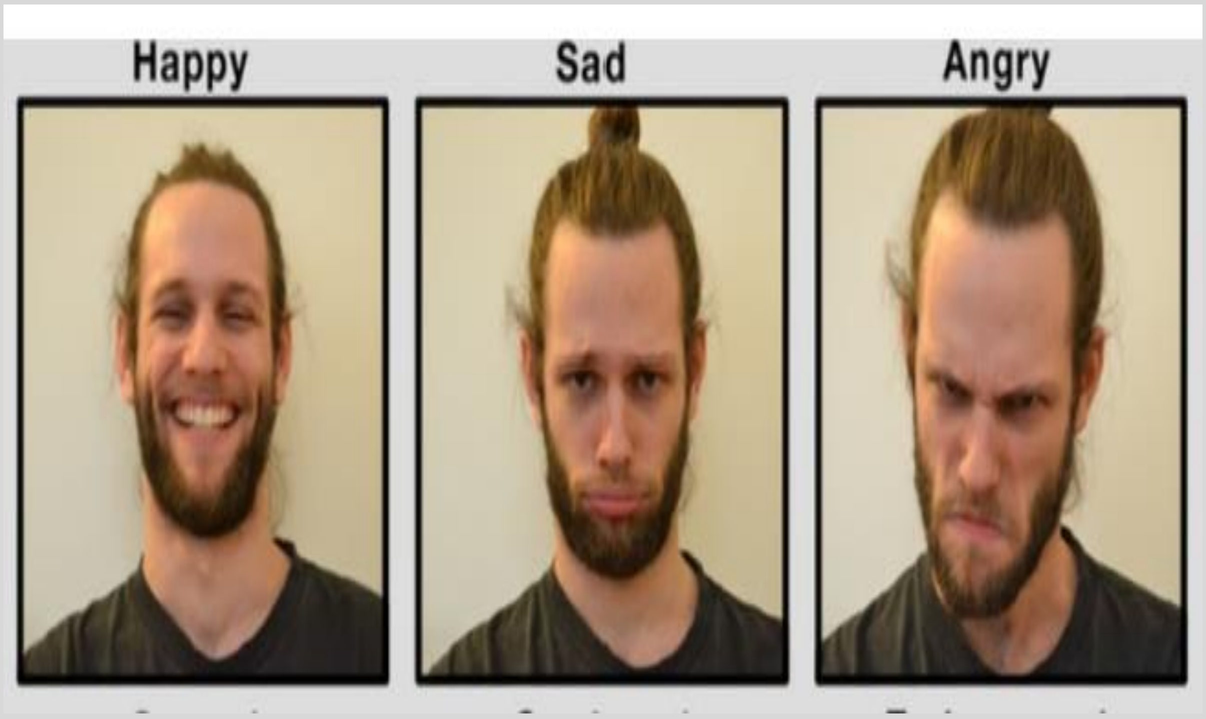
EMOTION CLASSIFICATION



EUNICE ABIEYUWA IGBINEDION

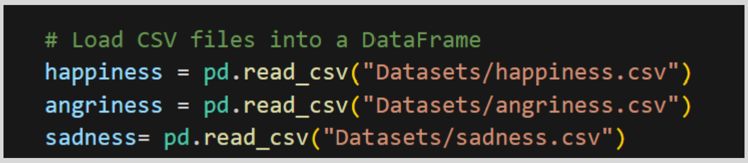
TABLE OF CONTENTS

1. Introduction

This report provides a comprehensive analysis of emotion classification using various natural language processing (NLP) models. The objective is to identify and predict emotions in text data, with a focus on enhancing accuracy and generalization through advanced modelling techniques.

2. Dataset Overview

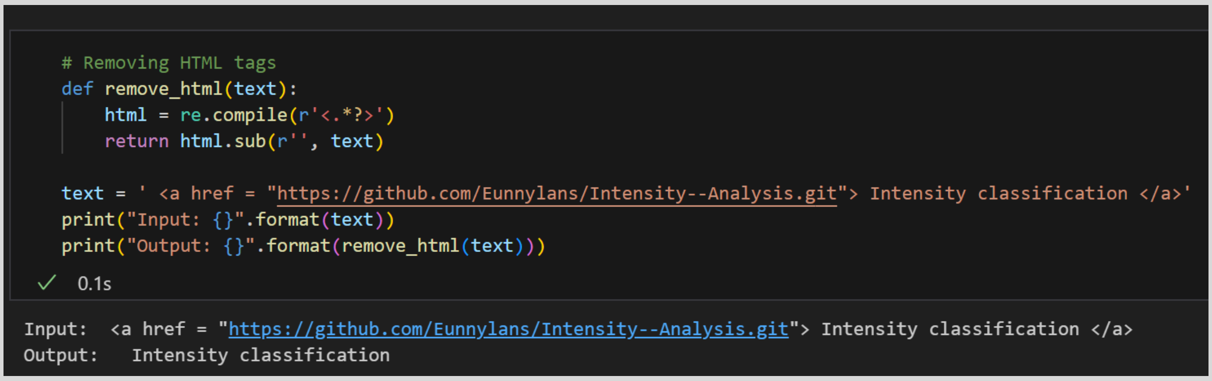
The dataset comprises multiple emotion categories, predominantly Happiness, followed by Anger and Sadness. The analysis reveals that people tend to send longer and more messages when happy, compared to when they are angry or sad.



```python
# Load CSV files into a DataFrame
happiness = pd.read_csv("Datasets/happiness.csv")
angriness = pd.read_csv("Datasets/angriness.csv")
sadness= pd.read_csv("Datasets/sadness.csv")
```

3. Text Normalization

Text normalization involves cleaning and standardizing the text data to ensure consistency and improve the performance of NLP models. This process includes removing special characters, converting text to lowercase, and eliminating stop words.



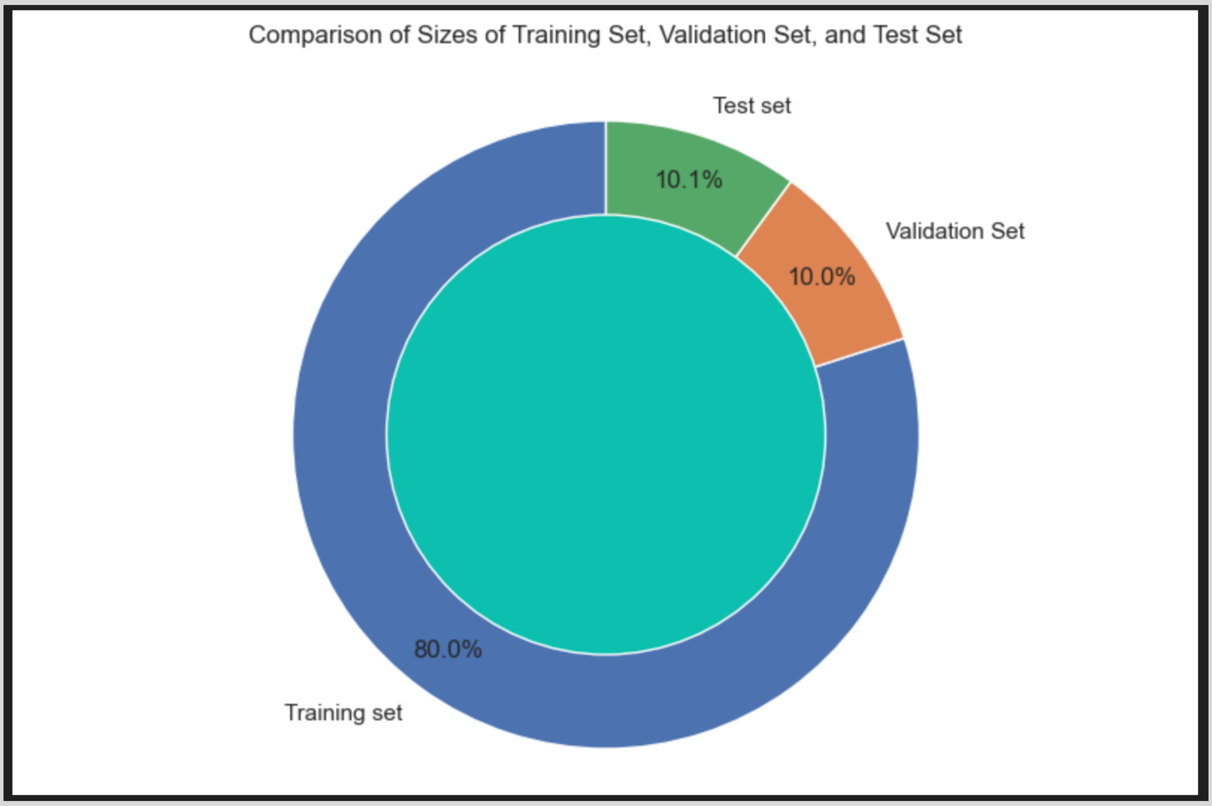
```python
# Removing HTML tags
def remove_html(text):
    html = re.compile(r'<.*?>')
    return html.sub(r'', text)

text = ' <a href = "https://github.com/Eunnylans/Intensity--Analysis.git"> Intensity classification </a>'
print("Input: {}".format(text))
print("Output: {}".format(remove_html(text)))
```

```
✓ 0.1s
```

```
Input:  <a href = "https://github.com/Eunnylans/Intensity--Analysis.git"> Intensity classification </a>
Output:   Intensity classification
```

4. Data Splitting

The dataset was divided into training, testing, and validation sets to evaluate the model's performance. This step is crucial for assessing how well the model generalizes to unseen data.

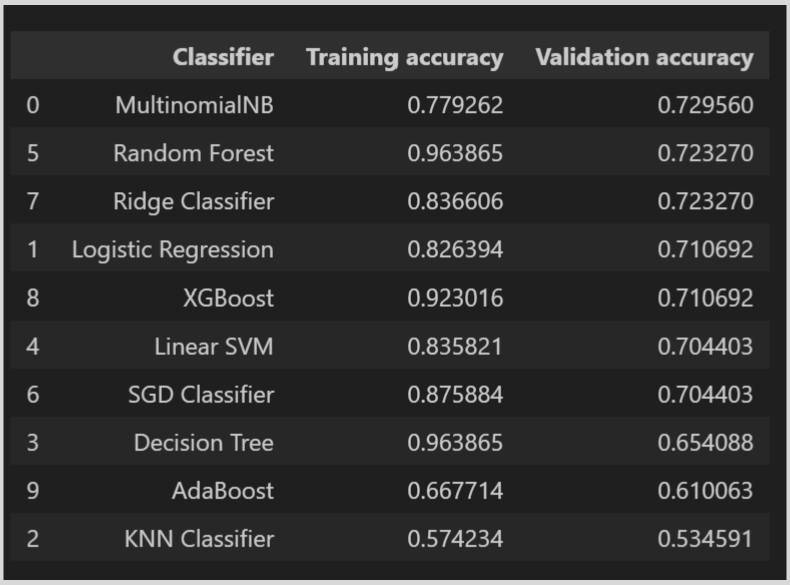


5. Bag of Words (BoW) Models

BoW models were employed to convert text data into numerical representations. Initial results indicated that these models perform well, particularly in identifying the "Happiness" category.

6. TF-IDF Models

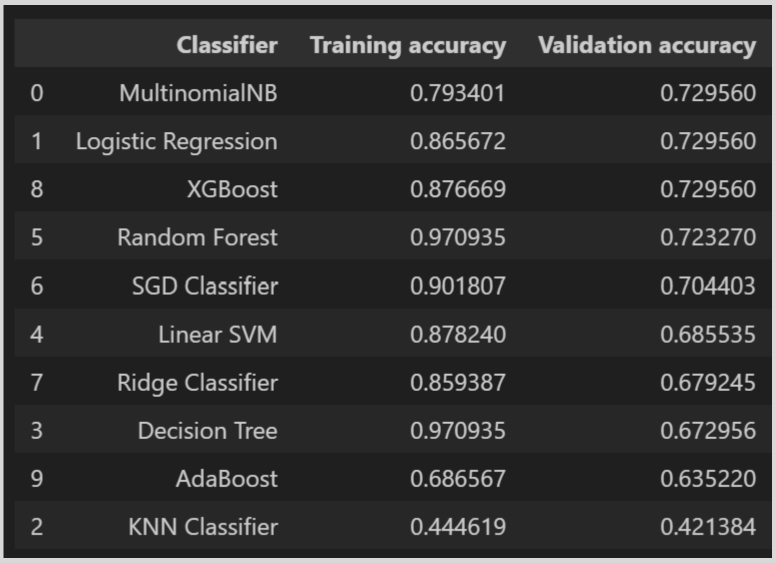
TF-IDF (Term Frequency-Inverse Document Frequency) models were explored to weigh the importance of words in the dataset. The XGBoost and Random Forest model using TF-IDF showed overfitting on the training data, but a comparative analysis revealed that models using CountVectorization outperformed those using TF-IDF.



	Classifier	Training accuracy	Validation accuracy
0	MultinomialNB	0.779262	0.729560
5	Random Forest	0.963865	0.723270
7	Ridge Classifier	0.836606	0.723270
1	Logistic Regression	0.826394	0.710692
8	XGBoost	0.923016	0.710692
4	Linear SVM	0.835821	0.704403
6	SGD Classifier	0.875884	0.704403
3	Decision Tree	0.963865	0.654088
9	AdaBoost	0.667714	0.610063
2	KNN Classifier	0.574234	0.534591

7. Hyperparameter Tuning for BoW Models

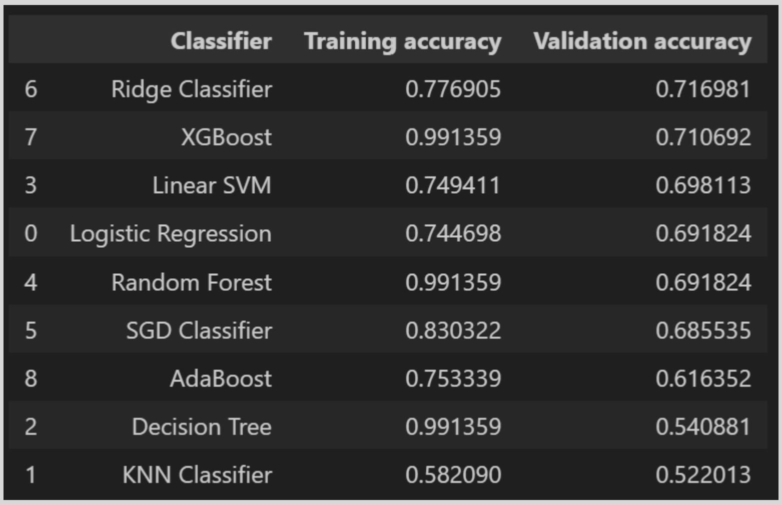
Hyperparameter tuning was conducted on the best-performing BoW model, SDG & XGBoost with CountVectorizer. The tuned model demonstrated good generalization, performing well on the test set with a reasonable balance between accuracy and overfitting. The Decision tree model using TF-IDF showed overfitting on the training data



	Classifier	Training accuracy	Validation accuracy
0	MultinomialNB	0.793401	0.729560
1	Logistic Regression	0.865672	0.729560
8	XGBoost	0.876669	0.729560
5	Random Forest	0.970935	0.723270
6	SGD Classifier	0.901807	0.704403
4	Linear SVM	0.878240	0.685535
7	Ridge Classifier	0.859387	0.679245
3	Decision Tree	0.970935	0.672956
9	AdaBoost	0.686567	0.635220
2	KNN Classifier	0.444619	0.421384

8. Word2Vec Models

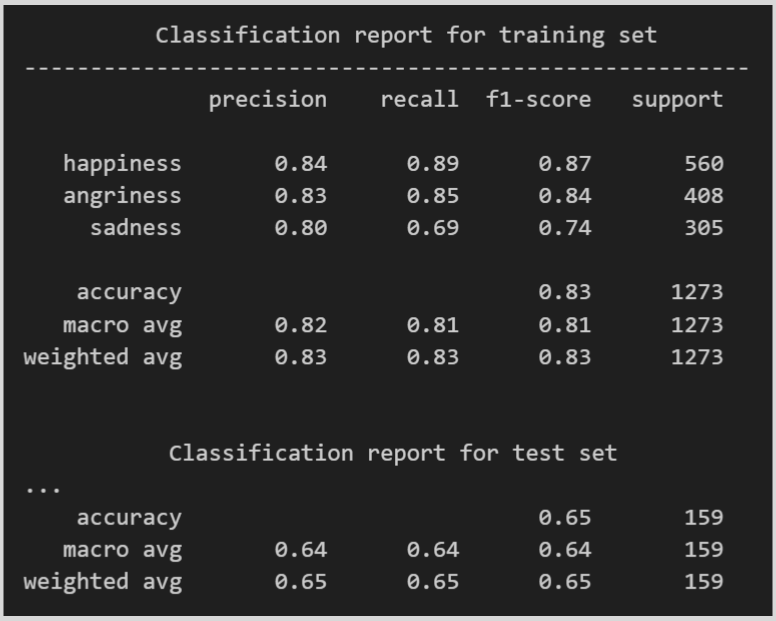
Word2Vec models were explored to capture semantic meaning in text data. The XGBoost and Random Forest and Decision Tree shows that there was overfitting in the test dataset with Word2Vec outperformed other models, showing better accuracy in emotion classification.



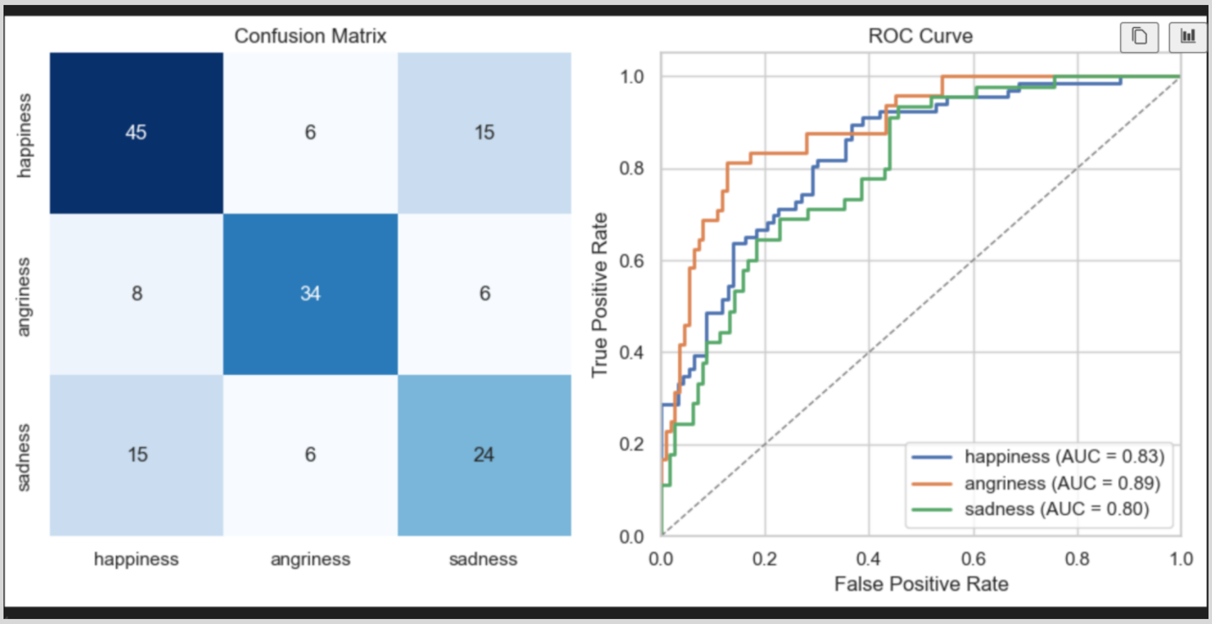
	Classifier	Training accuracy	Validation accuracy
6	Ridge Classifier	0.776905	0.716981
7	XGBoost	0.991359	0.710692
3	Linear SVM	0.749411	0.698113
0	Logistic Regression	0.744698	0.691824
4	Random Forest	0.991359	0.691824
5	SGD Classifier	0.830322	0.685535
8	AdaBoost	0.753339	0.616352
2	Decision Tree	0.991359	0.540881
1	KNN Classifier	0.582090	0.522013

9. Hyperparameter Tuning for Word2Vec Models

After hyperparameter tuning, the Linear SVM model achieved an accuracy of 87%, with minimal overfitting. This indicates that the model is effective in predicting emotions on new data.

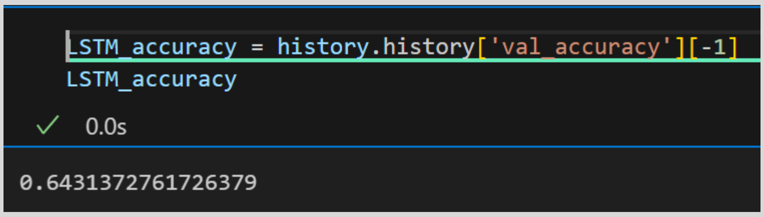


```
           Classification report for training set
-------------------------------------------------------
              precision    recall  f1-score   support

   happiness       0.84      0.89      0.87       560
   angriness       0.83      0.85      0.84       408
     sadness       0.80      0.69      0.74       305

    accuracy                           0.83      1273
   macro avg       0.82      0.81      0.81      1273
weighted avg       0.83      0.83      0.83      1273


            Classification report for test set
...
    accuracy                           0.65       159
   macro avg       0.64      0.64      0.64       159
weighted avg       0.65      0.65      0.65       159
```



10. GloVe and LSTM Models

The performance of the GloVe LSTM model was not optimal due to the small dataset size. To enhance emotion prediction and pattern identification, it is recommended to collect more data for better deep learning performance.



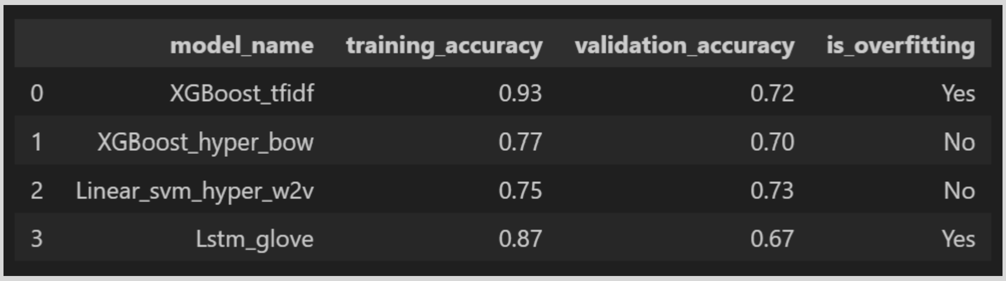
```
LSTM_accuracy = history.history['val_accuracy'][-1]
LSTM_accuracy
```

```
✓ 0.0s
```

```
0.6431372761726379
```

11. Conclusion

The Linear SVM with Word2Vec emerged as the best model, achieving a high validation accuracy of 73%. Despite the challenge of users typing native language slang in English, the model effectively predicted emotions. Future improvements will focus on expanding the dataset to further enhance performance.

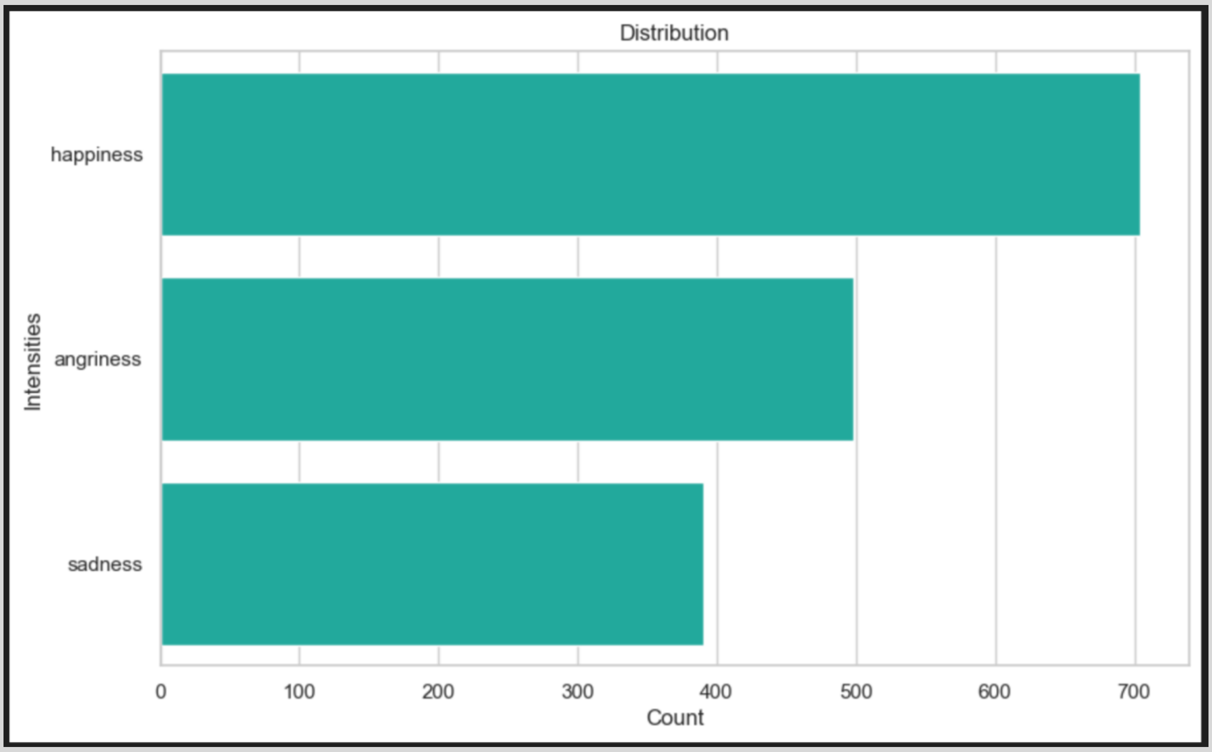


	model_name	training_accuracy	validation_accuracy	is_overfitting
0	XGBoost_tfidf	0.93	0.72	Yes
1	XGBoost_hyper_bow	0.77	0.70	No
2	Linear_svm_hyper_w2v	0.75	0.73	No
3	Lstm_glove	0.87	0.67	Yes

Visualisation and Insights

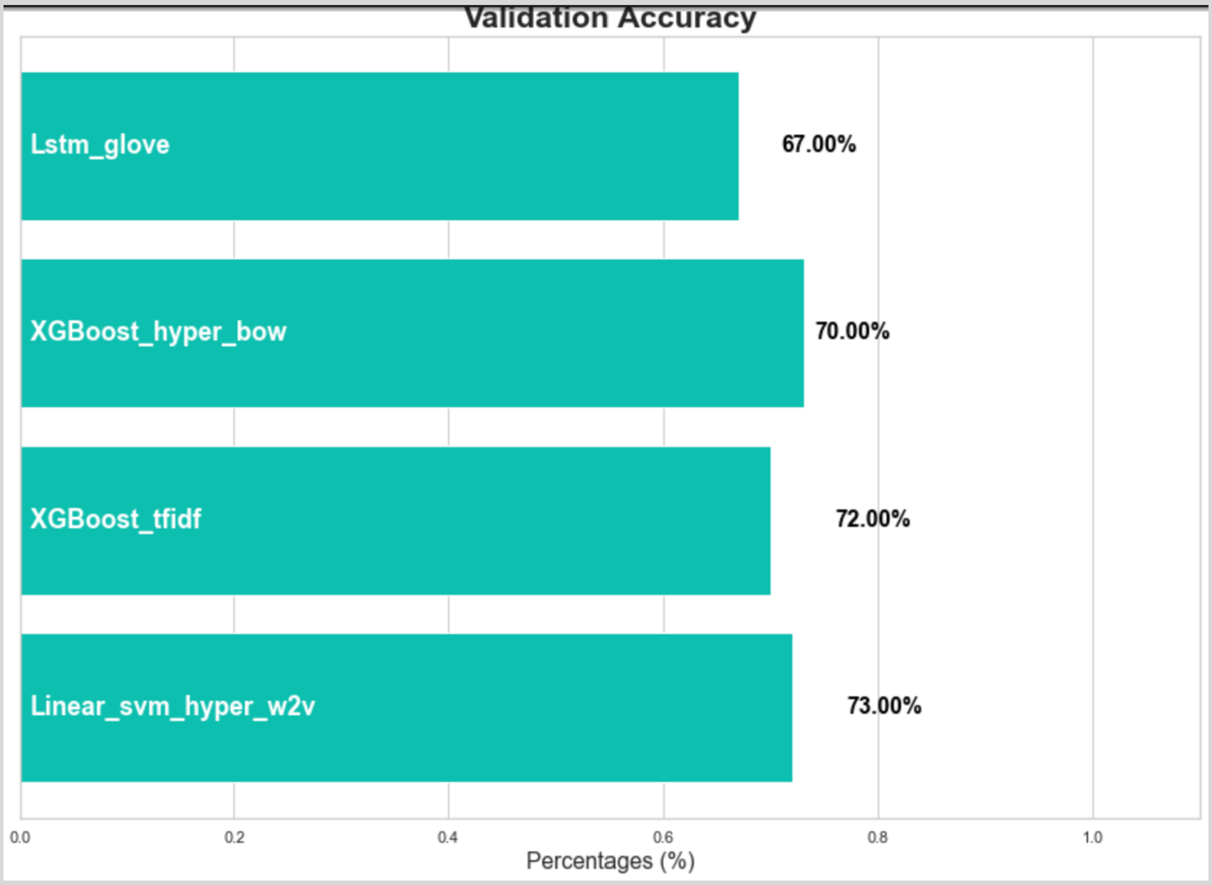
Data Distribution:

- **Happiness:** Highest number of messages.
- **Anger:** Moderate number of messages.
- **Sadness:** Fewest number of messages.



Model Performance:

- **BoW Models:** Good performance, particularly in identifying Happiness.
- **TF-IDF Models:** XGBoost showed overfitting; CountVectorization models performed better.
- **Word2Vec Models:** Linear SVM achieved the highest accuracy.
- **GloVe LSTM:** Requires more data for optimal performance.



Hyperparameter Tuning:

- **BoW (XGBoost):** Improved generalization with balanced accuracy.
- **Word2Vec (Linear SVM):** Achieved 74% accuracy with minimal overfitting.

Future Work:

- **Data Collection:** Gather more data to improve deep learning models.
- **Advanced Techniques:** Explore other NLP techniques and models.
- **Real-world Application:** Implement the best model in real-world applications to validate its effectiveness.

Happy Reading

1