Project Executive Summary

Project Name: Stock Behavior Prediction
Members: Xiaokang Wang, Xiangwei Peng
Github: https://github.com/jamesdrsteele/ErdosNewsFinanceProject

1. Introduction

Problem

Stock behavior forecasting is crucial in the finance industry. It can offer a range of practical benefits over several domains, such as ingesting investment opportunities and guiding risk exposure management. However, the stock market is influenced by many factors, such as economic status, policy, natural disasters, cash fluency, etc. Some data are structured, and some are unstructured. We aim to use machine learning (ML) and natural language processing (NLP) techniques to forecast stock behaviors.

KPIs

We focus on two stock behaviors: daily stock returns and movement directions. Stock returns reflect the absolute amount of price change compared to yesterday, and the stock movement is focused on the price-changing directions, up and down). We use separate metrics for evaluating the model performance as follows:

1. Stock Return Prediction: Mean Square Error (MSE).
2. Stock Movement Prediction: Accuracy.

Data

We use historical data from several sources from 2016 to 2019. Specifically, data from 2016 to 2018 are used to train several models and test the performance of the 2019 data.

1. Stock Data of 20 Major Companies ingested from Yahoo Finance API.
2. Market Data (Fama French Five Factors) is ingested from Dr. Kenneth R. Frech's website.
3. All News Data is downloaded from the website. It contains ~2 million news and articles from 27 major publications, covering the period from January 1, 2016, to April 2, 2020.
4. Headline News data is downloaded from the Kaggle website.

Key Assumptions

Finance and News data are fairly complex. To make them easier to model, we hold some key assumptions as follows:

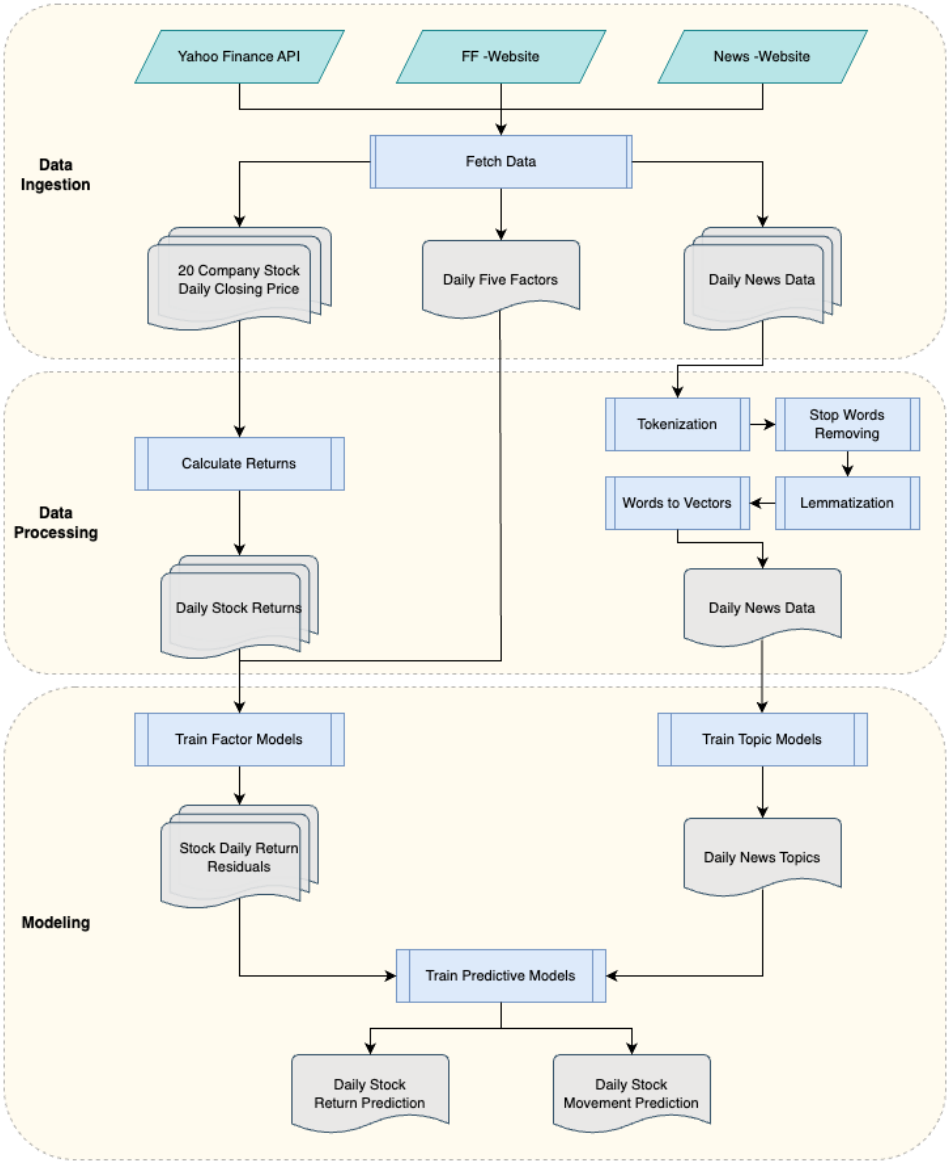
1. **Incomplete Information**. The market information captures the main pattern of stock behaviors, but it still needs to be completed. News data can reflect unstructured details and can be used as a supplement.

2. **Short-term dependency**. The most recent 1~3 days' price will impact current stock price behavior. No significant long-term dependency exists.
3. **Homogeneous pattern**. The primary impact of stock behavior is from the market and news information, and the pattern is fixed for periods.
4. **Stock Heterogeneity**. The patterns are different among different stocks.

Solution

We design the whole procedure as data ingestion, processing, and modeling

1. For ingestion, we download data from websites and pull data from Yahoo Finance API;
2. For processing, we convert the stock price data into returns and use NLP techniques to transform unstructured news data into a structured table.
3. For modeling, (1) we first build the factor model with market factors, and return residuals for further modeling, (2) then extract news topics by unsupervised ML algorithms, and (3) finally, we train the predictive models on news topics to predict the residuals from (1).



2. Implementation

Data Preprocess

We ingest the daily close price for each stock and then calculate the changing proportion as daily returns.

For the news data, we first run the standard preprocessing pipeline: tokenization, stop words removal, and lemmatization to clean text data into tokens. The cover text data into vectors. Specifically, for the category classification, we choose the term frequency-inverse document frequency (TF-IDF) embedding, and for the topic clustering, we select the bag-of-words (BoW).

Model Structure Design

We aim to use market and news information to predict future prices Y, such as stock returns and movement directions. To do this, we need both market factors and news topcis.

$$Y = g(Market\ Factors) + \epsilon_R \qquad \text{(Factor Model)}$$

Firstly, the Famma-French 5-factor (reduced) model is used to model the relationship between market trends and stock price change. This will be treated as the primary effect. This abnormal return part (residuals ϵ_R) is believed to be influenced by the news.

To extract news topics, we run a topic model h on news vectors from several past days.

$$News\ Topcis = g(News\ Vectors) \qquad \text{(Topic Model)}$$

Considering computing resource limitations, we also split the All News data into small portions by categories and built the topic model for each category. However, categorization information is lacking in All News data. To achieve this, we first train a supervised learning model to classify the news based on its headline. Then, we annotate news categories by their headlines.

We use unsupervised learning on the specific category dataset to cluster the news into topics. Then, we convert the topics into occurrence frequencies and use the most related and significant topic frequency to build a regression model for the residual.

Finally, we will use the news topics to model the residuals (“abnormal returns”) of the factor model to build the predictive model f.

$$Y = g(Market\ Factors) + f(News\ Topics)) + \epsilon_F \qquad \text{(Predictive Model)}$$

The whole model can be written as,

$$Y = g(Market\ Factors) + f(h(News\ Vectors)) + \epsilon_F.$$

Modeling Approach

1. Factor Model:

The Famma-French 5-factor model is a linear model that considers multiple systematic risk factors to predict the market's influence on individual stocks.

2. Topic Model:
   The category classification model is a weighted voting model consisting of the following:
   - Multi-logistic Classifier
   - Random Forest
   - XGBoost Classifier
   - CNN

   In the topic clustering model, we compare the most likely to be explainable algorithms:
   - Latent Dirichlet Allocation (LDA)
   - Hierarchical Dirichlet Process (HDP)

   We chose the HDP method because it has the advantage of model specification. With LDA, being too specific about the potential number of topics is risky. In practice, exploring a different number of topics with LDA and comparing results by coherence score is time-consuming.

3. Prediction Model:
   The final predicting model we compare:
   - Lasso Regression
   - Ridge Regression
   - Random Forest Regressor
   - XGBoost Regressor

   We chose the XGBoost Regressor because it fits the training set the best and has the lowest mean square error on the test set. The result contains more exceptional values, which can be used as a trading signal.

3. Results

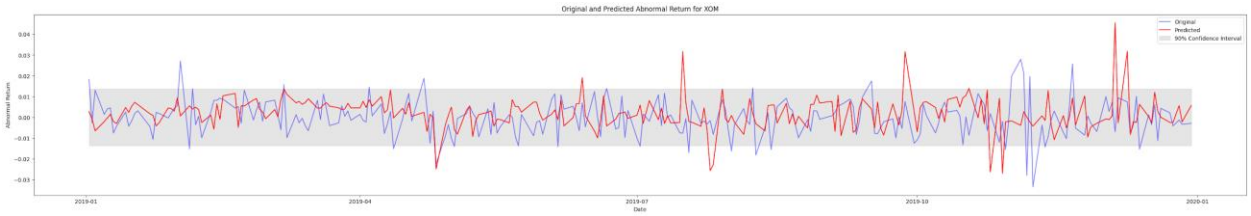
Return Prediction Model

We choose our stock to be Mobil Corp (XOM). Set two baseline models: one is the price of the test set as the first day's price, and the other is setting tomorrow’s price as today’s price. For this test set, the mean square error of the baseline model one is 1.976, and the baseline model two is 2.096. For our model, we have a mean square error of 1.241. This suggests that our model outbeat the two naive baseline models. Our predictive model improved the factor model with a mean square error of 1.243.

Model	Baseline1	Baseline2	Factor model	Predictive model
MSE	1.976	2.096	1.243	1.241

Stock Movement Prediction Model

When we convert the prediction into binary data, we predict whether tomorrow's price will go up or down by the sign of our prediction residual. The accuracy is 68.9%, which suggests that our model is better than the naive guess.



4. Discussion & Future Work

Data Limitation

Our dataset is just an assembly of news on all topics from many publishers. It contains too much news (~2 million) and too much noise. So, the topic clustering does not work well as expected, and our topic interpretation is poor. In the future, we can use the news API to get news directly related to a certain company and then run HDP on this much smaller dataset.

Sentiment Analysis

Our choice of topic feature is the frequency of certain topics. We treat the contribution of each piece of news evenly and ignore the actual content of the news. This oversimplified assumption should have included more crucial information. One may consider using sentiment analysis to convert news into a score. However, during the project, we tried different methods of sentiment analysis tools, like Vader and pre-trained BERT models from HuggingFace. All of these existing tools could not have performed better. One problem is that news contains only a little sentiment information. The words it uses are non-emotional, which makes all the word-based sentiment analysis tools perform poorly. One direction for the future is to fine-tune a BERT model so that it can detect the subtle sentiment of the news.

Topic Definition (LLMs for more generalized understanding interpretation)

Our topic definition is based on the unsupervised learning methods. It is hard to interpret the exact meaning of the topics. In the future, we can use the Large Language Model and run some prompts so that it can express the bag-of-words topics in natural language. Then, we can get more financial insight from the topics and the effect of the stock price.

Dynamic Assumption

Another oversimplified assumption is that we assume all the topics are time-homogenous, which means that the topics occurring before will occur in the future. They are distributed evenly on the time axis. However, this is not the case in reality; some news topics, such as the Paris

Olympics, will rarely occur again. So, the topics are time-sensitive. In the future, we will try a way to update the topics after a certain period, add new topics, and delete the old topics. With a certain amount of news in the database, we train a dynamic price-predicting model based on dynamic topics.

Online Deployment

The whole current pipeline and experiments are run in batch data. However, if we focus on the continuing performance of the forecasting performance, we should consider adding a mechanism to update the model corresponding to market changes. Meanwhile, the pipeline should be extended to be compilable to the stream data flow, which is easier to use in production environments.

Trading Strategy

A well-defined trading strategy can amplify the value of prediction power. In the future, we can also explore specific trading strategies to leverage the performance of stock behavior forecasting.