Leveraging XGBoost for Text Classification on Emotion Recognition

Diego Armando Salinas Lugo
Univeristy of Essex
MSC Artificial Intelligence
Student 2401168
ds24353@essex.ac.uk

Abstract—The study explores the application of XGBoost for emotion recognition in text data, focusing on preprocessing techniques, feature extraction, modelling, and performance evaluation. The model's results demonstrate high accuracy and effectiveness in distinguishing between six distinct emotional categories.

Keywords— text, classification, emotion, recognition, XGBoost, preprocessing, model, performance, metrics.

I. Introduction

Text classification plays a vital role in natural language processing, with uses like emotion detection and spam filtering. This project focuses on categorizing text using XGBoost, chosen for its efficiency with high-dimensional data and multi-classification tasks. Initially, linear regression was explored but later replaced. A Bag-of-Words approach was used for feature extraction, and performance was evaluated through metrics such as accuracy, precision, recall and F1-score using stratified K-fold cross-validation. This work highlights the model's effectiveness and areas for improvement.

II. Methodology

A. *Reading the file and splitting the data into trainning set and testing set.*

First, the csv file must be read and store to can be divided into the two catgories, one of them to train the model and the other one to predict. The dataset contains 416,809 entries, divided into training (90%) and testing (10%) subsets.

B. *Preprocessing the text*

Preprocessing is crucial to clean the text data and prepare it for the modelling.

Text preprocessing involves:

- Lowercasing
- Removing non-alphabetic characters
- Eliminating stopwords
- Stemming using the Porter Stemmer algorithm.

C. *Analysing the feature of the training set, reporting the linguisitc features of the training dataset.*

This includes word counts, unique words, frequent words and visualizations.

- Total words in training set: 3510229
- Average words per text: 9.36
- Maximum words in a text: 79
- Minimum words in a text: 0
- Voculary size (unique words): 49133

The visualizations are:

- A bar plot of the top 20 most common words.
- A histogram showing the distribution of text lengths.

In this section there are two functions to generate the following visualizations per emotion:

- A barplot demonstrating the top 20 words associated with fear.
- A word cloud with the same purpose of showing with more emphasis the most related words to the emotion.

D. *Building the text classification model, trainning the model on the training set and testing the model on the test set.*

For this part, it must be said that at the beginning linear regression was being applied for modelling trying to get good results, trainning the model with max iterations of 1000. However, after noticing the metric scores. The research for other models started. The second and actual model in use is XGBoost, due to its recognition of being an efficient gradient boosting algorithm, good for handling high-dimensional feature spaces and capturing complex, non-linear relationships between features

and the target which can be very helpful for text classification. Highlighting that XGBoost is specially designed for classification tasks, offering great perfomance in termns of accuracy, precision, recall, and F1-score, built-in regularization to prevent overfitting and scalability for large datasets, making it suitable for this project where the dataset consits of over 400,000 entries.

Key model parameters included:

- use_label_encoder=False: Disables automatic label encoding for better control.
- eval_metric='mlogloss': Evaluates multi-class loss, ideal for multi-class classification.
- random_state=42: Ensures consistent and reproducible results.

E. Summarizing the perfomance of the model.

After the modelling, a classification report is done showing the precision, recall, f1-score and support for the six emotions. Also, the overall accuracy, macro-averaged precision, recall, and F1-score are calculated and printed as well as confusion matrix for the model.

```
Classification Report (XGBoost):
              precision    recall  f1-score   support

     sadness       0.93      0.91      0.92     12093
         joy       0.90      0.89      0.89     14052
        love       0.76      0.77      0.76      3500
       anger       0.88      0.88      0.88      5695
        fear       0.88      0.82      0.85      4837
    surprise       0.67      0.92      0.77      1504

    accuracy                           0.88     41681
   macro avg       0.83      0.86      0.85     41681
weighted avg       0.88      0.88      0.88     41681

Overall Model Accuracy (XGBoost): 0.8773
Macro-Averaged Precision: 0.8341
Macro-Averaged Recall: 0.8646
Macro-Averaged F1-Score: 0.8457
```

Image 1.0 Results Report XGBoost model.

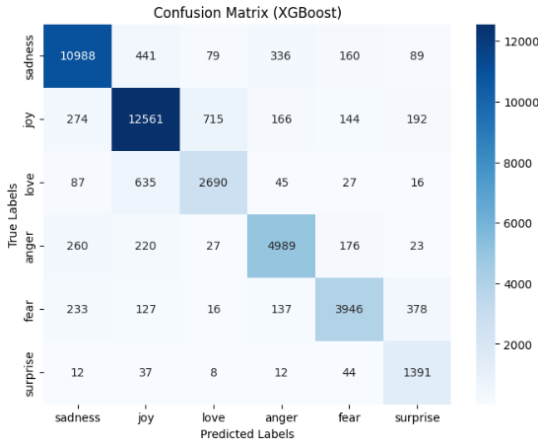

Image 1.1 Confusion Matrix XGBoost model.

To obtain a better understanding of the performance of the model, Cross-Validation for multiple runs and standard deviation of accuracy, precision, recall and F1 have been developed. This technique splits the data into training and testing sets multiple times using a stratified K-fold approach (5 in this case), ensuring balanced class distribution across folds. The model is trained and evaluated on different subsets of the data, and the metrics (accuracy, precision, recall, and F1-score) are calculated for each fold. Finally, the mean and standard deviation of these metrics are reported to provide a more robust evaluation of the model's performance and its consistency across different data splits.

The results:

```
Cross-Validation Results (5-Fold Cross-Validation):
Accuracy: Mean = 0.8750, Std = 0.0007
Precision: Mean = 0.8292, Std = 0.0015
Recall: Mean = 0.8629, Std = 0.0013
F1 Score: Mean = 0.8422, Std = 0.0007
```

Image 1.2 Cross-Validation results

For more visualization, a Precision-Recall Curve for each emotion is shown.

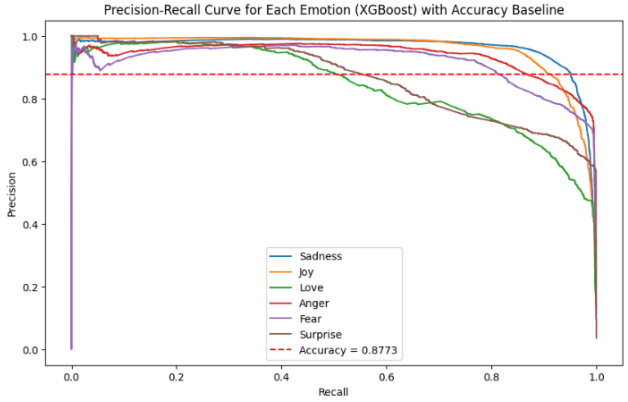

Image 1.3 Precision-Recall Curve for each emotion.

III. RESULTS

As previously shown the results after the 5-Fold Cross-Validation are:

- Accuracy: Mean = 0.8750, Std = 0.0007
- Precision: Mean = 0.8292, Std = 0.0015
- Recall: Mean = 0.8629, Std = 0.0013
- F1 Score: Mean = 0.8422, Std = 0.0007

IV. Discussion

First of all, it is important to remark that modelling with XGBoost performed better than with the linear regression model since with linear regression the results were:

- Accuracy: Mean = 0.8688, Std = 0.0009
- Precision: Mean = 0.8210, Std = 0.0014
- Recall: Mean = 0.8182, Std = 0.0019
- F1 Score: Mean = 0.8193, Std = 0.0013

Now, what could be tried to improve the model:

A. Feature Engineering

- Replacing Bag-of-Words with TF-IDF for feature extraction to give more importance to meaningful words and reduce the impact of the commonly used ones.
- Experimenting with n-grams (e.g., bigrams, trigrams) to capture word sequences and context.

B. Hyperparameter Tuning

- Using Grid Search or Randomized Search to optimize hyperparameters like:
    - learning_rate
    - max_depth
    - n_estimators
    - min_child_weight
- Experimenting with regularization parameters (lambda, alpha) to reduce overfitting.

V. Conclusion

This study highlights the potential of XGBoost for text-based emotion classification, achieving robust performance metrics and providing a foundation for further enhancements in NLP tasks. Remarking the importance of analyzing the context, the data and the purpose of the project to research between the different machine learning models to work with the right one. In this case, the project was first worked with linear regression but after researching and performing XGBoost, it was better not to ensure that is the best, since another one might perform better.

References

[1] Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

[2] OpenAI. (2024). Assistance provided via ChatGPT for text classification and analysis. Retrieved from https://openai.com/chatgpt