TABLE OF CONTENTS

CHAPTER NO TITLE PAGE NO

ABSTRACT

1. INTRODUCTION

* 1. PROBLEM STATEMENT
  2. MOTIVATION
  3. OBJECTIVES

1. LITERATURE

* 1. INTRODUCTION
  2. EXISTING SYSTEM
  3. SUMMERY

1. PREPOSED SYSTEM

* 1. INTRODUCTION
  2. ARCHITECTURE
  3. MODULES
  4. SUMMERY

1. PREPROCESSING

* 1. REMOVING STOPWORDS
  2. LEMMATIZATION
  3. N-GRAMS GENERATION
  4. WORD EMBEDDING
  5. SUMMERY

1. BUILDING MODEL
   1. SPLITING DATASET
   2. BUILDING MODELS
   3. TRAINING THE MODELS
   4. SUMMERY
2. MODEL EVALUTION
   1. INTRODUCTION
   2. MODEL EVALUTION METRICS
   3. CHOOSING THE BEST MODEL
   4. SUMMERY
3. RESULTS
4. CONCLUSIONG AND FUTURE SCOPE

**INTRODUCTION**

Humans have always had the innate ability to recognize and distinguish between faces. Now computers are able to do the same. This opens up tons of applications. Emotion Detection will play a promising role in the field of machine learning, especially in the case of Human-Computer Interaction development.

For Emotion Detection from a machine learning model different parameter should be taken into consideration. Various types of techniques are used to detect emotions from a human being like facial expressions, body movements, blood pressure, heart beat and textual information. This project focuses on the emotion detection from textual information. Emotion Detection in text documents is essentially a content - based classification problem involving concepts from the domains of Natural Language Processing as well as Machine Learning. In this paper emotion recognition based on textual data and the techniques used in emotion detection are discussed.

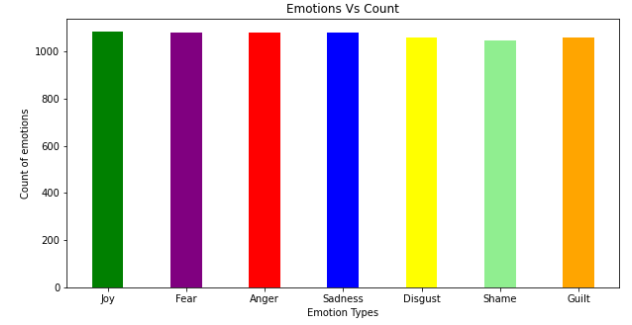
In this project preposes a better way for detecting the human emotions using textual data. such as joy, fear, anger, sadness, disgust, shame, and guilt.

**PROBLEM STATEMENT**

**OBJECTIVE**

Dataset:

Emotion text dataset contains the corpus which is collected from twitter. Total number of instances in the dataset are 7480.



Each sample is labeled with its corresponding emotion.

This dataset contains the sample text for seven emotions.

They are:

1. Joy
2. Fear
3. Anger
4. Sadness
5. Disgust
6. Shame
7. Guilt

Example - Line in data set:



Each line contains two parts, first part describes emotion of the text and the second part is the text.

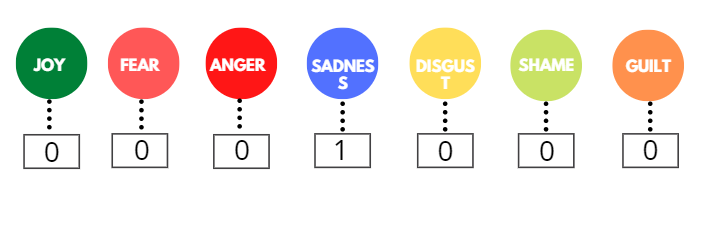
First part:

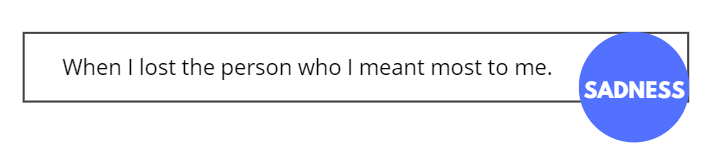
The first part(label) is represented as list of seven elements

Each element corresponds to the emotions joy, fear, anger, sadness, disgust, shame and guilt respectively.

Label list only contains Boolean values [ 0 or 1].

The emotion of the text is identified by the label list.





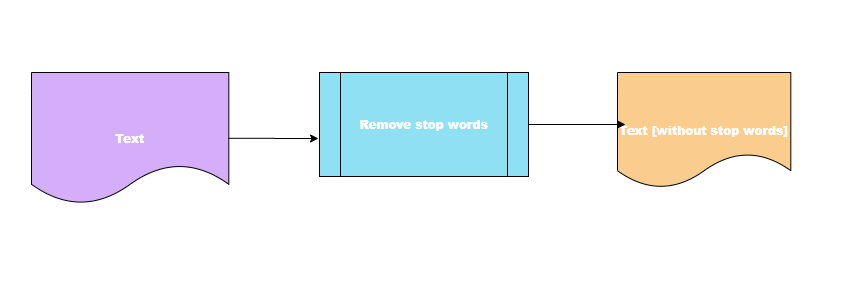
Preprocessing

1. Removing stop words

Stopwords are the most common words in any natural language. For the purpose of analyzing text data and building NLP models, these stopwords might not add much value to the meaning of the document.

For tasks like text classification, where the text is to be classified into different categories, stopwords are removed or excluded from the given text so that more focus can be given to those words which define the meaning of the text.

* On removing stopwords, dataset size decreases and the time to train the model also decreases
* Removing stopwords can potentially help improve the performance as there are fewer and only meaningful tokens left. Thus, it could increase classification accuracy
* Even search engines like Google remove stopwords for fast and relevant retrieval of data from the database



Stopwords removal using NLTK

The steps to import the library and the English stop words list is given below:

Removing stop words from the dataset:

Steps:

Step1: load the dataset text



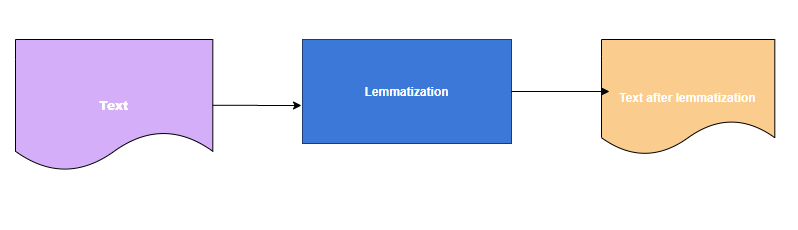
Step2: Send the text data to remove\_stop\_words() function

Step3: Store the resultant text

**Lemmatization**

Lemmatization are **Text Normalization** (or sometimes called **Word Normalization**) techniques in the field of **Natural Language Processing** that are used to prepare text, words, and documents for further processing.

*Lemmatization, unlike Stemming, reduces the inflected words properly ensuring that the root word belongs to the language.*



Removing stop words from the dataset:

Steps:

Step1: load the text in the dataset.

Step2: Send the text data to lemmatizer() function.

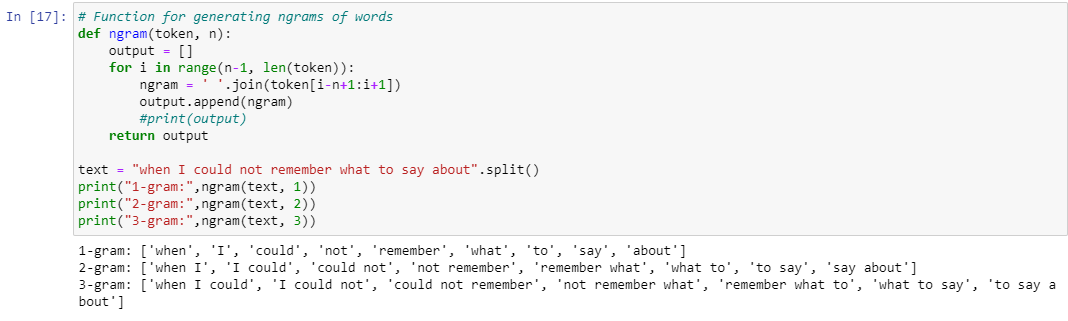
Step3: Store the resultant text

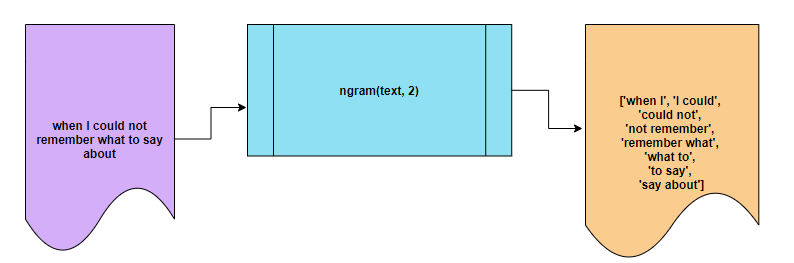
**N-grams Generation**

* n-grams are a set of co-occurring or continuous sequence of n items from a sequence of large text or sentence.
* The item here could be words, letters, and syllables.
* 1-gram is also called as unigrams are the unique words present in the sentence.
* Bigram(2-gram) is the combination of 2 words.
* Trigram(3-gram) is 3 words and so on.

**Generation of n-grams for the text in the corpus**

**Python function to generate n-grams**





Word Embedding

* **Word embeddings** (also called word vectors) represent each word numerically in such a way that the vector corresponds to how that word is used or what it means.
* Vector encodings are learned by considering the context in which the words appear.
* Words that appear in similar contexts will have similar vectors.

**Splitting the dataset for testing and training**

Dataset is splitted into two parts

1. Training
2. Testing

Training dataset is used to build the model and testing dataset is used to test the built model for its accuracy.

To split the dataset for testing and training sklearn module package is used.

import the train\_test\_split() function from sklearn.model\_selection package.

Set the options in the train\_test\_split() function to the following

Set random\_state = 123

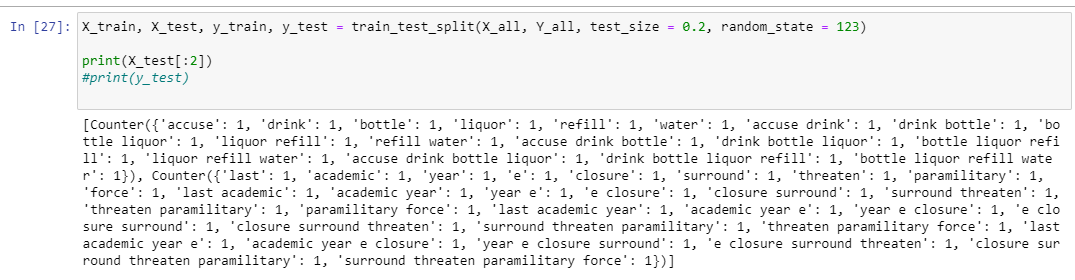
For shuffling the data before splitting the dataset

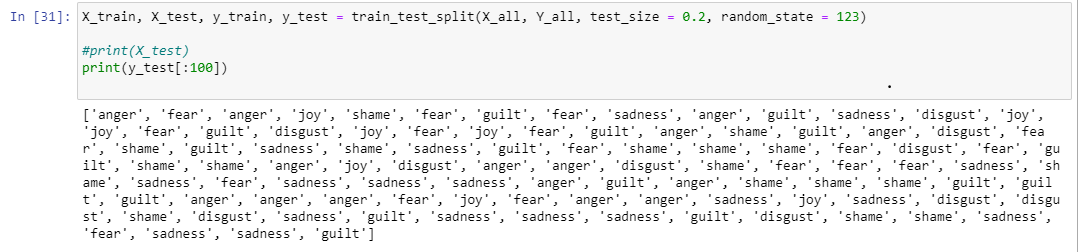
Set test\_size = 0.2

For splitting the data set, 80% data for training and 20% data for testing.

Pass the dataset to the function by setting the above options







**Building the model**

Emotion detection is a multi-class classification problem. Here, the sentence may contain the co-occurrence of words related to more than one emotion; hence, classification is challenging.

SVM has a better solution to deal with multi-class data. Support Vector Machine (SVM) scales relatively well on multi-class data.

We apply the vectorization process to convert text data into numerical feature vectors for training our SVM model. In the proposed system, our objective is to map the short text statement to a specific emotion.

In this project we are building SVM machine learning model with multiple kernels.

SVM kernel are used when the data has to be converted into another dimension to make it separable.

In this project the following kernel function are used.

1. Linear
2. Radial basis
3. Polynomial
4. Sigmoid

Building machine learning models with these kernels

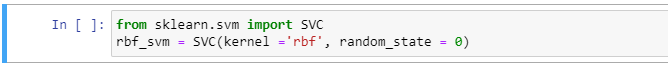
Import the SVC class from the sklearn.svc package

Create the machine learning models for each kernel

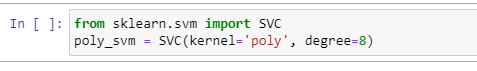
1. Linear SVM



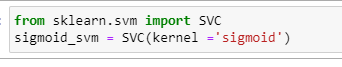
1. SVM with Radial basis kernel



1. SVM with Polynomial kernel



1. SVM with Sigmoid Kernel



**Training**

We train each model sequentially with the prepared dataset.

**Choosing the best model**

Out of the four models SVM with linear kernel performing well.

It has the best accuracy among the models.

**Testing**

**Accuracy**