**Infosys Springboard Internship Program 4.0**

****

Project Documentation on

**“Data Visualization – Text Classification”**

Submitted by Moulya. R

Under the Guidance of

Mentor – Sudheer Kumar

**Acknowledgement**

I am pleased to acknowledge the successful completion of the virtual **Data Visualization** Internship at Infosys Springboard. During this period, I completed a project titled **“Text Classification”** which is bona fide work carried out by Miss Moulya. R in the year 2024.

I would like to express my sincere gratitude to my mentor, **Mr.** **Sudheer Kumar**, for his invaluable support and encouragement throughout this internship. I also extend my heartfelt thanks to **Infosys Springboard** for providing me with this remarkable opportunity.

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **1** | **PROJECT INTRODUCTION** |
| **2** | **DATA COLLECTION** |
| **3** | **DATA PRE-PROCESSING**   * Lower Case * Remove links * Remove next lines * Words containing numbers * Extra spaces * Special characters * Removal of stop words * Stemming * Lemmatization * Eliminating features with extremely low frequency * Use Complex Features: n-grams and part of speech tags |
| **4** | **FEATURE ENGINEERING**   * TF-IDF * SMOTE |
| **5** | **MODEL BUILDING**   * Data Division * Classification Models |
| **6** | **DATA VIZUALIZATION**   * Bar graph * Confusion Matrix |
| **7** | **PROJECT OUTPUTS** |
| **8** | **RECOMMENDATIONS AND CONCLUSION** |

**PROJECT INTRODUCTION**

The project documentation describes the project on which focuses on developing and identifying the most accurate model for detecting emotions from textual data. The aim was to select the best accuracy of classification models.

**Objective:**

By achieving the objective of identifying the best-performing classification model, the project aims to contribute to the development of effective text classification systems for accurately understanding and analysing emotions conveyed in textual data.

**Methodology:**

Project is developed using agile methodologies, allowing for flexibility in adjusting to new insights gained during the project, accommodating changes in data characteristics, model performance. Agile encouraged collaboration and communication within the team, making informed decisions throughout the project lifecycle.

**Software Requirements:**

* Operating System: The project is developed on Windows.
* Python: The project is implemented using Python programming language, along with libraries such as NumPy, pandas, scikit-learn, seaborn, matplotlib and other machine learning frameworks.
* Integrated Development Environment (IDE): Popular IDE Jupyter Notebook used for code development and experimentation.
* Version Control: Git and platforms like GitHub and Git Desktop for uploading and pushing code.
* Conferencing applications: Microsoft Teams is used for collaboration among team members.

**Method:**

1. Data Preparation: Preprocessing the Emotion dataset to clean, tokenize, and normalize the text data for effective model training.
2. Model Selection: Experimenting with different classification algorithms such as Support Vector Machines (SVM), Naive Bayes, Random Forests, XG Boost and Logistic Regression to identify the best-performing model.
3. Training and Validation: Training the selected models on the training dataset and validating their performance using techniques like cross-validation to ensure robustness and prevent overfitting.
4. Hyperparameter Tuning: Optimizing the hyperparameters of the selected models to enhance their performance and generalization capabilities.
5. Evaluation Metrics: Assessing the performance of the models using evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrices to determine their effectiveness in classifying emotions accurately.
6. Comparative Analysis: Conducting a comparative analysis of the performance of different models to identify the model that yields the highest accuracy and robustness in classifying emotions on the Emotion dataset.

**DATA COLLECTION**

For a text classification the data collection phase is critical to ensure that the model is trained on high-quality and representative data.

**Customer Feedback**: Collect data from customer reviews, feedback forms, and surveys from platforms like Amazon or your own business's feedback channels.

Two sources where Emotion Training Dataset were received:

* **Mail:** Datasets for Text Classification was received through mail
* **GitHub:** Emotion training dataset were uploaded in GitHub also in CSV format

**DATA PRE-PROCESSING**

* **Lower Case:** Convert all characters in the text to lower case to ensure uniformity. This helps in treating words like "Happy" and "happy" as the same word.
* **Remove Links:** Eliminate URLs from the text. Links are usually not useful for text classification and can add noise.
* **Remove New Lines:** Remove newline characters (\n) to ensure that the text is continuous and to avoid unnecessary line breaks in the data.
* **Words Containing Numbers:** Remove words that contain numbers, as they often do not contribute to the understanding of the text's emotional content.
* **Extra Spaces:** Remove extra spaces, including leading, trailing, and multiple spaces between words, to clean up the text.
* **Special Characters:** Remove special characters such as punctuation marks, symbols, and emojis that do not add significant value to the text classification task.
* **Removal of Stop Words:** Remove common words like "and," "the," "is," etc., which do not carry significant meaning and are common across all documents.
* **Stemming:** Reduce words to their base or root form. For example, "running" becomes "run". This helps in treating different forms of a word as the same token.
* **Lemmatization:** Similar to stemming, but it reduces words to their base form (lemma) with a focus on returning valid words. For example, "better" becomes "good".
* **Eliminating Features with Extremely Low Frequency:** Remove words or tokens that appear very rarely in the dataset. These rare terms can add noise and do not contribute significantly to the model's learning.
* **Use Complex Features:** n-grams and Part of Speech Tags:
  + n-grams: Sequences of 'n' consecutive words. For example, bi-grams (n=2) in the sentence "I am happy" are "I am" and "am happy". N-grams capture context and relationships between words.
  + Part of Speech (POS) Tags: Label each word with its part of speech (e.g., noun, adjective). POS tags provide grammatical context, which can help in understanding the structure and meaning of the text.

These preprocessing steps and feature engineering techniques are essential to clean and prepare the text data for better performance in text classification tasks. They help in reducing noise, standardizing the text, and extracting meaningful features that improve the model's accuracy and generalization.

**FEATURE ENGINEERING**

**TF-IDF (Term Frequency-Inverse Document Frequency)**:

TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents (corpus). It is calculated as the product of two statistics: Term Frequency (TF), which measures how frequently a term appears in a document, and Inverse Document Frequency (IDF), which measures how important a term is by considering how common or rare it is across the entire corpus.

**Uses**:

* **Improved Relevance**: TF-IDF emphasizes terms that are important (frequent in a document) but not too common across all documents, helping to highlight distinctive words that are more likely to be relevant to the specific content of a document.
* **Dimensionality Reduction**: By assigning lower weights to common words, TF-IDF effectively reduces the noise in the data, which can help improve the performance of machine learning models by focusing on the more informative terms.

**SMOTE (Synthetic Minority Over-sampling Technique)**:

SMOTE is an over-sampling technique used to address class imbalance in datasets. It works by generating synthetic samples for the minority class by interpolating between existing minority class examples. This helps balance the dataset without merely duplicating minority class instances.

**Uses**:

* **Improves Model Performance**: By balancing the class distribution, SMOTE helps machine learning models perform better on the minority class, reducing bias towards the majority class and improving overall classification performance.
* **Prevents Overfitting**: Unlike simple over-sampling methods that duplicate minority class instances, SMOTE generates new, synthetic samples, which can help prevent overfitting and make the model more generalizable to unseen data.

**MODEL BUILDING**

* **DATA DIVISION:**

The process of splitting the collected dataset into separate subsets for training, validation, and testing. This step is crucial to ensure that the machine learning model is properly evaluated and generalized.

* **Training Set:** The portion of the data used to train the model. Typically, this comprises 60-80% of the total dataset. The model learns the patterns and relationships in the data from this subset.
* **Test Set:** Another portion of the data (typically 10-20%) that is kept separate from the training and validation sets. This set is used to evaluate the final model's performance, providing an unbiased assessment of how well the model will perform on new, unseen data.
* **Validation Set:** A smaller portion of the data (usually 10-20%) used to tune the model's hyperparameters and make decisions about model architecture. It helps in evaluating the model's performance during the training phase and prevents overfitting by providing a checkpoint for adjustments.

**Advantages:**

* **Prevents Overfitting:** By using a validation set, the model can be fine-tuned without being overly specific to the training data, thus improving its ability to generalize to new data.
* **Unbiased Evaluation:** The test set provides an unbiased evaluation of the final model, ensuring that the reported performance metrics are reflective of real-world performance.
* **CLASSIFICATION MODEL**
* **Logistic Regression:**

Logistic Regression is a linear model used for binary classification tasks. It predicts the probability that a given input belongs to a particular class by fitting data to a logistic function (sigmoid curve).

Advantages:

**Interpretable:** Logistic regression provides clear insights into the impact of each feature on the prediction, making it easy to interpret.

**Efficiency:** It is computationally efficient and performs well with linearly separable data.

* **Naive Bayes:**

Naive Bayes is a probabilistic classifier based on Bayes' Theorem. It assumes that the features are conditionally independent given the class, which simplifies the computation.

Advantages:

**Simple and Fast:** Naive Bayes is easy to implement and computationally efficient, making it suitable for large datasets.

**Good Performance with Small Data:** It performs well with small training datasets and is particularly effective for text classification tasks.

* **Random Forest:**

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Advantages:

**High Accuracy:** Random Forest generally provides high accuracy and robustness by averaging the results of multiple trees, reducing overfitting.

**Feature Importance:** It can handle a large number of features and provides estimates of feature importance, aiding in feature selection.

* **XG Boost** (Extreme Gradient Boosting):

XG Boost is an optimized gradient boosting algorithm designed to improve speed and performance. It builds an ensemble of decision trees in a sequential manner, optimizing for accuracy and speed.

Advantages:

**High Performance:** XG Boost often achieves superior performance due to its advanced optimization techniques and regularization methods.

**Scalability:** It is highly scalable and can handle large datasets efficiently, leveraging parallel and distributed computing.

**SVM (Support Vector Machine):**

SVM is a supervised learning model used for classification and regression tasks. It finds the hyperplane that best separates the data into different classes by maximizing the margin between the closest points of the classes (support vectors).

Advantages:

**Effective in High Dimensions:** SVM works well in high-dimensional spaces and is effective when the number of dimensions exceeds the number of samples.

**Robust to Overfitting:** With proper kernel choice and regularization, SVM is robust to overfitting, especially in high-dimensional feature spaces.

Each of these models has its strengths, making them suitable for different types of classification tasks and data characteristics. The choice of model depends on the specific requirements and constraints of the project, such as interpretability, computational resources, and the nature of the dataset.

**DATA VISUALIZATION**

* **Bar Graph**:

A bar graph (or bar chart) is a graphical representation of data using rectangular bars. The length of each bar is proportional to the value it represents. Bar graphs are used to compare different categories of data.

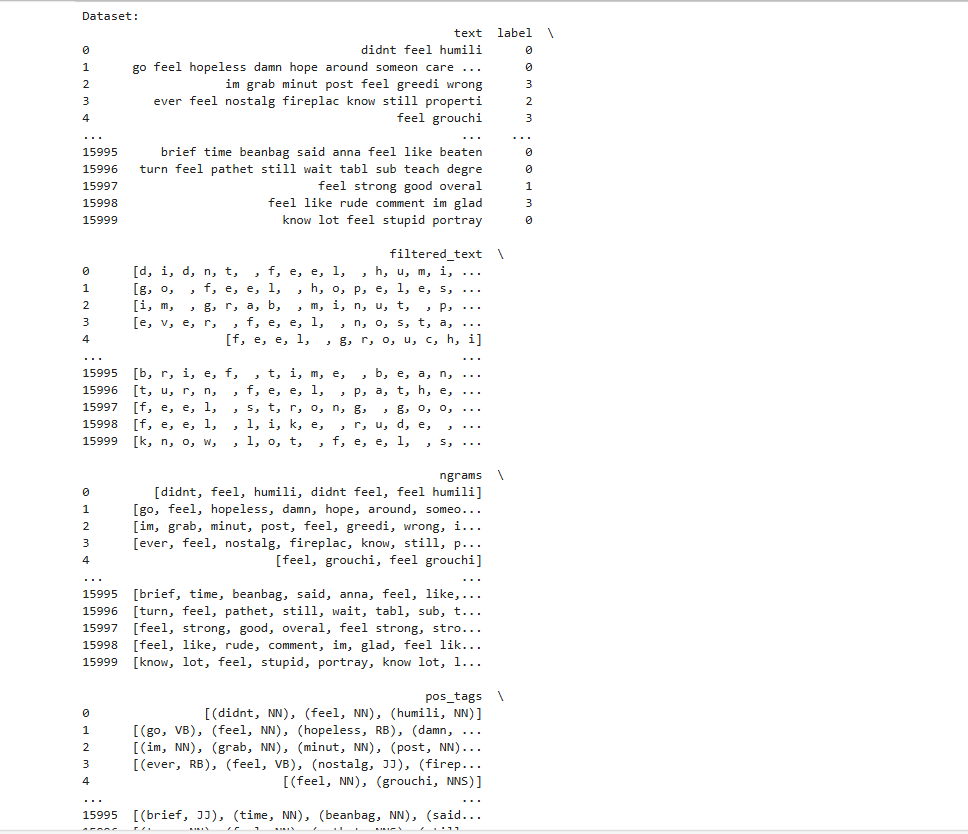
* + **Advantages**:
    1. **Comparison**: Bar graphs are excellent for comparing quantities across different categories, making it easy to visualize differences and trends.
    2. **Clarity**: They provide a clear and straightforward way to present categorical data, allowing for quick interpretation of the information.
* **Confusion Matrix**:

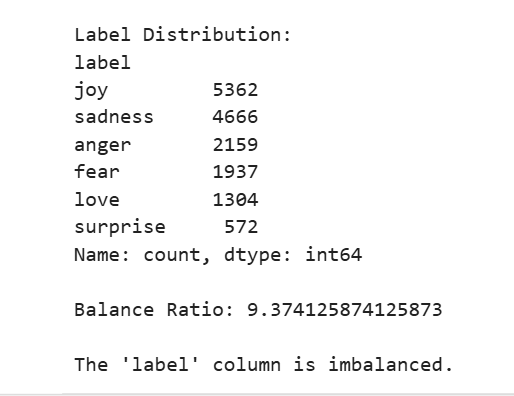
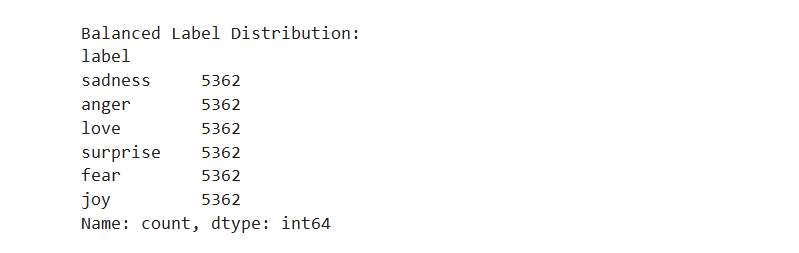
A confusion matrix is a table used to evaluate the performance of a classification model. It displays the true positive, true negative, false positive, and false negative predictions, allowing for a detailed analysis of the model's accuracy.

* + **Advantages**:
    1. **Detailed Performance Metrics**: The confusion matrix provides a comprehensive view of the model’s performance, highlighting specific areas where it is performing well or needs improvement.
    2. **Metric Calculation**: It facilitates the calculation of various performance metrics such as accuracy, precision, recall, and F1-score, which are essential for understanding the effectiveness of the classification model.

These visualizations were used in “Text Classification” project essential tools in data analysis and model evaluation, providing insights that are crucial for making informed decisions and improving model performance.

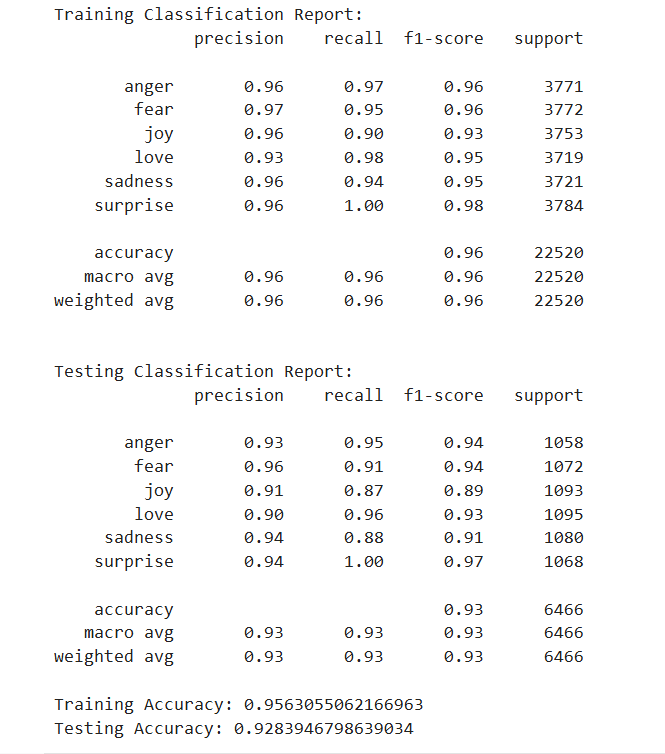
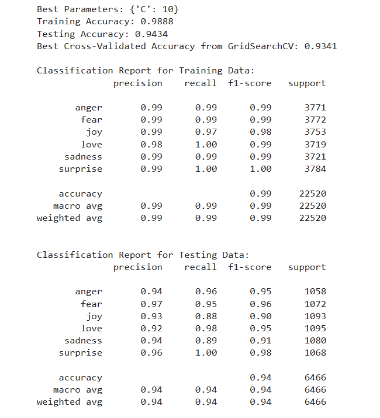
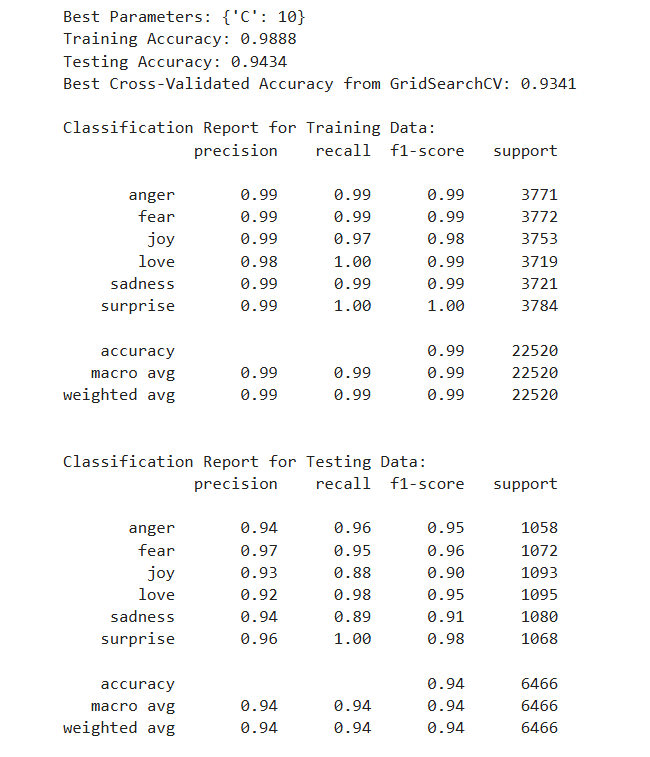
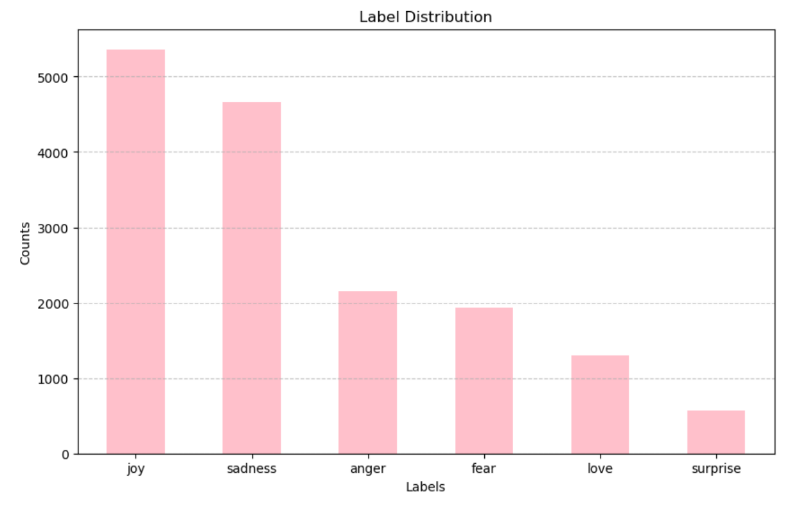
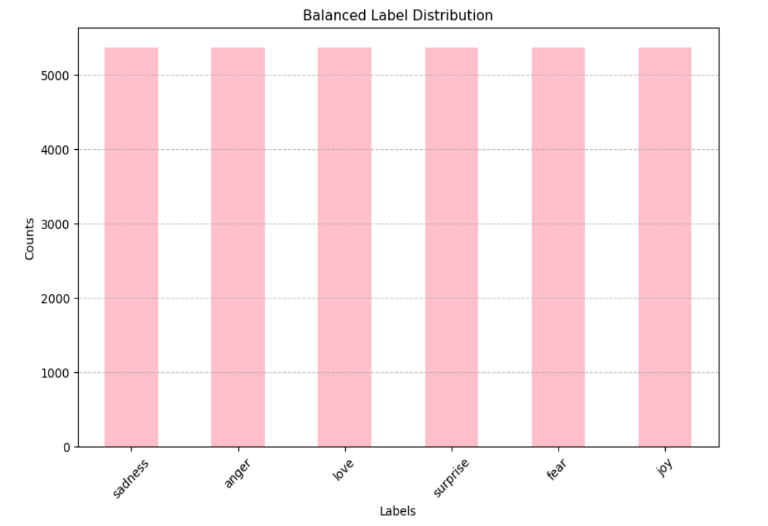
**PROJECT OUTPUTS**

Output Pre-processing Emotion Training Dataset



Balanced Data

Imbalanced Data

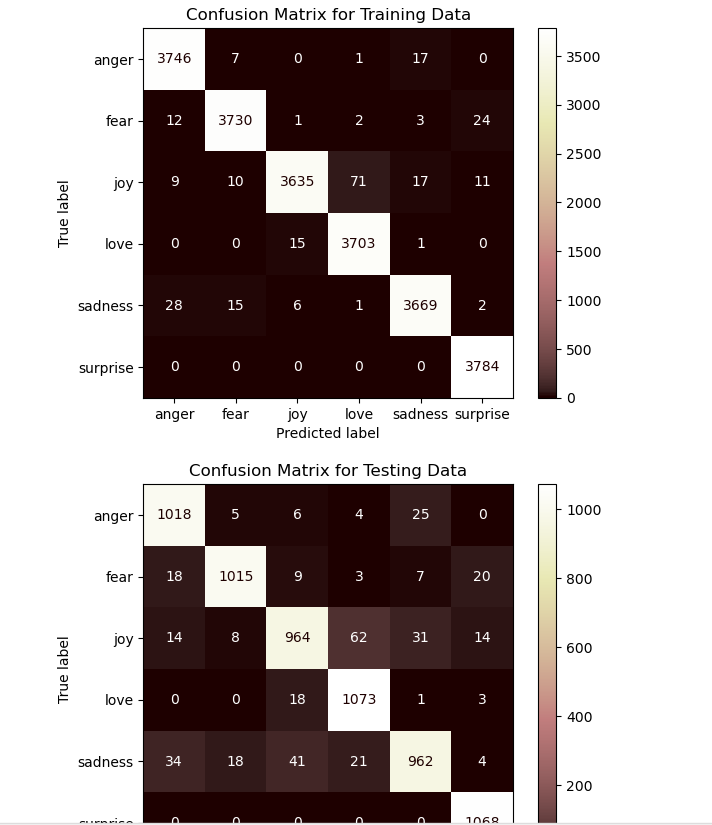
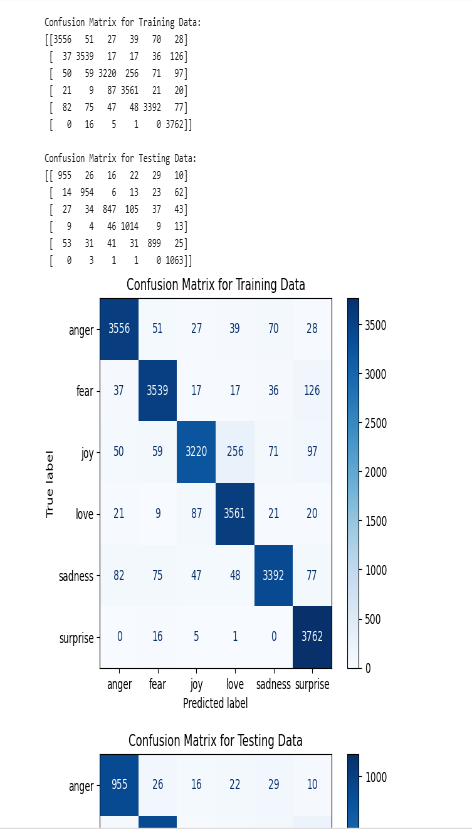
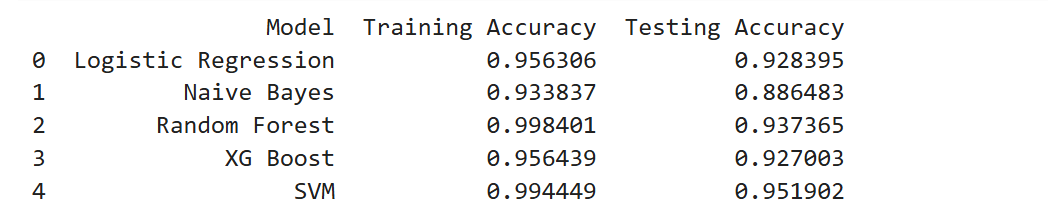
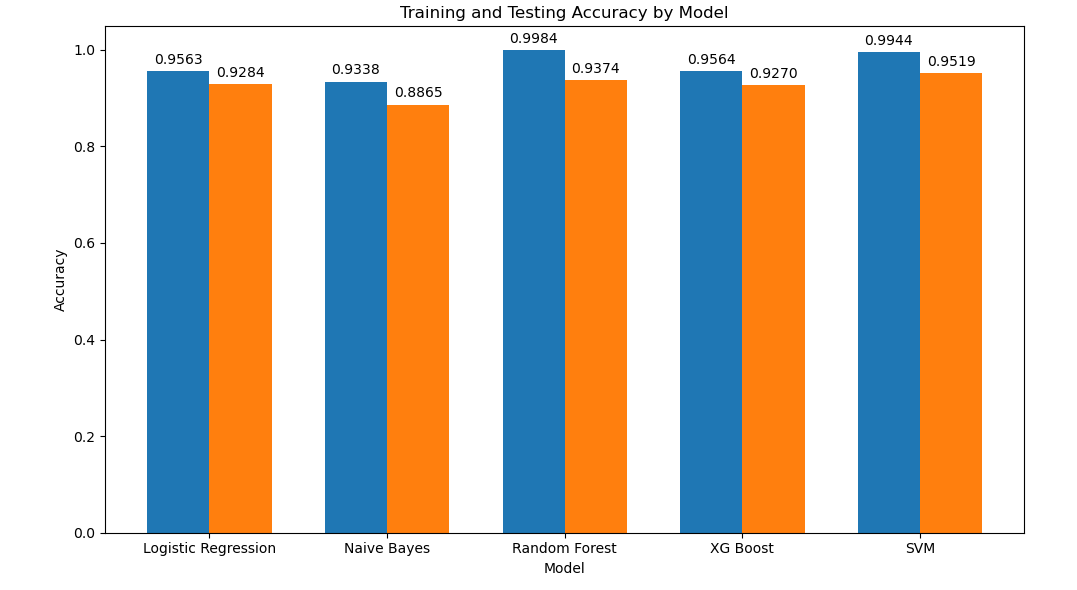
****

Classification Report

Imbalanced Data in Bar Graph Representation

Imbalanced Data in

Bar Graph Visualization

****

All Model Accuracy represented in Bar graph

All Models Accuracy

Confusion Matrix

**RECOMMENDATIONS AND CONCLUSION**

* **Data Enhancement**:
* **Augment Dataset**: Collect more data from diverse sources to improve the robustness of the model, especially for underrepresented emotion categories.
* **Regular Updates**: Continuously update the dataset to include recent data, ensuring the model remains relevant and accurate over time.
* **Model Optimization**:
* **Hyperparameter Tuning**: Perform extensive hyperparameter tuning for each model to achieve optimal performance.
* **Ensemble Methods**: Consider using ensemble methods like stacking or blending to combine the strengths of multiple models for improved accuracy.
* **Advanced Features**:
* **Incorporate Advanced NLP Techniques**: Explore advanced NLP techniques such as BERT or GPT-based embeddings to capture more complex language patterns and nuances.
* **Feature Engineering**: Implement additional feature engineering techniques like sentiment scores, word embeddings, and topic modelling to enhance model inputs.
* **Model Evaluation**:
* **Cross-Validation**: Use cross-validation techniques to ensure the model’s performance is consistent and reliable across different subsets of the data.
* **Real-world Testing**: Evaluate the model on real-world, unseen data to assess its practical applicability and robustness.
* **Deployment and Monitoring**:
* **Scalable Infrastructure**: Deploy the model on a scalable infrastructure to handle varying loads efficiently.
* **Continuous Monitoring**: Implement monitoring tools to track model performance in production and trigger re-training if performance degrades.

**CONCLUSION**

The text classification project on the emotion training dataset aimed to identify and categorize emotions in textual data. After comparing multiple models, including Logistic Regression, Naive Bayes, Random Forest, XG Boost, and Support Vector Machine (SVM), it was concluded that SVM provided the best performance in both training and testing phases.

SVM's effectiveness in handling high-dimensional data and its robustness to overfitting made it the optimal choice for this task. This project underscores the importance of selecting the right model and applying thorough preprocessing techniques to achieve high accuracy in emotion detection.

The successful implementation of the SVM model can greatly benefit applications such as customer sentiment analysis, personalized marketing, and improved customer support by accurately identifying emotional tones in text data.

THANK YOU