This project attempts to predict the performance of SPY ETF using financial news headlines on Reuters webpage on the previous trading day.

Data

News headlines between October 2018 and September 2020 are scraped from the Reuters financial news homepage[1]. Past versions of the webpage are retrieved from the Internet Archive[2]. The website snapshot closet to noon each day is used. Due to the limited scale of the project, data before October 2018 are not used because there are differences in the format of the HTML files.

The website is separated into 20 parts (figure 1). The text in each part is extracted and save into CSV files. Data validation is done to ensure the timestamp of the webpage is within the required date. Note that some fields may be empty since Reuters may have less than 6 stories for some days.

SPY ETF is chosen because it tracks the S&P500 index and can act as a proxy of the US stock market. The adjusted closing price and trade volume is retrieved from Yahoo Finance[3].

Returns for each trading day is calculated. Two Boolean variables is also created: whether the price moves up, and whether the trade volume is greater than USD 75,000,000 (median of recent 2 years).

Analysis

Sentiment analysis is performed on the text data using TextBlob and NLTK Vader. A polarity score, showing if the statement is positive or negative, is generated for each of the 20 parts of the website.

A random forest model is fitted to predict the direction of the stock price and the relative size of the daily trade volume. Random forest model is chosen because it is non-parametric and it can automatically perform feature selection. Due to the limited scale of the project, other models are not considered. Cross validation is performed to tune the complexity of the model to avoid overfitting.

The data is split into a training and a test set, which contains 17% and 83% of the data respectively. ETF price increases in 65.1% of the test set, and trading volume is greater than USD 75M in 44.2% of the test set.

6 models are fitted in an attempt to find predictive power as shown below.

Model	Features	Response Variable	Test Accuracy
Model 1	TextBlob polarity and subjectivity	ETF price direction	56.98%
Model 2	NLTK polarity	ETF price direction	62.79%
Model 3	NLTK polarity for news contents only	ETF price direction	46.51%
Model 4	TextBlob polarity and subjectivity	Size of trade volume	54.65%
Model 5	NLTK polarity	Size of trade volume	53.49%
Model 6	NLTK polarity for news contents only	Size of trade volume	47.67%

Results

All models do not show significant predictive power on both price direction and volume of trade as their accuracy is lower than a null model (predicting using the most occurring result).

Interestingly, for model 2, the content of the 3rd story has much more predictive power than other features (figure 2). This may be caused by pure luck; however, it can be a starting point for further analysis.

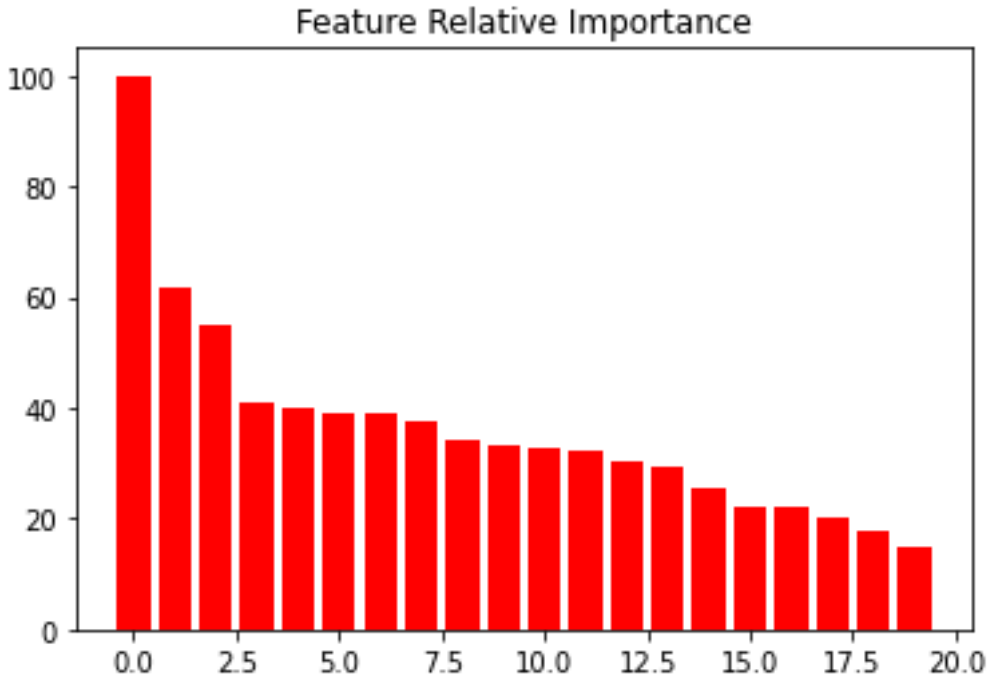
Due to the limited scale, this project simplifies multiple processes. More in-depth analysis can be done by extracting more data, generate more features from the text data, and use a more iterative approach for model building.

[1] https://www.reuters.com/finance
[2] https://archive.org
[3] https://finance.yahoo.com/quote/SPY/history/

Appendix

Figure 1: Reuters financial news homepage with annotations

Figure 2: Model 2 variable importance chart and table



Features	Relative Importance	Importance
stories_content_3	100%	13.74%
stories_title_0	62%	8.46%
video_title_1	55%	7.56%
stories_title_1	41%	5.65%
stories_content_4	40%	5.49%
stories_title_4	39%	5.38%
stories_content_5	39%	5.35%
feature_title_2	37%	5.15%
video_title_2	34%	4.72%
main_title	33%	4.57%
main_content	33%	4.47%
stories_content_0	32%	4.45%
feature_title_1	30%	4.16%
stories_content_1	29%	4.02%
stories_content_2	26%	3.53%
stories_title_2	22%	3.05%
feature_title_0	22%	3.02%
video_title_0	20%	2.76%
stories_title_3	18%	2.43%
stories_title_5	15%	2.02%

Attachment

Filename	Description
Step_1_scrape_web.py	Scrape text data from web and save into CSV files
Step_2_analyse_data.py	Data pre-processing and fitting random forest models
web_data/web_data_yyyymm.csv	Text data from Reuters financial news homepage
SPY.csv	Historic ETF prices from Yahoo Finance