Benzinga Headline Semantics vs Stock Price Variance

Hdpb88

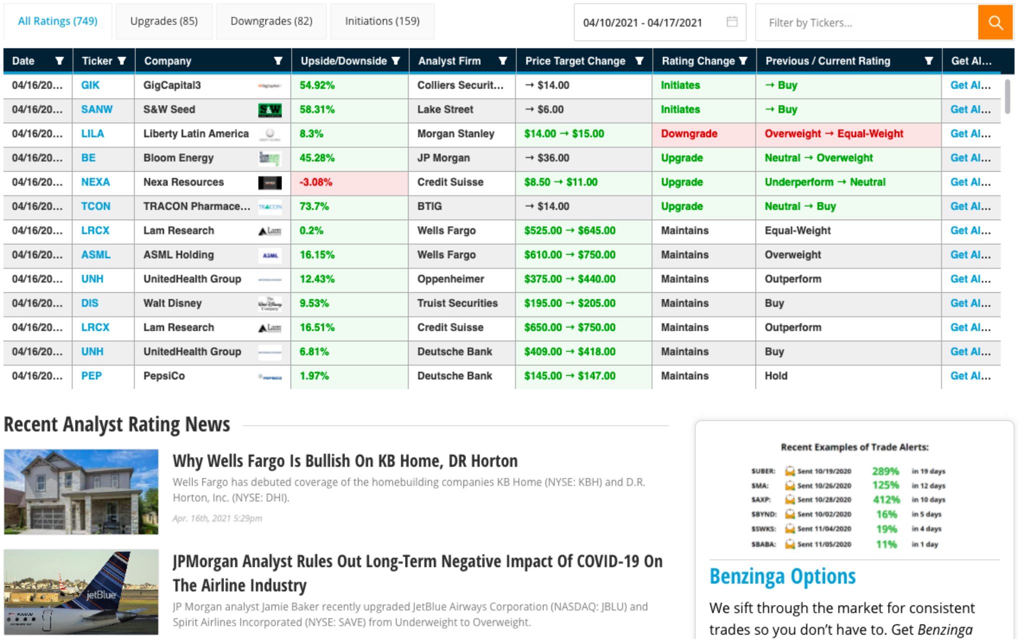
April 2021

1 Introduction

With recent years seeing a record number of investors in the market [5] a movement boosted by the ongoing pandemic providing employees working from home the added time and motivation to start individual trading [6]. With this increase in the number of investors, there is a consequent increase in the value of reliable media and news relating to the future performance of individual stocks. One such source of stock reporting media is Benzinga [2]: a media ecosystem with the mission to 'connect the world with news, data and education that makes the path to financial prosperity easier for everyone'.

One type of media delivered by Benzinga is Analyst Ratings, in which individual stocks are rated for future performance. After being rated by a human analyst, an article is written containing their findings and predictions before being published on the Benzinga website under a related headline [2]. There are 749 different analyst ratings available on just the homepage of Benzinga and with each of these containing long explanations and technically complex analysis, the simplistic, easily read headlines present what is supposedly a quick and accurate representation of the contents of an otherwise time-consuming and technical article. It is therefore understandable that these headlines could have a powerful impact on investor behaviours. An example might be an investor briefly scanning the page, seeing "stock XYZ is due to perform this summer" and hence be influenced to purchase XYZ.

With the potential influence of their headlines, analysts should strive to present their analysis alongside a headline relevant to not only the overall conclusion of the article but *ideally* also the stock's future performance. This paper presents a computational model representing a comparison between the sentiment expressed by Benzinga in their analysts' headlines and the real-world performance of the stocks discussed by those headlines. Such a model is constructed with the aim of evaluating the reliability of Benzinga headlines with regards to stock performance as well as the creation of a tool capable of analysing a headline and predicting the likelihood of that headline's sentiment being accurate.



Date	Ticker	Company	Upside/Downside	Analyst Firm	Price Target Change	Rating Change	Previous / Current Rating	Get Al...
04/16/20...	GIK	GigCapital3	54.92%	Colliers Securit...	→ $14.00	Initiates	→ Buy	Get Al...
04/16/20...	SANW	S&W Seed	58.31%	Lake Street	→ $6.00	Initiates	→ Buy	Get Al...
04/16/20...	LILA	Liberty Latin America	8.3%	Morgan Stanley	$14.00 → $15.00	Downgrade	Overweight → Equal-Weight	Get Al...
04/16/20...	BE	Bloom Energy	45.28%	JP Morgan	→ $36.00	Upgrade	Neutral → Overweight	Get Al...
04/16/20...	NEXA	Nexa Resources	-3.08%	Credit Suisse	$8.50 → $11.00	Upgrade	Underperform → Neutral	Get Al...
04/16/20...	TCON	TRACON Pharmace...	73.7%	BTIG	→ $14.00	Upgrade	Neutral → Buy	Get Al...
04/16/20...	LRCX	Lam Research	0.2%	Wells Fargo	$525.00 → $645.00	Maintains	Equal-Weight	Get Al...
04/16/20...	ASML	ASML Holding	16.15%	Wells Fargo	$610.00 → $750.00	Maintains	Overweight	Get Al...
04/16/20...	UNH	UnitedHealth Group	12.43%	Oppenheimer	$375.00 → $440.00	Maintains	Outperform	Get Al...
04/16/20...	DIS	Walt Disney	9.53%	Truist Securities	$195.00 → $205.00	Maintains	Buy	Get Al...
04/16/20...	LRCX	Lam Research	16.51%	Credit Suisse	$650.00 → $750.00	Maintains	Outperform	Get Al...
04/16/20...	UNH	UnitedHealth Group	6.81%	Deutsche Bank	$409.00 → $418.00	Maintains	Buy	Get Al...
04/16/20...	PEP	PepsiCo	1.97%	Deutsche Bank	$145.00 → $147.00	Maintains	Hold	Get Al...

Recent Analyst Rating News

Why Wells Fargo Is Bullish On KB Home, DR Horton

Wells Fargo has debuted coverage of the homebuilding companies KB Home (NYSE: KBH) and D.R. Horton, Inc. (NYSE: DHI).

Apr. 16th, 2021 5:29pm

JPMorgan Analyst Rules Out Long-Term Negative Impact Of COVID-19 On The Airline Industry

JP Morgan analyst Jamie Baker recently upgraded JetBlue Airways Corporation (NASDAQ: JBLU) and Spirit Airlines Incorporated (NYSE: SAVE) from Underweight to Overweight.

Recent Examples of Trade Alerts:

$UBER:	Sent 10/19/2020	289%	in 19 days
$MA:	Sent 10/26/2020	125%	in 12 days
$AXP:	Sent 10/28/2020	412%	in 10 days
$BYND:	Sent 10/02/2020	16%	in 5 days
$SWKS:	Sent 11/04/2020	19%	in 4 days
$BABA:	Sent 11/05/2020	11%	in 1 day

Benzinga Options

We sift through the market for consistent trades so you don't have to. Get *Benzinga*

Figure 1: Benzinga Analyst Ratings Homepage

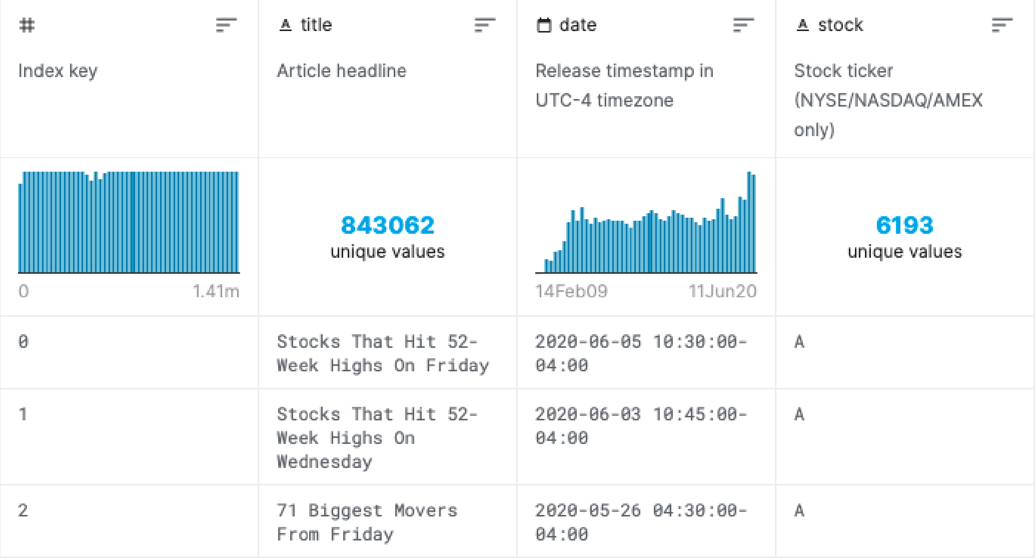
2 Data Sources

This project deals with two main sources of data used for comparison. The first is a collection of Benzinga analyst rating headlines [4]. Meanwhile, the second is a collection of historical stock price data obtained using a REST API [1].

2.1 Headline Data

In summer 2020, Bot_Developer uploaded a dataset of directly scraped analyst rating headlines from Benzinga. Despite containing 3 tables of data, the main one (and the one used in this project) contains 4 million headlines across 6000 stocks spanning 12 years. The main table:

analyst_ratings_processed.csv, contains the columns: title, date and stock. Referring to the headline, date published and stock in review, respectively.



#	title	date	stock
Index key	Article headline	Release timestamp in UTC-4 timezone	Stock ticker (NYSE/NASDAQ/AMEX only)
0 1.41m	843062 unique values	14Feb09 11Jun20	6193 unique values
0	Stocks That Hit 52-Week Highs On Friday	2020-06-05 10:30:00-04:00	A
1	Stocks That Hit 52-Week Highs On Wednesday	2020-06-03 10:45:00-04:00	A
2	71 Biggest Movers From Friday	2020-05-26 04:30:00-04:00	A

Figure 2: analyst_ratings_processed.csv

From initial experimentation in order to gain insight into the headline data, it became clear that utilising the entire dataset was simply not feasible. Examining the span of headlines across the available stocks revealed that while some stocks were extensively written about, others remained with only very few ratings, with a wide disparity in-between.

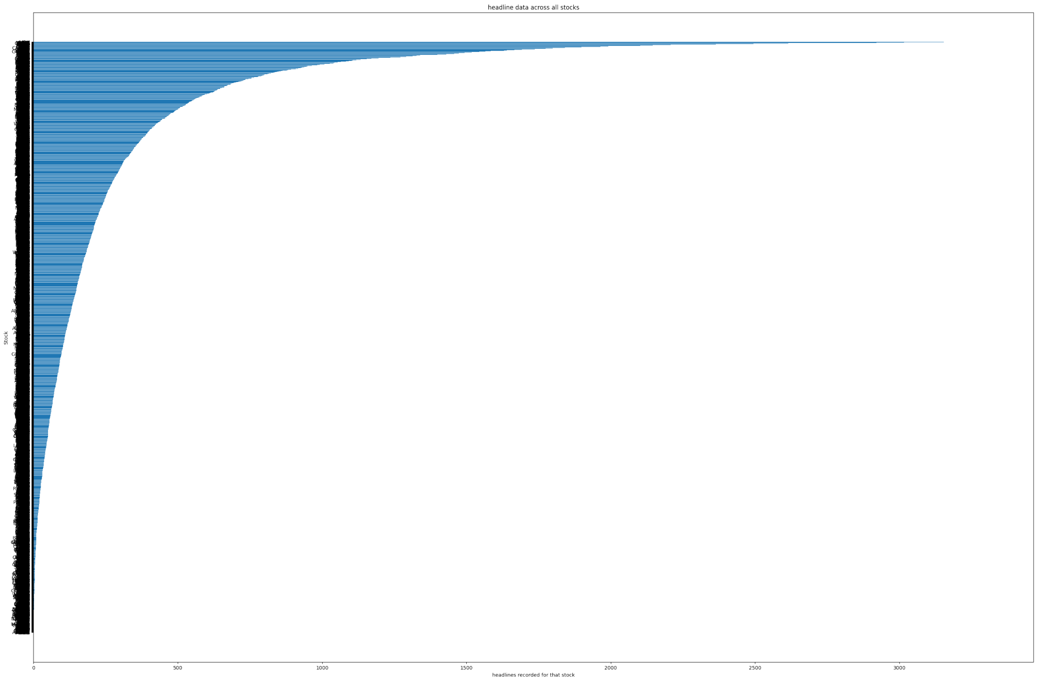


Figure 3: Number of headlines for all stocks

To capitalise on the enormous amount of headlines regarding just the first few stocks, it was decided that just the top 100 most written-about stocks in this dataset would be used in the project. This was to maximise the overlap between reliable stock price data and a strong number of headlines from which to gain sentiment readings.

After filtering down those stocks that were in the most popular 100, the spread of headlines per stock in the dataset is shown in Figure 4, a much more even spread despite still having over 175,000 headlines to use.

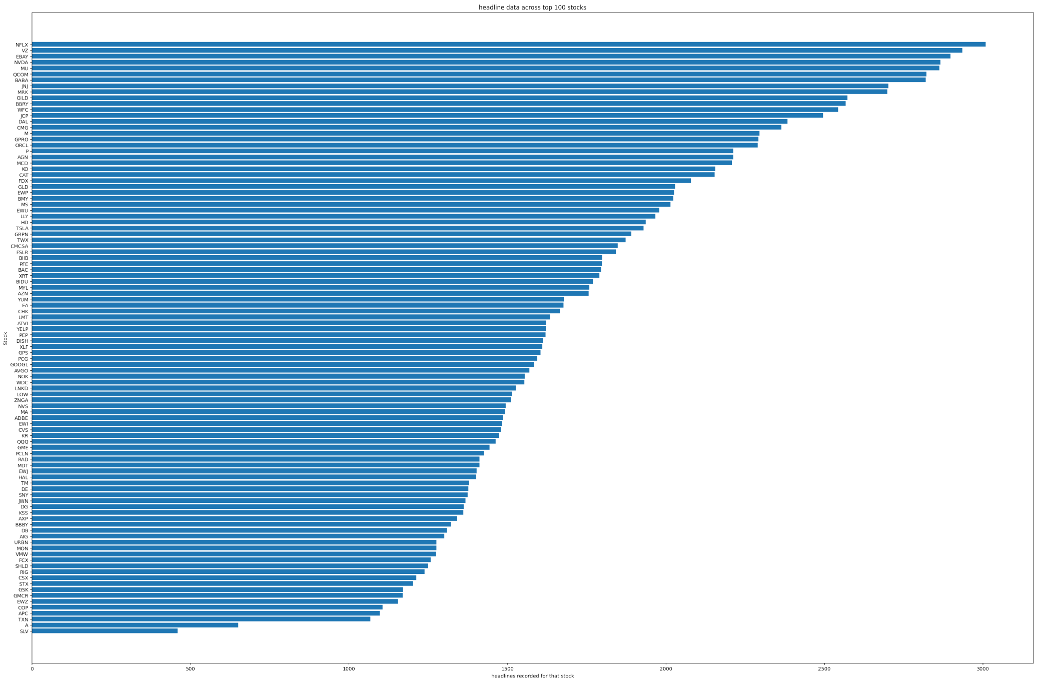


Figure 4: Number of headlines per stock for the top 100 stocks

2.2 API Data

The stock price data for this project was gathered from Marketstack [1]. After obtaining API keys, it was automatically retrieved from the API up to the maximum capacity permitted by the appropriate member tier. One concern addressed while preparing this data was the possibility of there being insufficient price data for the top 100 stocks identified earlier. After examining a near constant number of data-points across the top 100 stocks, it became clear that a merge between headline data and stock price data was a valid comparison.

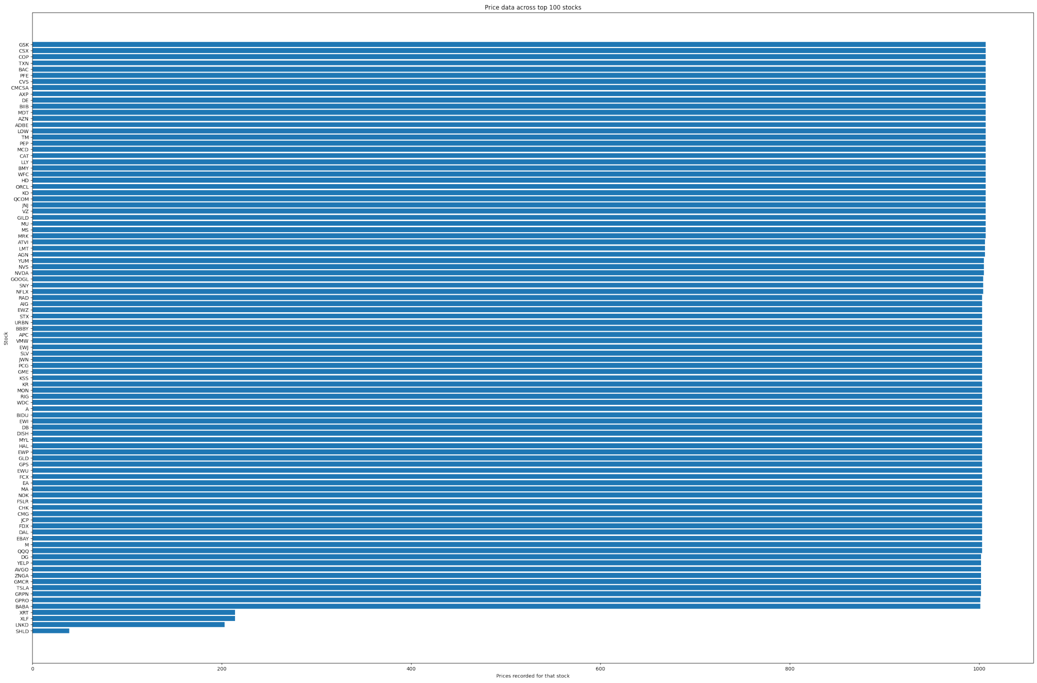
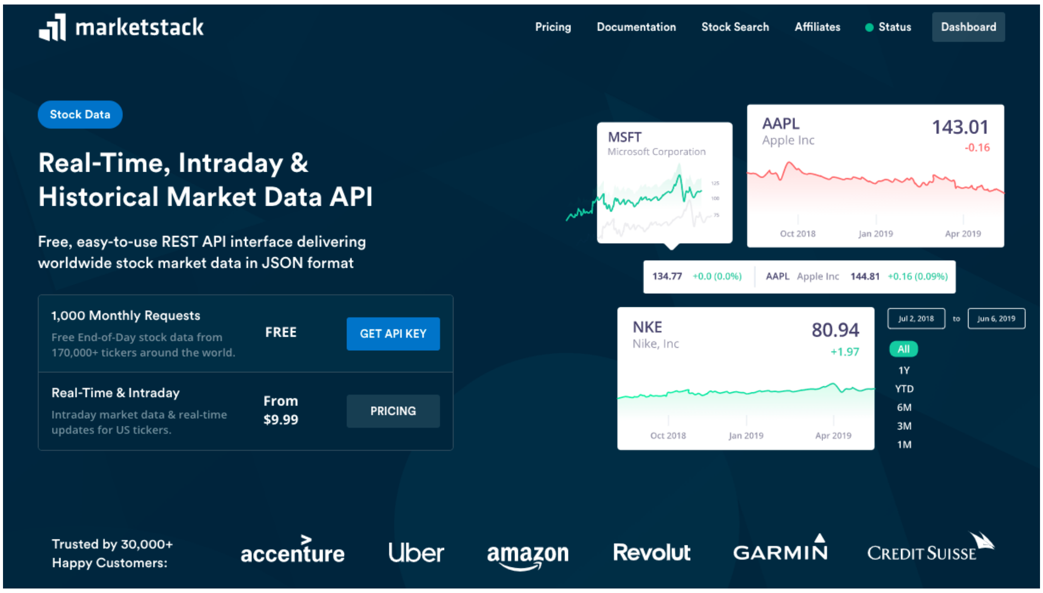


Figure 5: Price readings for each of the top 100 stocks

This data included multiple features, however those of direct interest to this model are: stock, closing price and date.



Real-Time, Intraday & Historical Market Data API

Free, easy-to-use REST API interface delivering worldwide stock market data in JSON format

1,000 Monthly Requests

Free End-of-Day stock data from 170,000+ tickers around the world.

Real-Time & Intraday

Intraday market data & real-time updates for US tickers.

Trusted by 30,000+ Happy Customers:

Figure 6: Marketstack REST API

2.3 Modelling Tools

For the semantics analysis, the Stanza tool was used. Stanza is a collection of natural language processing (NLP) tools provided by Stanford University [9]. Additionally, before inputting the headlines to the machine learning classifier, the Natural Language Toolkit (NLTK) [3] was utilised to stem words, as this has been found to improve classification of natural language features with machine learning models.

Following sentiment analysis with Stanza, Scikit Learn is used to fit then test a Random Forest Classifier (RFC). Scikit Learn is a collection of machine learning tools for the Python programming language and can be used to train a variety of models in classification, regression and clustering [8].

3 Implementation

3.1 Data Preparation

Before any steps could be made towards the creation of a model, the relevant data had to first be retrieved and analysed before finally being cleaned. Collecting the headline data was trivial in that it was pre-formatted in a csv format on the Kaggle website [4]. The stock pricing data required construction of an API request function as well as a system for applying this function iteratively in order to obtain all stored data about all 100 required stocks. Due to the number of requests required to the API, a business-level API key was purchased and once all the necessary requests were made, the price data was collated into a dataframe and saved into a local csv file.

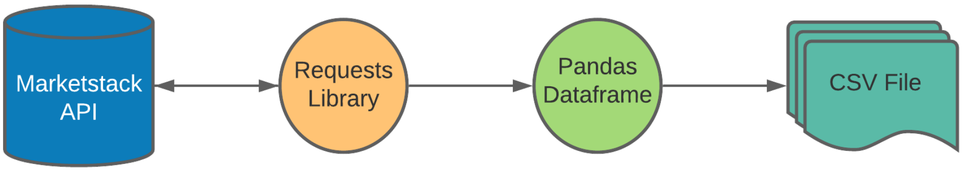


Figure 7: API data gathering function

Having gathered the relevant stock prices, the data cleaning and preparation began, the associated steps proceeded as follows:

3.1.1 Headline Data

1. Removed any undated headlines as they were irrelevant to the task.
2. Converted 'date' column from String to python.DateTime object.
3. Reduced headline data to contain only the top 100 stocks and covering only the time-span for which there is price data.
4. Added a column to the headline data and populated it with the result from Stanza's sentiment analysis.

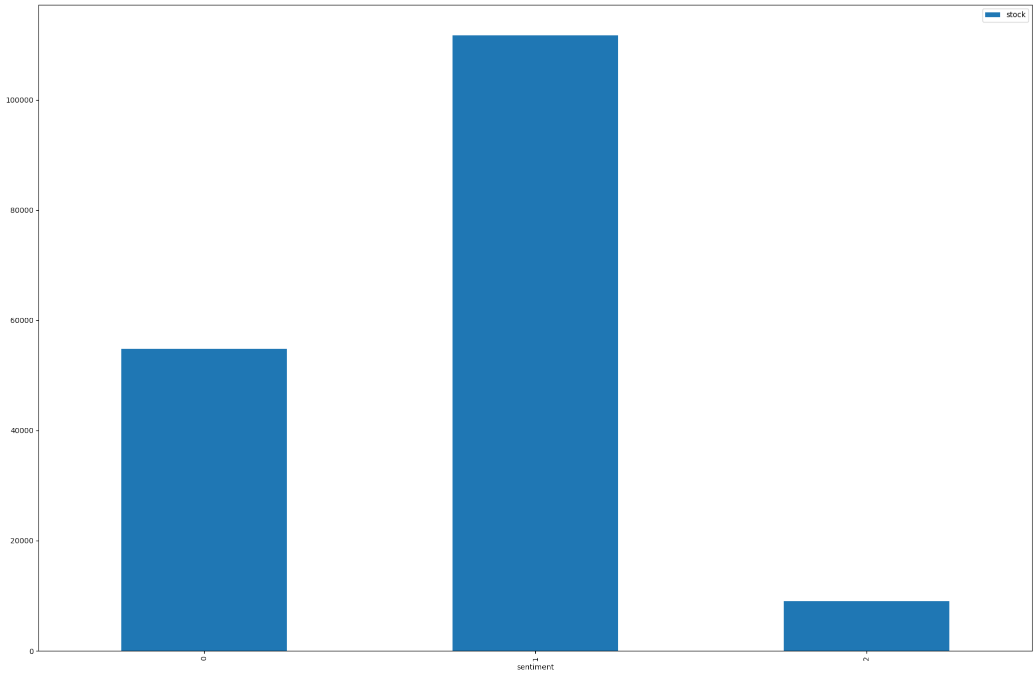


Figure 8: Number of headlines with each sentiment

3.1.2 Stock Price Data

1. The dataframe made from the API requests was checked for errors arising in the creation process such as duplicated rows.

2. Irrelevant features were removed.

3. The 'price_change' feature was added to the table, representing whether the closing price of the stock went up, down or stayed the same in the pre-determined time-window.

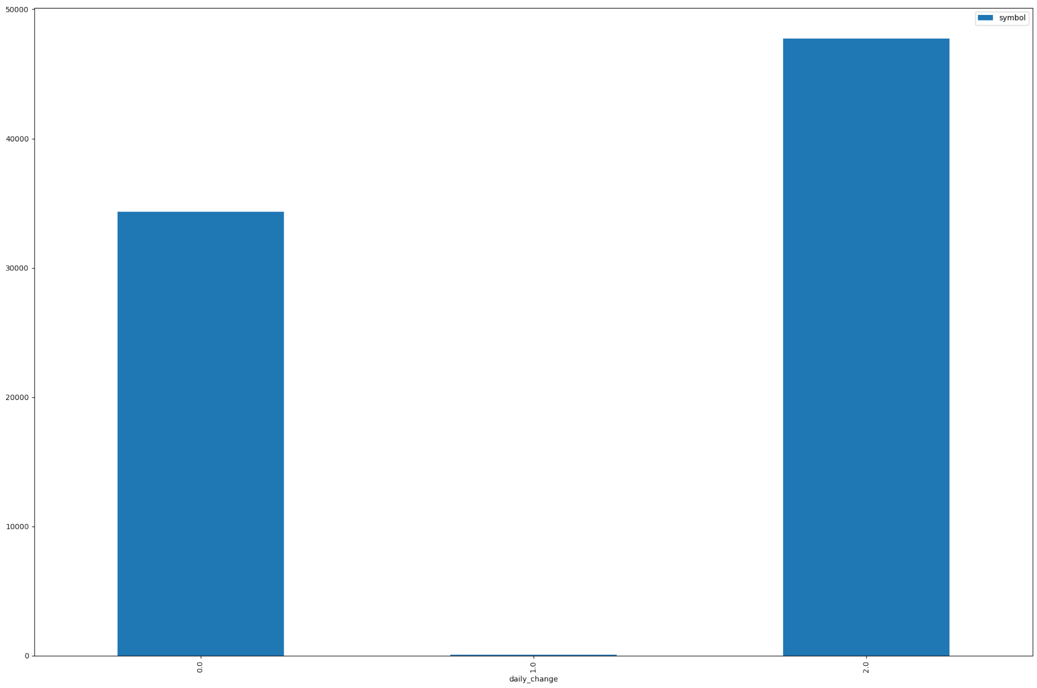
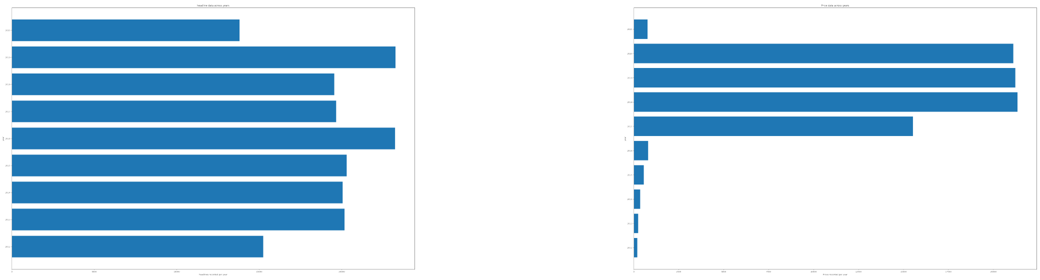


Figure 9: Number of stocks with each direction of price change

3.1.3 Data Merging

Following cleaning, the date spread of both tables was compared to check that the comparison between the two was valid.



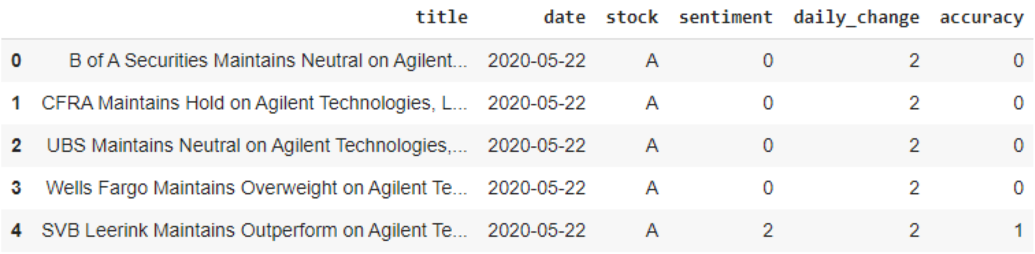
(a) Headline data per year

(b) Price data per year

Figure 10: Date spreads

This comparison showed great overlap in data availability for the years 2017-2020 however had little overlap in other years. While this lack of overlap in some years won't cause an error, it does bring to question the validity of the resulting model in years not between 2017 and 2020.

Merging the two pieces of data was performed with an inner join based on the stock and date features. The result of this join is that every headline relating to a stock is in a row containing that headline's sentiment and an indication as to whether that stock went up, down or remained the same price in 3 month's time. This table was then used to produce a feature 'accuracy', a simple binary flag representing whether the sentiment of the headline reflects the price change of the respective stock.



	title	date	stock	sentiment	daily_change	accuracy
0	B of A Securities Maintains Neutral on Agilent...	2020-05-22	A	0	2	0
1	CFRA Maintains Hold on Agilent Technologies, L...	2020-05-22	A	0	2	0
2	UBS Maintains Neutral on Agilent Technologies,...	2020-05-22	A	0	2	0
3	Wells Fargo Maintains Overweight on Agilent Te...	2020-05-22	A	0	2	0
4	SVB Leerink Maintains Outperform on Agilent Te...	2020-05-22	A	2	2	1

Figure 11: Accuracy table excerpt

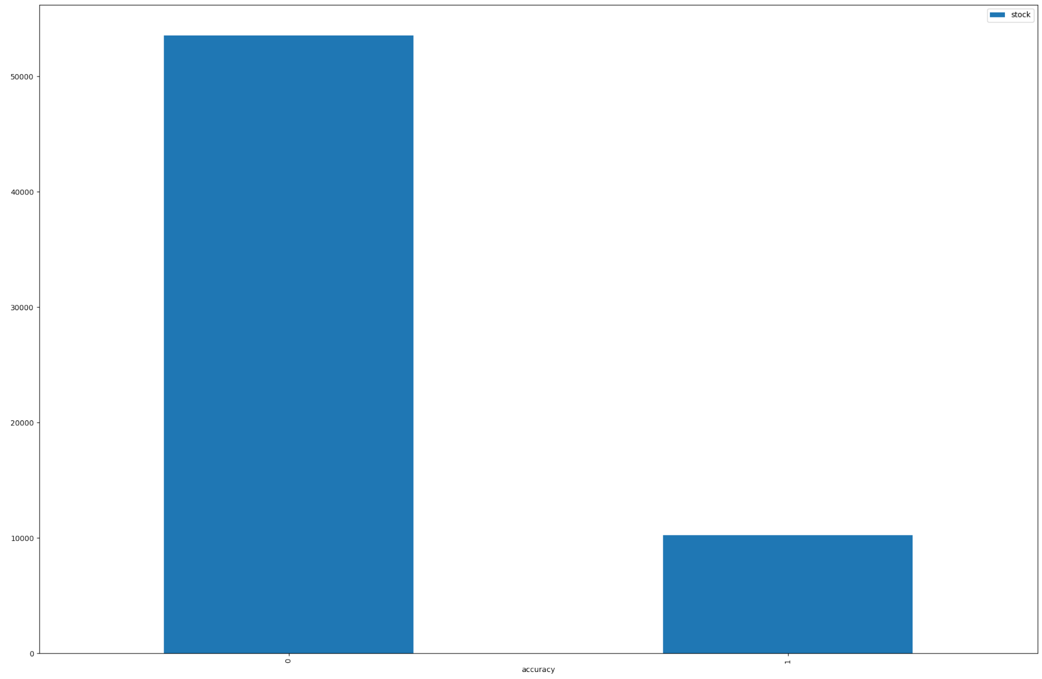


Figure 12: Number of inaccurate and accurate sentiments

3.2 Data Splitting

The data was then split into test and training sets using the following steps:

1. A set of purely positive instances was created.
2. A set of purely negative instances was created.
3. Each set was split into quarters and the first quarter of both was concatenated to form the test set.
4. The remaining three quarters of each set were concatenated to form the training set.
5. The result is two sets of data, of size ratio 1:3 and with equal distributions of positive and negative instances.

The data and labels for both sets consist of the headlines and accuracy, respectively.

3.3 Classifier Model

While the results gained thus far were interesting in their insight into the relationship between headline sentiment and real-world stock performance, to examine whether there were any underlying patterns within the vocabulary of the headlines that might point towards a headline's accuracy, a machine learning model was trained on the headline data and accuracy labels.

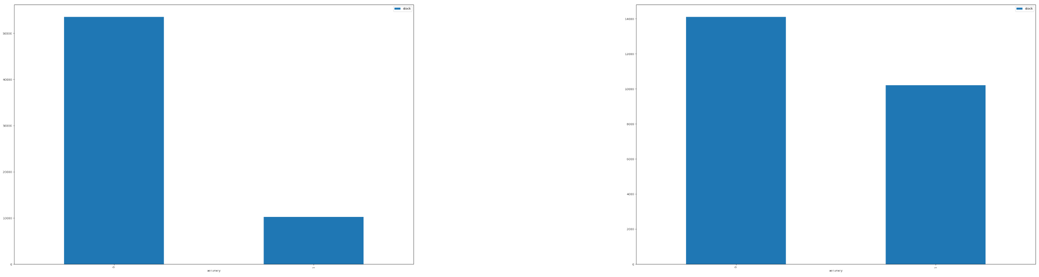
Firstly, following the example set out by [10], a snowball stemmer was defined using NLTK in Python [3]. The purpose of such a stemmer is to reduce each word in a phrase to its base (or root) form. An example of this might be to convert 'swimming' to 'swim' and so on. This process reduces variance in the words forming the headlines, bringing any trends in language/accuracy to the forefront and so leads to improved classification performance.

Using this stemmer in place of a standard CountVectorizer, the model then underwent a grid search to find optimal parameters for training, before sending the best model generated for evaluation.

4 Evaluation

4.1 Critical Analysis

One key insight gained regarding the data through this modelling is that, as shown in figure 8, a large majority of the headlines published on Benzinga have a neutral sentiment. This is unsurprising news for a news website and is to be both hoped and expected however it creates an issue when trying to classify headlines as having an accurate sentiment or not. This is the case because very few stocks remain at the same price across the 3 month window, as shown in figure 9.



(a) Headlines by accuracy with neutral semantics

(b) Headlines by accuracy without neutral semantics

Figure 13: Headlines by accuracy with and without neutral semantics

To explain briefly why this causes issues relating to prediction with machine learning, with so many headlines having a neutral sentiment but so few stocks having a neutral price change, the number of incorrect sentiments is driven up massively. The result of this is that a machine learner will essentially see any kind of neutral language in the headline and immediately predict it will be incorrect. While it is in-line with the objective of this project to have a machine learner use language to determine the correctness of the sentiment, in such a case the model would simply be exploiting a bias in the data.

The second issue arising from this situation is that just because a headline has a neutral sentiment, does not mean that the ultimate recommendation of that article is that a stock will remain level. For example, an article saying 'A has risen by 300% in the last 3 days' might have a neutral sentiment but clearly indicates positive stock movement. To counter this property and explore its effects, the model was evaluated on datasets both including and excluding neutral-sentiment articles.

This observation involves a majority of the conceptual issues with this model, listed succinctly as:

1. The model assumes the sentiment of a headline indicates it's prediction regarding the stock's performance.

2. The model presents a judgement on the accuracy of the headline's sentiment based on the stock's performance over 91 days, as this is the

average time in which a new analyst report is written. However, some reports may in fact refer to shorter or longer-term performance. Without full analysis of the corpus of articles, which is beyond the scope of this paper, such a dynamic judgement would be impossible.

3. One bias of the model addressed through it's implementation is the use of 'balanced' class weighting when training the random forest classifier. This implementation detail ensures that the model is punished proportionally for mis-classifying headlines belonging to a relatively uncommon class. As an example, when running evaluation metrics on the data set in which neutral sentiment remains used, inaccurate headlines are far more common, if this weighting of classes wasn't in place, the model would simply guess 'incorrect' every time and technically be around 80% accurate.

4.2 Formal Evaluation

The resulting classifier model was evaluated using the cross-validation method in which, the input dataset is split into equal chunks and the model is trained, then tested on differing chunks. This process produces an array of results across many different combinations of training and testing data, this array can then be averaged to find a conclusive result. The series of tests executed, were inspired by a similar evaluation of classifiers in [7].

Overfitting is a common issue with classifier models in which the model too closely learns the details of the data it is trained on, and so works excellently for that data but extremely poorly for new, unseen data. This method of validation was chosen due to it's ability to pick up on when a model is overfitting, because of the mixed sets the model is trained/tested on.

After calculating averaged values for the accuracy (percentage of correctly classified headlines) and root mean squared error (RMSE), the model was used to generate a confusion matrix and then a resulting ROC curve.

As discussed above, these tests were all conducted both on data that *included* and *excluded* neutral semantic headlines.

4.2.1 No Neutral Semantics

Random Forrest Classifier (No Neutral Semantics)	Predicted Negative	Predicted Positive
Real Negative	9638	4469
Real Positive	6115	4092

Table 1: Confusion Matrix on data without neutral semantics

- Precision: 0.478
- Recall: 0.401
- Accuracy: 0.566

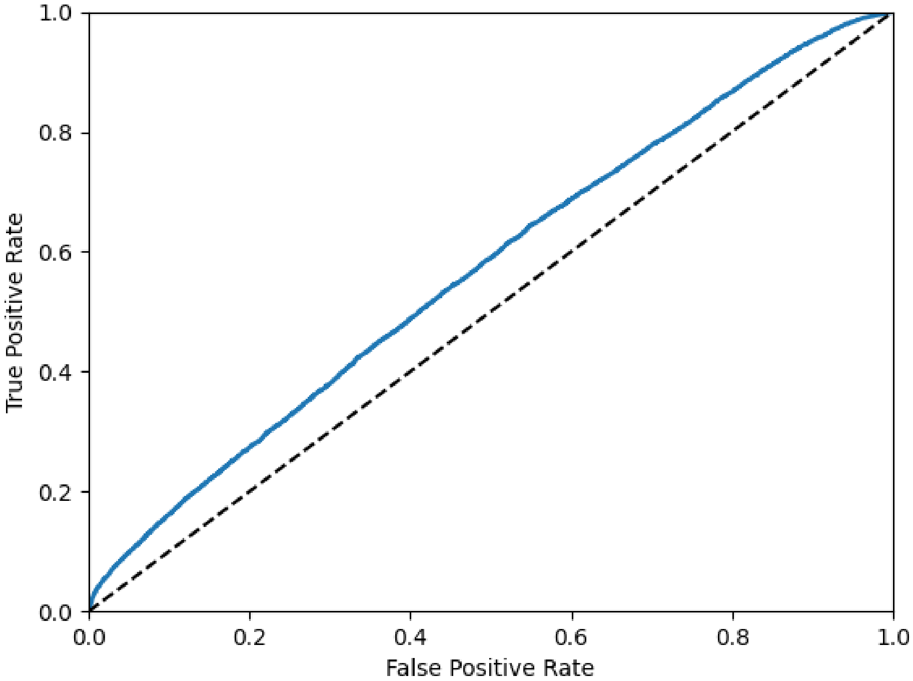


Figure 14: ROC curve for no neutral semantics data

4.2.2 With Neutral Semantics

Random Forrest Classifier (With Neutral Semantics)	Predicted Negative	Predicted Positive
Real Negative	51034	2519
Real Positive	7952	2270

Table 2: Confusion Matrix on data with neutral semantics

- Precision: 0.474
- Recall: 0.222
- Accuracy: 0.836

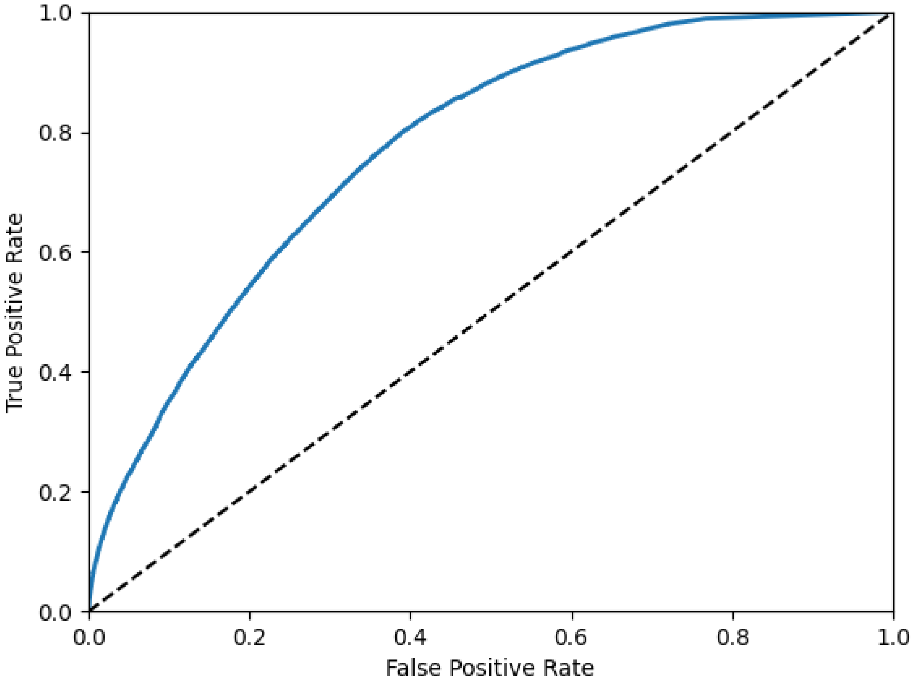


Figure 15: ROC curve for neutral semantics data

5 Conclusions

This project was founded on the question as to whether the semantics of a headline can be found to relate to a relevant stock's performance. This paper proposes that as of yet, there is inconclusive evidence in this regard. Despite achieving greater than 80& accuracy on one of the models proposed in this paper, it is proposed that this result relies largely on the inherent bias headlines to be neutral and stock prices to fluctuate.

In parallel to this however, the model in which no semantically neutral headlines were used, there was still some success. The accuracy achieved by this model was not conclusive, but undeniably better than random classification of headlines as accurate or not. This can be inferred by the difference between the plot in figure 13 and the normal, as well as by the confusion matrix. Due to this better-than-random performance, there is indication of a relationship between the language used in a news headline and the semantic accuracy of that headline re. the relevant stock's performance. This is however, not large enough of a result to conclusively answer the research question.

The proposed model was made in the knowledge of it's flaws and assumptions. Many of these assumptions are listed in 4.1. Were the model to be repeated and improved-upon, a scaling time-window by which to judge stock price difference could be implemented and experimented with, to see how results vary with time. Also, full corpus analysis of the articles could be implemented should the relevant data be scraped from Benzinga.com. As a further direction of experimentation, it is proposed that research should be undertaken to witness how model performance is affected by the use of other stemming, lemmatization and vectorization techniques.

References

[1] apilayer. *Real-Time Historical Stock Data API.* 2021. URL: `https://marketstack.com/`.

[2] Benzinga. *Actionable Trading Ideas, Real-Time News, Financial Insight.* Jan. 2010. URL: `https://www.benzinga.com/`.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[4] bot_developer. *Daily Financial News for 6000+ Stocks.* July 2020. URL: https://www.kaggle.com/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests?select=analyst_ratings_processed.csv.

[5] John Cassidy. *Have the Record Number of Investors in the Stock Market Lost Their Minds?* URL: https://www.newyorker.com/news/our-columnists/have-the-record-number-of-investors-in-the-stock-market-lost-their-minds.

[6] James Chen. *2020 Was a Big Year for Individual Investors.* Dec. 2020. URL: https://www.investopedia.com/2020-was-a-big-year-for-individual-investors-5094063.

[7] Joshua BF Harrison. *Learning Analytics Using OULAD Data,* Software Methodologies, Machine Learning Coursework. 2020.

[8] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[9] Peng Qi et al. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* 2020. URL: https://nlp.stanford.edu/pubs/qi2020stanza.pdf.

[10] Javed Shaikh. *Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK.* 2017. URL: https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a.