



slington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CU6051NI Artificial Intelligence

75% Individual Coursework

Submission: Final Submission

Academic Semester: Autumn Semester 2025

Credit: 15 credit semester long module

Student Name: Ranjana Silwal

London Met ID: 23050372

College ID: NP01CP4A230363

Assignment Due Date: 21/01/2026.

Assignment Submission Date: 21/01/2026

Submitted To: Mr. Binod Bhattarai

GitHub Link	https://github.com/ranzanas/Emotion-Detection-through-Text
--------------------	---

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

1. Introduction	1
1.1. Explanation of topic/AI concepts used	1
1.2. Introduction to chosen problem domain/topic	1
2. Background	2
2.1. Research Work done in Coursework 1	2
2.1.2. Detection of emotion by text analysis using machine learning	2
2.1.2. Journal Article on Comparative Studies on Emotion Detection Algorithms	3
2.1.3. Comparative Studies — Naive Bayes versus SVM.	4
2.1.4. Emotion Detection in Text: Leveraging Machine Learning for Sentiment and Emotional Intelligence Analysis.....	5
2.1.5. Surveys and Systematic Reviews	5
2.1.6. Academic Project Report in Emotion Detection From Text	6
Key findings from research	6
3. Solution	7
3.1. Explanation of used AI algorithm	7
3.1.1. Naïve Bayes	7
3.1.2. Logistic Regression	8
3.1.3. Support Vector Machine.....	9
3.2. Pseudocode.....	10
3.2.1. Pseudocode of Naïve Bayes for emotion classification	10
3.2.2. Pseudocode of SVM for emotion classification	12
3.2.3. Pseudocode of Logistic Regression for Emotion Classification.....	13
3.3. Flowchart	15
3.3.1. Flowchart for Naïve Bayes Emotion Classification	15
3.3.2. Flowchart for Logistic Regression Emotion Classification	17
3.3.3. Flowchart for SVM Emotion Classification.....	18
3.4. Development Process.....	19
3.4.1. Import necessary libraries.	19
3.4.2. Load the dataset	19
3.4.3. Dataset Understanding	20
3.4.4. Data Preprocessing.....	21
3.4.5. Label Encoding	22

3.4.6. Train Test Split	22
3.4.7. Feature Extraction using TF-IDF Vectorizer	23
3.4.8. Model Training	23
3.4.9. Model Evaluation.....	24
3.4.9.1. Evaluation Result of Naïve Bayes Model.....	25
3.4.9.2. Evaluation Result of Logistic Regression Model.....	26
3.4.9.3. Evaluation Result of SVM Model:	27
3.4.9.4. Overall Performance Comparison of Emotion Detection Models	28
3.4.10. Confusion Matrix	29
3.4.10.1. Confusion Matrix Visualization of Naïve Bayes through Heatmap	29
3.4.10.2. Confusion Matrix visualization of LR through heatmap.....	30
3.4.10.3. Confusion Matrix Visualization of SVM through heatmap	31
3.4.11. Cross Validation Score.....	32
3.4.12. Model Optimization with Hyperparameter Tuning Using GridSearch CV ...	35
4. Conclusion	38
4.1. Analysis of Work Done	39
4.2. How the application addresses real world problem.....	39
4.3. Further Work.....	40
References.....	41

Table of Figures

Figure 1: Screenshot of journal article researched for CW - 1	3
Figure 2: Screenshot of research work done in CS -1	4
Figure 3: Screenshot of research done in CW - 1 (Conference Paper)	5
Figure 4: Screenshot of research done in CW -1 (Academic Report)	6
Figure 5: Linear Regression vs Logistic Regression	8
Figure 6: SVM using graph.....	9
Figure 7: : Flowchart of Naive Bayes for Emotion Classification	16
Figure 8: Flowchart of Logistic Regression Emotion Classification	17
Figure 9: Flowchart of SVM emotional classification	18
Figure 10: Screenshot of dataset Loading.....	19
Figure 11: Screenshot of Emotion Distribution Analysis.....	20
Figure 12: Screenshot of text length analysis.....	21
Figure 13: Screenshot of Data Preprocessing.....	21
Figure 14: Screenshot of Label Encoding	22
Figure 15: Screenshot of Train Test Split.....	22
Figure 16: Screenshot of Feature Extraction	23
Figure 17: Screenshot of Model Training	23
Figure 18: Naive Bayes Model Evaluation Result	25
Figure 19: Logistic Regression Model Evaluation Result	26
Figure 20: Evaluation Result of Support Vector Machine	27
Figure 21: Overall Performance Comparison of Emotion Detection Models	28
Figure 22: Graphical Visualization of Performance Comparison of Emotion Detection Models.....	28
Figure 23: Confusion Matrix heatmap of Naïve Bayes	29
Figure 24: Confusion Matrix heatmap of Logistic Regression	30
Figure 25: Confusion Matrix heatmap of SVM.....	31
Figure 26: Mean Cross Validation F1-Score with Standard Deviation.....	32
Figure 27: Cross-Validation F1-scores for Each Fold.....	33
Figure 28: Cross-Validation F1-score Comparison Using Box Plot	34
Figure 29: Best Hyperparameter values for models	35

Figure 30: Naive Bayes Performance Before vs After Hyperparameter Tuning	36
Figure 31: Logistic Regression Performance before vs after hyperparameter tuning....	37
Figure 32: SVM Performance Before vs After Hyperparameter Tuning	38

Table of tables

Table 1: Used libraries for development.....	19
--	----

1. Introduction

Artificial Intelligence(AI) is a emerging technology which lets computers and machines to think like a human, respond in human languages, learn from the information and new experience and provide near to accurate result (Cole Stryker, 2025). It is the technology that has made life easier in many sectors such as health, science, education and even day to day life. Machine Learning(ML) is one of the important branches of AI. It is the subset of AI. Machine Learning is the way of making computer automatically learn and improve. Machine Learning works on train and test split process, where certain amount of data is used for training and certain amount of data is used for testing the accuracy. There are many ML types such as: Supervised Learning, Unsupervised Learning, Semi-supervised Learning, Re-inforcement Learning, Deep Learning, Ensemble Learning.

1.1. Explanation of topic/AI concepts used

This project uses **Supervised Machine Learning Model**. Supervised Learning works on the basis of labelled data. It uses labelled dataset to train the model where it identifies underlying patterns and relationships and predict output on new real world data (Cole Styker, 2025). In this learning model the trained model is provided with the input and predicted data and the model learns the relationship between inputs and outputs which then further predicts the unseen data.

As the project is emotion detection through text, it solely focuses on analysing the textual data. This project also involves the concept of Natural Language Processing (NLP) because it focuses on the identifying the human emotions through the written languages. The study involves processing the text input and predict the emotions such as: **anger, fear, joy, love, surprise and sadness**. This is **multi-class text classification** problem where the text is classified into one of the several pre-defined emotional categories. For this kind of text classification the algorithms used are: Naïve Bayes, Support Vector Machine and Logistic Regression.

1.2. Introduction to chosen problem domain/topic

Today's era is the era of technology and there is increasing use of digital communication platforms such as social media platforms, online review platform, messaging and chatting

applications such as whatsapp, viber, messenger, digital journaling platforms and many more. These platforms generate large volume of textual data each and everyday. These generated text often reflect valuable emotional information of users. Such kind of textual data is essential to understand human behaviour, mental states, opinions and even mental condition they are dealing with.

Detecting emotions from text aims to predict the emotions that are expressed in text form. In this project, the emotions considered are - anger, sadness, love, joy, surprise and fear. When emotions are detected accurately, it can support various applications such as mental health monitoring, mood tracking, customer feedback analysis, sentiment analysis and so on.

However, there may arise challenges while detecting emotions from text because of ambiguity, informal language. Hence, supervised machine learning techniques can be useful to find patterns in text and predicting the output.

2. Background

2.1. Research Work done in Coursework 1

The rise of social media, blogs, forums, and messaging platforms generates huge amounts of text data every day. This has led in rapid research into automated emotion detection since manually analyzing such vast amounts of text is not practical and efficient as well.

During coursework 1 many research papers were read about emotion detection through text. The analysis of such research are described below:

2.1.2. Detection of emotion by text analysis using machine learning

[Journal Article published in Frontiers in Psychology](#) (Kristína Machová, 2023)

What was done in the study:

This research focused on detecting human emotions from text data using supervised machine learning techniques. The authors applied text preprocessing steps like tokenization, normalization, and noise removal before turning text into numerical forms. They implemented and evaluated several machine learning models, including Naïve

Bayes, Support Vector Machines, and Neural Networks, for multi-class emotion classification.

The study also showed how the trained emotion detection model was used in a web application and chatbot system, emphasizing practical real-world use. The results indicated that traditional ML models are still effective, while neural networks performed better in recognizing complex emotions.

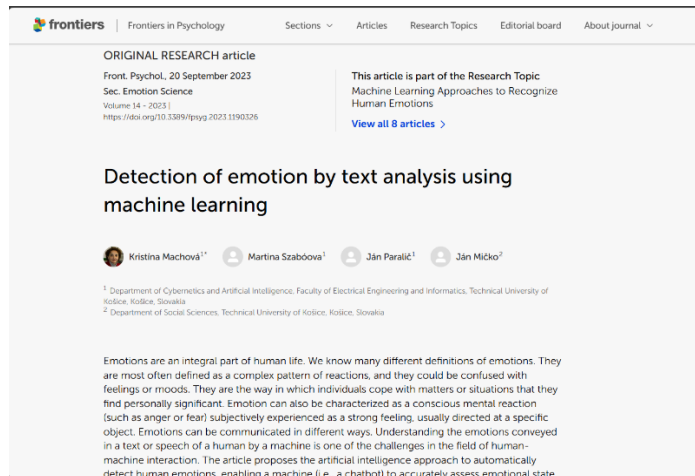
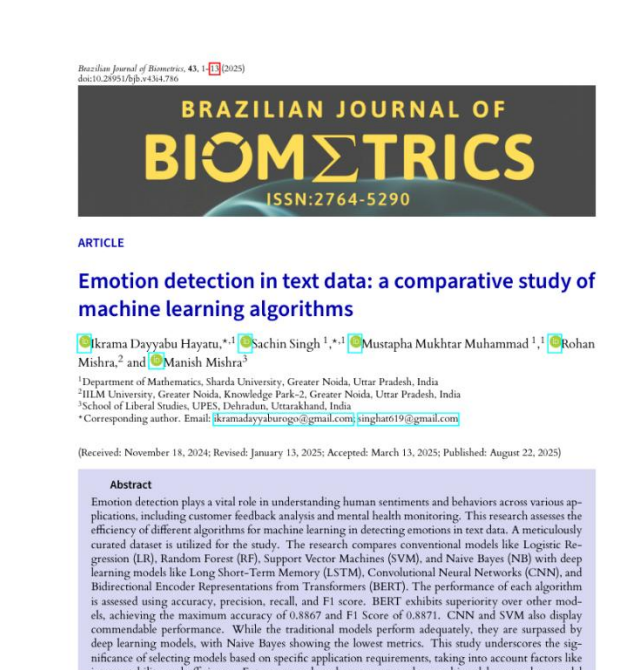


Figure 1: Screenshot of journal article researched for CW - 1

2.1.2. Journal Article on Comparative Studies on Emotion Detection Algorithms

Published on : [Brazilian Journal of Biometrics](#) (Ikrama Dayyabu Hayatu, 2025)



What was done in this study:

In this article, the comparative study of various machine learning models and deep learning models for emotion detection is included. The machine learning models like Logistic Regression, Naïve Bayes, SVM, Random Forest are compared with CNN, LSTM and BERT. To compare the results and evaluate the performance, standard evaluation metrics such as accuracy, precision, recall and F1-score are used. In this study, deep learning models are said to perform better but machine learning models are also seen as efficient. This study mainly emphasizes that model should be selected on the basis of application needs not just accuracy.

2.1.3. Comparative Studies — Naive Bayes versus SVM.



Figure 2: Screenshot of research work done in CS -1

What was done in the study:

The study published in [Edumatic Journal Pendidikan Informatika](#) showed a complete comparison of works in which Naive Bayes and SVM are implemented for the task of emotion recognition in social media texts. The study shows that SVM has a greater accuracy than Naive Bayes and this is observed when the TF-IDF and BoW techniques are implemented. The study confirms how the choice of technique, the method of text

preprocessing, and selection of features affect the performance of the classification (Rio Ferdinand Putra Pratama, 2025).

2.1.4. Emotion Detection in Text: Leveraging Machine Learning for Sentiment and Emotional Intelligence Analysis

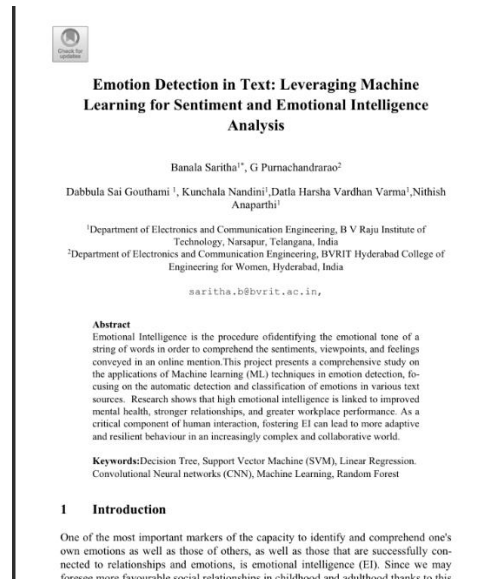


Figure 3: Screenshot of research done in CW - 1 (Conference Paper)

[Link](#)

What was done in the study:

In this conference paper, emotion detection from text is focused as a part of emotional intelligence and sentiment analysis. This paper gives more emphasis on how emotion detection is important for online communication, mental health and social interactions. There has been use of traditional machine learning models such as SVM, Logistic Regression, Random Forest and Decision Trees. The role of text preprocessing and feature extraction has been highlighted in this paper for development of emotion detection models (Saritha et al., 2025).

2.1.5. Surveys and Systematic Reviews

The reviews covering a larger area of research pertaining to the detection of emotions from text explain in detail classifications of emotions and various research methodologies. These reviews document that among the baselines ML models introduced in the

research, SVM and Naïve Bayes are the most popular. The reviews also document the vital importance of feature design and data preparation to achieve the best performance of the model (Sheetal Kusal, 2022).

2.1.6. Academic Project Report in Emotion Detection From Text

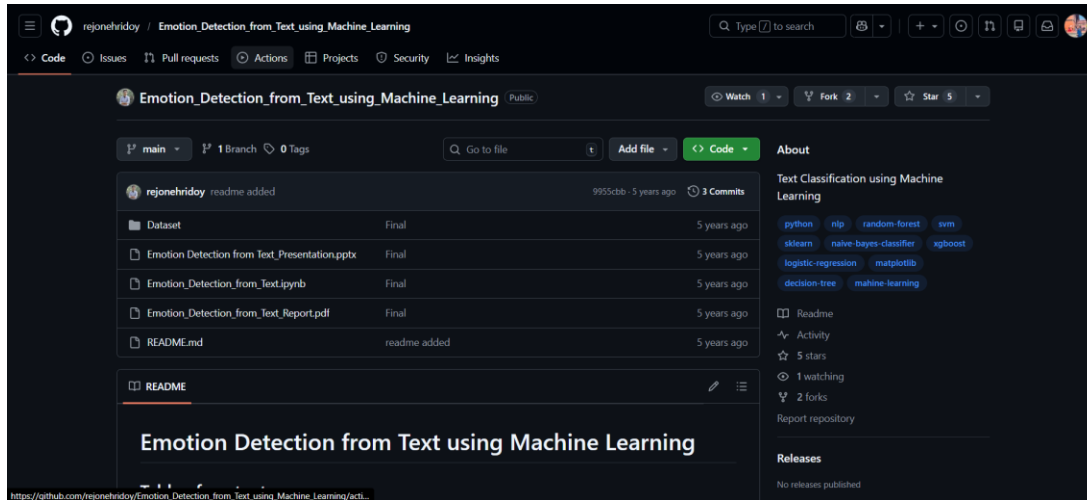


Figure 4: Screenshot of research done in CW -1 (Academic Report)

[Github Link](https://github.com/rejonchridoy/Emotion_Detection_from_Text_using_Machine_Learning)

What was done in the study:

This is the academic project report for emotion detection from text. It covers the collection of data, preprocessing, feature extraction and training of various machine learning models. 5 machine learning models : SVM, Logistic Regression, Naïve Bayes, Random Forest and Decision Trees are compared in this report. This report includes visual representation of confusion matrices, accuracy results and emotion distribution graphs. The main focus of this report is to perform experimental evaluation and result comparison.

Key findings from research

The key findings from research are:

- Classical ML models such as Naïve Bayes, SVM, and Logistic Regression are widely considered and widely used for text emotion detection.
- High quality preprocessing, feature extraction methods and dataset characteristics all contribute significantly to performance.

- No one single model is best, so systematic comparison can serve academic research well.
- But, most studies on sentiment analysis or emotion detection with small scale classes of emotion leaves room for multi-class emotion detection for a larger emotion set including anger, joy, fear, sadness, anxiety, and surprise.

3. Solution

3.1. Explanation of used AI algorithm

The AI algorithms that were used in this coursework are described as below:

3.1.1. Naïve Bayes

Naïve Bayes is a supervised machine learning algorithm that uses principles of probability to perform the classification of text. It is also known as probabilistic model and it is mostly used for working with natural language text documents (Kavlakoglu, 2025). This classifier is based on Bayes' Theorem or Bayes' rule represented by following formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

There are three types of Naïve Bayes Classifier:

- a. Gaussian Naïve Bayes
- b. Bernoulli Naïve Bayes
- c. Multinomial Naïve Bayes

For model training in emotion detection through text, **Multinomial Naïve Bayes** is used. This type of probabilistic classifier operates under the assumption that features follow a multinomial distribution, which is suited for discrete frequency counts (e.g., word frequencies in documents). It is widely used in text classification, including spam filtering (Kavlakoglu, 2025).

The key evaluation metrics for Naïve Bayes includes: Accuracy, Precision, Recall, F1-Score, Confusion Matrix.

3.1.2. Logistic Regression

Logistic Regression is supervised machine learning algorithm that predicts categorical outcome. Though it has the name regression it is used for classification task. Unlike linear regression it does not predict continuous value but the categorical outcome. For example: linear regression predict the exam score but logistic regression predict whether the student is pass or fail (Fangfang lee, 2026). Logistic regression uses sigmoid function represented by: $\phi(y) = \frac{1}{1+e^{-y}}$

Types of Logistic Regression:

- Binomial Logistic Regression
- Multinomial Logistic Regression
- Ordinal Logistic Regression

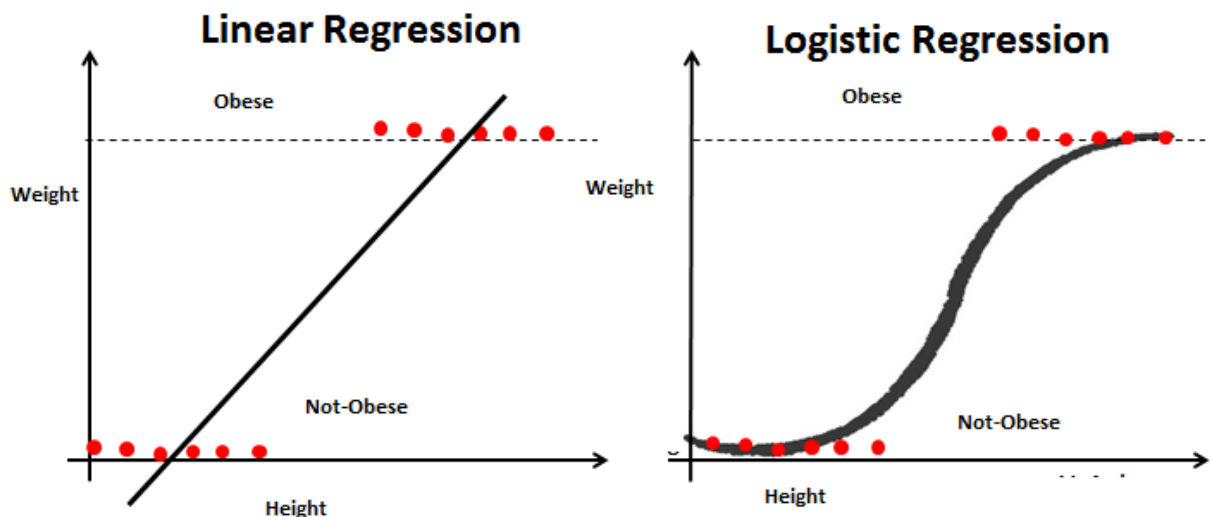


Figure 5: Linear Regression vs Logistic Regression

For emotion detection through text, multi class logistic regression is used since there are multiple emotion classes. Here, the model trains one classifier per emotion, each classifier learns how strongly words contribute to that emotion and emotion with highest probability is selected as final prediction.

Logistic Regression is commonly used in:

- Fraud Detection

- Disease Prediction
- Churn Prediction

The key evaluation metrics include: Recall, Accuracy, Precision and F1-Score and Confusion Matrix.

3.1.3. Support Vector Machine

A support vector machine is a supervised machine learning algorithm where data is classified by finding an optimal line or hyperplane. That optimal line or hyperplane maximizes the distance between each class in an N-dimensional space. SVM are commonly used in classification problem within machine learning. It can handle both linear and non-linear classification task.

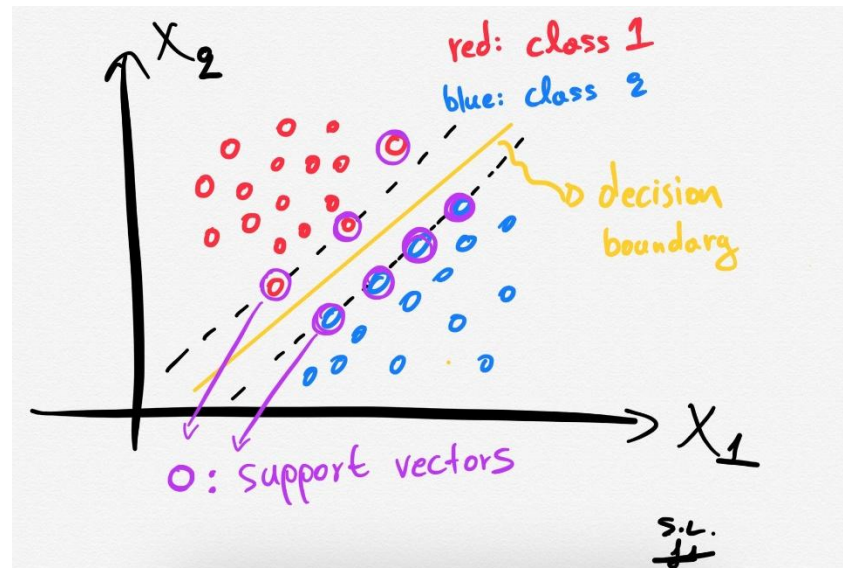


Figure 6: SVM using graph

By identifying the ideal hyperplane that optimizes the margin between the nearest data points of opposing classes, they are able to discriminate between two classes. Whether the hyperplane is a plane in an n-dimensional space or a line in a 2-dimensional space depends on the quantity of features in the input data. The approach can determine the optimal decision border between classes by maximizing the margin between points, since numerous hyperplanes can be identified to differentiate classes. As a result, it can accurately predict categorization and generalize well to fresh data. Because they pass

through the data points that establish the maximum margin, the lines that are next to the ideal hyperplane are referred to as support vectors (Eda Kavlakoglu, 2025).

The types of Support Vector Machine are:

- Linear SVM
- Non-Linear SVM
- Support Vector Regression

For emotion detection through text, Linear SVM is used. Maximizing the margin—the separation between the decision border and the nearest data points from each class—is the primary objective of linear support vector machines. The decision boundary is defined in large part by these nearest points, which are referred to as support vectors. A text is classified based on which side of this hyperplane it falls on.

The key evaluation metrics include: Recall, Accuracy, Precision and F1-Score and Confusion Matrix.

3.2. Pseudocode

3.2.1. Pseudocode of Naïve Bayes for emotion classification

BEGIN

IMPORT necessary libraries

LOAD emotion dataset

DISPLAY dataset shape and data types

CHECK for missing or null values

FOR each text entry in dataset:

 Convert text to lowercase

 Remove URLs

 Remove punctuation, numbers, and special characters

 Remove extra whitespace

END FOR

ENCODE emotion labels into numerical values using label encoding

SPLIT dataset into training set and testing set

INITIALIZE TF-IDF vectorizer
FIT TF-IDF on training text data
TRANSFORM both training and testing text into TF-IDF vectors
INITIALIZE Multinomial Naïve Bayes classifier with default alpha
TRAIN classifier using training TF-IDF features and labels
 CALCULATE prior probability for each emotion class
 CALCULATE conditional probability of each word given emotion class
FOR each text instance in testing set:
 COMPUTE probability score for each emotion class
 SELECT emotion with highest probability
END FOR
COMPUTE Accuracy
COMPUTE Precision, Recall, and F1-score
GENERATE Confusion Matrix
VISUALIZE Confusion Matrix using heatmap
PERFORM k-fold cross-validation ($k = 5$)
 FOR each fold:
 TRAIN model on training folds
 EVALUATE model on validation fold
 COMPUTE weighted F1-score
 END FOR
COMPUTE mean cross-validation score
APPLY hyperparameter tuning using GridSearchCV
 DEFINE range of alpha values
 PERFORM grid search with cross-validation
 SELECT best alpha value
TRAIN Naïve Bayes model using best alpha
RE-EVALUATE tuned model

COMPUTE Accuracy, Precision, Recall, and F1-score

VISUALIZE before vs after tuning performance using bar chart

END

3.2.2. Pseudocode of SVM for emotion classification

BEGIN

IMPORT necessary libraries

LOAD emotion dataset

DISPLAY dataset shape and data types

FOR each text entry in dataset:

Convert text to lowercase

Remove URLs

Remove punctuation, numbers, and special characters

Remove unnecessary whitespace

END FOR

ENCODE emotion labels into numeric values

SPLIT dataset into training and testing sets

INITIALIZE TF-IDF vectorizer

FIT TF-IDF vectorizer on training data only

TRANSFORM training and testing text into TF-IDF feature vectors

INITIALIZE Linear Support Vector Classifier (LinearSVC)

TRAIN SVM model using training TF-IDF features

IDENTIFY optimal linear decision boundaries

MAXIMIZE margin between emotion classes to improve generalization

FOR each text instance in testing set:

COMPUTE decision score for each emotion class

SELECT class with the highest decision score

END FOR

MEASURE Accuracy

MEASURE Precision, Recall, and F1-score

CONSTRUCT Confusion Matrix

DISPLAY evaluation results

PERFORM 5-fold cross-validation

COMPUTE weighted F1-score for each fold

COMPUTE average cross-validation score

APPLY hyperparameter tuning using GridSearchCV

DEFINE values for regularization parameter C and solver

PERFORM grid search with cross-validation

SELECT best hyperparameter combination

TRAIN Logistic Regression model using best parameters

RE-EVALUATE tuned model

COMPUTE Accuracy, Precision, Recall, and F1-score

VISUALIZE before vs after tuning results using grouped bar chart

END

3.2.3. Pseudocode of Logistic Regression for Emotion Classification

BEGIN

IMPORT necessary libraries

LOAD emotion dataset

DISPLAY dataset shape and data types

FOR each text entry in dataset:

Convert text to lowercase

Remove URLs

Remove punctuation, numbers, and special characters

Normalize whitespace

END FOR

CONVERT categorical emotion labels into numeric form using label encoding

SPLIT dataset into training and testing sets

INITIALIZE TF-IDF vectorizer with fixed vocabulary size

FIT vectorizer on training text

TRANSFORM training and testing text into numerical TF-IDF vectors

INITIALIZE Logistic Regression classifier

TRAIN model on training TF-IDF vectors

LEARN weight coefficients for each feature

OPTIMIZE weights to minimize classification error

FOR each text instance in testing set:

CALCULATE probability for each emotion class

ASSIGN emotion with the highest probability

END FOR

CALCULATE Accuracy

CALCULATE Precision, Recall, and F1-score

GENERATE Confusion Matrix

DISPLAY model performance metrics

PERFORM k-fold cross-validation ($k = 5$)

FOR each fold:

TRAIN model on training folds

EVALUATE model on validation fold

COMPUTE weighted F1-score

END FOR

COMPUTE mean cross-validation score

APPLY hyperparameter tuning using GridSearchCV

DEFINE range of C values

PERFORM grid search with cross-validation

SELECT best C value

TRAIN SVM model using optimal hyperparameter

RE-EVALUATE tuned model

COMPARE performance before and after tuning

VISUALIZE before vs after tuning performance

END

3.3. Flowchart

3.3.1. Flowchart for Naïve Bayes Emotion Classification

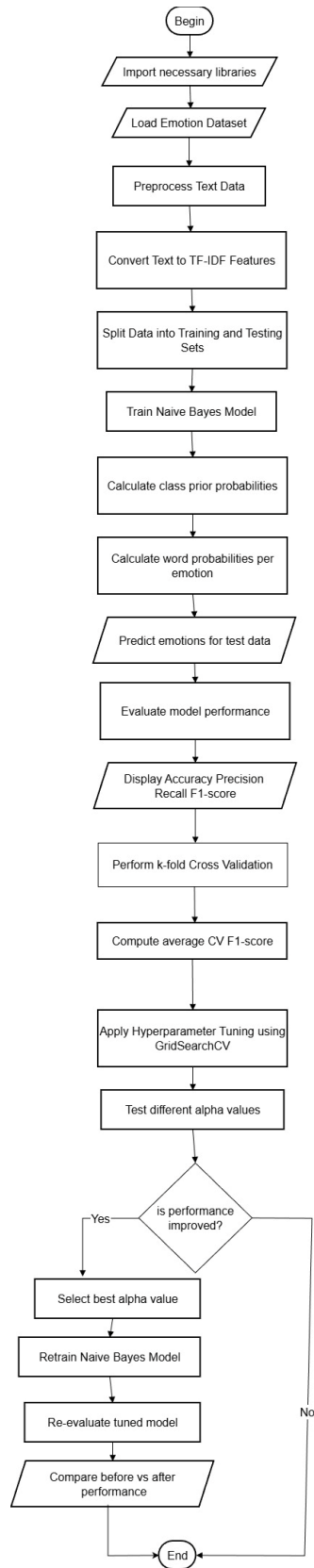


Figure 7: : Flowchart of Naive Bayes for Emotion Classification

3.3.2. Flowchart for Logistic Regression Emotion Classification

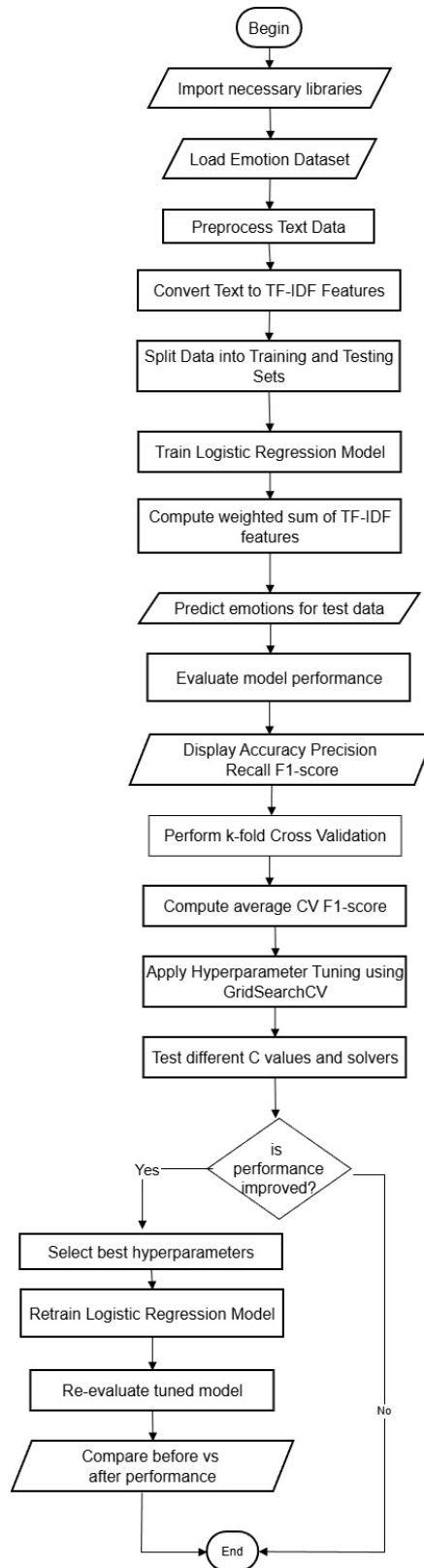


Figure 8: Flowchart of Logistic Regression Emotion Classification

3.3.3. Flowchart for SVM Emotion Classification

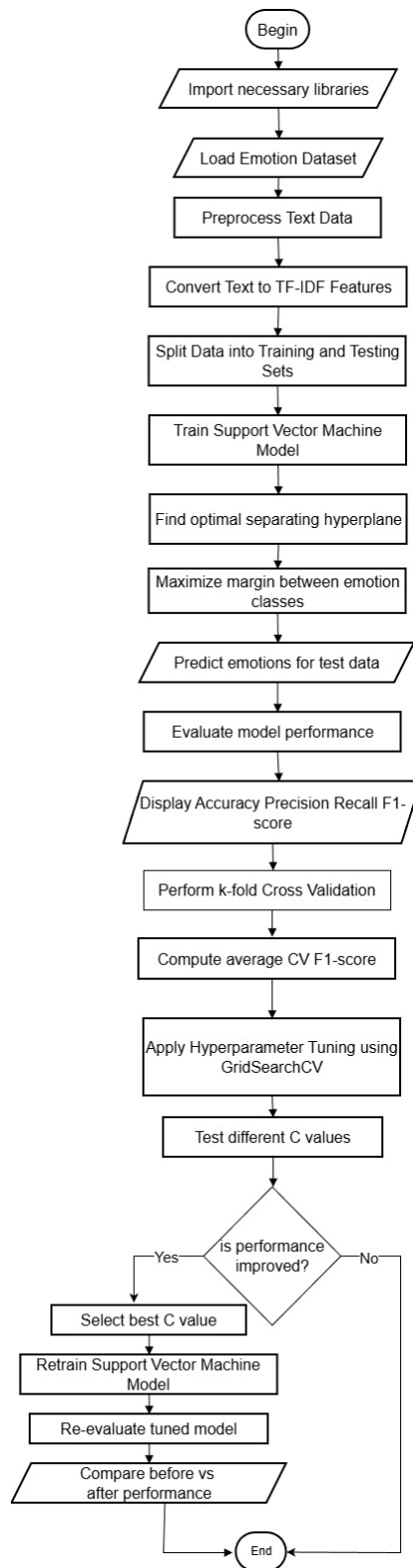


Figure 9: Flowchart of SVM emotional classification

3.4. Development Process

The development part was done in Jupyter Notebook using python language. The detailed explanation of development is presented below:

3.4.1. Import necessary libraries.

First of all, the libraries that were needed for development were imported. The libraries used are as follows:

Libraries	Purpose
numpy	Used for numerical computation support
pandas	Used for importing dataset and data manipulation
re	Used for text preprocessing
scikit-learn(sklearn)	Used for ML models, vectorization and evaluation
matplotlib	Used for plotting graphs
seaborn	Used for statistical data visualization

Table 1: Used libraries for development

Under these libraries sub modules such as **TfidfVectorizer** for feature extraction, **MultinomialNB**, **LogisticRegression**, and **LinearSVC** for model training, **GridSearchCV**, **cross_val_score** for model optimization and model evaluation and **classification_report** to analyze the performance of models from Scikit-learn were used.

3.4.2. Load the dataset

```
df = pd.read_csv("text_emotions.csv")
df.head()
```

	content	sentiment
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger

Figure 10: Screenshot of dataset Loading

Here, the dataset is loaded using pandas and stored in the variable named df. Later, to check if the dataset is imported, df.head() is used. df.head() provides the first five rows from dataset.

3.4.3. Dataset Understanding

To see how many columns and rows the used dataset contains, df.shape is used. and df.info() is used to see the datatypes and the memory usage. Also, to check if there contains any null values, df.isnull().sum() is used.

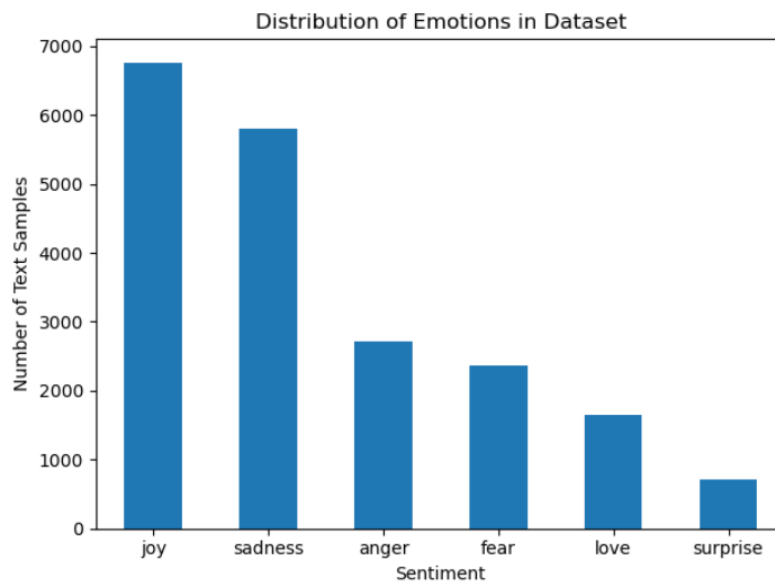


Figure 11: Screenshot of Emotion Distribution Analysis

To understand which emotion contains how much of rows, analysis of emotion distribution was done. This part was visualized using bar graph. It was done using matplotlib library and it was imported as plt alias. Many other graphs such as boxplot, histograms, piechart can be drawn using matplotlib library. It is the library widely used for plotting graphs.

```
[84]: df["text_length"] = df["content"].apply(lambda x: (len(x.split())))
      df["text_length"].describe()

[84]: count    20000.000000
      mean      19.135050
      std       10.972016
      min        2.000000
      25%       11.000000
      50%       17.000000
      75%       25.000000
      max       66.000000
      Name: text_length, dtype: float64
```

Figure 12: Screenshot of text length analysis

Along with emotion distribution, the text length in content column was also analysed using **.describe()**. The maximum length was of 66 words and minimum length was of 2 words. 25% of content contained 11 words, 50% contained 17 words and 75% contained 25 words.

3.4.4. Data Preprocessing

Data Preprocessing

```
[75]: def clean_text(text):
      text = text.lower()
      text = re.sub(r"http\S+", "", text)
      text = re.sub(r"^[a-z\s]", "", text)
      text = re.sub(r"\s+", " ", text).strip()
      return text

      df["clean_text"] = df["content"].apply(clean_text)
```

Figure 13: Screenshot of Data Preprocessing

This is one of the important step. Data preprocessing means removing the stopwords, urls, converting into lowercase, removing extra whitespaces. This is done using the library called **re**. Doing this improves feature quality for TF-IDF.

3.4.5. Label Encoding

]:

	Emotion	Encoded Value
0	anger	0
1	fear	1
2	joy	2
3	love	3
4	sadness	4
5	surprise	5

Figure 14: Screenshot of Label Encoding

After preprocessing the data next step is label encoding and it is done using **LabelEncoder()**. LabelEncoder falls under scikit-learn library under **sklearn.preprocessing module**. This step is necessary to convert the text under sentiment column into numeric values for model training. Here the label is encoded for each emotion. Machine learning models cannot understand words, they understand only numbers so each sentiment is encoded as number. When emotion is joy model sees it as 2, when sentiment is love model sees it as 3.

3.4.6. Train Test Split

Total samples: 20000
Train-Test Split Details:
Training samples: 16000
Testing samples : 4000
Split Ratio:
Training set: 80.00%
Testing set : 20.00%

Figure 15: Screenshot of Train Test Split

The `train_test_split` is imported from `sklearn.model_selection` to divide the dataset into training data and test data. The 80% of data is training data and 20% of data is testing data.

3.4.7. Feature Extraction using TF-IDF Vectorizer

```
TF-IDF Feature Extraction Successful  
Train shape: (16000, 5000)  
Test shape : (4000, 5000)  
Features   : 5000
```

Figure 16: Screenshot of Feature Extraction

For feature extraction, TF-IDF vectorizer is used. Here, this is used to convert text into numerical features. Only top **5000** important words are used and common english words such as the, is are removed because they do not help in emotion detection. Then the training text and testing text are converted to numerical vectors.

3.4.8. Model Training

Model Training

```
[50]: nb = MultinomialNB()  
      nb.fit(X_train_vec, y_train)  
  
[50]: ▾ MultinomialNB ⓘ ?  
      MultinomialNB()  
  
[52]: svm = LinearSVC()  
      svm.fit(X_train_vec, y_train)  
  
[52]: ▾ LinearSVC ⓘ ?  
      LinearSVC()  
  
[54]: lr = LogisticRegression(max_iter=1000)  
      lr.fit(X_train_vec, y_train)  
  
[54]: ▾ LogisticRegression ⓘ ?  
      LogisticRegression(max_iter=1000)
```

Figure 17: Screenshot of Model Training

After that, the models - Multinomial Naïve Bayes, LinearSVC and Logistic Regression are trained for emotion detection.

3.4.9. Model Evaluation

The evaluation metrics was used for model evaluation. The fundamental metrics such as Accuracy, Precision, Recall, F1-Score and support was used to evaluate the trained models.

- **Accuracy:** Accuracy is widely used evaluation metric for classification models. It shows the overall correctness made by the model while predicting the result. A high accuracy score indicates that the model is making a large proportion of correct predictions, while a low accuracy score indicates that the model is making too many incorrect predictions (Programmer, 2023).

Accuracy is calculated using following formula:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

where, TP = Number of true positive instances

TN = Number of true negative instances

FP = Number of false positive instances

FN = Number of false negative instances

- **Recall:** Recall is the performance metric for classification model that is the proportion of true positive prediction out of all actual positive instances. A high recall score shows that the model has the ability to identify a large proportion of positive instances, while a low recall score often means that the model is missing many positive instances (Programmer, 2023).

Recall is calculated using following formula:

$$\text{Recall} = TP / (TP + FN)$$

where , TP = Number of true positive instances

FN = Number of false negative instances

- **Precision:** Precision is the performance metric for classification models that measures the correctness of positive prediction made by the model. A high precision score indicates that the model is able to accurately identify positive

instances, while a low precision score indicates that the model is making too many false positive (FP) predictions (Programmer, 2023).

Precision is calculated using following formula:

$$\textbf{Precision} = \textbf{TP} / (\textbf{TP} + \textbf{FP})$$

where, TP = Number of true positive instances

FP = Number of false positive instances

- **F1- Score:** It is the harmonic mean of precision and recall, which balances both in a single metric. A high f1-score means the model is doing well in both precision and recall whereas low f1-score means the model is lacking in both precision and recall (Programmer, 2023).

F1- Score is calculated using:

$$\textbf{F1-score} = 2 * (\textbf{precision} * \textbf{recall}) / (\textbf{precision} + \textbf{recall})$$

3.4.9.1. Evaluation Result of Naïve Bayes Model

Naive Bayes Results

Accuracy : 0.7535

	precision	recall	f1-score	support
anger	0.92	0.58	0.71	542
fear	0.92	0.47	0.62	475
joy	0.70	0.97	0.82	1352
love	0.99	0.22	0.35	328
sadness	0.74	0.93	0.82	1159
surprise	1.00	0.07	0.13	144
accuracy			0.75	4000
macro avg	0.88	0.54	0.58	4000
weighted avg	0.80	0.75	0.72	4000

Figure 18: Naive Bayes Model Evaluation Result

The above figure represents the model evaluation result for Naïve Bayes Model that was trained to predict the emotion from textual data. Naïve Bayes showed the accuracy of 0.7535 which means it was able to predict **accuracy of around 75.35%**. Out of all emotion the precision, recall and f1-score of surprise emotion is very low because there is low test sample for this emotion. Emotions such as joy and sadness has given high

recall values which means model is able to identify most of the actual instances of these emotions. There has been such variation in performance across different emotion classes because Naïve Bayes model relies on strong independence assumptions making it difficult to grasp the less frequent emotional pattern. Overall, this evaluation result clearly shows that even though Naïve Bayes is efficient for common emotions, it lacks the efficiency for complex and less frequent ones.

3.4.9.2. Evaluation Result of Logistic Regression Model

Logistic Regression Results

Accuracy : 0.8755

	precision	recall	f1-score	support
anger	0.89	0.83	0.86	542
fear	0.88	0.78	0.83	475
joy	0.85	0.95	0.90	1352
love	0.85	0.71	0.78	328
sadness	0.90	0.92	0.91	1159
surprise	0.90	0.61	0.73	144
accuracy			0.88	4000
macro avg	0.88	0.80	0.83	4000
weighted avg	0.88	0.88	0.87	4000

Figure 19: Logistic Regression Model Evaluation Result

The accuracy of Logistic Regression is more than Naïve Bayes Model meaning Logistic Regression provides more accurate prediction than Naïve Bayes. Logistic Regression provided the **accuracy of 87.55%** .

The precision, recall and f1-score of different emotion classes are comparatively balanced. The precision of each emotion class is more than **0.8** which means that the model generally predicts emotions correctly and doesn't mislabel the emotion at a basic level.

The **recall of surprise emotion class is 0.61** in Logistic Regression making it slightly better than Naïve Bayes whose recall was only 0.07. This shows that even for a less frequent emotions logistic regression performs well and efficiently.

3.4.9.3. Evaluation Result of SVM Model:

SVM Results

Accuracy : 0.8905

	precision	recall	f1-score	support
anger	0.89	0.89	0.89	542
fear	0.87	0.82	0.84	475
joy	0.90	0.93	0.91	1352
love	0.82	0.80	0.81	328
sadness	0.92	0.92	0.92	1159
surprise	0.83	0.76	0.80	144
accuracy			0.89	4000
macro avg	0.87	0.85	0.86	4000
weighted avg	0.89	0.89	0.89	4000

Figure 20: Evaluation Result of Support Vector Machine

The above figure displays the evaluation result of Support Vector Machine model. The overall **accuracy of SVM is 89.05%** which is highest among Naïve Bayes and Logistic Regression. This accuracy shows that SVM provides most accurate emotion prediction for textual data as compared to other two models.

The precision, recall and f1-score of all the emotion classes are higher than the other two models that were trained. They are more consistent and balanced as well. Emotions such as joy and sadness have higher precision, recall and f1-score **which more than 0.90** meaning model predicted correct joy and sadness for textual data. Even for the less frequent test samples such as surprise, the precision, recall and f1-score is higher which is **more than 0.75**.

Overall, the evaluation result shows that SVM is more effective at handling complex and less frequent emotional patterns.

3.4.9.4. Overall Performance Comparison of Emotion Detection Models

	Model	Accuracy	Precision	Recall	F1-score
0	Naive Bayes	0.7535	0.802036	0.7535	0.718931
1	SVM	0.8905	0.890004	0.8905	0.889989
2	Logistic Regression	0.8755	0.876365	0.8755	0.872908

Figure 21: Overall Performance Comparison of Emotion Detection Models



Figure 22: Graphical Visualization of Performance Comparison of Emotion Detection Models

The overall performance comparison of three trained models is generated and visualized using bar chart. From the bar graph, it is clear that SVM has the highest scores across all evaluation metrics. This shows that among three models trained SVM is reliable and gives balanced performance.

From above generated bar graph, SVM has higher accuracy with similar percentage of precision, recall and F1-score. The precision, recall and F1-score of Logistic Regression is also on same level as accuracy with each other. There is a bit inconsistency in Naïve Bayes model as precision is slightly more than accuracy. Recall is less and F1-score is

lowest among all performance metric. Due to this reason, SVM shows more reliability for emotion detection from text data.

3.4.10. Confusion Matrix

A confusion matrix is a evaluation metric for a classification model that compares predicted values against actual values for a dataset. It summarizes the number of true positives, true negatives, false positives and false negatives which helps understand where the model is making correct or incorrect predictions (Jacob Murel, 2025). To evaluate the performance of three models for emotion prediction from textual data, confusion matrix was generated and visualized using heatmap.

3.4.10.1. Confusion Matrix Visualization of Naïve Bayes through Heatmap

```
plot_confusion_matrix(nb, X_test_vec, y_test, "Naïve Bayes Confusion Matrix")
```

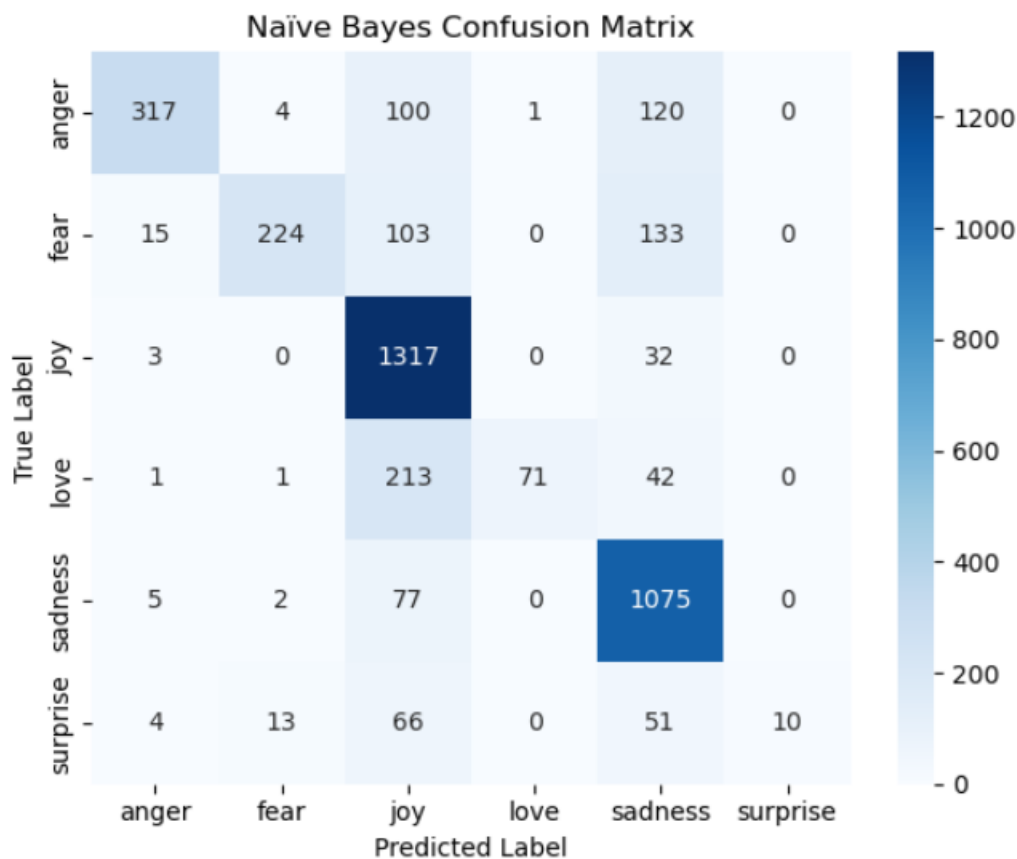


Figure 23: Confusion Matrix heatmap of Naïve Bayes

The vertical axis in above heatmap shows the true emotion in the dataset and horizontal axis shows the predicted emotion by the models. The number represented diagonally are the correct prediction made by the model and color that are darker means more correct prediction. The prediction for joy and sadness has darker color relative to other emotion classes meaning the model is very good at detecting joy and sadness. The model gets confuses emotions such as anger, fear, love, surprise with joy and sadness. For example, the model predicted **anger** for 4 text data but actually it was **surprise**. Similarly, the model predicted **joy** for 213 textual data but it was actually **love** emotion.

3.4.10.2. Confusion Matrix visualization of LR through heatmap

```
[68]: plot_confusion_matrix(lr, X_test_vec, y_test, "Logistic Regression Confusion Matrix")
```

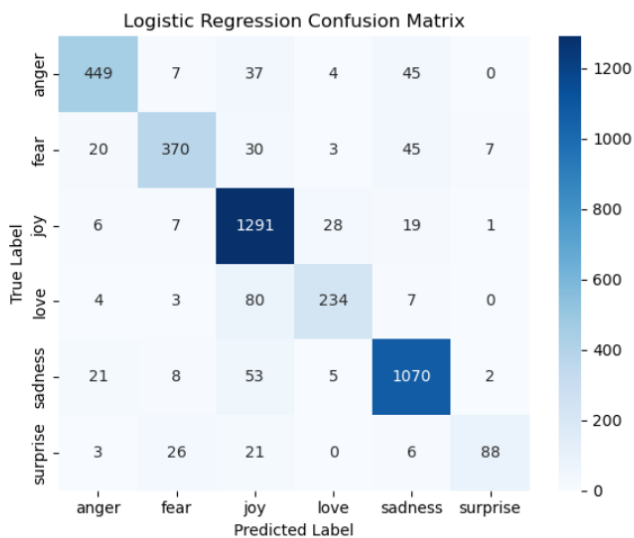


Figure 24: Confusion Matrix heatmap of Logistic Regression

The vertical axis in above heatmap shows the true emotion in the dataset and horizontal axis shows the predicted emotion by the Logistic Regression Model. The number represented diagonally are the correct prediction made by the model and color that are darker means more correct prediction. Observing the above heatmap, the joy and sadness emotion classes have darker than other emotion classes which means that the model performs well for these classes. It is because these emotions classes have more text data samples.

From the confusion matrix heatmap, it is clear that there is still confusion of some emotions. 80 texts that were predicted as joy were actually love, 45 text that were fear emotion classes were classified as sadness. Apart from these there are more confusion but overall diagonal values are stronger and more balanced than that of Naïve Bayes model

3.4.10.3. Confusion Matrix Visualization of SVM through heatmap

```
66]: plot_confusion_matrix(svm, X_test_vec, y_test, "SVM Confusion Matrix")
```

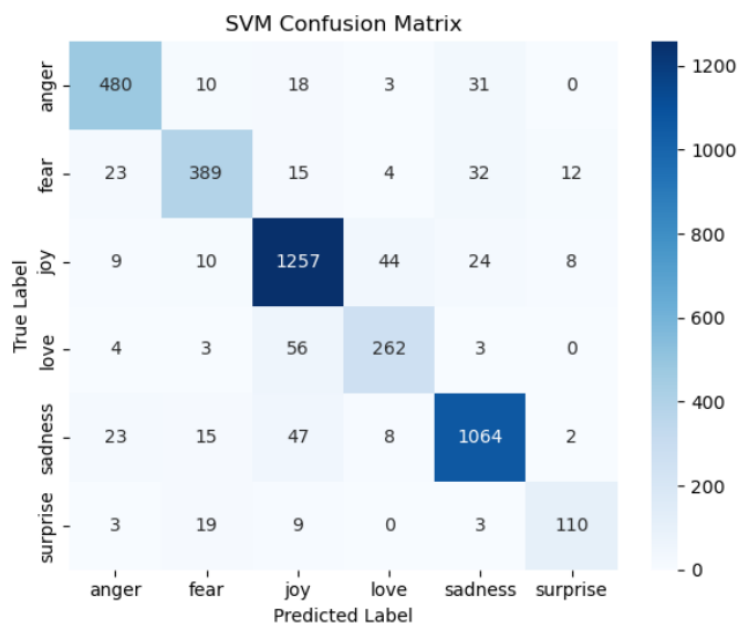


Figure 25: Confusion Matrix heatmap of SVM

In above generated heatmap, the actual emotion is presented in vertical axis and the predicted emotion by the model is presented in horizontal axis. The numbers in the diagonal shows the correct predictions and color intensity displays the correct classifications. As the other two models, the joy and sadness emotion classes have darker color intensity meaning it performs well for these emotion classes. But there have been improvement in other emotion classes also as compared to the other two previous trained models. The color intensity of anger is more darker than heatmap of previous two models meaning the model has predicted more number of text data as correct anger emotion.

Comparatively, there is less misclassification of emotion prediction done by the model. The text data that were actually love emotion, model predicted 3 of them as sadness, 56 as joy, 3 as fear, 4 as anger and 0 as surprise. This shows that SVM is comparatively performing better than other two models.

3.4.11. Cross Validation Score

The Cross Validation Score is a performance metric in Machine Learning model to check the consistency and reliability of the model. A model may perform well one time but that may be just one lucky time that it shows better performance. In Cross Validation Score, the data is trained and tested multiple time not just one lucky time. If the model is trained and tested 5 times then the average of score attained from each time is cross validation score. Doing this ensures the fair comparison and reliability of the model (Scikit-learn, 2025). There are multiple Cross Validation Score methods which are listed below:

- k-Fold Cross Validation
- Stratified k-Fold Cross Validation
- Leave-One-Out Cross Validation
- Time-Series Cross Validation

For this study, **k-Fold Cross Validation** is used. The value we have put of k is 5 meaning the model is trained and tested 5 times. In each iteration, the model is trained on four folds and tested on one fold. The performance of the model is measured using the weighted F1-score for each of the fold and the average of these five scores is taken as the final Cross Validation Score. Doing this ensures that the evaluation result is reliable and is not dependent on single train-test split (Scikit-learn, 2025).

```
Naive Bayes CV F1-score: 0.7005 ± 0.0062
SVM CV F1-score: 0.8873 ± 0.0058
Logistic Regression CV F1-score: 0.8603 ± 0.0069
```

Figure 26: Mean Cross Validation F1-Score with Standard Deviation

The above output represents the mean cross validation score with standard deviation. From the above picture the number left of the \pm represents the **average cross validation score** which means that it is **the average F1-score** obtained after testing the model 5

times on different data splits. The ± 0.0058 number is the standard deviation. It shows how much the model's performance changes across the five folds. The lesser the number, the model is more consistent and stable.

].:

	Naïve Bayes	SVM	Logistic Regression
0	0.692792	0.882606	0.851668
1	0.695707	0.887756	0.858058
2	0.710370	0.890153	0.865489
3	0.699816	0.879816	0.855354
4	0.704039	0.896299	0.870710

Figure 27: Cross-Validation F1-scores for Each Fold

The above table represents the F1-score of the models in each fold during 5-fold cross validation. In each fold, the value of F1-score changes meaning the model performance changes during each time. In first fold, the **F1-score of Naïve Bayes is 0.692792** but in next fold it changes to **0.695707**, then **0.710370** again changing to **0.699816** and on fifth fold the F1-Score is **0.704039**. In each fold of each model there is change in F1-score but the difference is not much which shows consistencies of model. Among all the models, the consistent model is seen to be SVM with higher F1-score in each fold.

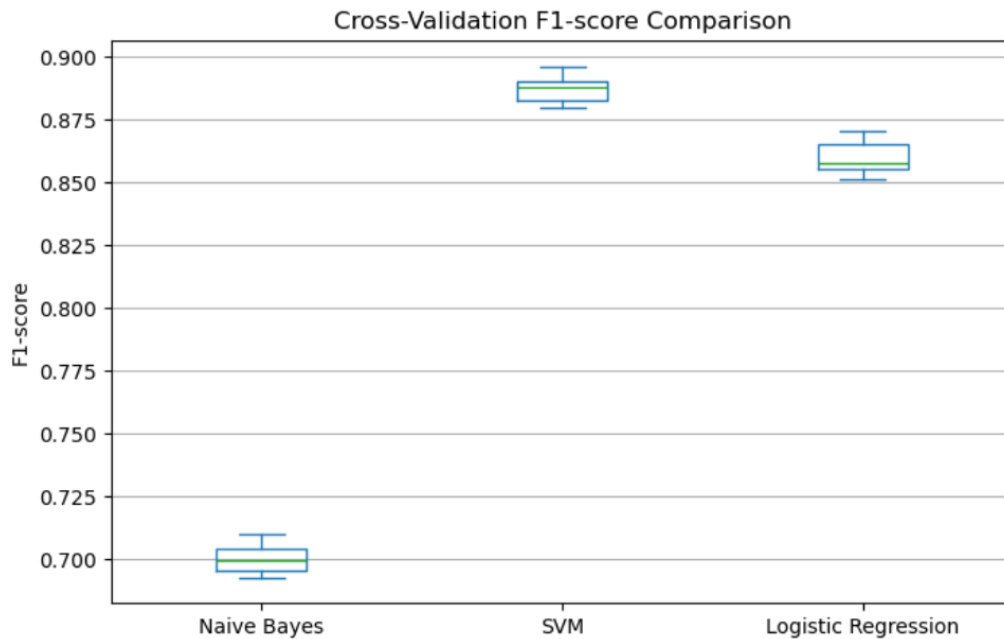


Figure 28: Cross-Validation F1-score Comparison Using Box Plot

The above generated boxplot depicts the comparison of Cross-Validation F1-Score of each models. The vertical axis represents F1-score and horizontal axis represents the models. Simply, the above box plot is all about how cross-validation F1-scores are distributed for each model.

The bottom line of box-plot is called lower whisker which represents the lowest F1-score obtained showing the worst performance across the fold. The bottom of the box is Q1 which represents the lower-range performance. The middle line inside the box is median. The top of the box is Q3 and shows upper range performance. The top line is called upper whisker which is the highest F1-score and shows the best performance across the folds.

The box of Naïve Bayes lies lowest on the graph. The **median is around 0.70** which shows lower performance and there is slight spread which means there is some variation. This means that Naïve Bayes may not perform consistently well.

The box of Logistic Regression is above than that of Naïve Bayes **with median around 0.86** with moderate spread showing the model is stable than Naïve Bayes.

The box of SVM is at highest with **median close to 0.89**. Also the box is very small and whiskers are short which clearly shows SVM gives the best performance, is stable across the folds and most reliable model.

3.4.12. Model Optimization with Hyperparameter Tuning Using GridSearch CV

Sometimes, a machine learning model may not achieve optimal performance due to the choice of hyperparameters. They are predefined settings that control how the model learns from data. In such cases models are further optimized using the technique called Hyperparameter Tuning. It improves the performance of machine learning models by selecting most appropriate parameter values (James et al., 2021). The hyperparameter values directly influences the model complexity, generalization ability and overall predictive performance so it is important to select suitable hyperparameter values.

Best Naive Bayes Params: {'alpha': 0.1}

Best Logistic Regression Params: {'C': 10, 'solver': 'liblinear'}

Best SVM Params: {'C': 1}

Figure 29: Best Hyperparameter values for models

Above displayed picture is the optimal hyperparameter values obtained by using **GridSearchCV** to perform hyperparameter tuning. GridSearchCV checks every blend of chosen hyperparameters and picks one that works best under a defined benchmark (Scikit-learn, 2024). In this study, it generated {'alpha': 0.1} as best hyperparameter value for Naïve Bayes, {'C' : 10, 'solver': 'liblinear'} for Logistic Regression and {'C' : 1} for SVM.

In **Naïve Bayes** model, the hyperparameter **alpha** controls the smoothing. The less alpha value means that the model relies more on the actual data and applies less smoothing (Scikit-learn, 2024). When hyperparameter tuning was performed for this model, GridSearchCV tested different alpha values and found that 0.1 gives the best performance for the dataset and optimize it as well. The visual representation of Naïve Bayes performing better after tuning is presented below:

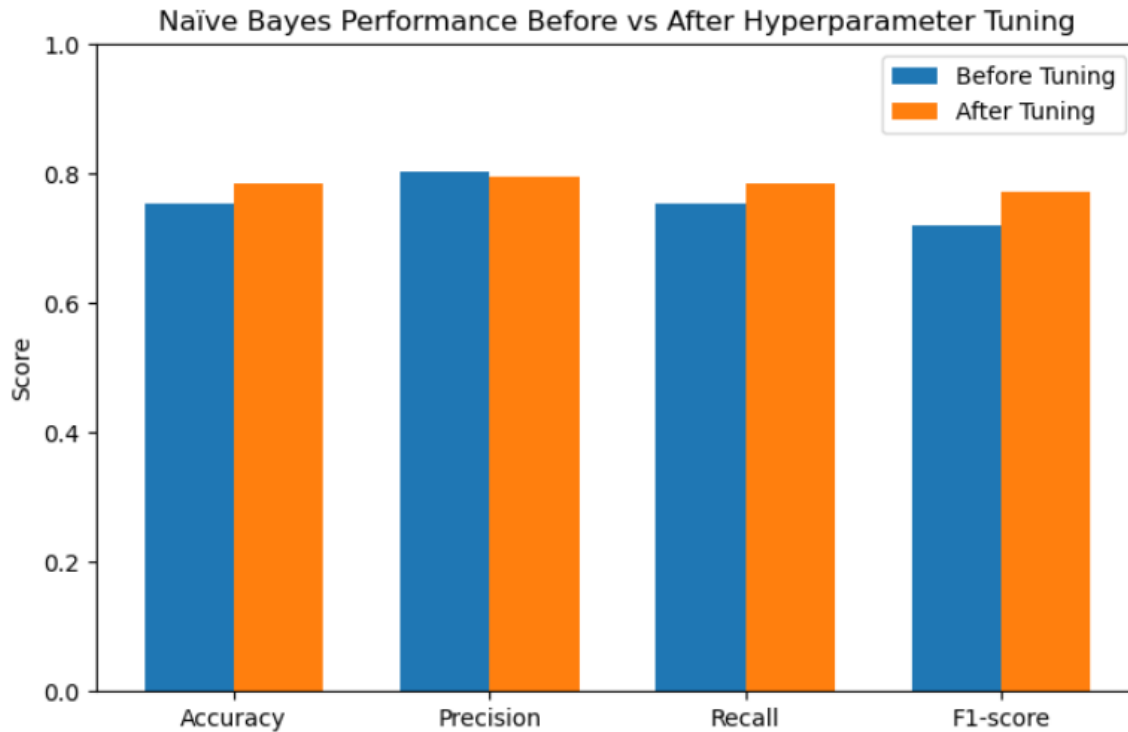


Figure 30: Naive Bayes Performance Before vs After Hyperparameter Tuning

Here, the graph shows that the accuracy, recall and f1-score of Naïve Bayes has improved after hyper parameter tuning and model is performing better than previous.

In **Logistic Regression**, {'C' : 10, 'solver': 'liblinear'} is shown as the best hyperparameter value. C controls the regularization strength. Higher C means lesser regularization and liblinear is chosen because it works well for text classification (Scikit-learn, 2024). The before and after tuning result of Logistic Regression is presented below:

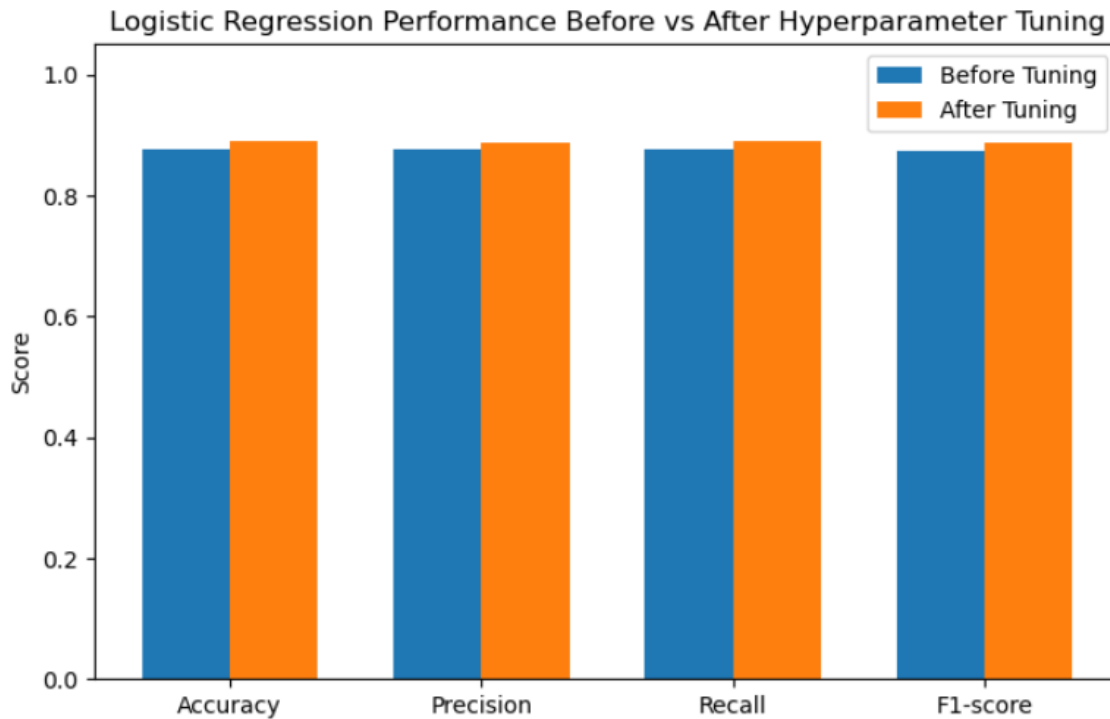


Figure 31: Logistic Regression Performance before vs after hyperparameter tuning

From the above presented graph it is clear that accuracy, precision, recall and f1-score of logistic regression is slightly increasing than before tuning. The model is optimized at its best.

In **SVM**, $C = 1$ is the best hyperparameter value. It is also the default value for LinearSVC. C controls the trade-off between margin size and classification errors (Scikit-learn, 2024). GridSearchCV chose 1 as best value which is default value because the default SVM setting was already optimal and model was performing at its best. The visual representation of SVM before and after tuning is presented below:

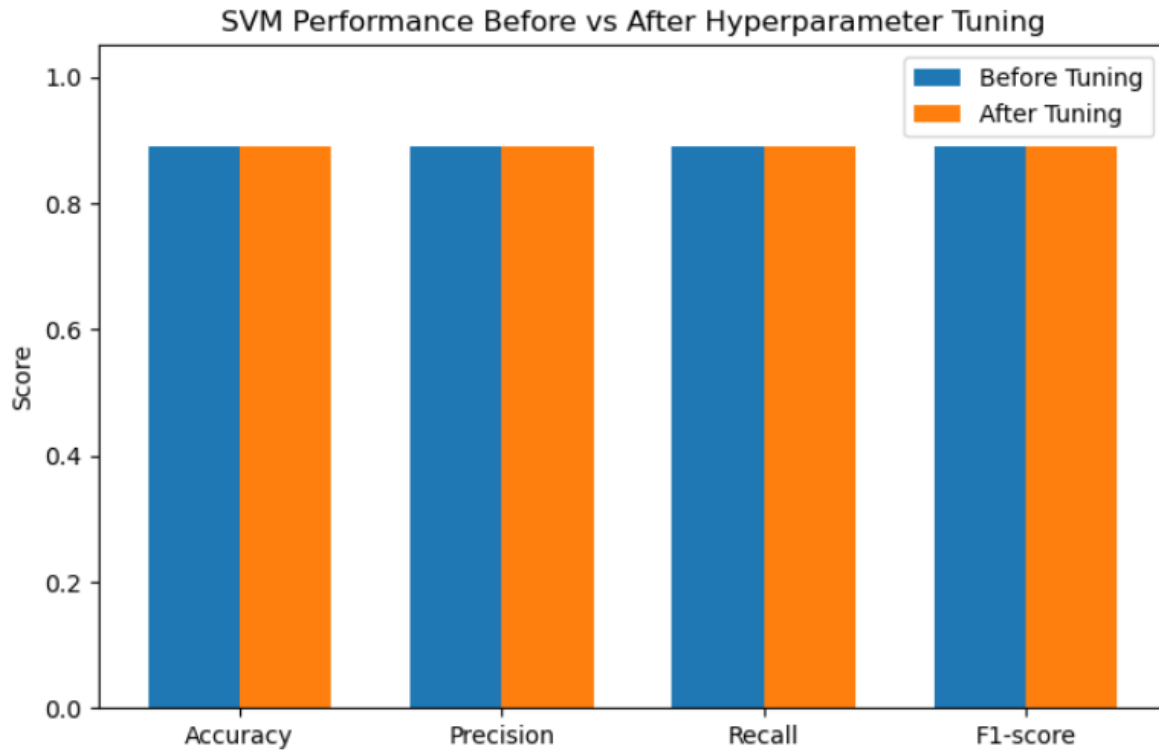


Figure 32: SVM Performance Before vs After Hyperparameter Tuning

The before and after results are same because the model as already at its optimal and was performing at its best.

In this way, hyperparameter tuning technique was performed using GridSearchCV to find the best params and optimize the models at their best.

4. Conclusion

The coursework is completed with the development and evaluation of an emotion detection system using machine learning models. Three models: Naïve Bayes, Logistic Regression, SVM were trained and evaluated to analyze the performance of each model. For the completion of the application part - text preprocessing, TF-IDF feature extraction, model training, evaluation and optimization was implemented. Standard performance metrics such as accuracy, precision, recall and F1-score were used to evaluate the trained model to see which one was better.

4.1. Analysis of Work Done

The system that was developed followed a systematic machine learning workflow. First and foremost thing that was done for the development was to import the necessary libraries. Then the dataset was visualized using graph to see the emotional distribution. Then the dataset was pre-processed to reduce the noise and retain only the meaningful textual information. Then to represent textual data numerically, TF-IDF vectorization was used which enabled models to identify important emotional pattern. Then models were trained and evaluated using different metrics. The evaluation was visualized using different graphs.

The comparative evaluation displayed the strength and limitations of the models that were trained. Among three models, Naïve Bayes was fast in computation but it showed lower performance among all three because of its independence assumptions. Among three models SVM consistently outperformed Naïve Bayes and Logistic Regression because of its ability to handle high dimensional feature spaces. For final phase, cross validation was used to strengthen the reliability of the evaluation so that the model performance was not dependent on single train-test split and hyperparameter tuning was implemented to enhance the performance of different models.

To conclude, the models were not just trained but was evaluated using the standard evaluation metrics to compare the models. The graphical visualization played vital role to understand how each model performed and avoid biased result. Such evaluation methods clearly showed why SVM was the best choice for emotion detection.

4.2. How the application addresses real world problem

The system such as emotion detection can be used to address real world problems such as:

- **Support for mental health:** These type of system would be a very supportive system for people dealing with anxiety and fear. They can track their mood on the basis of mood detected and apply necessary action for better well being.
- **Business clients:** Businesses can enhance their customers by utilizing emotion detection to understand client sentiments and reviews.

- **Social media analysis:** This type of system can be useful for social media monitoring and analyze how users use social media to communicate and provide opinions on various topics.
- **Large-scale text analysis:** The system can be used to automatically analyze a lot of text data that is too big to handle manually.
- **Useful for Chatbots:** This system can be useful for chatbot that can easily recognize human emotions without being involvement of humans.

4.3. Further Work

Some further works that awaits after developing this system are described below:

- **Learn from advanced models:** LSTM or BERT can be used to improve accuracy and understanding context. The system can be extended to detect more than one emotion in a single text.
- **Identify strong emotion:** Future research can be done on measuring strong emotion not only the emotion type.
- **Support multiple languages:** The solution could be extended to explore emotions in texts written in another language.
- **A real-time application:** This model can be embedded into a web or mobile application to detect real-time emotion.

References

Cole Stryker, E.K. (2025) *What is artificial intelligence (AI)?* [Online]. Available from: <https://www.ibm.com/think/topics/artificial-intelligence> [Accessed 15 December 2025].

Cole Styker, I.B. (2025) *What is supervised learning?* [Online]. Available from: <https://www.ibm.com/think/topics/supervised-learning#1509394340> [Accessed 15 December 2025].

Eda Kavlakoglu. (2025) *What are support vector machines (SVMs)?* [Online]. Available from: <https://www.ibm.com/think/topics/support-vector-machine#684929714> [Accessed 6 January 2026].

Fangfang lee. (2026) *What is logistic regression?* [Online]. Available from: <https://www.ibm.com/think/topics/logistic-regression#684929715> [Accessed 6 January 2026].

Ikrama Dayyabu Hayatu, S.S.M.M.M.R.M.M.M. (2025) Emotion detection in text data: a comparative study of machine learning algorithms. *Brazilian Journal of Biometrics*, 43, pp.1-13.

Jacob Murel, E.K. (2025) *What is a confusion matrix?* [Online]. Available from: <https://www.ibm.com/think/topics/confusion-matrix> [Accessed 16 January 2026].

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021) *An Introduction to Statistical Learning*. 2nd ed. New York: Springer.

Kavlakoglu, E. (2025) *What are Naïve Bayes classifiers?* [Online]. Available from: <https://www.ibm.com/think/topics/naive-bayes> [Accessed 6 January 2026].

Kristína Machová, M.S.J.P.J.M. (2023) Detection of emotion by text analysis using machine learning. *frontiers in psychology*, 14.

Pansy Nandwani, R.V. (2021) A review on sentiment analysis and emotion detection from text. 11.

Programmer, P. (2023) *Evaluation Metrics for Classification* [Online]. Available from: <https://medium.com/@mlmind/evaluation-metrics-for-classification-fc770511052d> [Accessed 16 January 2026].

Rio Ferdinand Putra Pratama, W.M. (2025) Comparative Analysis of Naive Bayes and SVM for Improved Emotion Classification on Social Media. *EDUMATIC Jurnal Pendidikan Informatika*, 9(1), pp.11-20.

Scikit-learn. (2024) *GridSearchCV — Exhaustive search over specified parameter values* [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Accessed 17 January 2026].

Scikit-learn. (2024) *Linear Support Vector Classification* [Online]. Available from: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html> [Accessed 17 January 2026].

Scikit-learn. (2024) *Logistic Regression* [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html [Accessed 17 January 2026].

Scikit-learn. (2024) *Multinomial Naive Bayes* [Online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html [Accessed 16 January 2026].

Scikit-learn. (2025) *Cross Validation* [Online]. Available from: https://scikit-learn.org/stable/modules/cross_validation.html [Accessed 16 January 2026].

Sheetal Kusal, S.P.J.C.K.K.D.V.I.P. (2022) *A Review on Text-Based Emotion Detection -- Techniques, Applications, Datasets, and Future Directions* [Online]. Available from: [https://www.researchgate.net/publication/360462407_A_Review_on_Text-Based_Emotion_Detection -- Techniques Applications Datasets and Future Directions](https://www.researchgate.net/publication/360462407_A_Review_on_Text-Based_Emotion_Detection_-_Techniques_Applications_Datasets_and_Future_Directions) [Accessed 15 December 2025].

Vimala Balakrishnan, W.K. (2019) String-based Multinomial Naïve Bayes for Emotion Detection among Facebook Diabetes Community. *Procedia Computer Science*, 159, pp.30-37.