Title: Sentiment Analysis and Stock Prediction using Knowledge Graphs

Team Name: Group 5

Members: Maneel Reddy Karri, Gurusankar Gopalakrishnan, Devendra Govil, Akshay Pamnani, Youshi Zhang

Problem/Motivation Statement:

The Russell 3000 stocks[1] contain large, mid & small-cap stocks which represent 98% of the US public equity market. Sentiment analysis on these stocks can be generated by using their 10K filing reports. Since 10K reports are a quarterly/yearly reporting exercise, the daily movement of stocks can be explained better by parsing social media posts/comments. Reddit was selected for this exercise. These sentiment scores generated from both Reddit and 10K filings are being used to predict stock prices by using a network graph that has the sentiment scores and company characteristics as its weights.

DataSet and analytical goals:

The APIs used for data collection are as follows:

1. Reddit PMAW
- API to scrape Reddit data
- This will provide textual data basis which we can compute Sentiment Scores
2. Hive Index of 28 Popular SubReddits
- This provides some popular investing subreddits that we plan to scrape
3. Edgar API by SEC
- This API provides SEC filings data for all companies regulated by SEC
- This will provide information to generate sentiment scores as well as to build knowledge graphs
4. Alpha Vantage
- This API provides stock price information, which is the ultimate prediction target

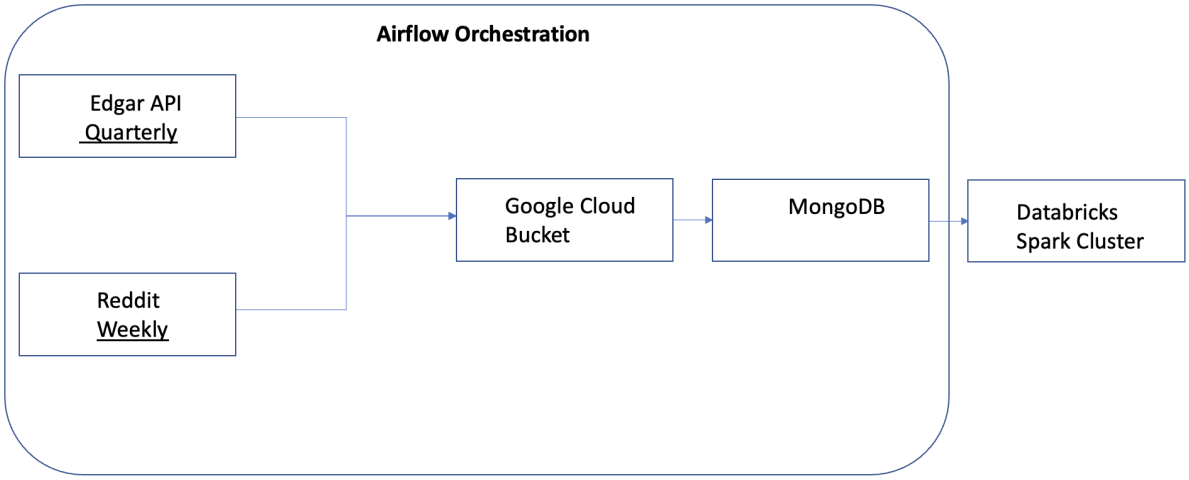
The analytical goals are as follows:

1. Generate sentiment analysis scores from Reddit and Form 10K filing and use it for stock prediction.
2. Predict stock prices using network graphs of companies binding them by sentiment scores.

---

[1] https://www.investopedia.com/terms/r/russell3000-value.asp#:~:text=It%20represents%20approximately%2098%25%20of,capitalization%20is%20at%20%242%20billion.

Overview of Data Engineering Pipeline:



1. **Edgar API:** Each stock has a CIK number associated with it. This CIK number can be thought of as the primary key for every stock on Edgar API. We used the attached github Repo[2] to parse data for every stock in the Russell 3000 Index. 10K filing reports are present quarterly.
2. **Reddit PMAW:** Each of the 28 subreddits posts have been scraped for the past 1 month as a weekly job.
3. **Google Cloud Bucket and MongoDB Atlas:** Data is uploaded to Google Cloud Bucket and then saved in MongoDB atlas with 10k filings as a collection and the reddit posts data as another collection.
4. **Databricks Spark Cluster:** Once data is in mongodb atlas, we fetch the data in a spark dataframe for running our Huggingface Pytorch model(FinBert).

Preprocessing goals:

1. **Edgar API:** The 10K report consists of several sub documents which are split into items such as Item 1(Business), Item 1A(Risk Factors) and so on. More can be found in this document.[3] In this sub document, item 7(Management Discussion of Financial Statements) contains the information that we wish to calculate our sentiment scores on. Beautiful soup and lxml libraries are used to parse this information. In this collection, every record is a unique identifier.
2. **Reddit PMAW:** Reddit PMAW is a relatively straightforward API. We chose Reddit PMAW over Praw which is the official Reddit scraping API because of rate limit issues. In this API, we have to mention the time frame as well as subreddit posts. In the collection, the id is the unique identifier of each post in the collection.
3. **Sentiment Score preprocessing:** Sentiment scores are generated for every sentence in the Form 10k/Reddit posts data. The FinBERT model we use generates positive/negative/neutral sentiment scores. In a collection of sentences, a wide majority of sentences have a high neutral score. The sentences filtered(filter on sentences with a neutral score < 0.5) will be few and we need to concentrate only on those sentences which have a high positivity/negativity score.

[2] https://github.com/nlpaueb/edgar-crawler
[3] https://www.sec.gov/files/reada10k.pdf

Algorithms:

FinBERT Model: This model is available on HuggingFace and is downloaded on Databricks. The Finbert Model is a BERT model which is trained specifically on financial analysis data.[4] We split the text of every company/reddit post data into a collection of paragraphs and send it in a batch to the model. The model generates a NX1 sentiment tensor for each company/reddit post, where N represents the number of paragraphs/chunks.

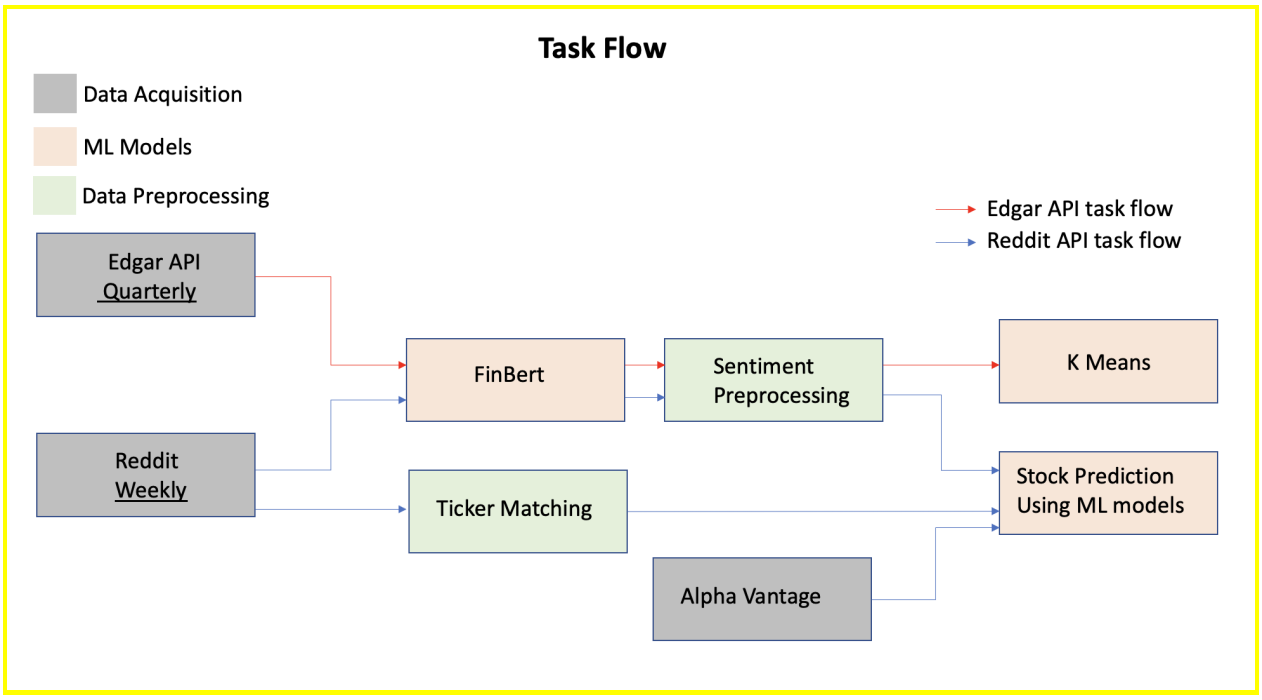
Sentiment Preprocessing: The collection of tensors are filtered(<0.5 neutrality score) and an average of the positive, negative and neutral sentiments is taken to form a composite sentiment score for every post/ticker.

Ticker Mention for Reddit Posts: Now, that we have Reddit posts, we need to find mention of stocks in these Reddit. We use Regex to find mentions of these tickers. However, in this algorithm, there are a number of false positives which occur. E.g. VISA has a ticker i.e. V. Searching for V in the Reddit posts can result in a number of posts being flagged as positive.

K means clustering of sentiment scores: A sentiment scores clustering is performed on stocks using Spark MLLib. This will help us identify companies with similar sentiments.

Stock Price Prediction: An attempt was made to predict stock prices for S&P 500 companies using sentiment scores as well as financial indicators. Linear Regression and Random Forest Models were chosen to make these predictions using Spark MLLib.

Graph Networks: Graph Networks were explored to improve stock price predictions, however the work is still ongoing and will be continued in the future.



[4] https://huggingface.co/ProsusAI/finbert

Time Efficiency

- Number of CPU cores: 4
- Number of GPU cores: 1
- PyTorch version: 1.13.1+cu117
- Databricks runtime : 12.1 ML (includes Apache Spark 3.3.1, GPU, Scala 2.12)
- Worker/Driver type - g4dn.xlarge, 16GB memory, 1 GPU
- Number of workers - 5
- A 15X increase in efficiency was achieved due to a more powerful cluster on Databricks/ better GPU/ parallel processing.
- For example, to predict the sentiment scores for 1970 10K Financial documents:
    - Local Machine[5]: 360 minutes
    - Databricks: 4.4 minutes
- To predict the sentiment scores for 13855 Reddit posts:
    - Local Machine[6]: ~170 minutes
    - Databricks: 11.16 minutes

ML Goals

1. Performing Sentiment Analysis using Pre-trained Models (Hugging Face FinBERT)
2. Use sentiment analysis to predict Stock Prices
3. Build Knowledge Graph for US Publicly listed firms

Outcomes

1. Able to generate sentiment scores for over 1960 companies out of an initial target of 3000 companies. Only 1971 companies were being scraped from Edgar API which suggests that:
    a. Companies being delisted
    b. Change in ticker information
    c. Inability to generate sentiment scores due to item 7 data being too large and resulting in GPU OOM issue.
2. Generating a K means clustering on sentiment scores so that similar companies can be grouped together. For expanding this work, a fundamental analysis can be done on the stocks with high sentiment scores to generate possible trading strategies in future iterations.

An example of stocks in the various clusters are as follows:-

	company prediction	cluster
0	STANLEY BLACK & DECKER, INC.	0
1	TRUSTMARK CORP	0
2	OMNICOM GROUP INC.	0
3	Hercules Capital, Inc.	0
4	COMCAST CORP	0

company prediction	cluster
GRANITE CONSTRUCTION INC	1
MidWestOne Financial Group, Inc.	1
Gevo, Inc.	1
Commercial Vehicle Group, Inc.	1
SANMINA CORP	1

company prediction	cluster
BERKLEY W R CORP	2
AUTONATION, INC.	2
FASTENAL CO	2
AVISTA CORP	2
DENTSPLY SIRONA Inc.	2

[5] Spec: M1 Pro, 16GB RAM, allocating 15G RAM to JVC
[6] Spec: M1 Pro, 16GB RAM, allocating 15G RAM to JVC

3. Baseline model to predict stock prices was a linear regression model with an RMSE of 2.717 on the test dataset, while the RMSE of a better model (Random Forest Regressor) was 2.38 on the test dataset.

Learnings

1. **Technical Learning Outcomes**: Learned how to create an airflow dag which would fetch data from APIs on a weekly/configurable basis. Storing data directly in Google Cloud and MongoDB atlas gave us the cloud computing skills which are platform agnostic(S3 for AWS/Azure Blob should have a similar process). Fetching data from MongoDB to a Spark Dataframe and using Spark MLLib to perform K means clustering involves the distributed computing aspect as well.
2. **Machine Learning/Data Science Learning Outcomes:** Learned how to use a HuggingFace model and test it on new data. Made aware of OOM memory issues on Pytorch issues and ways to resolve them(changing parameters like max_split_size/emptying torch cache/garbage collection on Python).

Conclusions

Our final RandomForest Regressor model was able to deliver an RMSE of 2.38 on the test dataset.

Future

1. Work to use Knowledge Graphs and Graph Data Science to improve stock market predictions is ongoing and will be carried out in the future. It presents an opportunity to significantly improve predictions.
2. Sentiment Scores can be augmented using News Data and Twitter Streams and will be explored in future.
3. Future work will also potentially explore newer data sources as well.