MSDS 697 - Distributed Data Systems

Group 14 - Ankit Gupta, Deepak Singh, Ity Soni

Predicting Stock Market behavior through various indicators

Motivation Statement

In this Report, we discuss several ways to predict stock-price movement that a stock market investor/trader may utilize to form their logical buy/sell decisions. Currently, there are various analyst reports available in the market that talk about which stocks to buy, based on their own analyses. However, it is hard to find a reliable and explainable source that could help an investor in making informed decisions. Hence the origin of this project.

Dataset and analytic goals

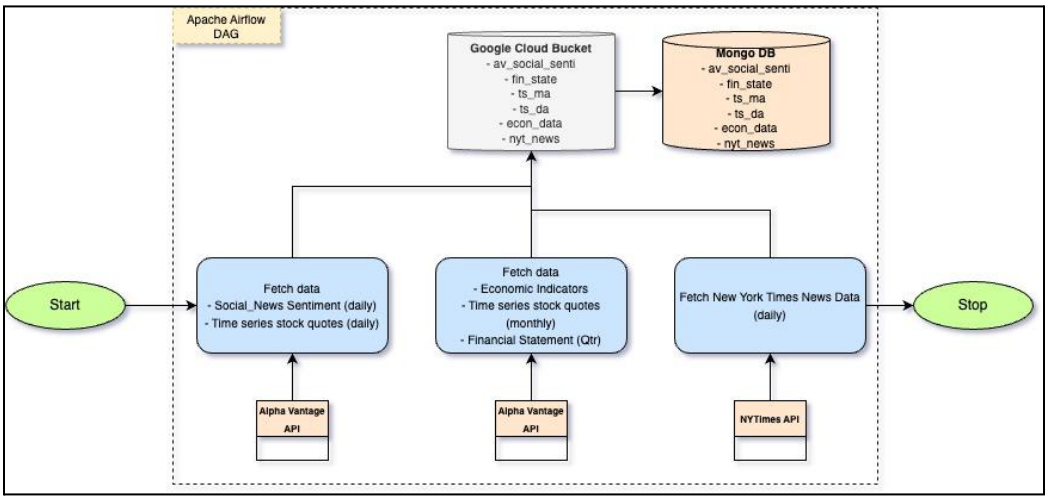
The dataset consists of the following different components:

- Econ_data: data for different economic indicators such as 'CPI', 'Brent price', 'federal rates' etc. spanning for over 20 years since 1997
- Fin_state: Data on quarterly and annual Financial statements, i.e. balance sheet, income statement and cash flow statement of various stocks under observation for past 5 years
- NY Times: News articles dataset for past 30 years
- Social media sentiment: We've utilized AlphaVantage API to retrieve this social media news and sentiment data from 50 different sources across the USA
- Ts_da, ts_ma: Time series daily/monthly adjusted close data for various stocks

Goals: Predict Stock price using various indicator features (mentioned separately below)

Overview of data engineering pipeline

We built our Data pipeline to retrieve various types of data as mentioned in the section below:



For the project, we tried to build a common pipeline using APache Airflow. We initially fetched data from alpha vantage social news API and aggregated it with the SEC filings per CIK (Central Index Key is a unique identifier used by the SEC to identify filing from the companies). Then fetched the form 10k data sections 1, 1a, 7, 7a etc. which were extracts from the annual financial reports of various companies on the SEC. However, this didn't work out as these datasets had different frequencies. We modified our pipeline to fetch these datasets individually to accommodate for the different update frequencies. Finally, all the datasets are being updated on the MongoDB cluster.

Preprocessing goals, algorithms, and Cluster Configuration

Deepak Singh

a) Preprocessing Goals: Fetch monthly Economic trends data, and Stock Quote trends data, and preprocessing to obtain the percent change in stock price and other indicators from previous close
b) Algorithms used: One Hot Encoding, String Indexer, Vector Assembler
c) Runtime: 41 + 30 + 0.5 = ~70 s

Ity Soni

a) Preprocessing Goals: Fetch daily news data from ny_times API, and Stock Quotes time series daily adjusted values data, and Preprocessing to get the percent change in stock price and news sentiment from one of the leading news services
b) Algorithms used: **BertForSequenceClassification, BertTokenizer** (UDF get_sentiment function) and convert function
c) Runtime: 570 + 330 = 900 s

Ankit Gupta

a) Preprocessing Goals: Fetch Stock Quotes time series daily adjusted values data, and Preprocessing to add the relevant features for further processing using ARIMA
b) Algorithms used: dataframe operations to transform the data
c) Runtime: ~20 s

Cluster Specifications:

MongoDB: Version: 5.0.15	**DataBricks**: Group_Project_14_GPU
Region: GCP / Oregon (us-west1) **Cluster Tier**: M30 (General) **Type**: Sharded Cluster - 2 shards **Backups**: Active **Linked App Services**: None	**Policy**: Shared compute **Runtime**: 12.1 ML (includes Apache Spark 3.3.1, GPU, Scala 2.12) **Worker**: 16 GB, 1 GPU, min workers = 2, max workers = 5 **Driver profile**: 16 GB, 1 GPU **Spark Config**: spark.jars.packages org.mongodb.spark:mongo-spark-connector_2.12:3.0.1

Bi Connector: Disabled **Online Archive**: None **Atlas Search**: Create Index	**JODBC**: usfca-2023-msds697.cloud.databricks.com **Libraries**: - Org.mongodb.spark:mongo-spark-connector_2.12:3.0.2 - Transformer - pystan==2.19.1.1 - fbprophet==0.7.1

ML Goals:

Deepak Singh

a) Predict the monthly buy vs sell decision target variable with respect to the available economic indicators

- Algorithm: Random Forest with Cross Validation and a final RF Model
- Runtime: ~120 s for Random Forest, ~1080 s for RandomForest with CV
- Test metric AUC: 66.06%

Ity Soni

b) Predict the daily buy/sell target variable with respect to the social news sentiment available for the stocks under observation

- Algorithm: Logistic Regression
- Runtime: 720 s to fit and predict using Logistic Regression
- Test Metric: Area Under ROC: 0.52

Ankit gupta

c) Estimate the predicted price using Time series forecasting models

- Algorithm: AutoARIMA and FBProphet Time Series Models
- Runtime:
    - AutoARIMA: 180 s
    - FBProphet: 28 s
- Test Metric:
    - AutoARIMA: MAE: 2.624
    - FBProphet: MAE: 19.9

Lessons Learned

- There was a lot of feature engineering involved in our case, where we dealt with converting categorical, and numeric data, converting to categorical feature vectors. We've also added a lot of features which are driven from existing features such as percentage change in price with respect to previous period close
- Simple models perform better than complex ones in terms of time series prediction

Conclusion

- We were able to generate machine learning models that could predict the effects of social news sentiment, economic indicators and previous time series trends. This could be utilized by potential investors in making buy/sell decisions for any upcoming trades
- Next Steps: It would be great to see the outcomes of a model that combines the effects of all the above features. Such a comprehensive model can be leveraged to provide users with a n interface where the investors can enter a stock's ID and get the relevant buy/sell decisions from various sources such as social news, economic indicators, SEC filings and past time series.

Reference:

- https://github.com/dianewoodbridge/2023-msds697-distributed-data-systems
- https://usfca.instructure.com/courses/1612247

- APIs and data sources:
  - Alpha Vantage news sentiment API: https://www.alphavantage.co/documentation/#news-sentiment
  - Alpha Vantage Time SEries data: https://www.alphavantage.co/documentation/#dailyadj
  - NY Times API: https://developer.nytimes.com/apis
  - Economic Indicators API: https://www.alphavantage.co/documentation/#economic-indicators