NLP: Sentiment Analysis of Financial News using Deep Learning

Anonymous Author
University College London
London WC1E 6BT
uczljg7@ucl.ac.uk

Abstract

In this project, multiple deep learning models are built to perform sentiment analysis on financial news headlines. The deep learning models include simple feedforward neural networks (FNN), convolutional neural networks (CNN) and long short-term memory (LSTM). A pre-trained BERT transformer model is also applied to evaluate its performance on financial sentiment analysis. The models are evaluated using confusion matrix, classification report, and ROC-AUC curve. The results show that the deep learning models outperform the baseline models in terms of accuracy and efficiency. The BERT transformer model achieves the highest accuracy among all models. The results demonstrate that deep learning models can be used to perform sentiment analysis on financial news headlines with high accuracy and efficiency.

1 Introduction

1.1 Background

Nowadays, a tremendous volume of information is generated (in the form of semi-structured/unstructured data) every day through financial reports, news headlines and market commentary within the financial market. This provides valuable insights in the process of decision-making for investors, traders, and financial analysts. However, this large amount of textual data makes it impossible for humans to process and analyse all of it manually. With the recent advancements in artificial intelligence and deep learning, especially in the field of natural language processing (NLP), it is now possible to automate the analysis of textual data and explore new methods of improving market forecasting for financial participants.

Natural language processing (NLP), the combination of computational linguistics and machine learning, is a type of artificial intelligence technique that can assist in filling the gap between human language and computer understanding. NLP can not only automate the analysis of textual information, but also help with the enrichment of unstructured data, making it more actionable and searchable. Among the various applications of NLP, sentiment analysis is one of the most popular and widely used technology in the financial market to perform the analysis of textual data.

Sentiment analysis, also know as opinion mining, was firstly introduced by Nasukawa and Yi in 2003, is a sub-field of NLP that focuses on extracting people's opinions, attitudes, and sentimental polarities from textual data. The outcome of sentiment analysis typically contains three labels (i.e., POSITIVE, NEUTRAL, NEGATIVE).

Beyond the traditional reliance on quantitative data in the financial market, for example, using financial ratios, technical indicators, historical prices and graphs, sentiment analysis can provide a new perspective on the market by analysing the qualitative data, such as news articles, social media posts, and earnings call transcripts to predict the future movement of financial markets.

Submitted to UCL Advanced Machine Learning in Finance (COMP0162) Project Report. Do not distribute.

Besides, one should be aware that sentiment analysis is not a new concept in the financial market. In fact, it has been used for many years by professional traders and financial analysts to make investment decisions. However, with the recent advancements in deep learning, it is now becoming more promising to build accurate and robust sentiment analysis models with deep neural networks that can outperform traditional sentiment analysis models, such as lexicon-based methods. As shown in the figure 1.1 below, lexicon-based model, also called rule-based approach, is typically based on pre-defined sentiment lexicons (e.g., dictionaries) through manually labelling words with sentiment polarities. It is therefore limited by the capacity of the lexicon and human bias when labelling words. This labour-intensive process can be time-consuming and costly, and it is now not used as widely as before. Whereas, automatic sentiment analysis approaches, such as machine learning and deep learning models, can learn the sentiment polarity of words from the training data and make predictions on unseen data with high classification accuracy. Therefore, in this project, **the aim is to build multiple deep learning models to implement sentiment analysis on financial news headlines and compare their performance with baseline and null models**. The corresponding objectives of this project are shown as follows: (1) Data Preprocessing - Perform data preprocessing on the Financial PhraseBank dataset, including text cleaning, tokenization, and padding. (2) Null Model - Utilise the majority class as the null model to predict the sentiment polarity of financial news headlines. (3) Baseline Model - Build baseline models using logistic regression and decision trees to perform sentiment analysis on financial news headlines. (4) Deep Learning Models - Build deep learning models using simple feedforward neural networks (FNN), convolutional neural networks (CNN) and long short-term memory (LSTM) to implement financial sentiment analysis. (5) BERT Transformer Model - Apply a pre-trained BERT transformer model to evaluate its performance on financial sentiment analysis, and then investigate how much improvement it can bring to the model in terms of accuracy and efficiency. (6) Model Evaluation - Evaluate the performance of all models using confusion matrix, classification report (including precision, recall, F1-score), and ROC-AUC curve.

Hence, this project report is structured as follows: Section 1 provides introductions and literature reviews. Then, section 2 firstly describes the methodologies of null model, baseline models, deep learning models and BERT transformer model. Afterwards, it shows the data preprocessing and exploratory data analysis process. Next, section 2 ends with explanations of the evaluation metrics (e.g., classification report, confusion matrix and ROC-AUC curve) utilised for evaluating the performance of models. After that, section 3 demonstrates experimental setup, model training and results evaluation with the use of performance metrics mentioned before. Finally, section 4 concludes the project with a summary of findings, limitations and future works.

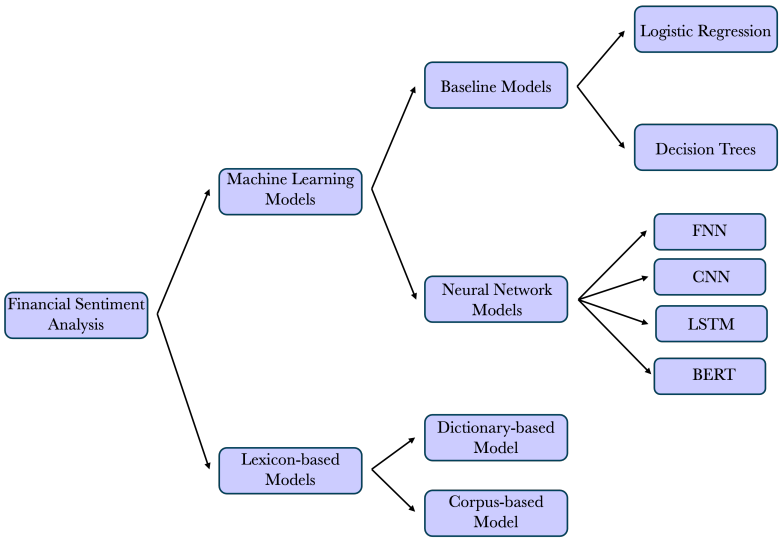

Figure 1.1: Financial Sentiment Analysis Models

1.2 Literature Reviews

In this part, several papers are reviewed to provide a comprehensive understandings of the sentiment analysis on financial news using different deep learning models. The papers are selected based on their relevance to this project's objectives and the methodologies. The papers are therefore summarised as follows:

Sohangir's paper [3] mainly focuses on the analysis of stock market opinions posted on StockTwits with the use of long short-term memory, doc2vec and CNN to evaluate the efficiency of sentiment predictions based on the StockTwits dataset, where CNN model achieves 90.93% of prediction accuracy. In comparison, the LSTM model leads to an accuracy of 69.23%, which indicates that the LSTM architecture used in the paper can still be improved as it has less capability of predicting sentimental information based on the StockTwits dataset. However, this research paves the way for further exploration into big-data and sentimental analytics in finance sector, especially with the use of deep learning models.

Memis and Yeniad [1] provide a comprehensive studies of sentiment analysis on financial tweets with deep learning methods. They utilise a combined GRU-CNN (Gated Recurrent Units and Convolutional Neural Network) model to perform sentiment analysis. They also uniquely use pre-trained word embeddings from fastText to accelerate the sentiment classifications, especially focusing on under-researched Turkish financial tweets dataset. Their model achieves 72.72% of prediction accuracy. Besides, they has acknowledged the limitations of modest size of the dataset and technical difficulties of implementing sentiment analysis of financial tweets in Turkish (given Turkish is a highly inflectional language).

Souma, Vodenska, and Aoyama [4] presents an innovative study to analyse the predictability of financial news sentiments with the use of Thomson Reuters News Archive and Dow Jones Industrial Average (DJIA) 30 Index high-frequency price data from 2003 to 2013. They leverage a combination of recurrent neural network (RNN) with long short-term memory (LSTM) units with a training dataset on news archive from 2003 to 2012. Then, they implement test validations with data on 2013. Besides, their paper employ GloVe embedding methods to vectorize the textual input data. They also suggest that different word embedding methods and polarity definitions could be explored in the future to further improve their prediction accuracy.

Shuhidan [2] investigates the application of machine learning models for sentiment analysis on Malaysian financial news headlines. The key objective of paper is to explore the impact of prevailing attitudes and opinions expressed in financial news on decision-making process. The study utilises two main methodologies: Opinion Lexicon-based algorithm and Naïve Bayes algorithm. Besides, this paper analyses 536 financial headlines from the New Straits Times over a 12-month period. It provides a new perspective of using Bayesian probabilistic model for classification of financial sentiment analysis, which achieves high efficiency in classifying sentiments by calculating the probability of headlines belonging to either category.

Yadav [6] explores the method of using unsupervised sentiment analysis techniques on financial sentiment analysis. This paper aims to find the correlation between the stock market and the sentiments extracted from the financial news headlines. The study conducts experiments on a dataset of dataset of financial news articles related to Tata Consultancy Services (TCS). It utilises three algorithms: Turney's original pattern (an accuracy of 75.00% and an F-score of 84.47%), a hybrid seed-set approach (an accuracy of 76.00% and an F-score of 84.81%), and a noun-verb combination approach (an accuracy of 79.00% and an F-score of 86.45%). This paper concludes that the noun-verb algorithm is proven to be most effective, indicating that traditional adjective-centric indicators are therefore not efficient enough for sentiment in financial news.

Shazmeen [5] presents a detailed investigation into the sentiment analysis of financial texts using supervised learning methods in in the thesis. This study mainly focuses on the exploration of BERT and FinBERT models for financial sentiment analysis classification efficiency with the use of Modular Finance (MFN) and combined with price indices from Bloomberg dataset. The outcome shows that FinBERT significantly outperforms both Naive Bayes and BERT models in sentiment prediction accuracy. Therefore, this thesis concludes that transfer learning models, particularly FinBERT, are highly promising in sentiment analysis of financial news, outperforming traditional machine learning models like Naive Bayes and BERT classifiers.

2 Methodology

In this section, methodologies (including mathematical formulas and equations) of the null model, baseline models, deep learning models and BERT transformer model are described in detail. Subsequently, the exploratory data analysis part (including dataset cleaning and text pre-processing) is also descried in detail at the end of this section.

Firstly, in this project, the null model is used as the reference to predict the sentiment polarity of financial news headlines with simply the most frequent class. It is therefore considered as the simplest model to compare the performance of other models. Secondly, the baseline models here are referred to as those traditional machine learning models with less complicated architectures. In this report, they includes logistic regression and decision trees models. Next, more complex deep learning models include feedforward neural networks (FNN), convolutional neural networks (CNN) and long short-term memory (LSTM) are demonstrated. Finally, the BERT model is therefore the most advanced deep learning model used in this project, which is a pre-trained transformer that can be fine-tuned on the Financial PhraseBank dataset to perform sentiment analysis on financial news headlines. Hence, BERT is a powerful model that can capture the complex context of words in a sentence and learn the sentiment polarity of words from the training data. It is therefore expected to outperform the baseline models in terms of accuracy and efficiency.

2.1 Null Model

For the null model under the context of financial news sentiment analysis, there's no specific equation, as it simply makes predictions based on the most frequent class. It is the simplest model for classification tasks in this project. Therefore there is no correlation between input features and the target variables when apply the most common class. For example, in this project, if $C = \{\text{positive}, \text{neutral}, \text{negative}\}$ and $f(c)$ represents the frequency of class c in the training dataset, then the predictions of null model is shown in equation 2.1 below:

$$\hat{y} = \underset{c \in C}{\operatorname{argmax}}\, f(c) \tag{2.1}$$

2.2 Baseline Models

2.2.1 Logistic Regression

In this project, a logistic regression classifier model is utilised for the sentiment prediction on the PyTorch platform. It works by simply fitting a logistic function to the input data, and then predicting the probability that an instance belongs to a particular class. The corresponding logic of calculating the probability under a given text input x belongs to a class (e.g., sentiment labels: positive = 2, neutral = 1, negative = 0) can be concluded as equation 2.2:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x} + b)}} \tag{2.2}$$

where x is the feature vector (i.e., TF-IDF values, will be explained in the section 2.4.1 below), w is the weight vector, b is the bias, and $y = 1$ represents for the neutral class. Besides, for multi-class (class greater than 3) classification tasks, it can be extended by using one-vs-rest(OvR) techniques.

2.2.2 Decision Trees

When it comes decision tree, it is a simple machine learning model which can use the feature thresholds to split the dataset into subsets recursively based on features that lead to the largest information gain. For example, under the context of financial sentiment classification, a decision criterion can be the TF-IDF scores mentioned above. Besides, at each node n, the splitting decision is based on the Gini impurity I_G, or entropy H, aiming to maximise the class purity in its child nodes. **Gini Impurity** is used to compute the probability of a randomly chosen element being incorrectly classified 2.3:

$$I_G(n) = 1 - \sum_{i=1}^{J} p_i^2 \tag{2.3}$$

Entropy measures the impurity in a group of examples 2.4:

$$H(n) = -\sum_{i=1}^{J} p_i \log_2 p_i \tag{2.4}$$

Information Gain is then used to decide which feature to split on, calculated as the difference in impurity before and after the split. For a dataset D, split on feature A into subsets $D_1, D_2, ..., D_k$ 2.5:

$$\text{Gain}(D, A) = H(D) - \sum_{m=1}^{k} \left(\frac{|D_m|}{|D|} H(D_m) \right) \tag{2.5}$$

2.3 Neural Network Models

2.3.1 Feedforward Neural Network (FNN)

Feedforward Neural Networks process input data through one or more hidden layers, each composed of nodes or neurons, to predict the output. Given an input vector x, the output $a^{(l)}$ for the l^{th} layer is calculated as 2.6:

$$a^{(l)} = g^{(l)}(W^{(l)} a^{(l-1)} + b^{(l)}) \tag{2.6}$$

where $W^{(l)}$ and $b^{(l)}$ are the weights and biases for the l^{th} layer, $g^{(l)}$ is the activation function (e.g., ReLU or sigmoid), and $a^{(0)} = x$ is the input layer.

2.3.2 Convolutional Neural Network (CNN)

For Convolution Neural Network (CNN) is primarily designed to implement computer visions and image processing tasks. However, for financial textual data classification, it needs to apply for a filter F to a window of k words to generate a feature map. Besides, a 1D convolution operation is applied across the sequence of words. The convolution for a filter F at position i along the input sequence x is 2.7:

$$o_i = g \left(\sum_{j=0}^{k-1} F_j \cdot x_{i+j} + b \right) \tag{2.7}$$

where k is the size of the filter, g is an activation function, and b is the bias term. This operation is applied for each position, capturing local dependencies in the data.

2.3.3 Long Short-Term Memory (LSTM)

LSTM architecture is a special variant of recurrent neural network (RNN) capable of learning long-term dependencies. It allows the model to remember or forget information over the input texts, particularly useful for understanding the textual data that depend on earlier appearing context in the news article. They process sequences by maintaining a cell state and applying several gates 2.13:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.8}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.9}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.10}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.11}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.12}$$
$$h_t = o_t * \tanh(C_t) \tag{2.13}$$

where σ is the sigmoid function, and $*$ denotes element-wise multiplication. These equations govern the forget gate f_t, input gate i_t, output gate o_t, cell state update $\tilde{C}_t$, and hidden state h_t, respectively.

2.3.4 BERT Transformer

BERT leverages the Transformer architecture, focusing on an attention mechanism to understand the context among words or sub-words in text. Unlike the full Transformer that uses both encoder

and decoder, BERT employs only the encoder for creating a language model. This bi-directional approach to pre-training on extensive unlabeled text data helps BERT to effectively learn language representations. These representations are then fine-tuned for specific tasks, leading to significant performance improvements in various NLP application. The success of BERT is mainly due to its bidirectional nature, innovative pre-training methods like Masked Language Model and Next Sentence Prediction, along with substantial data and computational resources from Google. The core of BERT is the self-attention mechanism, which for a single attention head is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.14}$$

where Q, K, and V represent the query, key, and value matrices derived from the input embeddings, and d_k is the dimensionality of the key vectors. It is noteworthy that this is simply a brief introduction of BERT model, detailed exploration of BERT can be found through Google Brain' s paper.

2.4 Exploratory Dataset Analysis (EDA)

2.4.1 Feature Extraction

The process of converting pre-processed data into numerical vectors is referred to as feature extraction. The widely used approach within feature extraction is called TF-IDF.

TF-IDF represents the term frequency–inverse document frequency, which is a commonly used feature extraction method for converting textual data into numerical values. Term Frequency (TF) indicates the scoring of the frequency of the word/term in the current document. It provides the frequency of the word/term each document in the corpus. TF is therefore defined as follows 2.15:

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.15}$$

where $f_{t,d}$ is the frequency of term t in document d, and the denominator is the sum of frequencies of all terms in the document. Whereas, Inverse Document Frequency (IDF) stands for the scoring of how rare the word is across the documents. It measures the importance of a word/term within the entire corpus. IDF is therefore defined as 2.16:

$$\text{IDF}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \tag{2.16}$$

where N is the total number of documents in the corpus D, and $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears. By combining TF and IDF together, terms with high TF-IDF score indicates high representation of a specific document in the corpus, making them useful in the financial news sentiment classification(e.g., distinguishing between positive, negative, or neutral sentiments) 2.17:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \tag{2.17}$$

2.4.2 Data Loading, Cleaning and Preprocessing

In this project, the dataset used for financial sentiment analysis is loaded from the Financial Phrase-Bank dataset. It contains 4840 English language financial news headlines from the perspective of a retail investor. The dataset has two columns: "Sentiment" and "text", labelled with three sentiment polarities (i.e., POSITIVE, NEGATIVE, NEUTRAL).

Then, data cleaning and text preprocessing procedures are implemented to clean and prepare the original dataset. As shown in the algorithm 2.1 below, data cleaning aims to prepare the data without irrelevant information by removing duplicate entries. Then, lower-casing, punctuation removal, stop-words removal, tokenization and stemming are implemented to transform raw textual data to more operational formats for machine learning models by removing noise and reducing complexity. Next, after implementing data pre-processing procedures, a plot of dataset visualisation can be generated in the figure 2.1 below. It contains a word cloud (demonstrate the most frequent words), sentiment count and proportions of this dataset. Finally, this pre-processed dataset is split into 80% of training and 20% of testing for the models in the following sections to train and test respectively.

Algorithm 2.1 Data Cleaning and Text Preprocessing Pseudocode

procedure CLEANDATAANDPREPROCESSTEXT(*documents*)
  for each *document* **in** *documents* **do**
    document ← DATACLEAN(*document*)
    document ← LOWERCASE(*document*)
    document ← REMOVEPUNCTUATION(*document*)
    tokens ← TOKENIZE(*document*)
    tokens ← REMOVESTOPWORDS(*tokens*)
    tokens ← STEMWORDS(*tokens*)
  end for
  return *documents*
end procedure

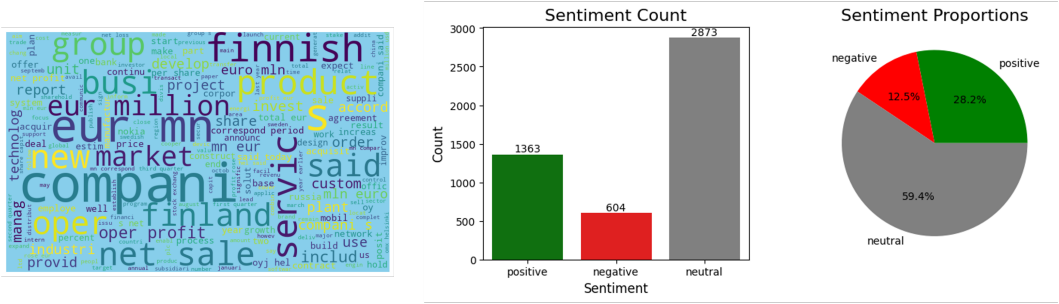

Figure 2.1: Visualisation of Dataset

2.5 Evaluation Metrics

Now, it is important to introduce the relevant performance metrics (including classification report, confusion matrix and ROC-AUC curve) for readers to understand how the models are evaluated and compared with each other before discussing the experiments and results in this section.

2.5.1 Classification Report

The classification report provides a comprehensive evaluation for the performance of classification models in this project. It contains key metrics, such as accuracy, precision, recall (sensitivity) and F1-score.

Accuracy is defined as the ratio of correctly predicted classes (both true positives (TP) and true negatives (TN)) to the total observations in the dataset. It therefore measures the general accuracy of a classifier when it makes correct predictions across all classes. Its equation 2.18 is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.18}$$

where TP is the number of true positives (model correctly predicts the positive class), TN is the number of true negatives (model correctly predicts the negative class), FP is the number of false positives (model incorrectly predicts the positive class), and FN is the number of false negatives (model incorrectly predicts the negative class).

Precision is defined as the ratio of the number of true positives to the total number of positives 2.19:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.19}$$

Recall is then defined as the fraction of the number of true positives to the total number of true positives plus false negatives 2.20:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.20}$$

F1-score is therefore the harmonic mean of precision and recall such that the F1-scores of the best classification distribution is 1.0 and the worst is 0.0. It keeps a balanced evaluation for the classification tasks, and therefore becomes particularly useful for financial sentiment analysis evaluation in this project report2.21:

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.21}$$

2.5.2 Confusion Matrix

The confusion matrix is a powerful tool that can visually demonstrate the performance of classification algorithms and indicate which classes are confused by the models. It provides explicitly information of observations between true and predicted labels where the matrix's diagonal shows the number of correctly predicted classes and off-diagonal represents the the number of misclassifications.

2.5.3 ROC-AUC Curve

The Receiver Operating Characteristic (ROC) curve is a plot of true positive rate (TPR) versus the false positive rate (FPR) under different threshold levels. Whereas, the Area Under the Curve (AUC) is another metric in the ROC curve to evaluate the classification capability for a classifier where a single value of 1 represents the 'perfect' classifier and 0.5 means a random classifier. The higher the AUC value, the better the model's ability of predicting positive as positive and negative as negative. Besides, TPR and FPR are defined as:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{2.22}$$

$$\text{FPR} = \frac{FP}{TN + FP} \tag{2.23}$$

3 Experiments and Results

3.1 Null Model Result

As mentioned in section 2.1, null model simply predicts the most frequent class for the financial sentiment. Therefore, after implementing the experiments, the accuracy of correctly making predictions is **59.36%**. F1-score of the null model is **44.22%**. Besides, the ROC-AUC curve is not applicable for a null model. Since a null model does not make any probabilistic predictions which means that there is no threshold settings, true positive rates and false positive rate to plot a ROC curve.

3.2 Baseline Model Results

3.2.1 Logistic Regression

After fitting the logistic regression model with the training dataset, one can find its prediction results of financial sentiment classification in the figure 3.1 below. Three plots of classification report, confusion matrix and ROC-AUC curve are displayed from left to right respectively in this figure. Particularly, classification report and confusion matrix are visualised by heatmaps to clearly distinguish between each sentiment classes. It is therefore noteworthy that weighted average f1-score and accuracy are **0.74** and **0.76** respectively. Compared with the results of null model, logistic regression achieves relatively high prediction accuracy. The total number of correct predictions of sentiment classes is **732**. In addition, the AUC scores of all three classes are all above **0.85**, indicating a high classification performance when applying logistic regression to financial sentiment analysis.

3.2.2 Decision Tree

As for the outcomes of decision tree models, it can be displayed in the figure 3.2 below. Weighted average f1-score of **0.69** is obtained. According to its confusion matrix, the total number of correctly predicted classes by the decision tree model is **669** (i.e., negative: 56; neutral: 445; positive: 168). Whereas, in the ROC-AUC curve, the minimum AUC score is **0.7**, which is just above 0.5, which means that it is slightly better than random classification.

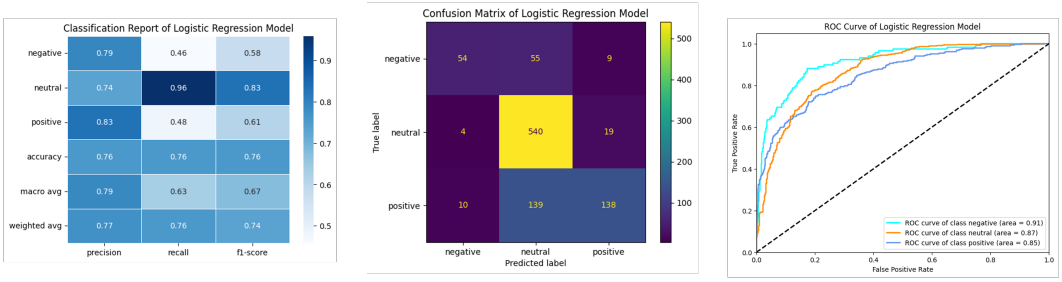

Figure 3.1: Logistic Regression Experimental Results

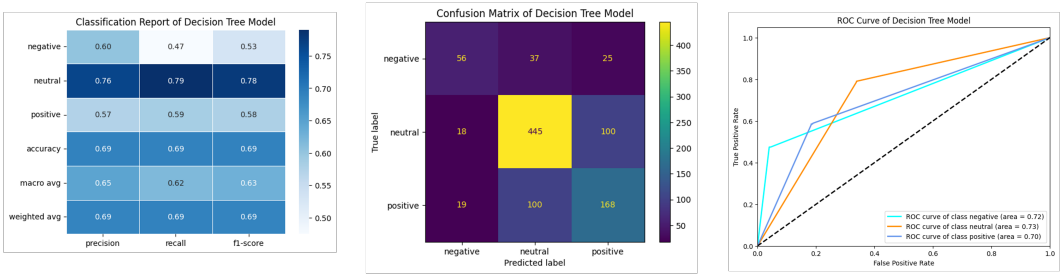

Figure 3.2: Decision Trees Experimental Results

3.3 Neural Network Model Results

3.3.1 FNN

When it comes to the main methodologies introduced in this report, the performance of neural network models depends on the complexity of each architecture. Starting from the feedforward neural network model, which has the simplest architecture. The structure of this applied FNN is: input of size 6337, two hidden layers (with 128 and 64 neurons, respectively), one dropout layer for regularization, and finally output of size 3, which indicates the classification for three classes.

Then, by fitting this model with the training data and making predictions on testing data on PyTorch, one can obtain the following results shown in figure 3.3 below. From the result of classification report of FNN, it achieves **0.71** in weighted-average f1-score. Its prediction accuracy is also **0.71**. Besides, the total number of correct predictions is **690**, which is slightly better than that of decision tree model. Next, a minimum AUC score of **0.80** is achieved.

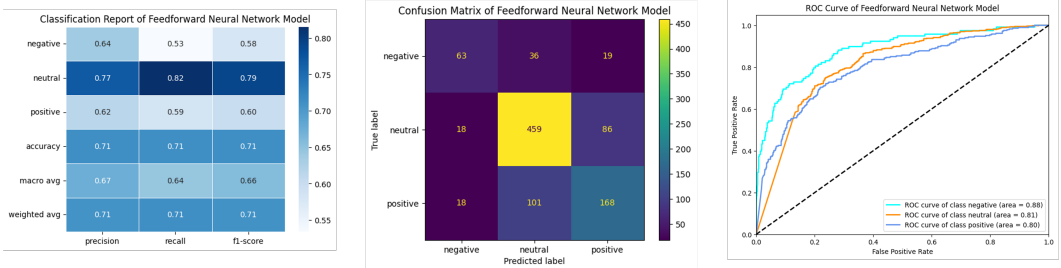

Figure 3.3: FNN Experimental Results

3.3.2 CNN

Subsequently, CNN model is implemented and evaluated its performance of sentimental classification on the testing dataset. This CNN model has a structure of: 1 embedding layer, 2 convolutional layers, 1 fully connected layer and 1 dropout layer. In particular, one embedding layer is used to transform input word index into dense vectors. Then, two convolutional layers are applied to filter the embedded inputs and extract features by capturing local dependencies with a sequence of 3 words each time. Finally, one fully connected layer and one dropout layer are utilised to make predictions and prevent overfitting respectively.

As illustrated in the figure 3.4 below, CNN only achieves **0.70** in f1-score and **0.71** in prediction accuracy. Then, a total number of **687** is displayed in the confusion matrix. Finally, a **0.79** of AUC score is obtained in the ROC curve. Compared with the results in FNN above, CNN does not provide significant improvements in terms of prediction accuracy and f1-score. Considering that its main strength relies on visual computing and image processing in the field of computer visions, CNN may not be suitable for textual data classification tasks.

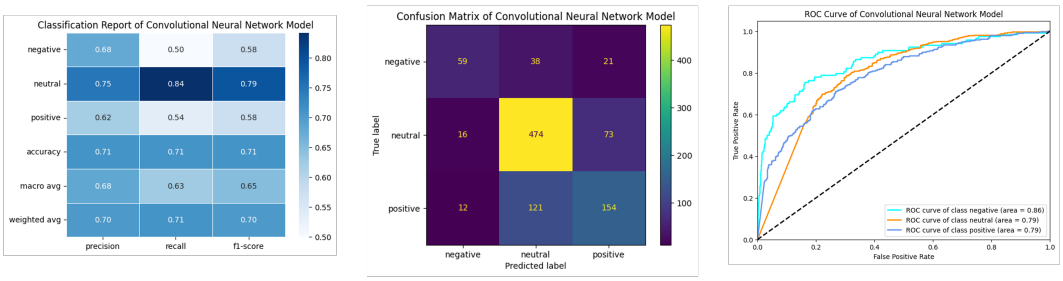

Figure 3.4: CNN Experimental Results

3.3.3 LSTM

The architecture of LSTM applied for this project can be summarised as: 1 embedding layer, 1 bidirectional LSTM layer, 1 self-attention layer, 1 fully-connected layer and 2 dropout layer. The embedding layer of this LSTM is used to extract semantic information of words by converting token index to dense vectors. It has a vocabulary size of 6356 (the number of unique tokens) with output dimension of 128 to represent 812568 (i.e., 6356×128) vectors. As for the LSTM layer, it is utilised to bidirectionally process sequences of embedded vectors to extract past (backward) and future (forward) contexts within the sequence. This LSTM layer also has an input size of 128 which matches the output of embedded layer. More importantly, a self-attention layer is used in this LSTM model for the project to include weighted-words in sequence and focus the model's attention more on the informative data. It has an input size of 128 from the output of LSTM layer. Finally, a fully-connected layer and a dropout layer are used to map the weighted features to the output classes and help prevent overfitting respectively.

After fitting this LSTM model with training data and making prediction on the testing data, one can receive the following results on figure 3.5 below. As shown in the figure, both f1-score and accuracy achieve **0.75** on classification predictions. Besides, the total correctly predicted number in the confusion matrix of LSTM model is **729**, which is far more than that of CNN model in figure 3.4 above. Then, an average AUC score of **0.86** is obtained from the ROC curve of LSTM model. These results indicate that the LSTM architecture used in this project successfully outperforms the FNN and CNN models, showing good potentials for future research.

3.3.4 BERT

Finally, the BERT model is implemented as the most complicated architecture within this project to evaluate its performance. The exact version of BERT used in the experiment is called BertForSequenceClassification model (i.e., use a BERT model as the backbone and add a classification layer on top), which is pre-trained by a large corpus of texts and then fine-tuned for classification tasks with three output classes. Specifically, the embedding layer inside the BERT uses a vocabulary of

30,522 tokens and outputs 768 word embeddings. Besides, the BERTEncoder is composed of 12 identical layers of BERTLayer to carry out multi-head self-attention mechanism and feed-forward networks. It is noteworthy that each BertLayer in the encoder consists of a BertAttention module (deal with self-attention mechanism) and a BertIntermediate module (transform the output of the attention mechanism).

Experimental results of this BERT model is illustrated in figure 3.6 below. It can be noticed that **0.8** of f1-score is achieved in the classification report. Then, the total number of correct predictions increases to **778** for BERT model in the confusion matrix. Besides, the average AUC score in the ROC curve is **0.91** which is the top score achieved among this project. Therefore, with the use of contextual embeddings and attention mechanism, this BERT architecture is a powerful tool for understanding complex language patterns, which makes it significantly useful for solving financial sentiment classification tasks.

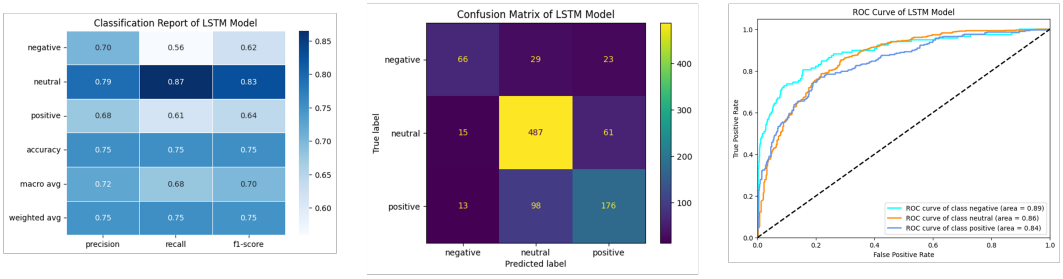

Figure 3.5: LSTM Experimental Results

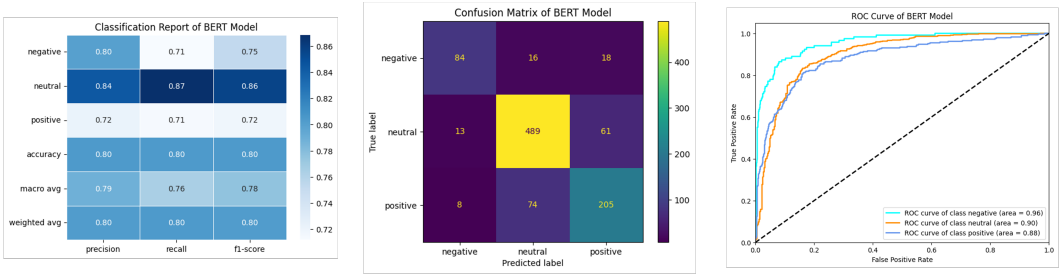

Figure 3.6: BERT Experimental Results

4 Conclusions

4.1 Achievements

Finally, a summary including f1-score, prediction accuracy and AUC score has been generated in figure 4.1 below to compare the performance of each model. The figure of f1-score has been particularly emphasized as it is a balanced figure of merit for evaluating the model. As shown in the figure, BERT is undoubtedly the best model with the top classification performance of **0.80** in f1-score. Then, LSTM is the second best model with **0.75** f1-score. In addition, logistic regression models achieves **0.74** in f1-score which is higher than FNN (**0.71**) and CNN (**0.70**)models. This reveals that a simpler machine learning model can still achieve better classification performance especially when the dataset is small (e.g., only 4840 data points in this case). Furthermore, when applying deep learning models to solve simple three-classes classification tasks, it does not guarantee better performance than traditional machine learning models. This is because deep learning models are more complex and require more data to train effectively. In this case, the deep learning models may not have enough data to learn the complex relationships in the data, which results in lower classification performance compared to the logistic regression model. Afterwards, decision tree

model achieves **0.69** in f1-score, which is slightly worse than FNN and CNN. However, all machine learning models used in this project are significantly better than the performance of null model (only achieves **0.44** in f1-score). This indicates that machine learning algorithms can effectively learn the sentiment polarity of financial news headlines and make accurate predictions on unseen data. The results demonstrate that machine learning models can be used to perform sentiment analysis on financial news headlines with high accuracy and efficiency.

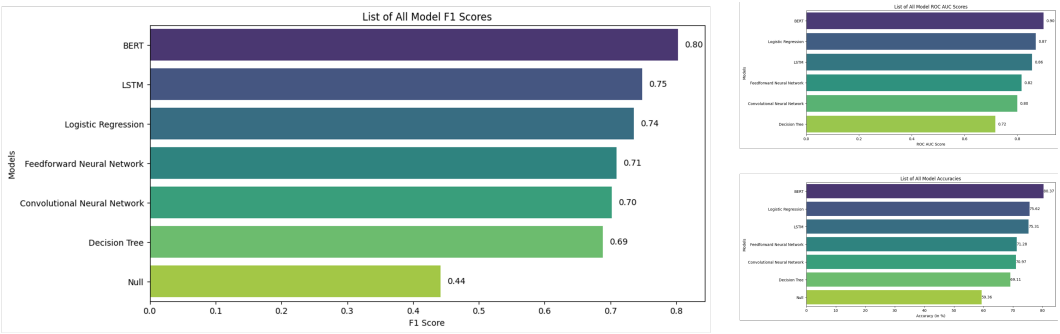

Figure 4.1: A Summary of All Models' Performance

4.2 Limitations and Further Improvements

When it comes to limitations, the author of this report would acknowledge that the design of neural networks for FNN and CNN is based on previous project experience and may not fine-tune their specific type and number of layers in the architecture to properly adapt them into this financial sentiment analysis tasks. Therefore, the first limitation of this project comes from the design of neural network model architectures, namely, feedforward neural network and convolutional neural network can be further fine-tuned to improve the performance in f1-score and accuracy.

Secondly, the training process of BERT transformer requires high GPU computational resources and long computation time, which makes it difficult to apply more advanced version of BERT, such as FinBERT, which is currently state-of-the-art approach in analysing financial sentiment texts. Hence, if more computational resource is allowed in the future, FinBERT is the next goal of experiment. Hopefully, it could achieve even better results than the basic version of BERT used in this project.

References

[1] Erkut Memiş, Hilal Akarkamçı (Kaya), Mustafa Yeniad, Javad Rahebi, and Jose Manuel Lopez-Guede. Comparative study for sentiment analysis of financial tweets with deep learning methods. *Applied Sciences*, 14:588, 1 2024.

[2] Shuhaida Mohamed Shuhidan, Saidatul Rahah Hamidi, Soheil Kazemian, Shamila Mohamed Shuhidan, and Maizatul Akmar Ismail. Sentiment analysis for financial news headlines using machine learning algorithm. volume 739, pages 64–72. Springer Verlag, 2018.

[3] Sahar Sohangir, Dingding Wang, Anna Pomeranets, and Taghi M. Khoshgoftaar. Big data: Deep learning for financial sentiment analysis. *Journal of Big Data*, 5, 12 2018.

[4] Wataru Souma, Irena Vodenska, and Hideaki Aoyama. Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*, 2:33–46, 1 2019.

[5] Farha Shazmeen Syeda. Sentiment analysis of financial news with supervised learning, 2020. Dissertation.

[6] Anita Yadav, C. K. Jha, Aditi Sharan, and Vikrant Vaish. Sentiment analysis of financial news using unsupervised approach. volume 167, pages 589–598. Elsevier B.V., 2020.