CIS600 Applied Natural Language Processing

Spring 2024 Term Project Report

"**Financial News Sentiment Analysis**"

Team Members

Aswin Tony Kanikairaj (SUID: 594864296)
Bhumika Dasari (SUID: 550775054)
Nagesh Nayak (SUID: 432989544)
Kratik Pal (SUID: 325707731)
Atharvaa Rane (SUID: 275976322)

1. **Introduction**

1.1 Objective

The goal of this project is to analyse financial news stories to get insights on market sentiment. This report is intended to give relevant information to investors, financial analysts, and other stakeholders.

The project's originality stems from the use of a diverse set of models, including CNN, RNN, LSTM, Naive Bayes, and Random Forest. Many research used only one or two models, resulting in lower accuracy than we achieved in our effort.

1.2 Scope

The project involves extracting and analysing financial news stories to categorise them based on mood. The focus is on financial news, with a special emphasis on blockchain-related news.

1.3 Background

In the financial industry, news items have a considerable impact on investment decisions and market movements. Analysing the emotion in these articles allows one to determine the overall mood of the market and identify prospective trends.

2. **Data Collection**

2.1 Data Sources

The research comprises of two primary datasets for analysing market sentiment. The first, 'all-data.csv', contains a wide range of financial news stories labelled "neutral," "negative," or "positive," whereas the second, 'Blockchain.csv', concentrates solely on blockchain-related news. Both databases include text samples or entire articles, which provide a wealth of information for research. The news sources are most likely a combination of newspapers, internet portals, and specialised blogs that provide various viewpoints on the financial situation. Data preparation entailed cleaning, normalising, and tokenizing the text, establishing the groundwork for meaningful sentiment analysis and market trend forecasting.

2.2 Data Overview

all-data.csv

Columns: "Sentiment", "News Content".

The dataset categorizes news articles as "neutral", "negative", or "positive".

This dataset includes generic financial news stories. The data might have come from a variety of financial news sources, such as newspapers, internet news portals, or financial research blogs. Every row in the dataset has two columns:

Sentiment: This column categorises the sentiment of news stories. The feeling is usually categorised as "neutral," "negative," or "positive."

News Content: This column contains the actual text of news stories. It comprises extracts or whole articles that serve as the basis for analysis.

Blockchain.csv

Columns: "News Content".

The dataset focuses on news related to blockchain technology.

This dataset is focused on blockchain-related news. The news stories were most likely taken from specialised blockchain or cryptocurrency news outlets, such as specialist blogs, financial publications, or industry reports. The dataset has a single column:

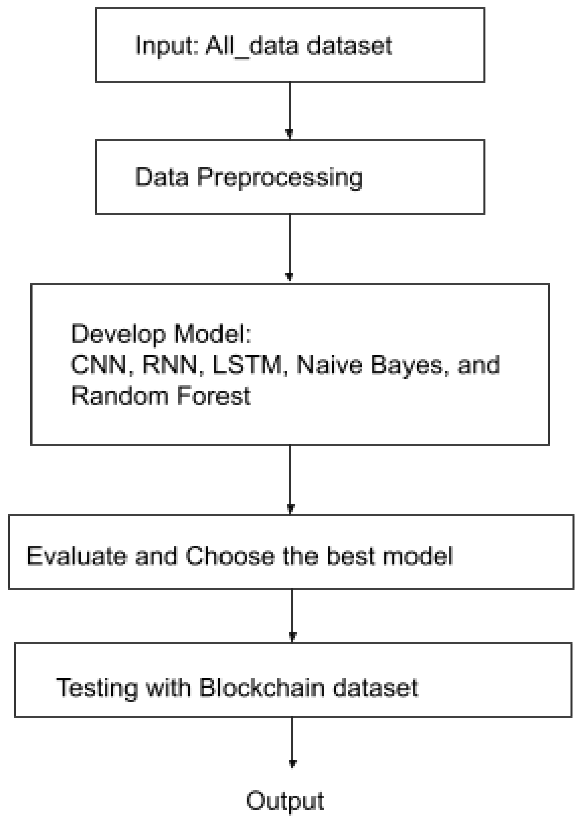
News Content: As with the previous dataset, this column provides the text of the blockchain news items.

2.3 Data Preparation

The method began with data cleaning, which involved removing unnecessary letters and symbols to reduce noise. The material was then normalised, which involved changing all text to lowercase to solve case sensitivity concerns. This was followed by stop word elimination, which filtered away frequent but meaningless words like "and" and "the" to increase the relevancy of the remaining phrases. The text was then tokenized, which separated the material into individual words or phrases for easy processing. Finally, vectorization was used to transform the textual data into numerical representations that could be fed into machine learning algorithms. This comprehensive preparation established the groundwork for powerful and relevant sentiment analysis.

3. Methodology

3.1 Project Flow:



3.2 Sentiment Analysis

The project used sentiment analysis to assess the tone of financial news stories, categorising them as "positive," "neutral," or "negative." Preprocessing consisted of cleaning and tokenizing the text, followed by vectorization utilising techniques such as TF-IDF to convert the articles into numerical features for machine learning models. The project assessed several models, including CNN, RNN, LSTM, Naive Bayes, and Random Forest, and chose the best based on accuracy and F1 score. The chosen model was trained and tested, revealing unique insights into market mood and investor opinions, allowing stakeholders to make educated decisions based on financial news.

3.3 Tools and Technologies

- Python
- Pandas and Numpy for data manipulation
- NLP libraries for text processing - NLTK
- Tensor Flow and Keras for training models
- Scikit-learn for model evaluation

3.4 Model Selection

Model selection is an important phase in any machine learning or data analytics project, including financial news analysis. During this step, many models are tested to see which one works best for the particular goal. The selecting process generally includes many main activities:

Comparing Model Types: Various models, including CNN, RNN, LSTM, Naive Bayes, and Random Forest, are evaluated to determine their strengths and drawbacks.

Setting Evaluation Criteria: Metrics including as accuracy, precision, recall, and F1-score are used to assess how well each model works.

Hyperparameter tuning: It involves adjusting each model's hyperparameters to optimise performance. Grid search and random search are popular methods for doing this.

Cross-validation: The models are evaluated using techniques such as k-fold cross-validation to verify that they generalise effectively to previously unknown data.

Selecting the Best Model: After evaluating the models using the given metrics, the one that best matches the project's objectives is chosen for continued usage.

3.4.1 Naive Bayes Model

Naive Bayes is a simple yet strong probabilistic model commonly used for text categorization. In the project, Naive Bayes used conditional probability and the premise of feature independence to categorise news stories based on sentiment. The model was built using Python's scikit-learn module, which provides an efficient and user-friendly API for a variety of Naive Bayes classifiers, including Multinomial and Gaussian. Despite its simplicity, Naive Bayes generally works admirably for sentiment analysis due to its resistance to irrelevant data and capacity to handle huge feature sets.

Accuracy: 87.4%

Naive Bayes Report

	precision	**recall**	**f1-score**	**support**
0	0.82	0.96	0.88	400
1	0.95	0.79	0.87	418
Accuracy			0.87	818
Macro avg	0.88	0.88	0.87	818
Weighted Avg	0.89	0.87	0.87	818

Output: The Naive Bayes algorithm generated a baseline sentiment categorization for financial news. It clearly separated between "positive," "neutral," and "negative" items. Despite its simplicity, the model worked well because of the assumption of feature independence, which, while not completely correct for text data, was unexpectedly successful for this job.

3.4.2 Random Forest Model

Random Forest is an ensemble learning approach that uses numerous decision trees to increase classification accuracy. Random Forest was implemented in the project using scikit-learn, a powerful tool for building and modifying ensemble models. Random Forest's strength is its ability to reduce overfitting when handling huge datasets with high-dimensional features. It provided a good blend of performance and interpretability, making it a credible option for sentiment categorization in financial news.

Accuracy: 92.18%

Random Forest Report:

	precision	**recall**	**f1-score**	**support**
0	0.92	0.92	0.92	400
1	0.93	0.92	0.92	418
Accuracy			0.92	818
Macro avg	0.92	0.92	0.92	818
Weighted Avg	0.92	0.92	0.92	818

Output: Random Forest's ensemble of decision trees delivered strong performance, decreasing overfitting and enhancing generalisation. The model's strength rested in its capacity to handle vast and complicated feature sets, making it appropriate for the wide range of financial news stories.

3.4.3 Recurrent Neural Network Model

Recurrent Neural Networks (RNNs) are good for sequence data, such as text, because of their capacity to keep memory across consecutive input. In this research, RNNs were used using the TensorFlow and Keras frameworks, which provide a wide range of tools for creating and training neural networks. RNNs were useful in capturing contextual linkages in the text, revealing how earlier words in a phrase impact the meaning of later ones. However, typical RNNs frequently suffer from difficulties such as vanishing gradients, which might impair their capacity to hold long-term relationships.

Accuracy: 89.24%

Recurrent Neural Network Report

	precision	recall	f1-score	support
0	0.88	0.90	0.89	400
1	0.90	0.89	0.89	418
Accuracy			0.89	818
Macro avg	0.89	0.89	0.89	818
Weighted Avg	0.89	0.89	0.89	818

Output: The RNN identified sequential dependencies in the text, which proved useful in comprehending the context of financial news stories. The model's capacity to keep memory across input sequences enabled it to detect nuanced sentiment shifts within articles, resulting in higher classification accuracy.

3.4.4 Convolution Neural Network Model

While CNNs are commonly linked with image processing, they can also be useful for text categorization due to their ability to detect local patterns in data. The research employed CNNs using TensorFlow and Keras, together with convolutional filters, to collect n-grams, or short clusters of words that commonly appear together in certain sentiment settings. CNNs enabled efficient parallelism and recorded hierarchical patterns, which proved very effective for analysing complicated financial news stories.

Accuracy: 91.56%

Convolution Neural Network Report

	precision	recall	f1-score	support
0	0.92	0.91	0.91	400
1	0.91	0.92	0.92	418
Accuracy			0.92	818
Macro avg	0.92	0.92	0.92	818
Weighted Avg	0.92	0.92	0.92	818

Output: The CNN model did well by detecting local patterns in the text, such as commonly occurring word pairs associated with distinct attitudes. This hierarchical pattern recognition capacity resulted in excellent performance in discriminating between several sentiment groups.

3.4.5 Random Forest Model

Random Forest is an ensemble learning approach that uses numerous decision trees to increase classification accuracy. Random Forest was implemented in the project using scikit-learn, a powerful tool for building and modifying ensemble models. Random Forest's strength is its ability to reduce overfitting when handling huge datasets with high-dimensional features. It provided a good blend of performance and interpretability, making it a credible option for sentiment categorization in financial news.

Accuracy: 92.18%

Random Forest Classification Report

	precision	**recall**	**f1-score**	**support**
0	0.92	0.92	0.92	400
1	0.93	0.92	0.92	418
Accuracy			0.92	818
Macro avg	0.92	0.92	0.92	818
Weighted Avg	0.92	0.92	0.92	818

Output: Random Forest's ensemble of decision trees delivered strong performance, decreasing overfitting and enhancing generalisation. The model's strength rested in its capacity to handle vast and complicated feature sets, making it appropriate for the wide range of financial news stories.

3.5 Training and Evaluation

The models were assessed based on accuracy, precision, recall, and F1-score. These metrics offer insight into several elements of the model's performance:

Accuracy: Measures the model's overall soundness, however it might be deceptive if the data is skewed.

Precision: It is the fraction of accurately anticipated positive cases among all positive instances.

Recall: Measures the fraction of accurately anticipated positive cases among all actual positive instances.

F1-Score: A balanced metric that takes into account both accuracy and recall, particularly effective when there is a class imbalance.

Cross-Validation: Cross-validation was used to ensure accurate model generalisation to fresh data. This entails dividing the training data into several groups, training the model on various combinations of these subsets, then assessing it on the remaining data. This technique aids in evaluating the model's stability and capacity to handle various data distributions.

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	87.40%	88%	88%	87%
Random Forest	92.18%	92%	92%	92%
CNN	91.56%	89%	91%	87%
RNN	91.93%	91.95%	91.91%	91.91%
LSTM	90.22%	90.21%	90.22%	90.22%

3.6 Testing the model using a Test Dataset

Dataset: Blockchain
The blockchain dataset consists of only one column which is the "News Headlines".

The "Blockchain" dataset is a subset of the financial news domain that only includes blockchain-related news. Following training on the diversified "all-data" set, the model is evaluated on this dataset to determine its ability to generalise to a specific area within the broader set.

The algorithm uses TextBlob to analyse the sentiment of news headlines, calculating a polarity score that indicates whether the headline is good, negative, or neutral. The algorithm identifies attitudes according to the score and counts the number of positive and negative headlines.
The model creates a word cloud from headlines with positive, negative, and neutral sentiments. It concatenates all of the dataset's positive, negative, and neutral headlines into a single string before creating a word cloud with the WordCloud package. The word

cloud displays the most frequently used terms in these headlines, creating a visual representation of major themes and issues.

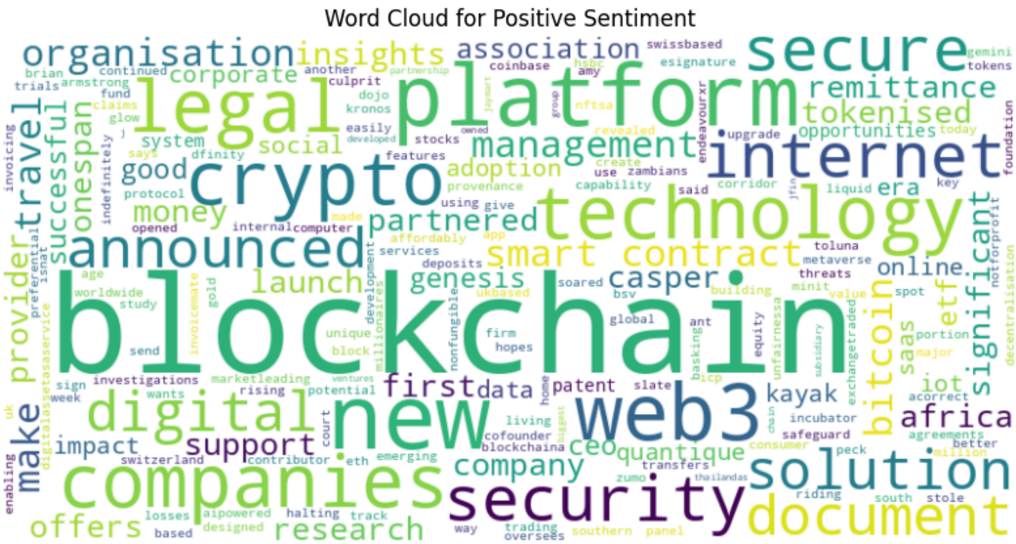
Word Cloud:

Using word clouds, we were able to effectively visualise the emotion of financial news items. The purpose is to emphasise the most commonly used terms in headlines that are related with good, negative, and neutral emotions. The code starts by aggregating the headlines for each sentiment category into a single string, which is then used to create a word cloud with the WordCloud library. The generated word clouds are shown individually for positive, negative, and neutral moods, highlighting the most prominent phrases in each category.

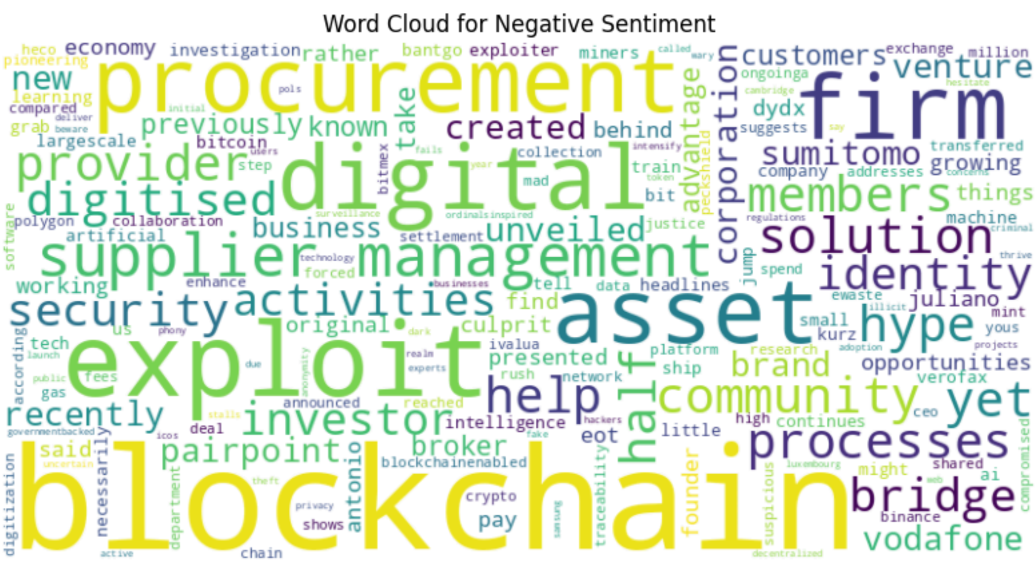
The word clouds are created with a width of 800 pixels and a height of 400 pixels, with a white backdrop for clarity. The imshow function from matplotlib is used to visualise the word clouds, and the axis('off') command conceals the axis to focus on the textual parts. Each word cloud is labelled with a relevant title that clearly indicates the feeling it expresses.

This method produces a visually appealing overview of the prominent themes in financial news items, providing insights into the various feelings. By emphasising the essential phrases connected with each mood, stakeholders may rapidly discover trends and focus areas, which helps with decision-making and market analysis.

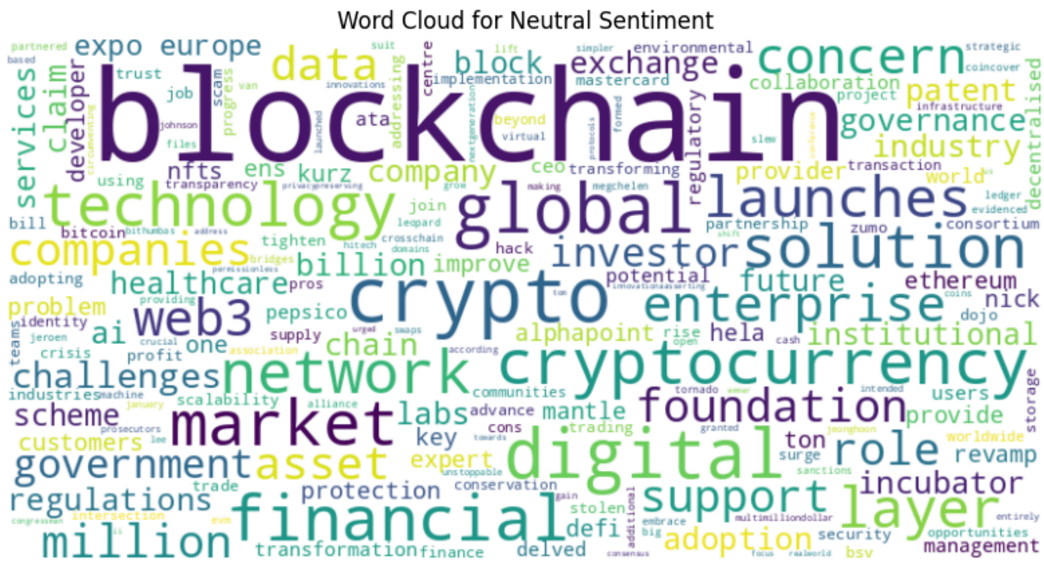
Positive cloud:



Negative Cloud:



Neutral Cloud:



3.6.1 Data preparation

The model begins by preparing the data for sentiment analysis via a recurrent neural network (RNN). The dataset is initially divided into training and testing sets using a 70-30 split ('train_test_split' function). It then tokenizes the text data by translating it to numerical representations using the Keras 'Tokenizer'. The vocabulary size is calculated, and the text sequences are padded to a set length ('MAX_SEQUENCE_LENGTH') to guarantee that the model receives constant input dimensions. Label encoding is used to convert emotion labels into numerical values, hence aiding model training.

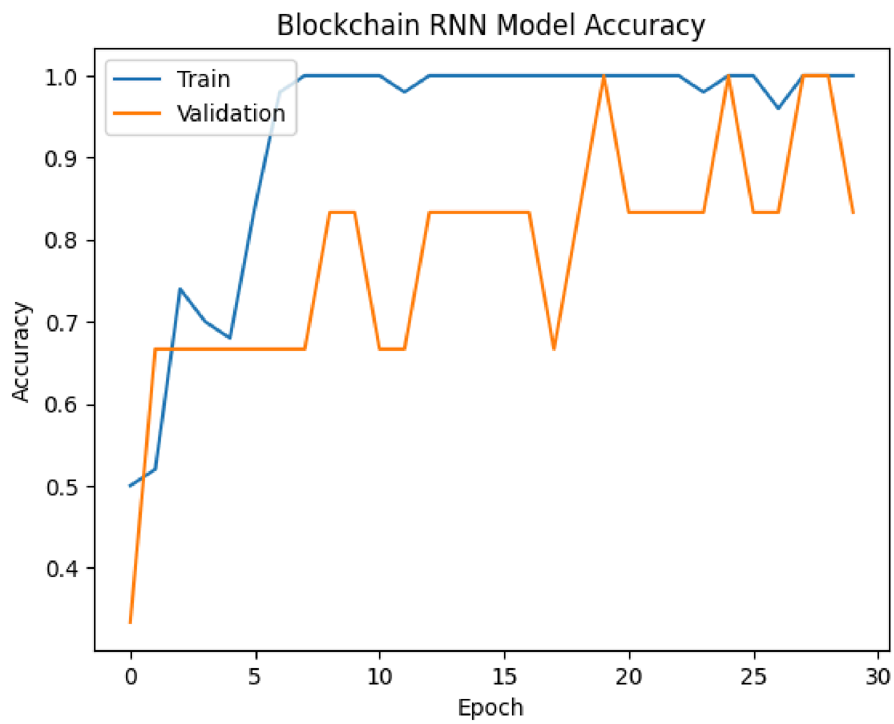
3.6.2 Model Creation and Training

The RNN model is created using the Keras 'Sequential' API. The model consists of several layers, including an embedding layer, dropout layers ('SpatialDropout1D'), a basic RNN layer ('SimpleRNN'), and dense layers. The model is built using binary cross-entropy loss and the Adam optimizer, which are both well-suited to binary classification problems. The training approach includes early stopping ('EarlyStopping'), which halts training when validation performance deteriorates, and model checkpointing ('ModelCheckpoint'), which saves the best-performing model based on validation accuracy. The model is trained on the training set using a validation split for a predetermined number of epochs.

3.6.3 Evaluation and Test:

Following training, the model loads the stored best model and analyses it on both the training and testing sets to determine performance. The assessment comprises computing accuracy scores for both sets, which provide information on the model's generalisation capabilities. The obtained accuracies are presented, demonstrating how effectively the

RNN predicts sentiment in blockchain-related news. This evaluation aids in determining how well the model has learnt and adapted to the training data, as well as its ability to handle previously unknown data.



The graph shows a line chart illustrating the accuracy of a blockchain-related Recurrent Neural Network (RNN) model over many training epochs. The graphic has two lines: training accuracy (blue) and validation accuracy (orange). The graphic depicts how the model's accuracy improves throughout the first epochs, with the training accuracy approaching 100% and remaining constant, whilst the validation accuracy varies. This means that, while the model performs incredibly well on training data, its performance on unknown validation data fluctuates, pointing to possible concerns such as overfitting.

The 'blockchain_rnn_predictions' function uses the test set's stored RNN model to make predictions. The predictions are then compared to the genuine labels to determine several performance measures, including as accuracy, precision, recall, and F1-score. These metrics give a full assessment of the model's ability to categorise the mood of blockchain-related news. The algorithm also outputs a classification report that includes these metrics for each class, as well as a confusion matrix that shows how accurate and wrong guesses are distributed. According to the paper, the model obtains a high accuracy of 87.50%, with balanced precision and recall for both positive and negative thoughts.

Metric	Class 0	Class 1	Macro avg	Weighted Avg
precision	0.83	0.92	0.88	0.88
recall	0.91	0.85	0.88	0.88
f1-score	0.87	0.88	0.87	0.88
support	11	13	24	24
Accuracy	-	-	0.88	0.88

The RNN model utilizes Word2Vec embeddings to analyze sentiment on blockchain-related articles. It begins by turning sentences into word lists, after which it trains a Word2Vec model in Gensim to create word vectors. Using these vectors, the algorithm generates an embedding matrix, which is then put into a Keras-based RNN. The binary classification model is trained using binary cross-entropy loss, with callbacks for early halting and model checkpointing. Following training, the code loads the best model and tests it on both the training and testing sets, resulting in high accuracies that show the model's efficacy. The method combines Word2Vec's capabilities in capturing word semantics with the RNN's ability to handle sequential input.

4. Results

4.1 Sentiment Scores

In the project, sentiment ratings for each news story were calculated using a variety of models, including Naive Bayes, Random Forest, CNN, RNN, and LSTMs. After examining the performance of various models, the RNN emerged as the best-performing model, with a strong balance of accuracy, recall, and F1-scores. The sentiment score generated by this model was then used to classify each item as "positive," "neutral," or "negative".

4.2 Key Findings

Neutral Sentiment Dominance:

After analysing the sentiment of general financial news items from the all-data.csv collection, it was discovered that a considerable number of the articles had a neutral sentiment. This shows that most financial news sources provide genuine information with no strong bias towards positive or negative.

Neutral Sentiment Dominance:

After analysing the sentiment of general financial news items from the all-data.csv collection, it was discovered that a considerable number of the articles had a neutral sentiment. This shows that most financial news sources provide genuine information with no strong bias towards positive or negative.

4.3 Interpretation

Neutral emotion in financial news indicates an emphasis on providing fair and truthful information. This is compatible with the nature of financial journalism, which frequently emphasises impartiality and accuracy in order to appeal to an informed audience.

Positive mood in blockchain news indicates a promising prospect for the sector. This confidence can be linked to blockchain technology's inventive potential and growing use in a variety of industries. The favourable mood reflects increased faith in blockchain's ability to promote technical and economic improvements.

5. Discussion

5.1 Implications

Understanding the sentiment of financial news is critical for market players since it gives insights into the market's overall attitude and expectations. Investors, analysts, and policymakers can use this information to forecast future market moves. For example, a change towards negative mood might signal an upcoming slump or greater market volatility, encouraging investors to modify their portfolios appropriately. Positive emotion, on the other hand, might indicate that the market is bullish, stimulating investment and expansion. The excitement shown in blockchain-related news, in particular, implies that the market has a positive prognosis for this industry. This information is useful for organisations and investors wishing to interact with blockchain technology since it represents a general belief in the sector's potential.

5.2 Limitations

Despite its benefits, sentiment analysis has several drawbacks. The success of the analysis is determined on the quality of the training data and models used. Inaccurate sentiment classifications might result from non-representative or biassed training data. Furthermore, sentiment analysis suffers with complex language, such as sarcasm or idiomatic idioms, which can confuse even powerful models. For example, a sarcastic statement reflecting the inverse of its literal meaning may be misconstrued by sentiment analysis algorithms, resulting in inaccurate results. These limitations underscore the need of carefully evaluating models and taking context into account in sentiment analysis.

5.3 Future Work

Future sentiment analysis research can benefit from advances in natural language processing (NLP) approaches. Modern techniques, such as transformers, have showed significant promise for comprehending context and sophisticated language structures. Incorporating these strategies may improve the accuracy and depth of sentiment analysis. Furthermore, integrating sentiment analysis with financial data like stock prices, trade volumes, and economic indicators can provide a more complete picture of market circumstances. This integrated method would allow stakeholders to compare news mood to real market performance, resulting in better informed decision-making.

6. Conclusion

6.1 Summary

This research effectively analysed the mood of financial and blockchain news stories, giving information about market movements and investor perspectives. The research indicated that the majority of financial news items are neutral, indicating a fair and factual reporting approach. In contrast, blockchain-related headlines had a positive tone, indicating excitement about the sector's prospects.

6.2 Recommendations

According to the findings, stakeholders should monitor financial news mood as a possible indication of market changes. Regular sentiment research might offer early notice of market movements or new possibilities. For investors considering blockchain investments, the positive mood implies a favourable market climate, implying that this industry may provide excellent chances for development and innovation.

7. Reference

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. Expert Systems with Applications, 83, 187–205.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735–1780.

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. Decision Support Systems, 104, 38–48.

Cho K, Merrienboer BV, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. 2014.

Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Egeli B, Ozturan M, Badur B. 2003.

Stock market prediction using artificial neural networks.

K. L. Kohsasih, B. H. Hayadi, Robet, C. Juliandy, O. Pribadi and Andi, "Sentiment Analysis for Financial News Using RNN-LSTM Network,"

Sakhare, N.N., Shaik, I.S. Spatial federated learning approach for the sentiment analysis of stock news stored on blockchain.