A Sentiment Analysis of Financial Headlines using multiple Machine Learning Models

Module Title: Predictive Analytics 23/24
Module Code: MSIN0097
Word Count: 1992
Candidate Number: FYWW6

Table of Contents

Introduction ***3***

EDA ***3***

Figure 1. Class Distribution 3
Figure 2. Word Cloud for negative category 4
Figure 3. Word Cloud for positive category 4
Figure 4. Word Cloud for neural category 4

Pre-processing ***5***

Classical Machine Learning Models ***5***

Figure 5. Effect of Min Samples Split and Min Samples Leaf on Performance 6
Figure 6. Effect of Number of Estimators and Max Depth on Performance 6
Figure 7. Correlation Matrix from best Random Forest 7

Multi-Layer Perceptron ***7***

Figure 8. Baseline MLP model Training & Validation Loss 8
Figure 9. Reduced Network Capacity MLP model Training & Validation Loss 8
Figure 10. L2 Regularised MLP model Training & Validation Loss 9
Table 1 MLP Model's Accuracies 9

Recurrent Neural Network ***10***

Conclusion ***10***

Limitations ***11***

Introduction

In a world where there is an abundance of information and not enough time to consume it, being able to understand market sentiment quickly is crucial for investors seeking to make informed decisions. This project focuses on the natural language processing challenge of analysing sentiment in a collection of financial headlines, from the Financial Phrasebank (Hugging Face's dataset repository)

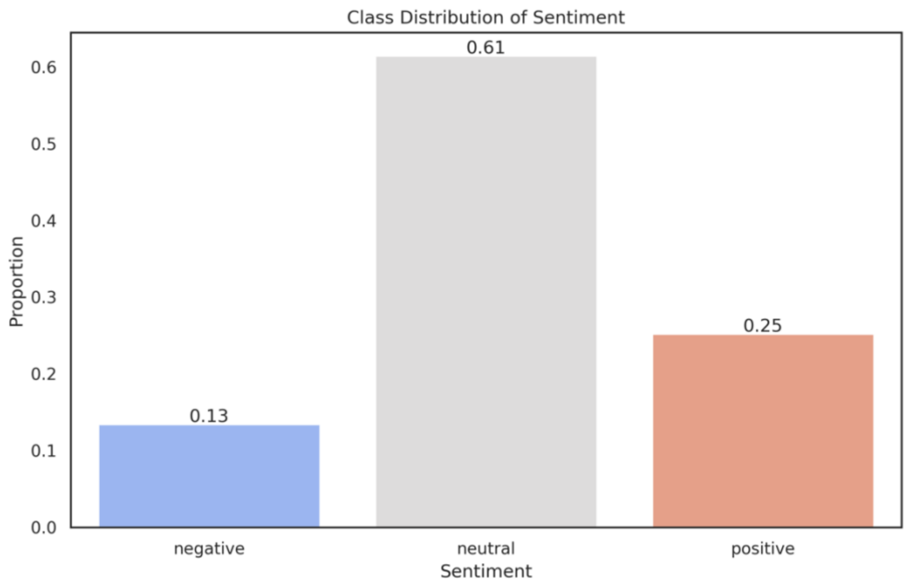
This dataset comprises 4,840 sentences extracted from financial news articles, each one classified by sentiment. Classification was manually completed, with original annotators assessing each sentence's sentiment as either positive, negative, or neutral. This evaluation was based solely on the sentence's content, with a specific focus on the potential impact of the news on stock prices from an investor's perspective. Consequently, sentences that do not carry an economic or financial sentiment are deemed neutral, providing a nuanced approach to understanding market sentiment.

The modeling process begins with constructing a regression model to establish a baseline for data relationships. Subsequently, a decision tree model will examine more complex variable interactions. This progresses to employing a Random Forest Classifier attempting to improve accuracy and robustness by leveraging multiple trees to reduce overfitting. The final step involves deploying neural networks, significantly increasing complexity to uncover deeper insights and capture intricate patterns that simpler models might miss.

EDA

The first step in our exploratory data analysis was to check for class imbalances, which are crucial to identify because they can lead to biased models if one class dominates. Recognising these imbalances can be important in the modelling phase if the model is showing signs of high bias.

Figure 1. Class Distribution



Word clouds offer valuable insights in sentiment analysis by highlighting the most frequent words within each category. When examining the clouds, the largest words aren't always the most telling. However, certain terms do stand out. In the negative word cloud, phrases like 'decreased' and 'profit fell' are prominent. Conversely, the positive cloud features optimistic terms such as 'profit rose', 'grew', and 'increased'. The neutral category is characterized by more objective language, with words like 'production', 'service', and 'include', reflecting words without a positive or negative sentiment attached to it. Although the distinctions between categories are not stark, there's a discernible pattern that could aid a model in distinguishing between sentiments.

Figure 2. Word Cloud for negative category

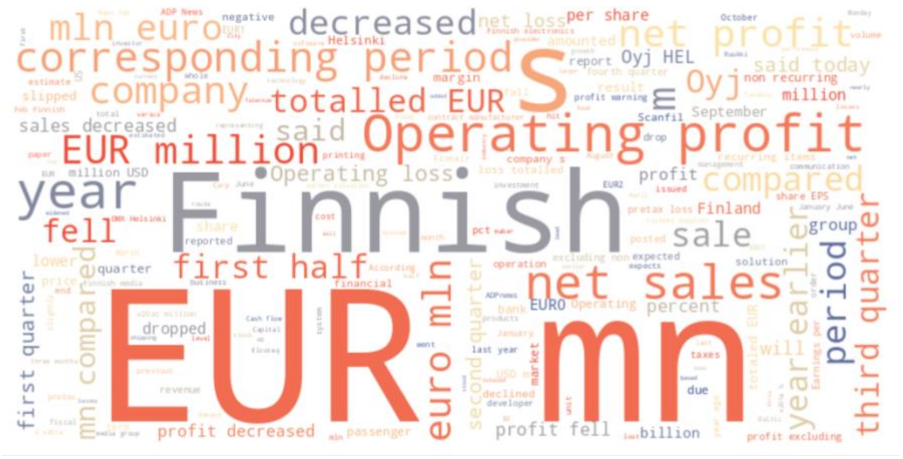


Figure 3. Word Cloud for positive category

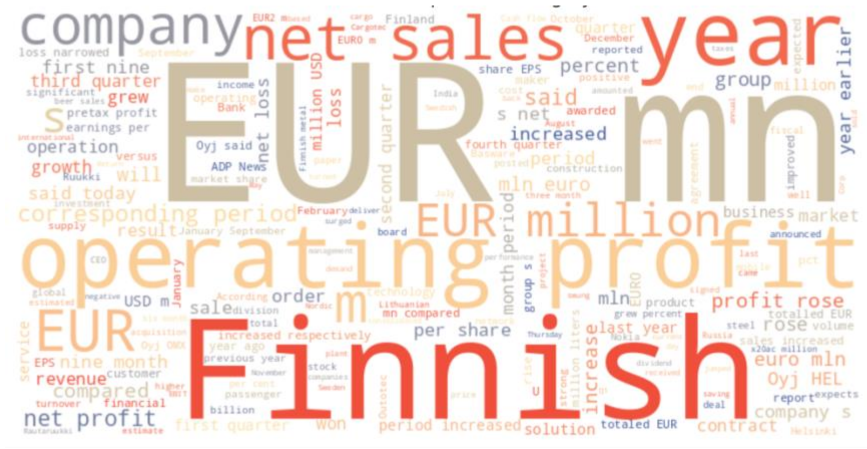
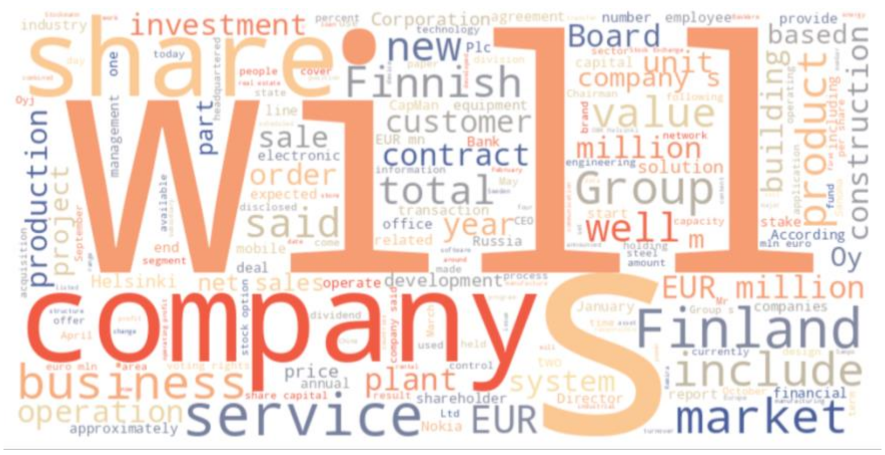


Figure 4. Word Cloud for neural category



Pre-processing

In the data pre-processing phase, the text data was cleaned by: lowercasing, removing URLs and punctuation, and exclusion of words containing numbers. Following this, the data was shuffled to guarantee a random distribution of data. Subsequently, split the dataset into a 90% training set and a 10% testing set, ensuring that each set had a corresponding number of samples for features and labels.

Next, a tokenizer was fitted on to the text to create a comprehensive word dictionary and then converted into binary matrix representations suitable for model input. Then, sentiment labels were then one-hot encoded in order to be included in the model. A subset of the training data was earmarked as a validation set to monitor model performance during training. Lastly, the textual data was vectorised using a TF-IDF approach, transforming it into a feature matrix that highlights the importance of each term in the corpus. After these steps, our the data was prepared and ready for the model building phase.

Classical Machine Learning Models

The model-building process began with a logistic regression model, chosen for its simplicity and effectiveness as a baseline. This model helped gauge the baseline accuracy, providing early insights into potential challenges and helping us understand our data’s characteristics. It achieved an initial accuracy of 83.70%, suggesting the data is well-suited for predictive modelling and showing potential for further improvements with more complex models.

Following the logistic regression model, a decision tree was used due to the model’s ability to handle nonlinear relationships and categorical data. This model achieved a slightly better accuracy of 85.02%, confirming its capacity to capture more complex patterns and prompting further exploration of intricate models.

To take this one step further a random forest was considered, this model is known for reducing overfitting by creating multiple decision trees trained on various data subsets. Despite its strengths, the random forest model achieved a slightly lower accuracy of 84.58% compared to the decision tree. This unexpected result could stem from the model's complexity making it less responsive to our dataset's nuances, or potentially suboptimal parameter settings like the number of trees or their depth.

In response to the unexpected decrease in accuracy from the random forest model, grid search was implemented, to fine-tune the models parameters. This technique systematically explores a range of configurations with the goal of this optimisation to identify the best combination of parameters that could enhance the model's predictive accuracy. By conducting a grid search, we aim to rectify the underperformance seen in the random forest model compared to the decision tree and to achieve higher.

The modest increase in accuracy of the random forest model to 85.03% after the grid search may have been as a result of the optimization of parameters such as the number of trees or their maximum depth. It was an early sign of the benefits of fine tuning and how they can improve model performance.

After conducting a grid search on the random forest classifier, the results presented in Figures 5 & 6 indicate that within the tested range of parameters, neither the minimum samples split nor the number of estimators significantly impacted accuracy. However, the data shows that increasing the minimum number of samples per leaf and the maximum depth of the trees correlates with a slight improvement in model accuracy. This suggests that deeper trees and more substantial leaf samples may enhance the classifier's ability to capture more complex patterns in the data, potentially leading to better performance on this dataset.

Figure 5. Effect of Min Samples Split and Min Samples Leaf on Performance

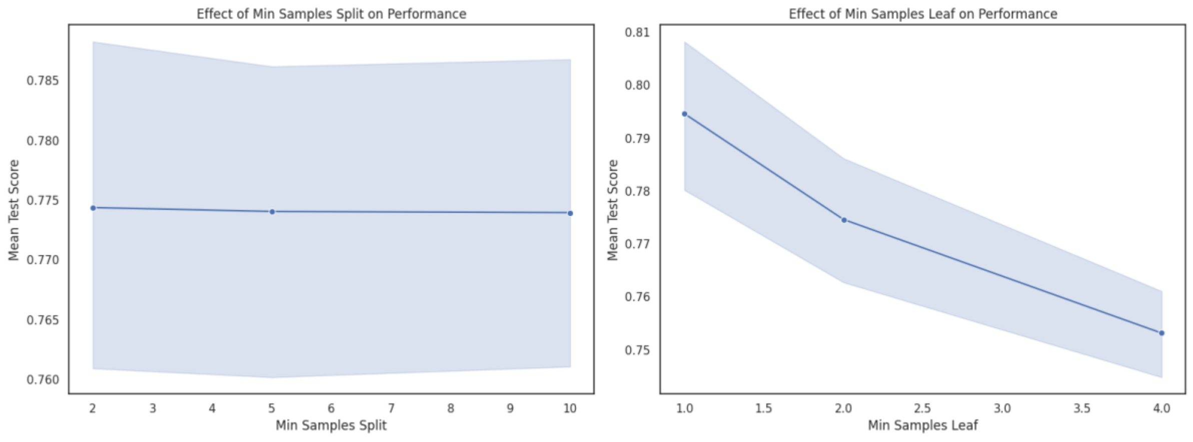
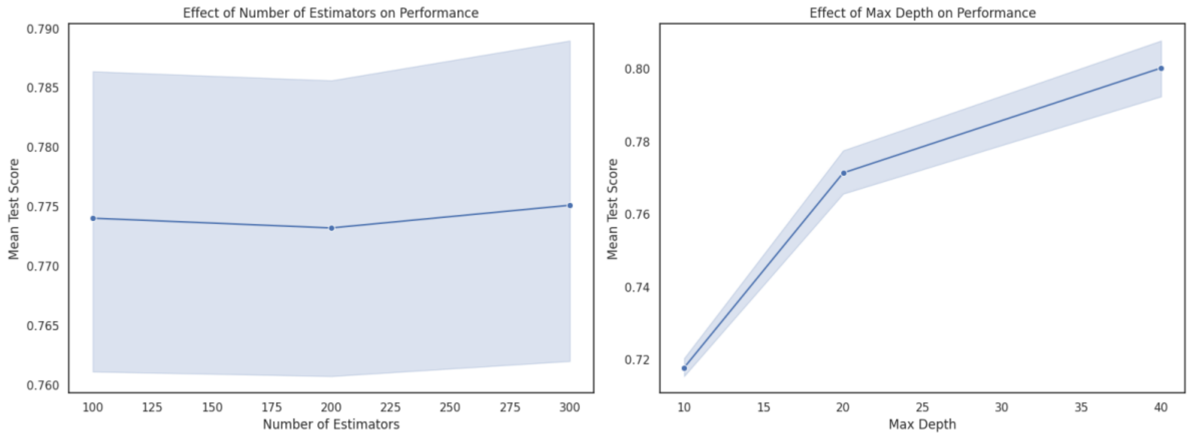
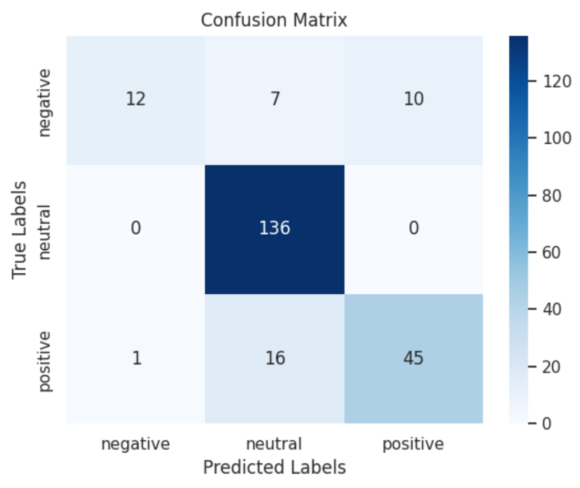


Figure 6. Effect of Number of Estimators and Max Depth on Performance



The confusion matrix for the random forest classifier primarily highlights its proficiency in correctly identifying the neutral category without any misclassifications. Yet, it shows a trend of confusion between negative and positive predictions, often mislabelling one as the other. The positive class, in particular, seems to be more frequently mistaken for neutral, suggesting a possible bias towards the neutral class in the model's predictions. These trends indicate there may be room for improvement in distinguishing between the sentiment classes, especially for the more nuanced negative and positive categories.

Figure 7. Correlation Matrix from best Random Forest



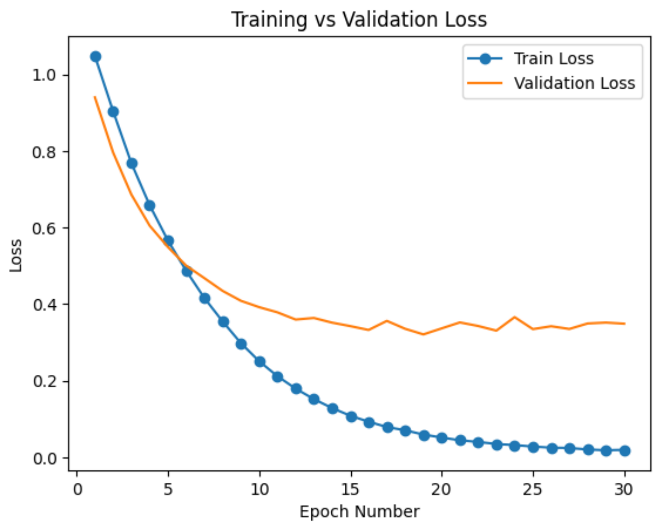
Multi-Layer Perceptron

The first neural network architecture built was a Multilayer Perceptron (MLP) due to its comparatively simpler architecture and its fundamental status among neural networks. Similar to beginning with logistic regression, starting with an MLP, helped gauge the achievable performance with a basic architecture, setting a clear benchmark for subsequent, more complex models.

The architecture of the first model is a succinct MLP with three layers, designed for a multi-class classification. It has two intermediate layers of 64 ReLU-activated neurons each, and a final SoftMax output layer with 3 neurons to classify the inputs into three categories. Compared to the classical ML models constructed so far, this model has the highest accuracy at 89.43%. One reason for this may be due to the neural networks ability to capture complex patterns due to their deep architectures and non-linear activation functions.

Looking at Figure 4 suggests quite a large gap between the training loss and the validation loss. Where the model has performed well on the training set by less well on the validation, this may be a result of overfitting, so in the next phase of this different methods to reduce this will be attempted

Figure 8. Baseline MLP model Training & Validation Loss



To combat overfitting, three strategies will be tested. First, reducing the model's complexity (Géron, 2022), by replacing two intermediate layers of 64 ReLU-activated neurons with a single layer of 16 neurons, aiming to prevent the model from learning overly intricate patterns that do not generalise well (see Figure 9.). Secondly, implementing L2 regularization, by adding a penalty proportional to the sum of the squares of the weights to the loss function to discourage the model from fitting noise in the training data (James, 2013)(see Figure 10.). Lastly, incorporating dropout layers (see Figure 11.), which help reduce overfitting by randomly deactivating a subset of neurons during training, thus preventing the network from becoming overly dependent on any specific neuron and promoting the development of robust features that generalise better to new data (Srivastava, 2014).

Figure 9. Reduced Network Capacity MLP model Training & Validation Loss

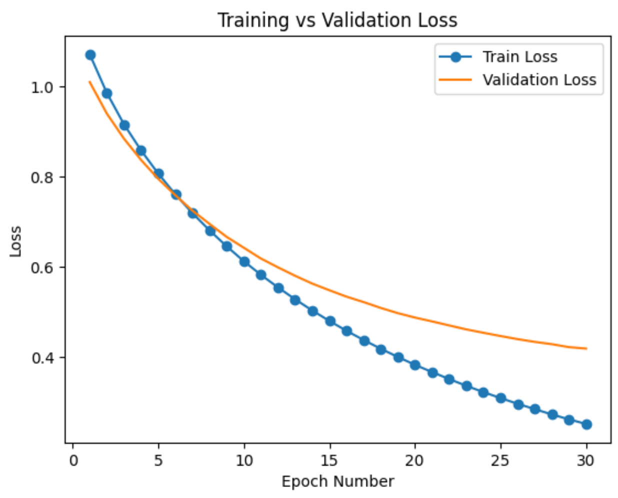


Figure 10. L2 Regularised MLP model Training & Validation Loss

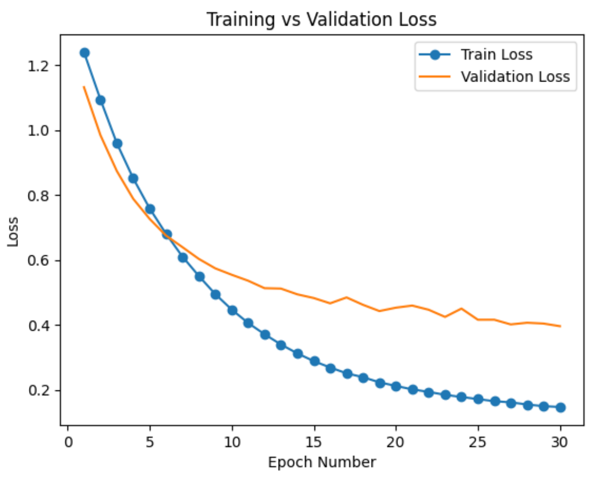
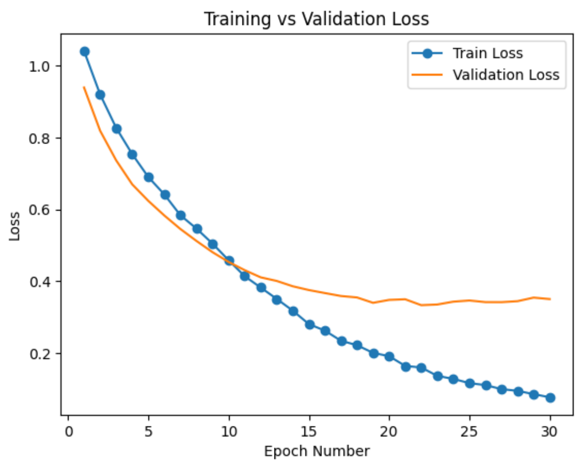


Figure 11. MLP model with dropout layers Training & Validation Loss



The results of various strategies implemented to reduce overfitting are reflected in the test accuracies of model variations. The baseline model recorded an accuracy of 88.98%. Simplifying the model's complexity slightly decreased accuracy to 88.55%, demonstrating a balance between simplicity and effectiveness. Similarly, L2 regularization also resulted in an 88.55% accuracy, effectively controlling overfitting without reducing performance. The largest improvement was when including dropout layers, increasing accuracy to 90.31%, confirming their effectiveness in enhancing model robustness and generalization. What is notable is that all 3 strategies seem to reduce overfitting to some degree.

Table 1 MLP Model's Accuracies

Model	Test Accuracy
Base	88.98%
Reduced Capacity	88.55%
L2 Regularisation	88.55%
Dropout Layers	90.31%

As the dropout layers model achieved the highest accuracy, hyperparameter tuning using random search to further enhance its performance and achieve even higher accuracy. The highest accuracy achieved through Random search was 92.15% which makes this the best model so far. This is a strong result, however there is still some room for improvement, so I wanted to try one more slightly more complex model, to see if there could be any further improvements.

Recurrent Neural Network

The model architecture selected to test next was a Recurrent Neural Network with Long Short-Term Memory layers included, as they have been found to be particularly strong in NLP classification tasks (Minaee et al., 2020). Especially in sentiment analysis classification tasks (Goldberg, 2016). The model starts with an embedding layer that reduces dimensionality to a 16-dimensional vector space, essential for managing large input vocabularies efficiently. An LSTM layer with 84 units captures temporal patterns, and a SoftMax output layer maps these to classification categories.

In theory the RNN should be better at processing sequential data, the initial deployment of our LSTM-based model yielded a lower-than-expected accuracy of 78.36%. This underperformance could be due to, the characteristics of the dataset, not fully utilising the LSTM's capabilities, potential overfitting due to the model's complexity, and potentially suboptimal initial hyperparameter settings. Nevertheless, by applying random search for hyperparameter tuning, a 7% increase to 85% accuracy was achieved. This improvement highlights the critical role of fine-tuning in optimising complex models, demonstrating that with the right adjustments, the sophisticated mechanisms of RNNs can be effectively harnessed.

Conclusion

This project developed a range of ML models, from straightforward regression to intricate neural networks. The classical models performed reliably well, yet greater complexity did not consistently enhance accuracy. A similar trend was observed with neural networks, where the sophisticated LSTM-based RNN initially fell short of expectations, underscoring the challenges inherent in complex models.

The outcomes of these experiments emphasize the trade-off between model complexity and overfitting. However, they also demonstrate the value of fine-tuning. By adjusting model parameters, notable improvements were achieved, showcasing the critical role of this process in optimising model performance. The MLP with dropout layers is the preferred model due to its superior accuracy. This model effectively combats overfitting and demonstrates strong generalisation on unseen data, making it ideal for this task.

Limitations

While the project achieved promising results, there were some limitations. The use of basic embeddings might not have fully captured the complexity required for nuanced sentiment analysis in the financial domain. Advanced embeddings from models like BERT or GPT, designed to grasp contextual nuances, could enhance accuracy. The architecture choice, an MLP with dropout layers, while effective, might not be the most optimal. Alternatives such as Convolutional Neural Networks (CNNs) or more complex LSTM configurations could potentially offer improvements.

References

Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (3rd ed.). O'Reilly Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, (https://dl.acm.org/doi/10.5555/2627435.2670313)

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J., 2020. Deep Learning Based Text Classification: A Comprehensive Review. arXiv preprint arXiv:2004.03705.

Goldberg, Y. (2017). Modeling with Recurrent Networks. In: Neural Network Methods for Natural Language Processing. Synthesis Lectures on Human Language Technologies. Springer, Cham. https://doi.org/10.1007/978-3-031-02165-7_16

Link to Code

https://drive.google.com/drive/folders/1GSEXsRIEXwXFGDZMxXoyxr3DiUtMJvLw?usp=sharing

Appendix

The dataset offers variations based on the consensus among annotators, ranging from a 50% to 100% agreement rate. The versions with lower agreement rates contain more instances; the dataset with a 50% agreement rate has 4,217 sentences. In contrast, the dataset with a 100% agreement rate includes 2,264 sentences, offering a more refined but smaller pool of data. Despite having roughly half instances as the 50% agreement, the 100% set worked much better in the modelling process.