BT4222 Mining Web Data for Business Insights

SEMESTER I 2020-2021

Predicting Stock Prices Using Natural Language Processing

Group 4

Andy Low Wei Liang, A0187919H (E0323503@u.nus.edu)
Chee Zhong Quan, A0183873R (e0310668@u.nus.edu)
Tan Tze Kai, Nathaniel, A0167736R (e0176169@u.nus.edu)
Qiao Shizhe, A0167001W (E0175434@u.nus.edu)
Yang Zhi Xiong, Alex, A0168518U (E0176951@u.nus.edu)

TABLE OF CONTENTS

01 Introduction

1.1 The DJIA index

The stock market index is an index that measures the stock market or a subset of the stock market that helps investors compare price levels with past prices to calculate stock performance. The stock market index is usually a weighted arithmetic mean of a basket of selected stocks, usually the major ones. The Dow Jones Industrial Average (DJIA) index that we used for this project is a stock market index that measures the performance of the top 30 large companies in the US. Although there are debates as to whether the DJIA is a good indicator of actual market performance, it is one of the most commonly used indexes.

There are many ways that people trade and monitor stocks in the 21st century. Some have complex algorithms to predict stock prices using thousands of inputs, some monitor essential company ratios over the years (such as price-earning, price-book, P/B ratios) while the most accessible and perhaps the easiest one is to read newspapers and figure out what real world events could affect the stock prices. This method requires no background regarding machine learning and can be used by anyone at any place.

1.2 Variables explained

In this project, our goal is to predict the direction on today's market movement of the DJIA, whether it:
Increases or remains the same - as indicated by a 1
Decreases - as indicated by a 0

To help us achieve this goal, we have sourced a dataset from Kaggle, which contains the daily top 25 headlines from any part of the world crawled from Reddit WorldNews Channel (/r/worldnews) from years 2006 - 2016. The dataset also consists of DJIA index daily for each of the days.

Stock prices fluctuate constantly due to public sentiment, random noise and other factors such as demand and supply. Even though we aim to predict the price movement of stocks based on news headlines, we do not attempt to prove causation but simply correlation - that certain topics mentioned in the headlines will invoke a certain psychological response, which leads investors to trade stocks in a predictable manner. We believe that a few examples that invoke such a response includes: recession, impending war at certain countries and terrorism.

1.3 Hypothesis

In this project, we aim determine whether the following hypothesis is true:
Hypothesis: Chasing the news is a good stock-picking strategy for the individual investor.

02 Exploratory Data Analysis (EDA)

2.1 Data Overview

The data used is extracted from Reddit WorldNews Channel (/r/worldnews) and Yahoo Finance across a span of 8 years - 2008 to 2016.

News Data was ranked by reddit users' votes, and only the top 25 headlines are considered in our model for a single day.

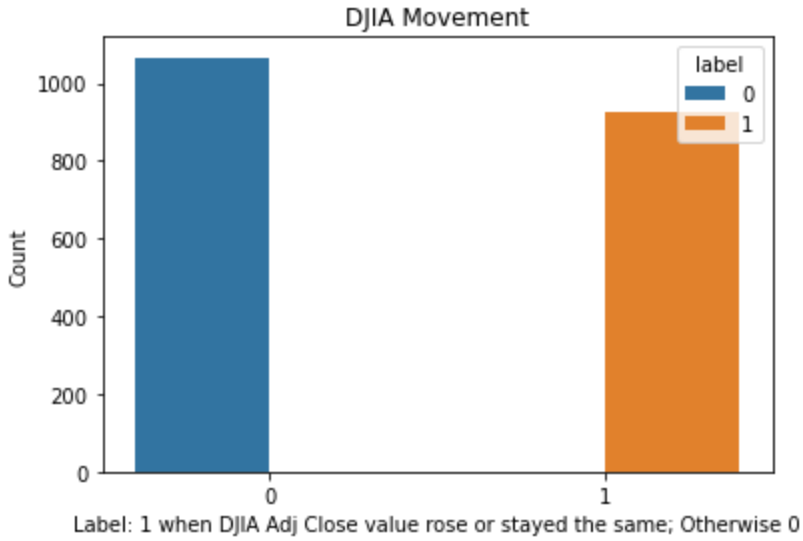


Figure 1: DJIA Movement

Stock Data follows the performance of the Dow Jones Industrial Average (DJIA) as a benchmark to evaluate the performance of our model. We simplified the problem into a binary classification model with only 2 labels: "1" when DJIA Adj Close value rose or stayed the same; "0" when DJIA Adj Close value decreased. Our target variable is fairly balanced, and thus we would not require further sampling from our training and test dataset split.

Next, we will analyze our variables.

2.2 Analysing the variables

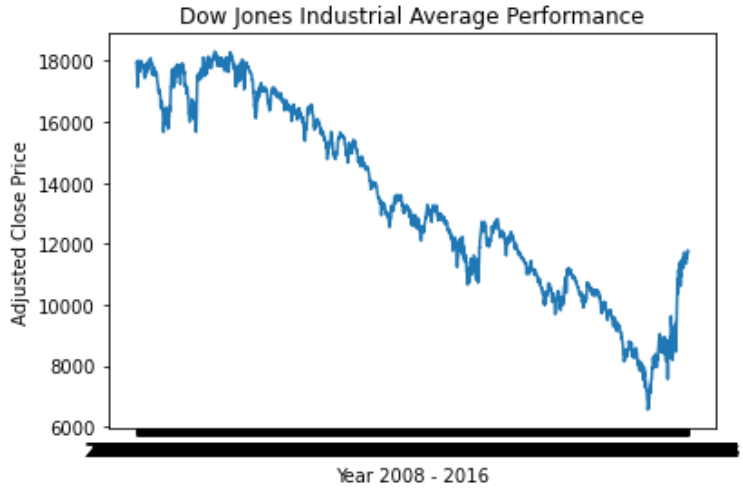


Figure 2: Dow Jones Industrial Average Performance

Market Trend: The Dow Jones Industrial Average (DJIA) is a stock index of 30 blue-chip industrial and financial companies in the U.S. (Seagal, 2020). The index performance gives us a signal on the health of the stock market and economy as a whole. For the year 2008 up to 2016, we can see a general downward trend in DJIA Adjusted Close prices. There is a steeper dip in adjusted closing price from around 2011-2012 and an increase in around 2013 before falling again. In 2016, we can see a steep increase in adjusted closing price of the stock.

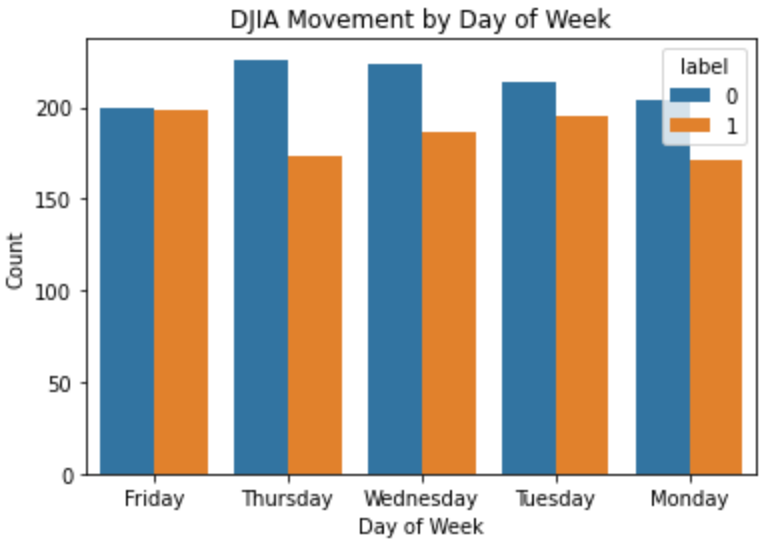


Figure 3: DJIA Movement by Day of Week

Dow Jones Industrial Average (DJIA) Performance by Day of Week

A closer look at daily price changes from a day of week perspective, we can see that the market seems to have the least volatility on Fridays compared to the other days of the week which trend towards negative movements. This might be due to investors looking to close their position ahead of the weekend. On average, Thursdays represent the least desirable day of the week for DJIA stock investments with the highest stock-price-falling to stock-price-not-falling ratio.

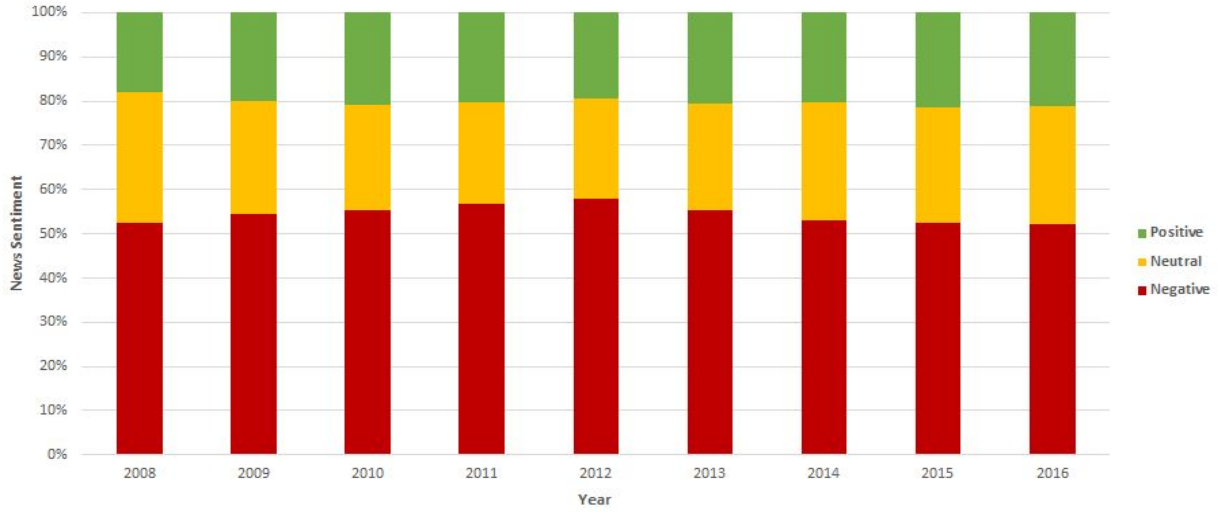


Figure 4: General News Sentiment by Year

Analysis of General News Sentiment with DJIA Movement by Year

News Sentiment: To accurately evaluate the general sentiment of the news articles (i.e. Positive, Negative and Neutral), we used the lexicon and rule-based sentiment analysis tool VADER (Valence Aware Dictionary and sEntiment Reasoner) (source) to evaluate the news headlines on a daily basis. We can see that news sentiment trends were negative (i.e. >50% Negative), and the negative sentiment seems to follow some trends of the stock market. The downward trend correlates to the increasing trend in negative news sentiment, with the steepest decrease on the DJIA falling in line with the peak negativity in 2012. From 2013 onwards, negative sentiment shows a falling trend until 2016, while DJIA increases in 2013 before falling until 2016, in which the fall in negative sentiment in which correlates with the stock market growth. Throughout this period, positivity fluctuates marginally around 20%, while the fluctuations in negativity are compensated by changes in neutral sentiment.

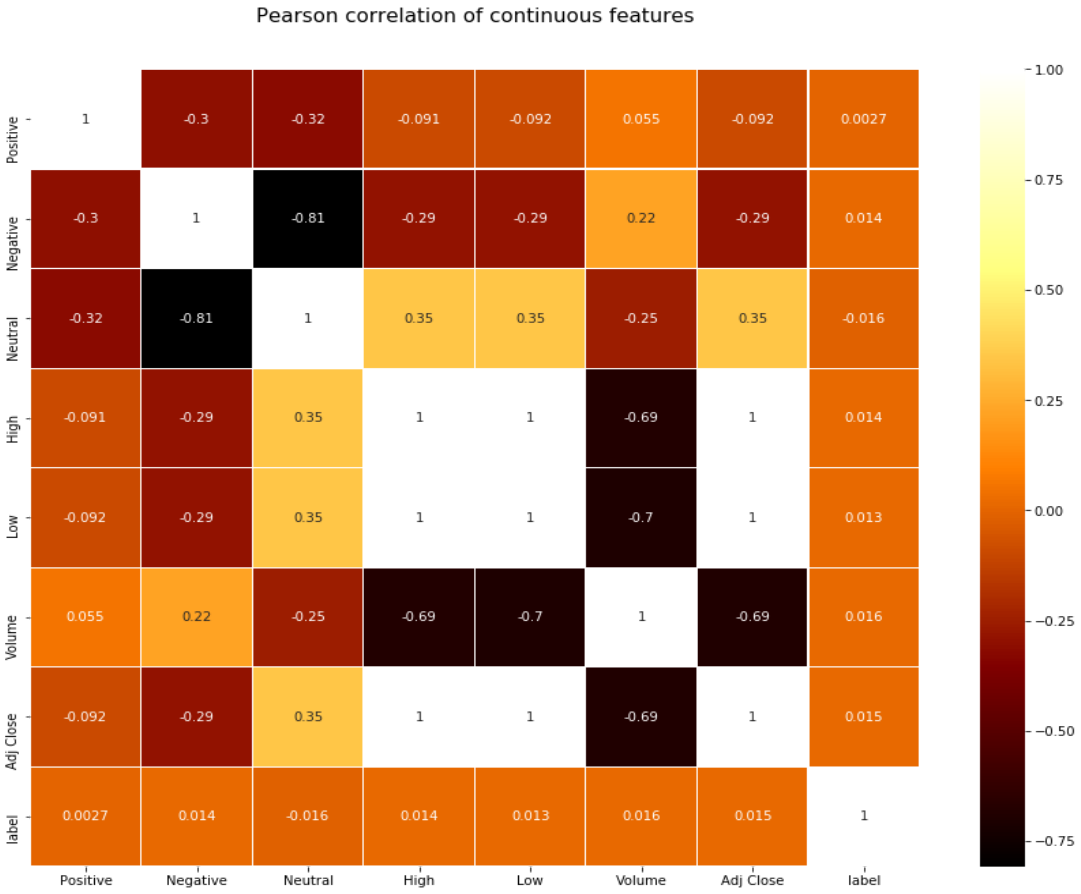


Figure 5: Correlation matrix of continuous features

We can summarize the relationships of our key continuous variables by constructing a correlation matrix. This matrix allows us to isolate and extract relationships between specific variables. The strongest relationships denoted by this chart are as follows:

1) Positive correlation value of **0.35** between the neutrality of news and the range of stocks (high and low), and also the adjusted closing value of the stock.
2) Negative correlation value of **-0.29** between the negativity of news and the range of stocks (high and low), and also the adjusted closing value of the stock.

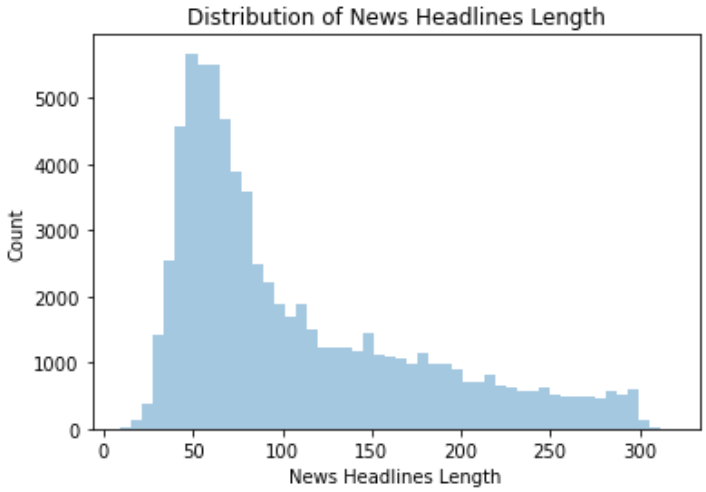


Figure 6: Distribution of News Headlines Length (Overall)

News Headlines: Text length has been shown to be influential towards overall model performance (Amplayo et al., 2019). From the dataset, we can see that the overall distribution of all the news headlines length is left skewed, and these outliers may affect our model performance - especially regression-based models.

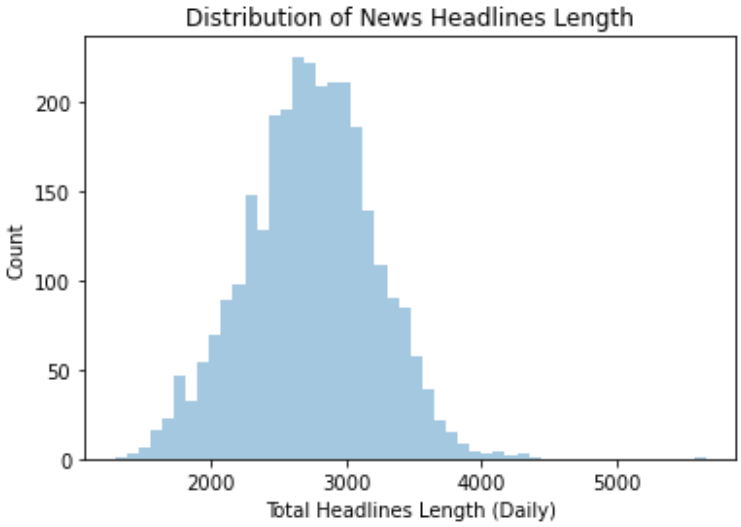


Figure 7: Distribution of News Headlines Length (Daily)

By grouping the news headlines by their date, we see a much more normal distribution of headline length.

Figure 8:Word Cloud for Headlines for Days Where Stock Decreased (left)

Figure 9: Word Cloud for Headlines for Days Where Stock Increased or Remained the Same (right)

In the word cloud above, we see that it is not very informative because neutral worlds like "georgia" and "iran" do not convey much information on their own. Negative words like "war" and "conflict" were also found in days where stock price increased or remained the same, which was not what we expected according to intuition. These figures suggest that there is no clear relationship with sentiment (positive or negative words) and the increase or decrease of stock. To confirm this hypothesis, we will look at n-gram frequencies.

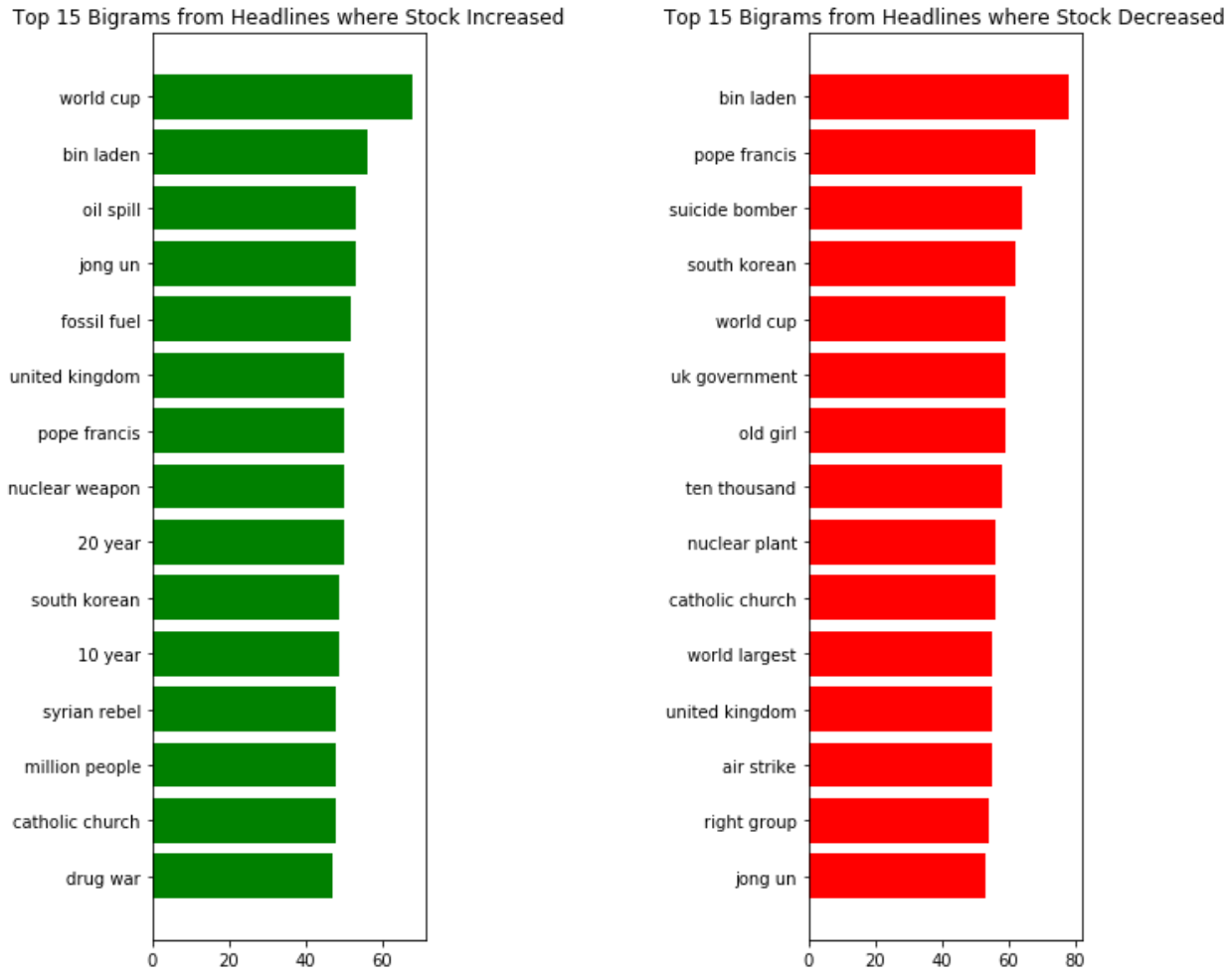


Figure 10: Top 15 Most Common Bigrams

The headlines were lemmatized (for greater consistency), vectorized into bigrams and studied in relation to the target variable (whether stock prices decreased or not) to provide us with the 15 most common bigrams from headlines in each category of the target variable.

Studying the top bigrams from headlines where stock increased and decreased respectively, many of the top 15 bigrams from headlines where stock increased are shared with headlines where stock decreased. Some examples include Bin Laden, the United Kingdom, Jong Un and Pope Francis.

This analysis further solidifies our previous findings where days of increase and decrease are not necessarily associated with more positive or negative sentiments on headlines.

03 Data Pre-processing

Homogenization of Corpus

To pre-process the news headlines, we first **converted all characters to lowercase** using regex. We then converted **word contractions** (eg. ain't, aren't) into their root forms (eg. am not, are not). These steps allow us to homogenize our text vocabulary for more effective analysis.

Removal of Non-Essential Features

Next, we **removed all symbols** (eg. ?!"/) **and stopwords** (eg. the, an, in) from our text. This is an important step as symbols and stopwords are both commonly used and irrelevant for our purposes of analysis. We kept currency figures ("$") in the text as we believe it might be of significance to financial markets.

Lemmatization and vectorization

For our analysis, we **lemmatized** the words in the corpus, removing inflectional endings and returning the base or dictionary form of the words, known as the lemma. This enables us to consider different inflections of the same word together and form a more coherent analysis. Next, to allow us to count the number of unique word pairs occurring in the corpus, we then use the function Count Vectorizer to **vectorize the corpus as bigrams**.

04 Model Selection

Train-Test Split

Since our dataset is a form of panel data, we should optimally train on older observations and make predictions on the test set, composed of later observations. This models reality where we are predicting stock movements based on currently available news. Opting for a train-test split of 80-20, we sorted the dataset by date, obtaining the 80th percentile at 2014-12-03. Hence, any observations equal to or before this date will be allocated as the train set while any observations after will be allocated as a test set. The classification problem is hence based on predicting the stock movements between 03 December 2014 to 01 July 2016.

In training the models, we tried varying the following variables:

- Top n news, n = 5,10,25
  New headlines are widely believed to be sensationalised. We believe that with the addition of more news headlines, it need not provide additional information on stock

movements but create more noise instead. This may lead to a drop in accuracy. Hence, we trained our models based on the top 5, 10, 25 news headlines to determine if this assumption holds true.

- Respective kernels and parameters
  Through GridSearchCV we were provided the means of hyperparameter tuning, obtaining the best parameters for the simpler classification models.

4.1 Logistic Regression

Logistic Regression is a classification model where the dependent variable can only take two values, 0 (false) and 1 (true) (Twala, 2010), perfect for our dependent variable of binary stock movement. However, it is important to note that Logistic Regression will not provide exact values of 0 or 1. Instead, it returns the probabilistic value which lies between 0 and 1 (Gurucharan, 2020).

Under the Logistic Regression, instead of a regression line, we fit a S-shaped curve with the formula:

$$1/(1+e^{-value})$$

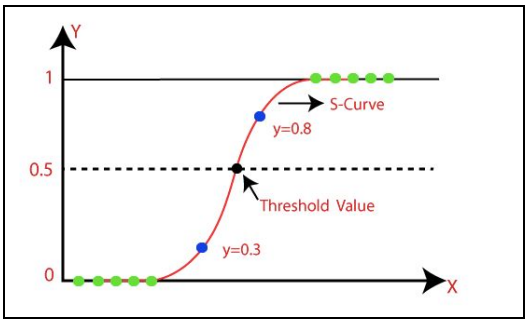


Figure 11: Logistic Function (Source: Gurucharan, 2020)

As seen in Figure 11, the logistic function can take on any values approaching 0 or 1 but never exactly. In Linear Regression, when predicting if the stock price rises/stays the same or decreases, we would get continuous values between 0 and 1. However, under the Logistic Regression model, we set a threshold of 0.5 as seen in Figure 11. As such, the change in stock price will be classified as "increase/remain the same" if the value is greater than the threshold of 0.5, and "decrease" if the value is less than 0.5

4.1.1 Model Evaluation

Top n news	Accuracy	Precision score	F1 score	Recall score
5	0.497	0.503	0.627	0.833
10	0.505	0.509	0.611	0.766
25	0.516	0.529	0.220	0.138

4.2 Support Vector Machine (SVM)

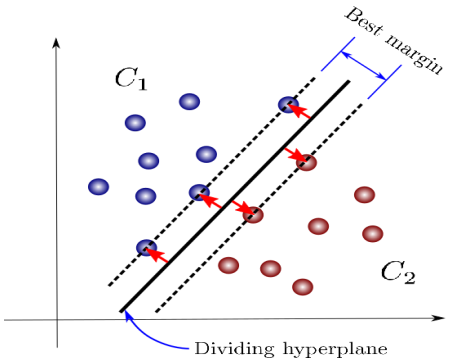


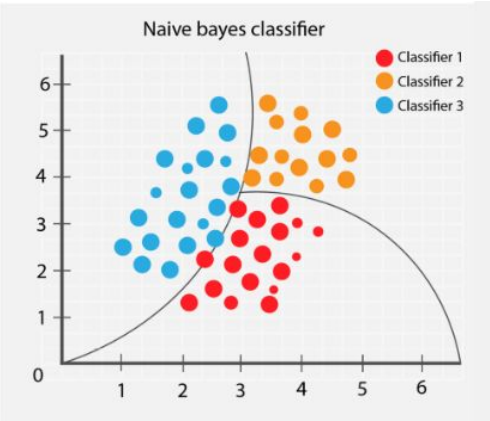
Figure 12: Support Vector Machine Diagram (Source: Oscar, 2019)

For this model, we would plot each data item as a point in a n-dimension space with the value of every characteristic being a coordinate. We would then conduct classification by finding out what would be the optimal hyper-plane for the selected characteristics.

4.2.1 Model Evaluation

Top n news	Accuracy	Precision score	F1 score	Recall score
5	0.506	0.506	0.672	1.0
10	0.506	0.506	0.672	1.0
25	0.509	0	0	0

4.3 Naive Bayes Classifier



$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Figure 13: Naive Bayes Classifier Model (Source: Yang, 2019)

Mathematically, the Bayes theorem is represented as P(A|B). The Naive Bayes Classifier belongs to the family of probability classifier, using Bayesian theorem. This method solves classification problems using a probabilistic approach. However, it has a strong assumption that all the variables are independent of one another. This might not be the case for real-life examples. This is also why the model requires much less training data. Even if the assumption does not hold, this method could still prove to be an effective one.

4.3.1 Model Evaluation

Top n news	Accuracy	Precision score	F1 score	Recall score
5	0.460	0.473	0.510	0.552
10	0.487	0.496	0.538	0.589
25	0.520	0.521	0.391	0.313

Evaluation of Top n news

Given the results thus far, reducing the number of news headlines did not provide significant improvement in accuracy. Hence, we concluded the number of news headlines does not provide sufficient incentive for the additional training time and determined that the optimal course of action is training our remaining models with all the top 25 news headlines.

4.4.Random Forest Classifier

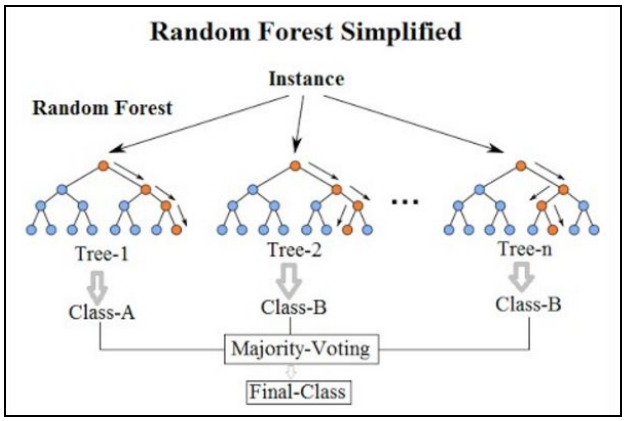


Figure 14 : Random Forest (Source: Jagannath, 2020)

Random Forest consists of many individual decision trees that operate together. Each decision tree represents an independent variable and generates a class prediction, which contributes to a vote in the final prediction. Since decision trees are highly sensitive to the data they are trained on, small changes to the training set can lead to significantly different tree structures. Thus, random forest builds on this by allowing each individual tree to randomly sample from the dataset with replacement (bagging/bootstrap aggregation), resulting in different trees. With the low correlation between trees, they are able to protect each other from potential errors that might occur.

4.4.1 Hyperparameter Tuning

Hyperparameter Tuning is defined as choosing a set of optimal hyperparameters for a learning algorithm. We will be employing Grid Search as our tuning strategy. Grid Search works by searching exhaustively through a specified subset of hyperparameters (Boyle,2019), taking a Caertesian Product of the parameters, optimising for a chosen metric. For the Random Forest Classifier, our sets of hyperparameters include n_estimators, criterion, bootstrap, max_depth, max_features, min_samples_leaf and min_sample_splits. GridSearchCV will thus provide the best sets of hyperparameters for training our Random Forest model.

4.4.2 Model Evaluation

Hyperparameter Tuning	Accuracy	Precision score	F1 score	Recall score
No	0.511	1.0	0.01	0.005
Yes	0.514	0.750	0.03	0.015

After using hyperparameter tuning, our accuracy of our Random Forest model did improve(from 0.511 to 0.514).

4.5 Boosting Classifier

4.5.1 Gradient Boosting Classifier

Gradient Boosting is a forward stagewise method that gradually improves a weak learner, producing a prediction model in the form of weaker prediction models. Gradient Boosting consists of three elements, a loss function to be optimised, a weak learner to predict and an additive model to add weak learners while minimising the loss function. Firstly, we consider a tree that minimises our desired loss function. Additively, we fit a tree to a model that reduces the loss function, partitioning the tree, where existing trees remain unaltered. We achieve this by parameterising the tree, modifying the parameters then moving towards reducing the residual loss.

The algorithm is as follows:

1. Initialize $f_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
2. For m = 1 to M
    a. Compute $r_{im} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_1)}]_{f=f_{m-1}}$ for i = 1, … , N
    b. Fit regression tree to $\{r_{im}\}$ giving terminal partitions $\{R_{jm}\}$, for $j = 1, \ldots, J_m$
    c. For $j = 1, \ldots, J_m$, compute $\gamma_{jm} = argmin_\gamma \Sigma_{x_i \varepsilon R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$
    d. Update model to $f_m(x) = f_{m-1}(x) + \lambda \sum_{j=1}^{J_m} \gamma_{jm} 1(x \varepsilon R_{jm})$

4.5.1.1 Hyperparameter Tuning

We will use the Randomized search as our hyperparameter tuning method that we used for our Random Forest Classifier in 4.4.1. For this Gradient Boosting model, the sets of hyperparameters that we will be tuning are learning_rate,max_depth and max_features.

4.5.1.2 Model Evaluation

Hyperparameter Tuning	Accuracy	Precision score	F1 score	Recall score
No	0.456	0.409	0.303	0.241
Yes	0.484	0.438	0.255	0.179

After using hyperparameter tuning, our accuracy of our XGBoost model did improve(from 0.456 to 0.484).

4.5.2 XGBoost Classifier

XGBoost (eXtreme Gradient Boosting) is a software library which provides a gradient boosting framework . It implements the gradient boosting tree decision algorithm, and contains other key algorithms to automatically handle missing data, support parallelisation of tree construction, and allow for continued boosting of an already fitted model with new data. It supports both regression and classification predictive modelling problems.

Compared to the original Gradient Boosting algorithm above, XGBoost includes regularisation that penalises more complex models and 'learning' the best missing value based on training loss. The algorithm also comes with a built-in cross-validation at each iteration to find the best hyperparameters without the need to explicitly program the validation.

4.5.2.1 Hyperparameter Tuning

Due to immense runtime for GridSearch, we decided to use Randomized search as our hyperparameter tuning method for this model. For this XGBoost model , the sets of hyperparameters that we will be tuning are, n_estimators, colsample_bytree, max_depth, reg_alpha, reg_lambda , subsample, learning_rate, gamma, min_child_weight, sampling_method.

4.5.2.2 Model Evaluation

Hyperparameter Tuning	Accuracy	Precision score	F1 score	Recall score
No	0.466	0.441	0.373	0.323
Yes	0.514	0.750	0.003	0.015

After using hyperparameter tuning, our accuracy of our XGBoost model did improve(from 0.466 to 0.514).

Classifier Models Evaluation

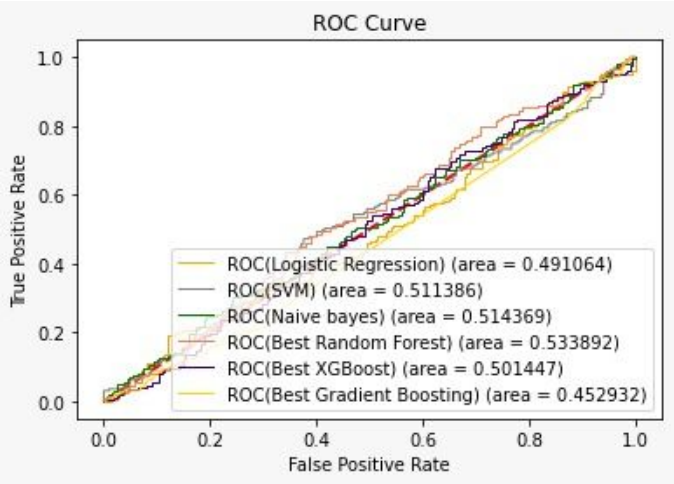


Figure 15: ROC curve

From our ROC curve above, we observed that the performances of our classifier models are roughly about the same. Hence, there is a need for us to explore more models of higher complexity such as Neural Network models which we will be going through in the next section.

4.6 Neural Network

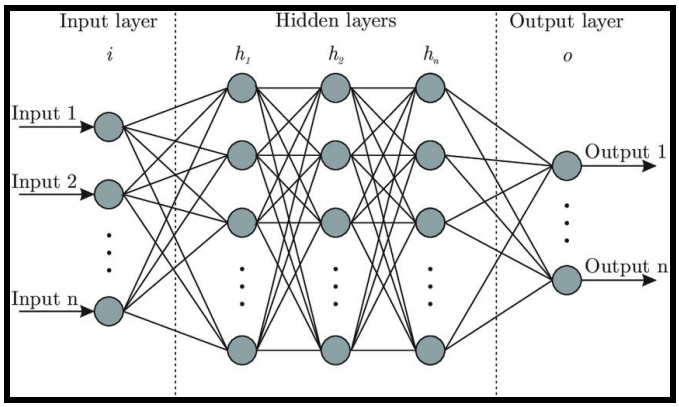


Figure 16: ANN Network Diagram (Source: Shukla, 2020)

Artificial neural networks are the workhorses of deep learning. They are black boxes trying to achieve good predictions. A neural network consists of both input and output neurons which are

weighted with varying hidden layers. The weights will affect the degree of forward propagation that goes through the algorithm. When the back propagation happens, the weights are flexible enough to change and this is when the neural network learns.

The constant process of forward and backward propagation is conducted iteratively for all data in the training set. The larger the dataset, the more the neural network will learn, and therefore the more accurate the algorithm will be at forecasting outputs.

For the prediction of stock movement, we employed three forms of neural networks. Namely basic Artificial Neural Network (ANN), Convolutional Neural Network (CNN) and
Long Short-Term Memory (LSTM), a form of Recurrent Neural Network (RNN). We will be building the above neural networks using the keras sequential model, carrying out parameter tuning as well as addition and alteration of layer size accordingly. All models will run on a fit with batch size = 32 over 100 epochs and early callback with monitor = val_loss at patience = 10 and min_delta= 0.0001. We have utilised the sigmoid and reLU activation functions, however, due to the problem of vanishing gradients, we are compelled to limit the number of layers in the model. As weights are initialised randomly, we obtained our results from the average of 10 runs.

4.6.1 Artificial Neural Network

Artificial Neural Networks (ANN) consists of a group of multiple perceptrons in each layer. Inputs are only processed in a forward manner, where information is only passed in a singular direction from the input node to the output node as shown in figure 5 above. ANNs are the simplest of neural networks but provide the ability to work with incomplete information.

In this case, the input being the 25 top news headlines are tokenized, before being vectorized with CountVectorizer or TFIDVectorizer respectively and converted into a dense matrix. The input passes through 2 Dense and 1 Dropout layer for regularisation before reaching the output node.

4.6.1.1 Model Evaluation

Preprocessing	**Parameters**	**Loss**	**Accuracy**
CountVectorizer	9946 Total and Trainable Params (2 Dense, 1 Dropout Layers)	0.701	0.506
	10418 Total & Trainable Params (2 Dense, 1 Dropout Layers)	**0.742**	**0.539**

TFIDVectorizer	9946 Total and Trainable Params (2 Dense, 1 Dropout Layers)	0.707	0.506
	10418 Total & Trainable Params (2 Dense, 1 Dropout Layers)	0.744	0.521

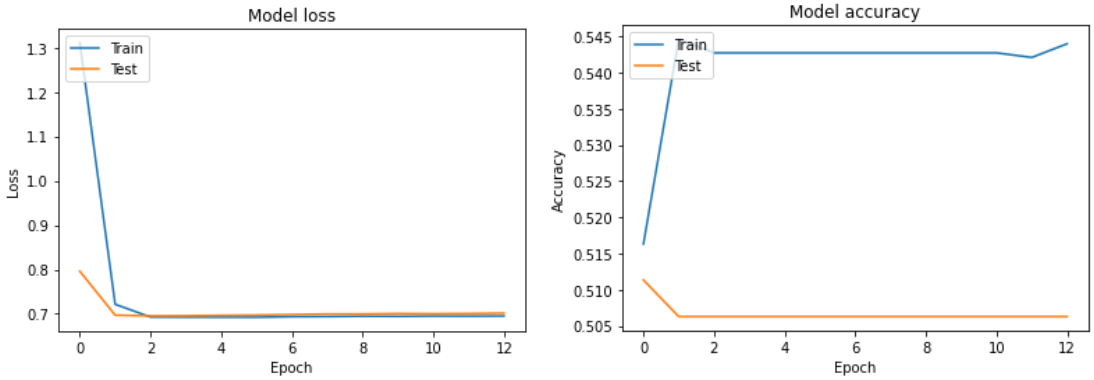


Figure 17: Plot of Loss and Accuracy for CountVectorizer Preprocessing with 10418 Params

The traditional ANN has displayed promising results and showed that increasing layer sizes negatively impacts the loss and accuracy of the model. This observation is something we kept in mind for the following models.

4.6.2 Convolutional Neural Network

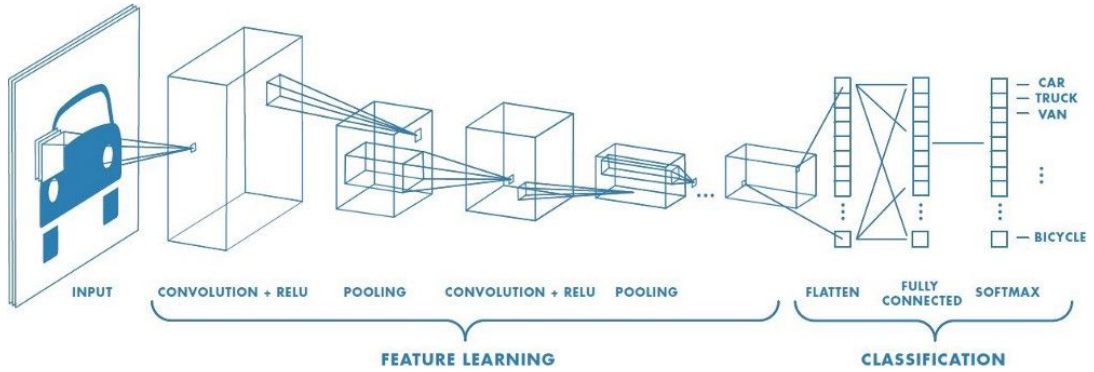


Figure 18: CNN (Source: Stanford University, 2018)

Convolutional Neural Networks (CNN) are different from traditional ANNs in the composition of hidden layers. Hidden layers in CNNs include convolutional, pooling and normalisation layers not found in ANNs. This implies that convolution and pooling are used as activation functions instead of common sigmoid, reLU and tanh and is typically used to study "spatial correlation" in data and Computer Vision.

Convolution works by applying a filter over inputs as shown in Figure 6 above, and extracting information, which is multiplied by the kernel to obtain an output signal, where mathematically, the convolution of two functions is defined as a product of the input and kernel function as shown in the equation below.

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + m/2)$$

Whereas Pooling is a sample-based discretization process, down-sampling the input, reducing the dimensions, allowing assumptions to be made about the features contained in the convolution output.

For our model, we processed the news headlines with trained Word2Vec embeddings and made use of Conv1D, effective for CNNs to derive features from fixed-length segments of the dataset. However, in Natural Language Processing (NLP), the proximity of words is not taken into account. Following which, we made use of GlobalMaxPooling1D, an alternative to the Flattening block after the last pooling block and replacing the fully connected blocks of CNN, before passing them through two more Dropout and Dense layers to obtain the output.

4.6.2.1 Model Evaluation

Parameters	**Loss**	**Accuracy**
170,300 Total & Trainable Params (1 Conv1D, GlobalMaxPooling, 2 Dense, 2 Dropout)	1.438	0.511
160276 Total & Trainable Params (1 Conv1D, GlobalMaxPooling, 2 Dense, 2 Dropout)	**0.898**	**0.569**

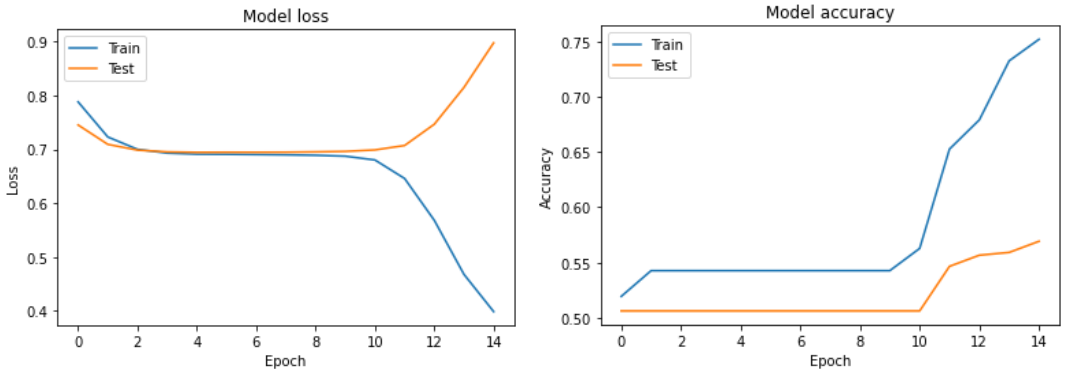


Figure 19: Plot of Loss and Accuracy for 160276 Params

With CNN, we have obtained the model with the best accuracy at 0.569 at a relatively low loss of 0.898. However, an observable trait of our model is the instability where test loss increases exponentially even as train loss decreases, as both train and test accuracy increases.

4.6.3 Long Short-Term Memory

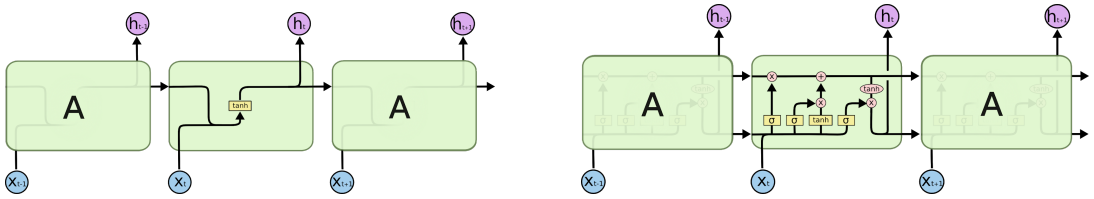


Figure 20: RNN vs LSTM (Source: Colah, 2015)

Long Short-Term Memory (LSTM) is a form of Recurrent Neural Network (RNN). LSTMs retain information over long sequences, allowing the significance of word proximity in a sentence to be retained, which is especially relevant to the study of news headlines. As shown in Figure 7 above, LSTM introduces a forget gate which remembers a sequence if it has observed a similar one before and resolves the problem of exploding gradient by limiting the range of the signal to between 0 and 1. However, a drawback of LSTM over CNN is the amount of training time due to the sequential nature of LSTM.

For our model, we made use of Word2Vec embeddings as well, followed by a LSTM layer, Dropout and Dense layer before reaching output. We have found this to work best through our attempts in minimising loss and maximising accuracy.

4.6.3.1 Model Evaluation

Parameters	**Loss**	**Accuracy**
160,562 Total & Trainable Params (1 LSTM, 1 Dense, 1 Dropout Layers)	2.11	0.537
160722 Total & Trainable Params (1 LSTM, 2 Dense, 2 Dropout Layers)	**2.49**	**0.542**

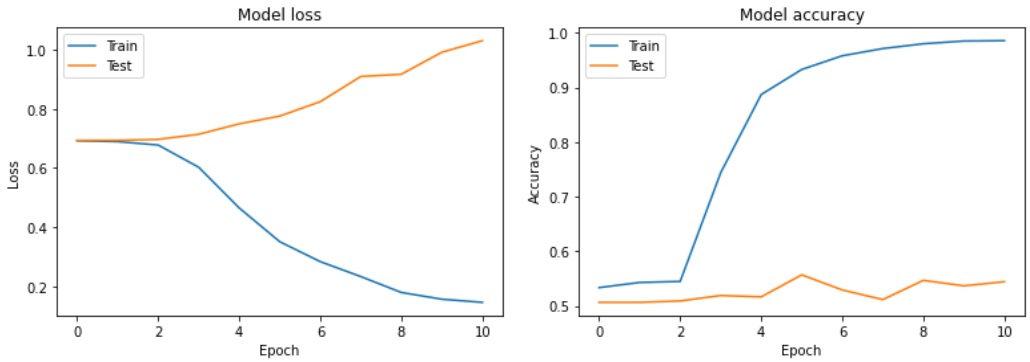


Figure 21: Plot of Loss and Accuracy for 160722 Params

With LSTM we expected to achieve results that surpasses CNN, however this is not the case. The effectiveness of LSTM in analysing news headlines may be limited by the temporal changes in sentiments affecting the word embeddings, as shown in the high loss values which will be further elaborated in the limitations. Notably, LSTM still performed very well on average compared to all other models so far.

An interesting observation of LSTM is that as the number of params increase, the loss is minimised and accuracy is maximised, which is conversely true for both CNN and ANN.

05 Models Evaluation

Logistic Regression

Top n news	Accuracy	Precision score	F1 score	Recall score
5	0.497	0.503	0.627	0.833
10	0.505	0.509	0.611	0.766
25	0.516	0.529	0.220	0.138

Support Vector Machine (SVM)

Top n news	Accuracy	Precision score	F1 score	Recall score
5	0.506	0.506	0.672	1.0
10	0.506	0.506	0.672	1.0
25	0.509	0	0	0

Naive Bayes Classifier

Top n news	Accuracy	Precision score	F1 score	Recall score
5	0.460	0.473	0.510	0.552
10	0.487	0.496	0.538	0.589
25	0.520	0.521	0.391	0.313

Random Forest Classifier

Hyperparameter Tuning	Accuracy	Precision score	F1 score	Recall score
No	0.511	1.0	0.01	0.005
Yes	0.514	0.750	0.03	0.015

Gradient Boosting Classifier

Hyperparameter Tuning	Accuracy	Precision score	F1 score	Recall score
No	0.456	0.409	0.303	0.241
Yes	0.484	0.438	0.255	0.179

XGBoost Classifier

Hyperparameter Tuning	Accuracy	Precision score	F1 score	Recall score
No	0.466	0.441	0.373	0.323
Yes	0.514	0.750	0.003	0.015

Traditional Artificial Neural Networks (ANN)

Preprocessing	Parameters	Loss	Accuracy
CountVectorizer	9946 Total and Trainable Params (2 Dense, 1 Dropout Layers)	0.701	0.506
	10418 Total & Trainable Params (2 Dense, 1 Dropout Layers)	0.742	0.539
TFIDVectorizer	9946 Total and Trainable Params (2 Dense, 1 Dropout Layers)	0.707	0.506
	10418 Total & Trainable Params (2 Dense, 1 Dropout Layers)	0.744	0.521

Convolutional Neural Networks (CNN)

Parameters	Loss	Accuracy
170,300 Total & Trainable Params (1 Conv1D, GlobalMaxPooling, 2 Dense, 2 Dropout)	1.44	0.511
160276 Total & Trainable Params (1 Conv1D, GlobalMaxPooling, 2 Dense, 2 Dropout)	**0.898**	**0.569**

Long Short-Term Memory (LSTM)

Parameters	Loss	Accuracy
160,562 Total & Trainable Params (1 LSTM, 1 Dense, 1 Dropout Layers)	2.11	0.537
160722 Total & Trainable Params (1 LSTM, 2 Dense, 2 Dropout Layers)	**2.49**	**0.542**

In conclusion, the best model for predicting stock movement is CNN at an accuracy of 56.9% and loss of 0.8979. The model is a remarkable feat as it outperformed most of the kaggle entries with an accuracy of 52~53%, all without introducing stock variables as predictors.

It is worth noting that despite LSTM falling short slightly at 54.2%, it remains viable due to the nature of the model in the NLP context and similarly outperforming entries on kaggle. Additional work can be carried out on CNN and LSTM for parameter tuning with the possibility of improving the model slightly.

However, after conducting extensive review of news headlines being a predictor of stock movement. We believe that chasing the news may not be the best stock-picking strategy for the individual investor at least in the machine learning context.

06 Limitations and Recommendations

6.1 Limitations

6.1.1 Limited Data: News Selection

Reddit WorldNews Channel is not financial news. There could be a selection bias in the news headlines we considered in our model. This is because there might be a possibility for news to rank highly on these pages which might not be directly related to the financial markets, or even heavily skewed towards US investors, rather than a global audience. This results in a lack of relevant news data on the companies represented in the DJIA as many investors of the DJIA might also be global investors who are affected by news in their region.

Also, news are increasingly sensationalized, and these news events are shown to drain liquidity and reduce volatility in stocks (Peress, J. & Schmidt, D.,2016). This style of news report encourages biased impressions of events rather than neutrality, and may cause a manipulation to the sentiment data behind our model. Our model did not consider other sources of information such as those on social media, and thus our model might only be considering systemic risk where probability of a loss is associated with the entire market or segment rather than the unsystematic risk associated directly with the companies listed on DJIA.

6.1.2 Limited Data: Financial Market Performance

Despite accounting for over 8 years of stock movement, the data set is rather limited in size, with only 1989 observations which might not be representative of the entire business cycle from pre to post correction of the markets. Ideally, we would also want to include a broader range of asset classes as investors will often balance their assets between different asset classes in times of financial volatility. Not only that, we did not account for the change of components in the DJIA, where there were 6 changes within this period.

Many attempts have been made over the years to extract useful patterns of stock market movements (Hirshleifer and Shumway, 2003), however no method has been discovered to accurately predict stock movement. In order for news headlines to accurately predict stock movements, the Efficient Market Hypothesis has to hold true (Lowe and Webb, 1991), however, given our attempts at prediction, the results are in strong support of the Random Walk Theory (Malkiel, 1996), where prices are predicted randomly and attempts at prediction is virtually infeasible as of present. Moving forward, we should consider effective features into the dataset that reflect investor psychology, such as the anticipatory nature of investors, in the dataset.

Lastly, We did not consider the scale of the market changes, but only analyzed and predicted for absolute positive/ negative growth, and negative growth might represent price corrections after a period of growth for the companies.

6.1.3 Temporal Changes in Sentiments and Word Embeddings

Over time, words, events and names linked to positive sentiments might become conversely true. In such cases, word embeddings such as Word2Vec, gensim and gloVe may not be able to accurately reflect such change in sentiments in vector form and an updated corpus may have to be trained. As such, the further the news is from the present, the informative it is towards current sentiments.

6.2 Recommendations

6.2.1 Rolling Timeframe for Analysis

In order to minimise the perceived changes in sentiments of words, models could be trained on 10 months of a year and tested on the following 2 months.

6.2.2 New Sources of Headlines

Instead of scraping news from reddit, attempts could be made to scrap news from more reliable sources, provided that the timestamp is available. This effectively reduces the number of sensationalised news which could play a role in the sentiment of sentences.

6.2.3 Hybrid Models and Ensembles

Alternative indicators of stock market performance can be studied, forming other models. From which, the models can be combined forming a new forecast through methods such as the Granger-Ramanathan combination, a regression method to combine forecasts by attaching weights to each model (Granger and Ramanathan, 1984).

6.3 Applications

6.3.1 Implementation into Roboadvisors

Robo-advisory investing offers ease of use, convenience, and affordable fees, as an attractive alternative to low-interest savings accounts for those who prefer to employ a "hands-off" approach towards investing. Implementing our model with roboadvisors can help them to identify and manage risk faster to force rebalancing whenever required.

Not only that, the quicker consumption and sentiment scoring of news can allow roboadvisors to transition from a passive to a more active investing approach, and further scale the service to include a wider range of asset classes across geographies and regions within each portfolio - as we would be able to factor in different global news sources for their individual market impacts.

07 Conclusion

Our group created a model to predict whether chasing the news is a good stock-picking strategy for the individual investor. The dataset is from Kaggle and consist of two main datasets – News Headlines from Reddit WorldNews Channel and DJIA Daily Index data from Yahoo Finance. After which we converted the direction of each day's DJIA movement into binary as our target variable to determine the impact of news headlines on the stock market.

There were 9 models built – Logistic Regression, Support Vector Machine, Naïve Bayes, Random Forest, Boosting Classifiers (XGBoost and Gradient Boost) and Neural Network Models (ANN, CNN and LSTM) to evaluate the best model for making stock market predictions. Using Accuracy, Precision, Recall and F1 Score as performance metrics across different subsets of our news headlines (Top 5, 10 and 15), we have selected CNN as the final model with an accuracy of 56.9%. Although it has an accuracy score 2.7% higher than that of LSTM model, we feel that with the introduction of more stock market predictors and further parameter tuning, the LSTM model will have greater potential for improvement due to the nature of its model in preserving the information for word proximity. Hence, we conclude that chasing the news is not a good stock-picking strategy, at least with regards to reddit news.

In the future, our model accuracy can be further improved by training on a rolling time frame with more reliable news sources. We could also improve model generalisability by including alternative stock indicators to form new models such as the Granger-Ramanathan combination regression method. Recognising these factors, our model can then be further applied and scaled with Roboadvisors for better risk management and transition into a more active investing option for investors.

08 References

Amplayo, R. K., Lim, S., & Hwang, S. (2019). Text Length Adaptation in Sentiment Classification. 647-658.

Boyle, T(2016). Hyperparameter Tuning. Retrieved from https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624

Colah. (2015). Understanding LSTM Networks. Retrieved November 18, 2020, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

D. Lowe and A. R. Webb, "Time series prediction by adaptive networks: a dynamical systems perspective," in IEE Proceedings F - Radar and Signal Processing, vol. 138, no. 1, pp. 17-24, Feb. 1991, doi: 10.1049/ip-f-2.1991.0004.

Granger, C.W.J. and Ramanathan, R. (1984) Improved Methods of Combining Forecasts. Journal of Forecasting, 3, 197-204. http://dx.doi.org/10.1002/for.3980030207

Gurucharan, M. (2020, August 06). Machine Learning Basics: Logistic Regression. Retrieved November 10, 2020, from https://towardsdatascience.com/machine-learning-basics-Logistic-regression-890ef5e3a272

Hirshleifer, D. A., & Shumway, T. (2001). Good Day Sunshine: Stock Returns and the Weather. *SSRN Electronic Journal*. doi:10.2139/ssrn.265674

Jagannath, V. (2020, August 06). Random Forest Template for TIBCO Spotfire. Retrieved from https://community.tibco.com/wiki/random-forest-template-tibco-spotfire

Mahanta, J. (2017, July 10). Introduction to Neural Networks, Advantages and Applications. Retrieved from https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-Applications-96851bd1a207

Malkiel, B. G. (2020). *Random walk down wall street.* 6th ed. LondonL W.W. Norton Co.

Oscar, C. C. (2019). Support Vector Machines for Classification. Retrieved from: https://towardsdata science.com/support-vector-machines-for-classification-fc7c1565e3

Peress, J., & Schmidt, D. (2016). Glued to the TV: Distracted retail investors and stock market liquidity. Unpublished working paper, INSEAD from https://www.bi.edu/globalassets/forskning/center-for-asset-pricing-research/seminars/peress_attention33_jf_withinternetappendix.pdf

Segal, T. (2020, August 28). How Does the Dow Jones Work? Retrieved from https://www.investopedia.com/investing/what-moves-the-djia/

Shukla, L. (2020). Weights and Biases. Retrieved from https://www.kdnuggets.com/2019/11/designing-neural-networks.html

Simpson, M. (2018, November 19). Machine Learning Algorithms: What is a Neural Network? Retrieved from https://www.verypossible.com/insights/machine-learning-algorithms-what-is-a-neural-network

Sun, J.(2016). Daily News for Stock Market Prediction, Version 1. Retrieved November 10,2020 from https://www.kaggle.com/aaron7sun/stocknews

Twala, B. (2010). Multiple classifier application to credit risk assessment. *Expert Systems with Applications, 37*(4), 3326-3336. doi:10.1016/j.eswa.2009.10.018

Yang, S. (2019).An Introduction to Naive Bayes Classifier. Retrieved from: https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf