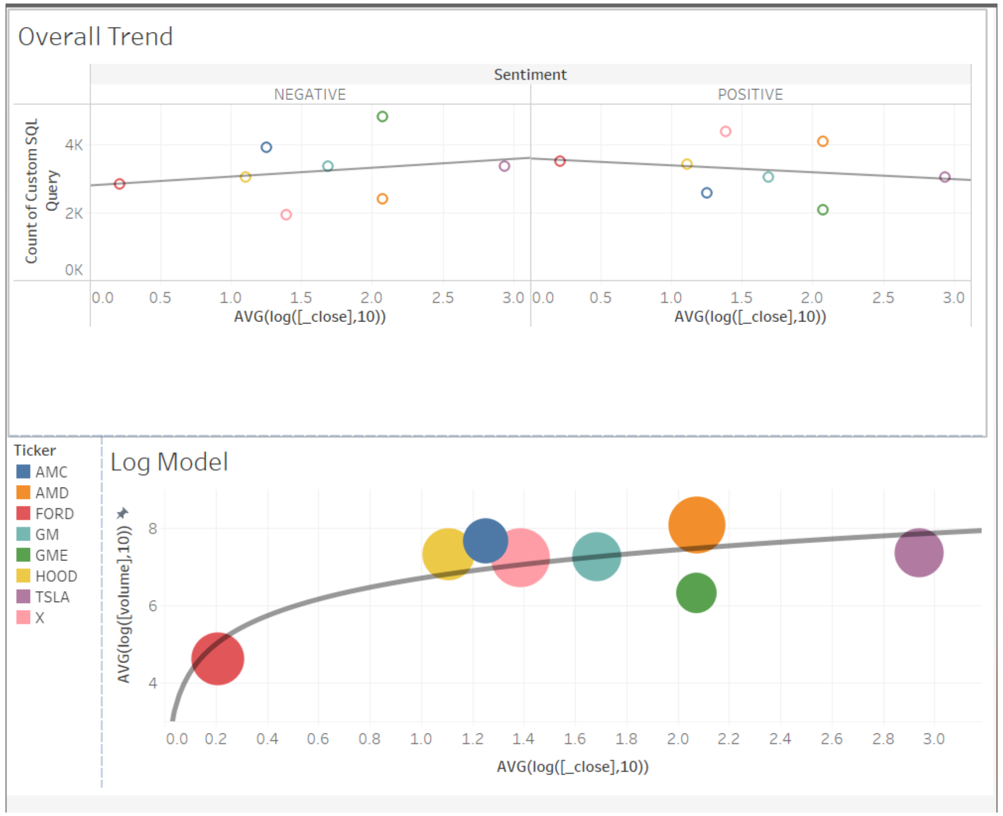

RDS > Databases

Databases

Filter by databases

DB identifier	Role	Engine	Region & AZ	Size	Status
database-1	Regional cluster	Aurora PostgreSQL	us-east-2	1 instance	✓ Available
database-1-instance-1	Writer instance	Aurora PostgreSQL	us-east-2b	db.r6g.large	✓ Available

Tableau Dashboard example


Overall Trend
Sentiment
NEGATIVE
POSITIVE
Count of Custom SQL Query
4K
2K
0K
AVG(log([_close],10))
Ticker
AMC
AMD
FORD
GM
GME
HOOD
TSLA
X
Log Model
AVG(log([volume],10))
AVG(log([_close],10))