
NLP: Sentiment Analysis of Financial News using Deep Learning

Anonymous Author
University College London
London WC1E 6BT

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 known 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.

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

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

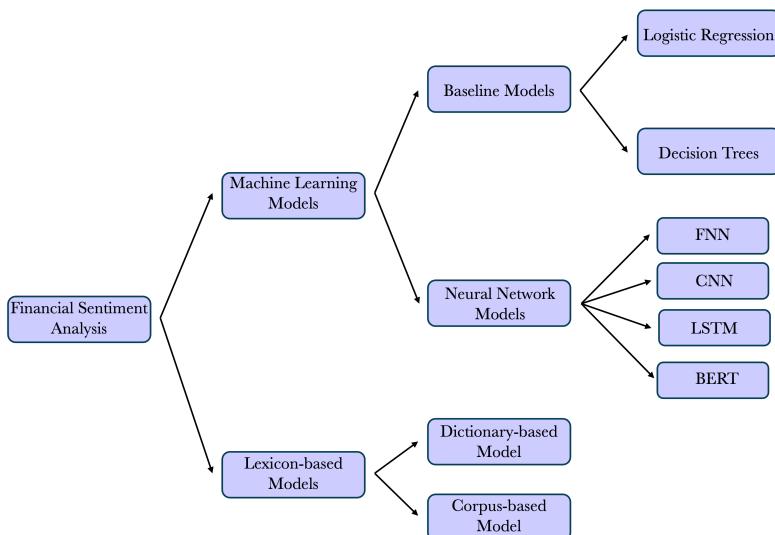


Figure 1.1: Financial Sentiment Analysis Models

70 **1.2 Literature Reviews**

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

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

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

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

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

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

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

124 **2 Methodology**

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

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

141 **2.1 Null Model**

142 For the null model under the context of financial news sentiment analysis, there's no specific equation,
143 as it simply makes predictions based on the most frequent class. It is the simplest model for
144 classification tasks in this project. Therefore there is no correlation between input features and
145 the target variables when apply the most common class. For example, in this project, if $C =$
146 $\{\text{positive}, \text{neutral}, \text{negative}\}$ and $f(c)$ represents the frequency of class c in the training dataset, then
147 the predictions of null model is shown in equation 2.1 below:

$$\hat{y} = \operatorname{argmax}_{c \in C} f(c) \quad (2.1)$$

148 **2.2 Baseline Models**

149 **2.2.1 Logistic Regression**

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

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

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

158 **2.2.2 Decision Trees**

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

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

166 **Entropy** measures the impurity in a group of examples 2.4:

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

167 **Information Gain** is then used to decide which feature to split on, calculated as the difference in
168 impurity before and after the split. For a dataset D , split on feature A into subsets D_1, D_2, \dots, D_k
169 2.5:

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

170 2.3 Neural Network Models

171 2.3.1 Feedforward Neural Network (FNN)

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

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

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

177 2.3.2 Convolutional Neural Network (CNN)

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

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

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

185 2.3.3 Long Short-Term Memory (LSTM)

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

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.8)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.9)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.10)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.11)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.12)$$

$$h_t = o_t * \tanh(C_t) \quad (2.13)$$

190 where σ is the sigmoid function, and $*$ denotes element-wise multiplication. These equations govern
191 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.

192 2.3.4 BERT Transformer

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

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

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

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

205 2.4 Exploratory Dataset Analysis (EDA)

206 2.4.1 Feature Extraction

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

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

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

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

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

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

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

222 2.4.2 Data Loading, Cleaning and Preprocessing

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

227 Then, data cleaning and text preprocessing procedures are implemented to clean and prepare the
 228 original dataset. As shown in the algorithm 2.1 below, data cleaning aims to prepare the data without
 229 irrelevant information by removing duplicate entries. Then, lower-casing, punctuation removal, stop-
 230 words removal, tokenization and stemming are implemented to transform raw textual data to more
 231 operational formats for machine learning models by removing noise and reducing complexity. Next,
 232 after implementing data pre-processing procedures, a plot of dataset visualisation can be generated
 233 in the figure 2.1 below. It contains a word cloud (demonstrate the most frequent words), sentiment
 234 count and proportions of this dataset. Finally, this pre-processed dataset is split into 80% of training
 235 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

```
1: procedure CLEANDATAANDPREPROCESSTEXT(documents)
2:   for each document in documents do
3:     document  $\leftarrow$  DATACLEAN(document)
4:     document  $\leftarrow$  LOWERCASE(document)
5:     document  $\leftarrow$  REMOVEPUNCTUATION(document)
6:     tokens  $\leftarrow$  TOKENIZE(document)
7:     tokens  $\leftarrow$  REMOVESTOPWORDS(tokens)
8:     tokens  $\leftarrow$  STEMWORDS(tokens)
9:   end for
10:  return documents
11: end procedure
```

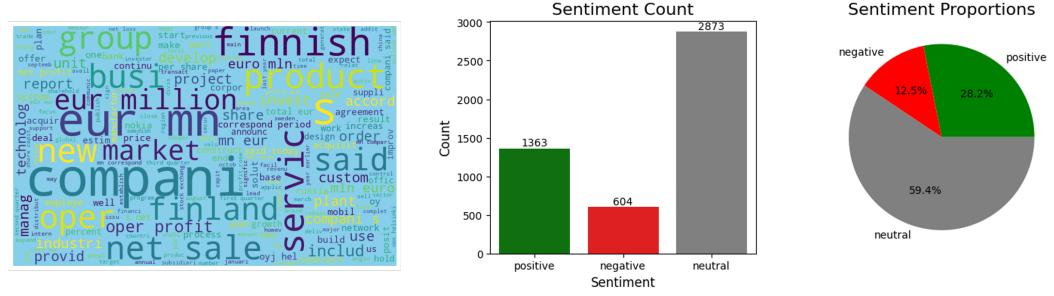


Figure 2.1: Visualisation of Dataset

236 **2.5 Evaluation Metrics**

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

240 **2.5.1 Classification Report**

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

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

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.18)$$

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

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

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.19)$$

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

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.20)$$

254 F1-score is therefore the harmonic mean of precision and recall such that the F1-scores of the best
255 classification distribution is 1.0 and the worst is 0.0. It keeps a balanced evaluation for the classifica-
256 tion tasks, and therefore becomes particularly useful for financial sentiment analysis evaluation in
257 this project report^{2.21}:

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

258 2.5.2 Confusion Matrix

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

263 2.5.3 ROC-AUC Curve

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

$$\text{TPR} = \frac{TP}{TP + FN} \quad (2.22)$$

$$\text{FPR} = \frac{FP}{TN + FP} \quad (2.23)$$

270 3 Experiments and Results

271 3.1 Null Model Result

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

277 3.2 Baseline Model Results

278 3.2.1 Logistic Regression

279 After fitting the logistic regression model with the training dataset, one can find its prediction results of
280 financial sentiment classification in the figure 3.1 below. Three plots of classification report, confu-
281 sion matrix and ROC-AUC curve are displayed from left to right respectively in this figure. Particularly,
282 classification report and confusion matrix are visualised by heatmaps to clearly distinguish between
283 each sentiment classes. It is therefore noteworthy that weighted average f1-score and accuracy are
284 **0.74** and **0.76** respectively. Compared with the results of null model, logistic regression achieves
285 relatively high prediction accuracy. The total number of correct predictions of sentiment classes is
286 **732**. In addition, the AUC scores of all three classes are all above **0.85**, indicating a high classification
287 performance when applying logistic regression to financial sentiment analysis.

288 3.2.2 Decision Tree

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

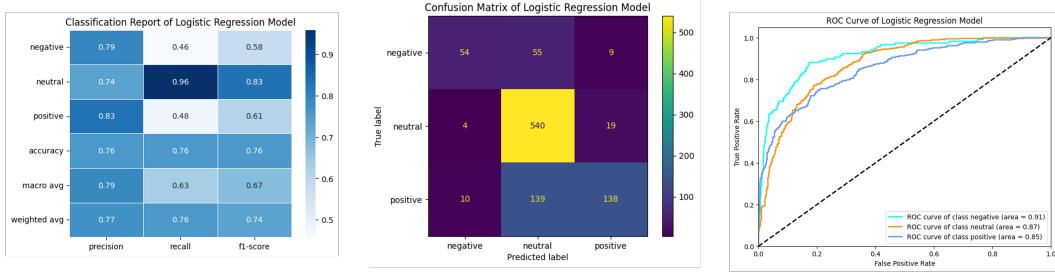


Figure 3.1: Logistic Regression Experimental Results

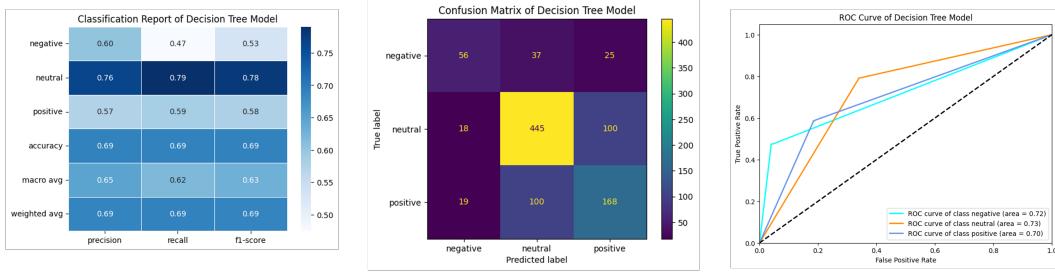


Figure 3.2: Decision Trees Experimental Results

294 3.3 Neural Network Model Results

295 3.3.1 FNN

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

301 Then, by fitting this model with the training data and making predictions on testing data on PyTorch,
 302 one can obtain the following results shown in figure 3.3 below. From the result of classification report
 303 of FNN, it achieves **0.71** in weighted-average f1-score. Its prediction accuracy is also **0.71**. Besides,
 304 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.

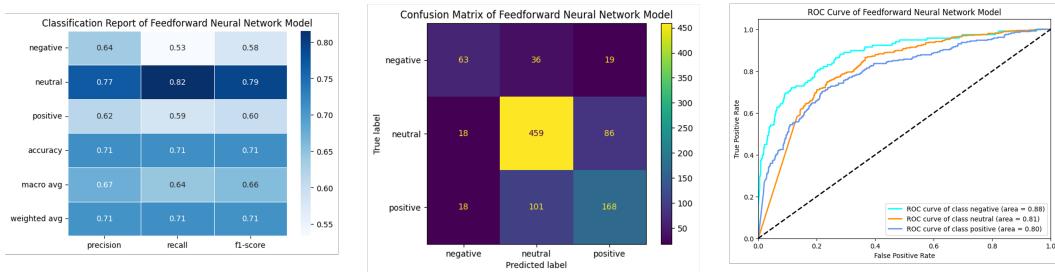


Figure 3.3: FNN Experimental Results

306 **3.3.2 CNN**

307 Subsequently, CNN model is implemented and evaluated its performance of sentimental classification
 308 on the testing dataset. This CNN model has a structure of: 1 embedding layer, 2 convolutional layers,
 309 1 fully connected layer and 1 dropout layer. In particular, one embedding layer is used to transform
 310 input word index into dense vectors. Then, two convolutional layers are applied to filter the embedded
 311 inputs and extract features by capturing local dependencies with a sequence of 3 words each time.
 312 Finally, one fully connected layer and one dropout layer are utilised to make predictions and prevent
 313 overfitting respectively.

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

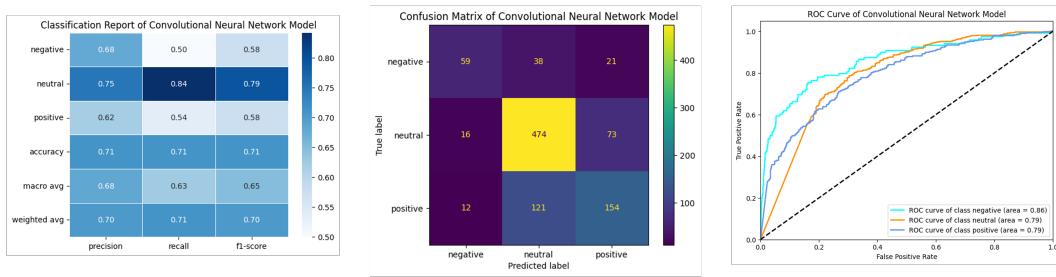


Figure 3.4: CNN Experimental Results

319

320 **3.3.3 LSTM**

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

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

340 **3.3.4 BERT**

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

346 30,522 tokens and outputs 768 word embeddings. Besides, the BERTEncoder is composed of 12
 347 identical layers of BERTLayer to carry out multi-head self-attention mechanism and feed-forward
 348 networks. It is noteworthy that each BertLayer in the encoder consists of a BertAttention module
 349 (deal with self-attention mechanism) and a BertIntermediate module (transform the output of the
 350 attention mechanism).

351 Experimental results of this BERT model is illustrated in figure 3.6 below. It can be noticed that
 352 **0.8** of f1-score is achieved in the classification report. Then, the total number of correct predictions
 353 increases to **778** for BERT model in the confusion matrix. Besides, the average AUC score in the
 354 ROC curve is **0.91** which is the top score achieved among this project. Therefore, with the use
 355 of contextual embeddings and attention mechanism, this BERT architecture is a powerful tool for
 356 understanding complex language patterns, which makes it significantly useful for solving financial
 sentiment classification tasks.

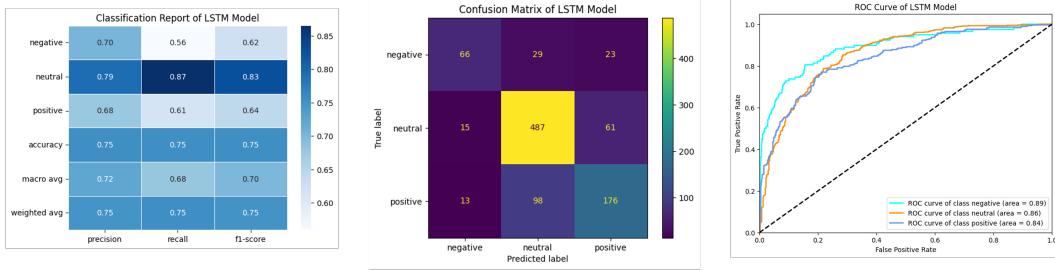


Figure 3.5: LSTM Experimental Results

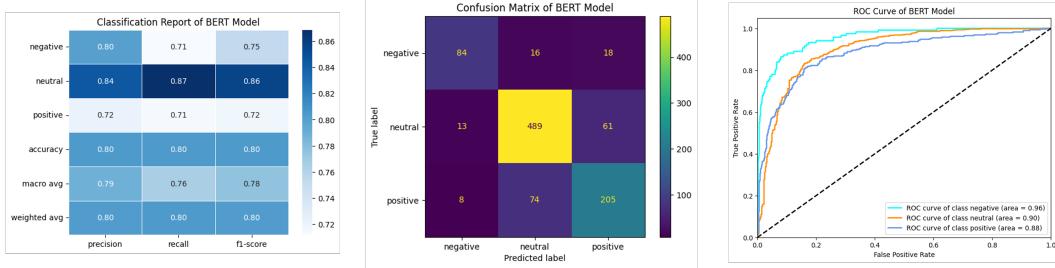


Figure 3.6: BERT Experimental Results

357

358 4 Conclusions

359 4.1 Achievements

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

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

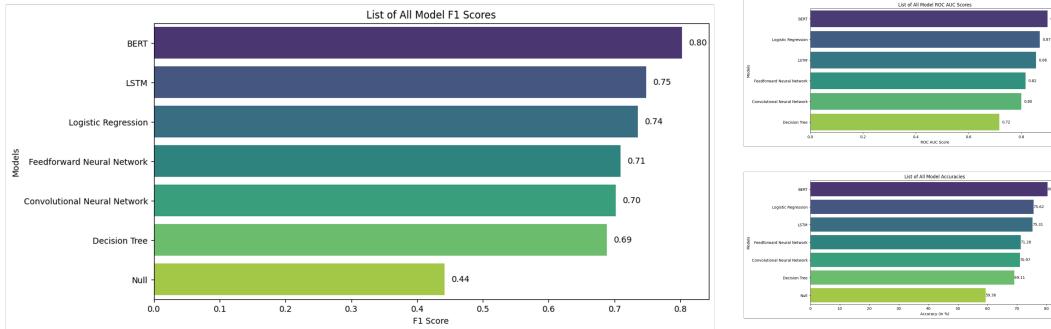


Figure 4.1: A Summary of All Models' Performance

378

379 4.2 Limitations and Further Improvements

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

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

391 References

- 392 [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.
- 395 [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.
- 398 [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.
- 400 [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.
- 402 [5] Farha Shazmeen Syeda. Sentiment analysis of financial news with supervised learning, 2020. Dissertation.
- 404 [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.