Emotion Detection from Text

Project Report

Course ID: CSE 4214

Course Name: Pattern Recognition Lab

Semester: Fall 2020

Submitted by

Nowshin Rumali	**160204107**
Rejone E Rasul Hridoy	**170104116**
Mehedi Hasan Sami	**170104118**

Submitted To

Faisal Muhammad Shah, Associate Professor

Farzad Ahmed, Lecturer

Md. Tanvir Rouf Shawon, Lecturer

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

September 2021

ABSTRACT

Emotion is one of the basic instincts of a human being. Emotion detection plays a vital role in the field of textual analysis. At present, people's expressions and emotional states have turned into the leading topic for research works.In this project, Our primary goal is to detect human's emotion from text input through some machine learning techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Emotion is one of the basic instincts of a human being. Emotions refer to various types of consciousness or states of mind that are shown as feelings. They can be expressed through facial expressions, gestures, texts, and speeches.

Emotion detection plays a vital role in the field of textual analysis. At present, people's expressions and emotional states have turned into the leading topic for research works. Emotion Detection and Recognition from texts are recent fields of research that are closely related to Emotion Analysis.

Emotion Analysis aims at detecting and recognizing feelings through the expressions from sentences, such as anger, surprise, joy, disgust, sadness etc. For detecting emotions we will use some natural language processing (NLP) and classification algorithms of machine learning.

Chapter 2

Literature Reviews

We have chosen emotion detection from tweets from IEEE as our literature. They have taken "SemEval-2018 Affect in Tweets Distant Supervision Corpus (AIT-2018 Dataset)" as their dataset. This dataset contains lots of emotions but they mainly clustered those emotions into 4 basic emotions which are anger, fear, joy, and sadness.

The dataset contains tweets and we know tweets contain so much noisy data such as website URL, person mentions, short form etc. That's why they have preprocessed the whole dataset. The preprocessing contains some steps such as removing tweets which are not english, converting texts to lower cases, removing URL, removing mentions, removing retweet mentions, removing unnecessary numbers, separating hashtags, changing short forms with their full forms, word tokenizing, removing punctuation, removing stopwords, stemming and lemmatization, making parts of speech tag for those tokenized words etc.

They also have balanced their dataset for each emotions.

After the data is preprocessed they have used WordNet-Affect which is a subset of WordNet with only words that have emotions. They mapped those preprocessed words and retrieved the words that have emotions only. They also have used EmoSenticNet for the same purpose.

After that they used feature vector conversion in the preprocessed text. They used Term frequency- Inverse Document Frequency as its the one of the best methods for machine learning models. After performing feature extraction they split the dataset into train and test and run them on traditional machine learning algorithms such as Naive Bayes, Decision Tree, and Support Vector Machine.

They got the highest accuracy using **SVM** which is **88.23%**. They have also measured precision and recall of each emotions for each model.

Chapter 3

Data Collection & Processing

3.1 Dataset

Here tweet emotions from SemEval-2018 Affect in Tweets Distant Supervision Corpus (AIT-2018 Dataset) is used.This dataset has two columns content and sentiment.It has 20000 unique text classified with 6 emotions such as anger, love, surprise, fear, joy, sadness. This dataset contains lots of emotions but they mainly clustered those emotions into 4 basic emotions such as anger, fear, joy, sadness.

Data Distribution:
There are total 20000 of text data with emotion labels.We categorize the texts of each emotions. so that, we can easily visualize the data of each emotion labels. Here we can notice that data are not properly distributed. Text with emotion 'Surprise' is lowest on the other hand emotion of 'joy' is highest. Only emotion of joy and sadness is very large in number compared to other emoition labels.

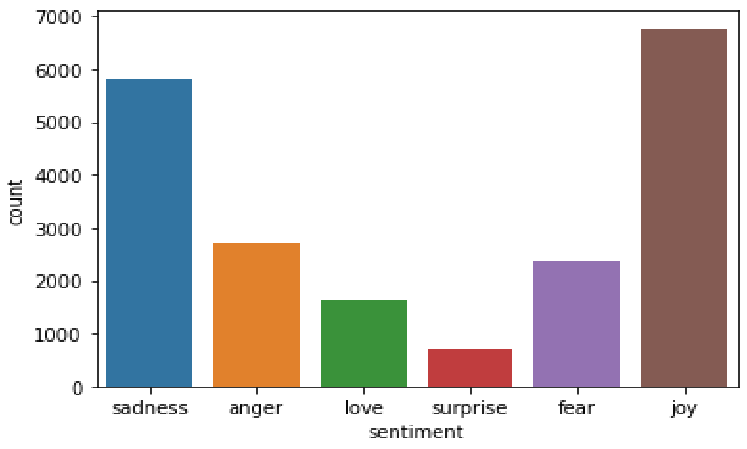


Figure 3.1: Distribution of emotions

3.1.1 Text Collection

Text is collected from the Dataset [1]. For an example:

Doc-1: I didn’t feel humiliated.
Doc-2: I’m grabbing a minute to post I feel greedy wrong.
Doc-3: I have been feeling the need to be creative.

3.1.2 Text Pre-Processing

Before Feature-Extraction we need to pre-process the text data in-order to get a good result. before train our model firstly we need to do these kind of pre processing like:

- Removing Punctuation
- Tokenize Words
- Removing Stop Word
- Lemmatization

Remove Punctuation:
Firstly, we need to remove punctuation from the textual input. For example:
!"#$%&´()*+,-./:;<=>?[ ]_‘{|} . we need to remove such character from our text. Because these symbols doesn’t help us to detect someone’s emotion.
So our Doc-1 will be:
Before removing punctuation: i didn’t feel humiliated
After removing punctuation: i didnt feel humiliated

Tokenize words:
Tokenization is the process of turning a meaningful piece of data.Tokenization of a sentence simply Converts a sentence into list of words.
For example:
Before tokenization: i didnt feel humiliated
After tokenization: [i, didnt, feel, humiliated]

The total number tokens of the entire dataset are shown in the below figure. So we can see the distributions of tokens after preprocessing.

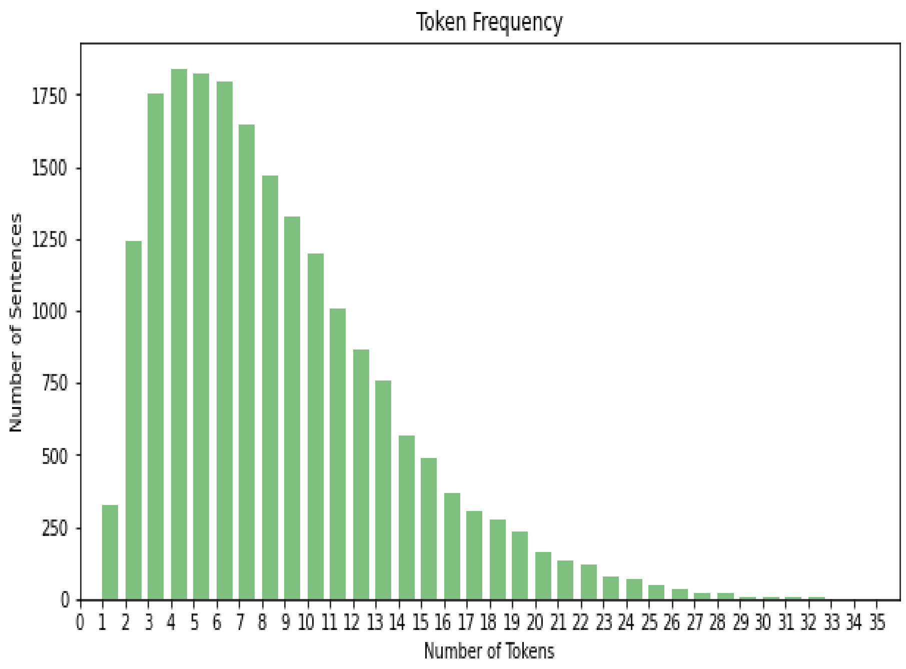
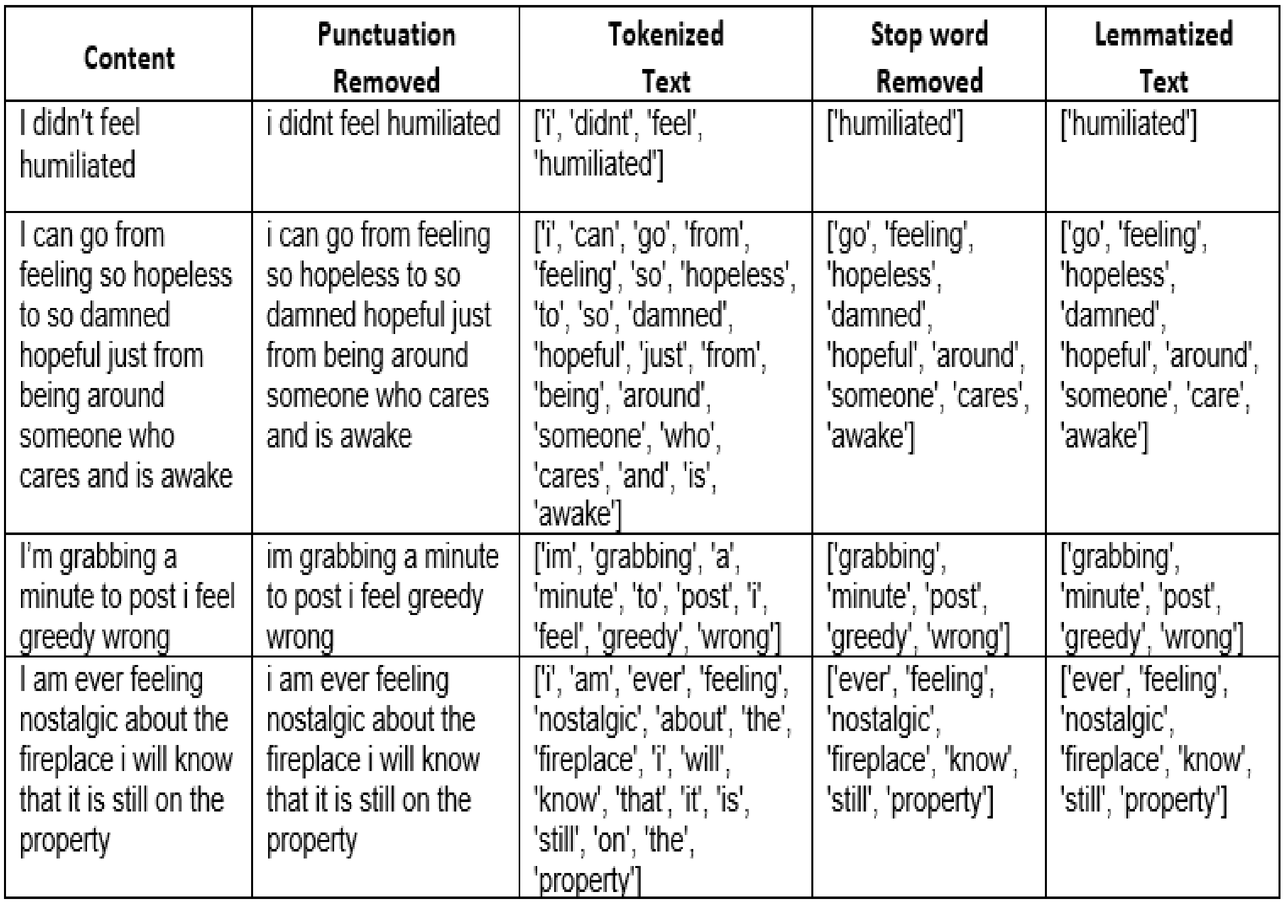


Figure 3.2: Distribution of Tokens

Stop words:
Stop words are a set of commonly used words in a language. Examples of stop words in English are “a”, “the”, “is”, “are” and etc. Stop words are commonly used in Text Mining and Natural Language Processing (NLP) to eliminate words that are so commonly used that they carry very little useful information.we can also extend stop word list. For example:
before removing Stopword: [i, didnt, feel, humiliated]
after removing Stopword: [humiliated]

Lemmatization:
Sometimes it also called Word Normalization) techniques in the field of Natural Language Processing. It takes into consideration the morphological analysis of the words. It Tranforming any form of a word to its root word like ‘Playing –> Play’, ‘Going –> Go’, etc.
In our Doc-1:
Before lemmatization: [humiliated]
After lemmatization: [humiliated]
Though here the output is same but it will not happen always.

Here is the sample, after doing all these pre-processing stuff our dataset looks like this:



Content	Punctuation Removed	Tokenized Text	Stop word Removed	Lemmatized Text
I didn't feel humiliated	i didnt feel humiliated	['i', 'didnt', 'feel', 'humiliated']	['humiliated']	['humiliated']
I can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	['i', 'can', 'go', 'from', 'feeling', 'so', 'hopeless', 'to', 'so', 'damned', 'hopeful', 'just', 'from', 'being', 'around', 'someone', 'who', 'cares', 'and', 'is', 'awake']	['go', 'feeling', 'hopeless', 'damned', 'hopeful', 'around', 'someone', 'cares', 'awake']	['go', 'feeling', 'hopeless', 'damned', 'hopeful', 'around', 'someone', 'care', 'awake']
I'm grabbing a minute to post i feel greedy wrong	im grabbing a minute to post i feel greedy wrong	['im', 'grabbing', 'a', 'minute', 'to', 'post', 'i', 'feel', 'greedy', 'wrong']	['grabbing', 'minute', 'post', 'greedy', 'wrong']	['grabbing', 'minute', 'post', 'greedy', 'wrong']
I am ever feeling nostalgic about the fireplace i will know that it is still on the property	i am ever feeling nostalgic about the fireplace i will know that it is still on the property	['i', 'am', 'ever', 'feeling', 'nostalgic', 'about', 'the', 'fireplace', 'i', 'will', 'know', 'that', 'it', 'is', 'still', 'on', 'the', 'property']	['ever', 'feeling', 'nostalgic', 'fireplace', 'know', 'still', 'property']	['ever', 'feeling', 'nostalgic', 'fireplace', 'know', 'still', 'property']

Figure 3.3: Text Pre-processing

Chapter 4

Methodology

4.1 Feature Extraction

After text pre-processig all text input need to be converted to numbers before using this in our model to predict emotions specifically, convert the textual input into vector of words. Text is messy in nature and machine learning algorithms prefer well defined fixed-length inputs and outputs.
For supervised learning there are two popular forms of **Feature Extraction** techniques.

- Count Occurrence
- TF-IDF

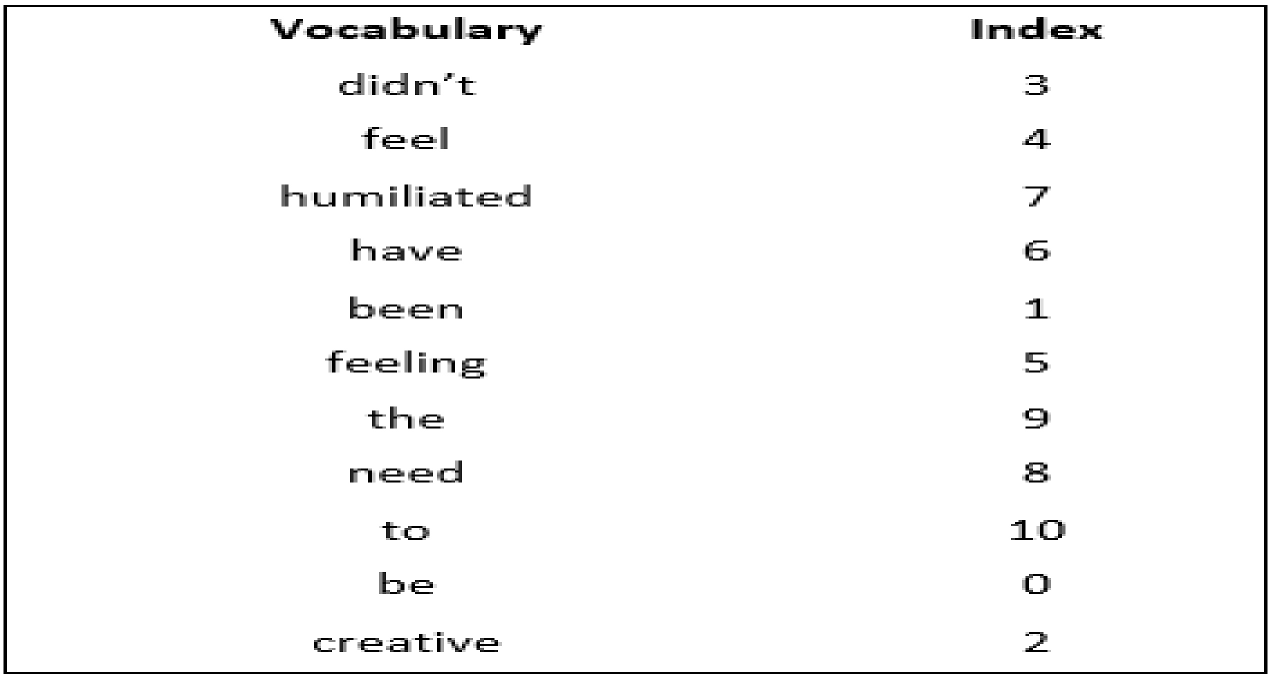
4.1.1 Count Occurrence

The Count Occurrence is a way of representing text data when modeling text with machine learning algorithms. It converts a textual input into word vector. It involves two things:

- Vocabulary of Words
- Frequency of Vocabulary

For example we take two sentences from our dataset:
doc 1: I didn’t feel humiliated
doc 2: I have been feeling the need to be creative
vocabulary for these two sentences are:

After getting the vocabulary we need to create a document vector for each sentence to find the frequency of each word. The length of document vector will be same as vocabulary size.



Vocabulary	Index
didn't	3
feel	4
humiliated	7
have	6
been	1
feeling	5
the	9
need	8
to	10
be	0
creative	2

Figure 4.1: Vocabualary

So the document vector for given sample will be-
Sentence 1: [0 0 0 1 1 0 0 1 0 0 0]
Sentence 2: [1 1 1 0 0 1 1 0 1 1 1]
In this Document vector '1' is placed at the index of that corresponding vocabulary if that specific word is present in that particular sentence otherwise '0' is placed.

4.1.2 TF-IDF

TF (Term Frequency)

Term Frequency (TF) is a measure of how frequently a term (t) appears in a document, d

$$TF_{t,d} = \frac{n_{t,d}}{Total\ number\ of\ terms\ in\ document\ d} \tag{4.1}$$

Here,
$n(_{t},_{d})$=Number of times a term (t) appears in a document (d). Thus, each document and term would have its own TF value.
Lets consider last two documents again:
doc 1 : I didn't feel humiliated
doc 2 : I have been feeling the need to be creative
Now we will calculate the TF values for the Doc-1
Number of words in Doc-1 is 3.

So, TF will be:
TF(feel)=$\frac{1}{3}$
TF(didn't)=$\frac{1}{3}$
TF(humiliated)=$\frac{1}{3}$
Similarly for all the vocabulary TF of doc 1 will be calculated by this way.

IDF (Inverse Document Frequency)
IDF is a measure of how important a term is. We need the IDF value because computing just the TF alone is not sufficient to understand the importance of words.

$$IDF_t = log(\frac{TotalNumberofDocuments}{TheNumberofDocumentWithTermt}) \tag{4.2}$$

Now If we calculate the IDF vlaues for Doc-1, it will be-
IDF(feel)=$log(\frac{2}{1})$
IDF(didn't)=$log(\frac{2}{1})$
IDF(humiliated)=$\frac{2}{1}$
As there is no common term in both documents so denominator of IDF remains same for these three words. Similarly for all the vocabulary IDF of doc 1 will be calculated by this way.

We can now compute the TF-IDF score for each word in the corpus. Words with a higher score are more important, and those with a lower score are less important:

TF-IDF$_{t,d} = TF\ X\ IDF$

We can easily visualise the importance of each word through heat-map:

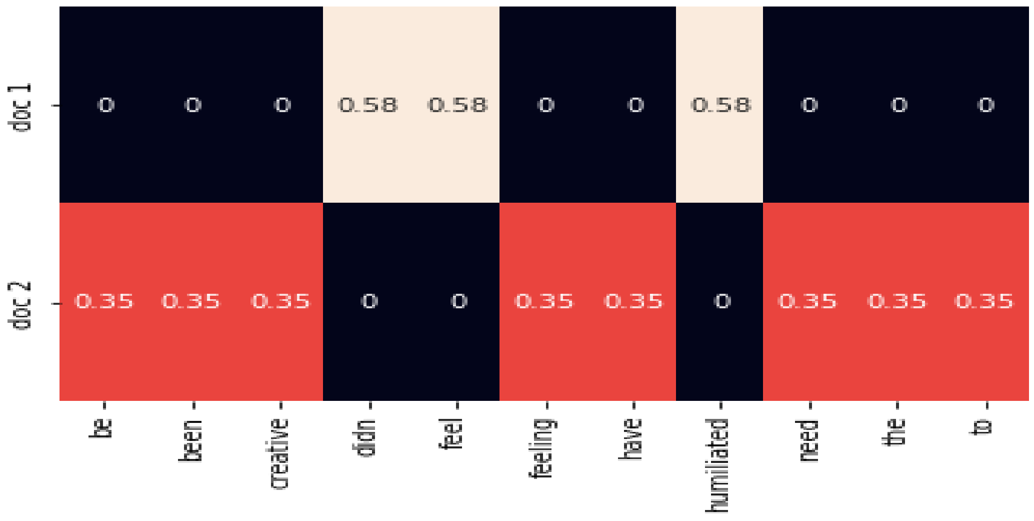


Figure 4.2: TF-IDF

4.2 Implemented Models

In order to detect emotion from a text input we need some models to train with our dataset so that we can classify emotions of different labels. Here we used six models to train our dataset-

1. Support Vector Machine (SVM)
2. Logistic Regression
3. Random Forest Classifier
4. XGBoost Classifier
5. Multinomial Naive Bayes
6. Decision Tree Classifier

4.2.1 Support Vector Machine (SVM)

The objective of SVM is to find a hyperplane that maximizes the separation of the data points to their actual classes in an n-dimensional space. In its most basic type, SVM doesn't support multiclass classification. For multiclass classification, the same principle is utilized after breaking down the multi-classification problem into smaller subproblems, all of which are binary classification problems like-

- One vs One (OVO) approach
- One vs All (OVA) approach
- Directed Acyclic Graph (DAG) approach

Here in our model SGDClassifier is used which used One VS rest approach.

4.2.2 Logistic Regression

Multi class classification is implemented by training multiple logistic regression classifiers, one for each of the K classes in the training dataset. Once training all our classifiers, we can now use it to predict which class the test data belongs to. For the test input, here we compute the "probability" that it belongs to each class using the trained logistic regression classifiers, then give the highest probability of a class at which our data point belongs.

4.2.3 Random Forest Classifier

Random forests can be used both for classification and regression.Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.There is the n_estimators hyperparameter, which is just the number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions.

4.2.4 XGBoost Classifier

XGBoost is a boosted tree based ensemble classifier. Like ‘RandomForest’. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting. XGBoost improves upon the base GBM (Gradient Boosting Machines) framework through systems optimization and algorithmic enhancements.

4.2.5 Multinomial Naive Bayes

Naïve Bayes classifiers are a family of probabilistic classifiers based on Bayes Theorem. Multinomial Naive Bayes classifiers has been used widely in NLP problems. They are probabilistic classifiers uses Bayes theorem to calculated the conditional probability of the each label given a given text, and the label with highest will be output.

4.2.6 Decision Tree Classifier

The decision tree is a tree-like structure of decisions made based on some conditional statements. Here we have to decide which node will be the root node, for splitting a node and deciding threshold for splitting, we use entropy.

Chapter 5

Experiments and Results

5.1 Accuracy

Accuracy is defined as the percentage of correct predictions for the test data.Here Accuracy of 6 models are shown.So that we can easily understand which model is performing better.

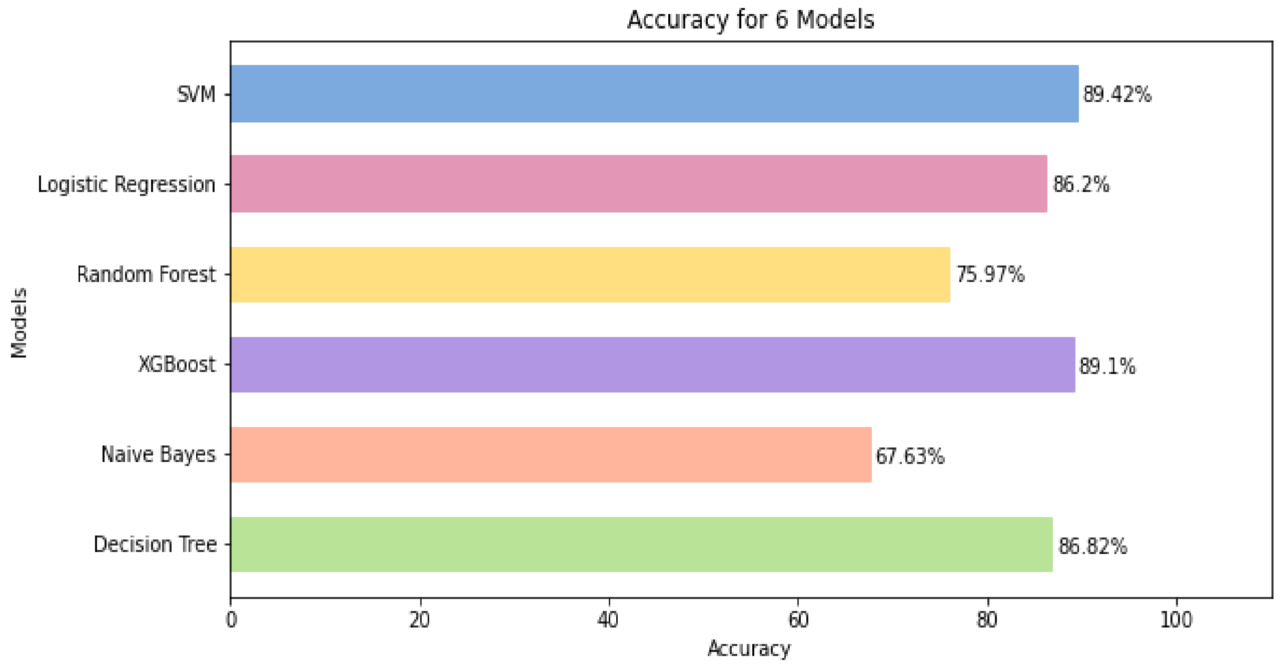


Figure 5.1: Accuracy of all models

5.2 Classification Report

It's a performance evaluation metric in machine learning. It is used to show the precision, recall, F1 Score, and support of trained classification model.

Precision of all Models:
Precision is the ability of a classification model to identify only the relevant data points. Mathematically, precision the number of true positives divided by the number of true positives plus the number of false positives.

	SVM	Logistic Regression	Random Forest	XGBoost	Naive Bayes	Decision Tree
anger	90.549	90.704	77.675	90.372	94.649	88.265
fear	88.494	87.404	75.882	87.819	87.963	83.641
joy	88.832	82.406	78.812	89.451	63.378	90.984
love	84.028	86.197	68.269	79.832	100.000	75.337
sadness	91.619	88.403	74.973	92.857	67.402	88.560
surprise	86.145	93.333	70.952	76.126	100.000	70.213

Table 5.1: Precision of all models

Recall of all Models:
Recall is the ability of a model to find all the relevant cases within a data set. Mathematically, we define recall as the number of true positives divided by the number of true positives plus the number of false negatives.

	SVM	Logistic Regression	Random Forest	XGBoost	Naive Bayes	Decision Tree
anger	85.628	77.778	71.014	85.024	34.179	85.386
fear	84.076	76.788	69.636	83.671	25.641	85.560
joy	94.552	95.344	77.563	92.818	97.028	86.974
love	75.000	63.223	73.347	78.512	7.025	80.785
sadness	94.344	94.227	80.700	93.236	92.595	90.729
surprise	67.136	52.582	69.953	79.343	1.878	77.465

Table 5.2: Recall of all models

F1-Score of all Models:
F1-Score of the six models - Support Vector Machine (SVM), Logistic Regression, Random Forest Classifier, XGBoost Classifier, Multinomial Naive Bayes and Decision Tree Classifier are 85.675%, 80.847%, 75.805%, 85.386%, 43.572% and 83.434% respectively.

5.3 Confusion Matrix

Confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class or vice versa.

Confusion Matrix of SVM and Logistic Regression:

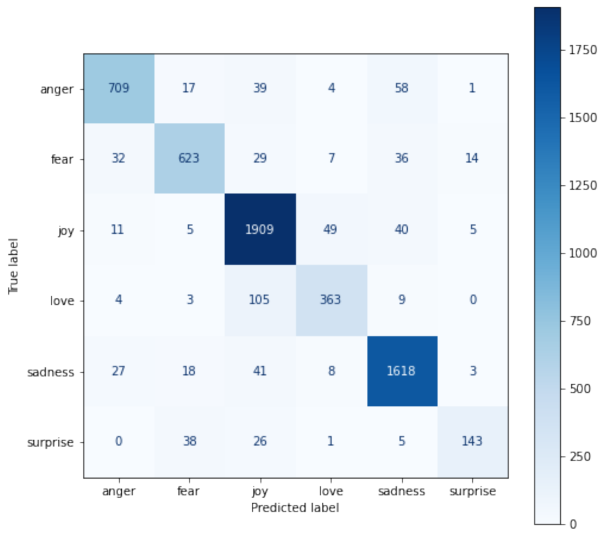


Figure 5.2: SVM

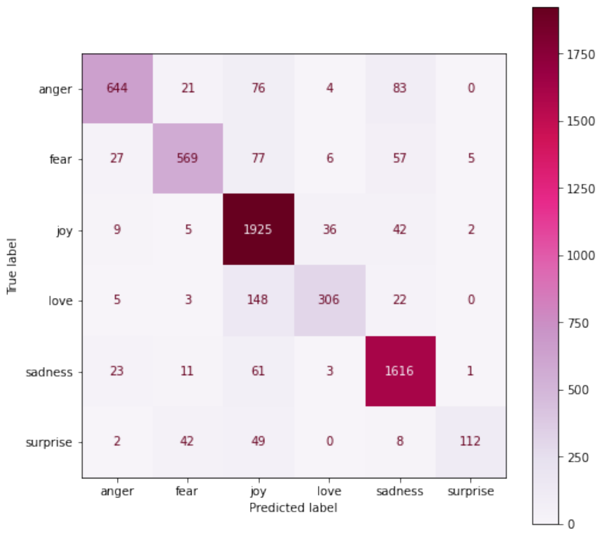


Figure 5.3: Logistic Regression

Confusion Matrix of Random Forest and XGBoost:

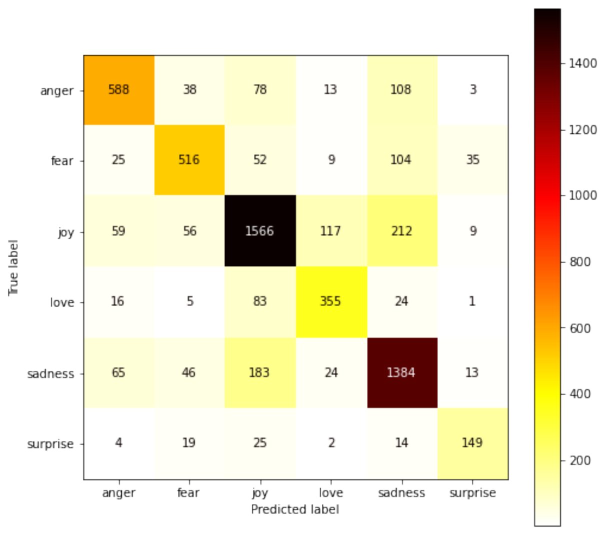


Figure 5.4: Random Forest

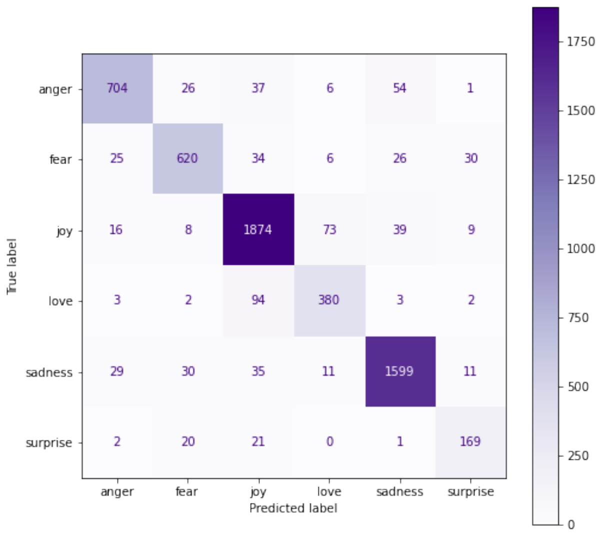


Figure 5.5: XGBoost

Confusion Matrix of Multinomial Naive Bayes and Decision Tree:

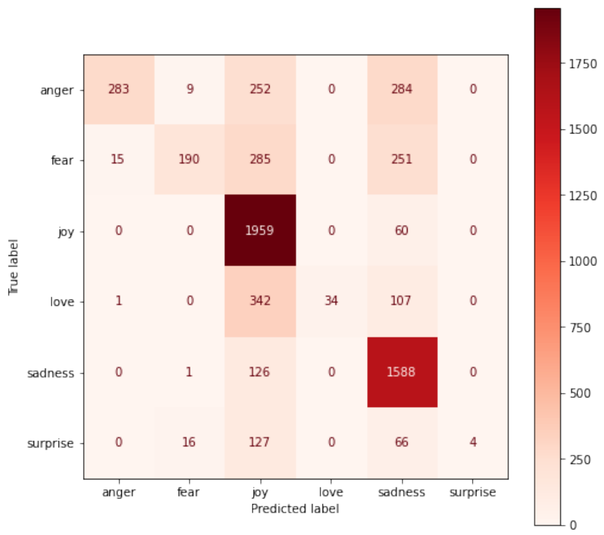


Figure 5.6: Multinomial Naive Bayes

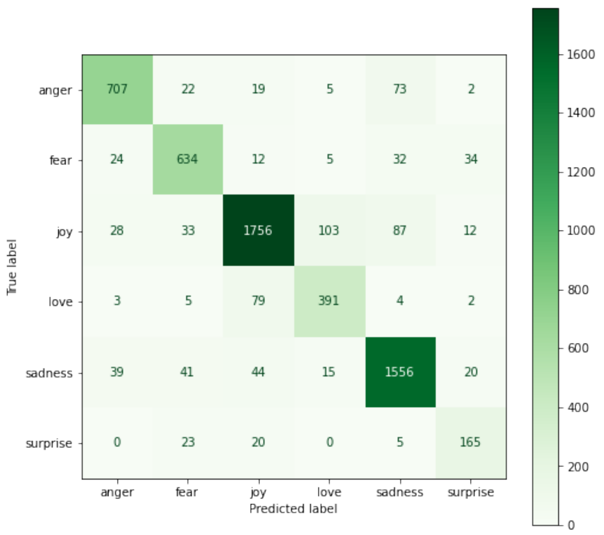


Figure 5.7: Decision Tree

Chapter 6

Future Work and Conclusion

6.1 Conclusion

In this project, we have used "tweet emotions from SemEval-2018 Affect in Tweets Distant Supervision Corpus (AIT-2018 Dataset)". As it's a noisy dataset, first we had to do the preprocessing such as punctuation remove and converted emojis into texts, tokenization, remove stopwrods, and lemmatization.

Then we have implemented six models and compared them. After comparing them, we observed that **SVM** model gave the highest accuracy which is **89.52%**.

After completion of our project, we observed that our project can detect emotions from texts.

6.2 Future Work

In future we want to research and make improvements of this project.

At first we want to make a chatbot that can be used in various platforms such as gaming platforms, retailer apps and websites to get customer feedback, social media platforms where people can chat and know the emotions of the other person.

Secondly, We will use a balanced dataset or we will make our current dataset balanced so that our model can learn better.

Thirdly, if our model can learn better than now, we can also improve our accuracy and this project will give better performance.

References

[1] “Emotions in text.” https://www.kaggle.com/ishantjuyal/emotions-in-text.