**IST 736**

**Emotional Classification on Tweets**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Jaishree Palaniswamy**  **Karthik Thokala**  **Priyanka Reddy Yellati**  Applied Data Science  Syracuse University  Syracuse, NY 13210 | |  |  | |
|  | |  | | |

**Abstract**

The purpose of this project is to classify the tweets collected from twitter based on four different emotions such as “Joy”, “Anger”, “Fear” and “Sadness”. Hence, it is an classification problem.Multinomial Naive Bayes and Linear support vector classifier algorithms are used to classify tweets using different vectorization techniques.Insights are obtained from each model and calculated precision, recall, F-measure and accuracy for each model. Performed error analysis to know the reason behind the incorrect classification. Tweets collected has hashtags as well, analysed and found the most common hashtags that are used for each emotion.

**1 Introduction**

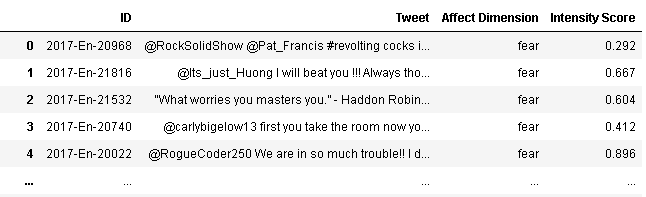
Increase in technology, increased the use of social media platforms. Social media platforms like Facebook, Twitter, Instagram and many more are creating opportunities for users to communicate with each other, express thoughts and post their day-to-day activities. Apart from these, it is also providing users with wide range of information. Users thus join the groups that interest them and that best suit their needs.

Twitter has at least 500 million users and 400 million tweets posted in its site every day. Each of these millions of tweets are used to express their emotions on the current affairs. These tweets are about 140 characters in length and are in the form of texts that contain opinions, expressions, and emotions of the users. Usually, the tweets are not structured and contain many spelling mistakes and non-alpha characters. Analyzing these tweets is a difficult task. However, we have many inbuilt python packages like “nltk” and “scikit learn” that help us in easy analysis of the unstructured tweets. Also, Application of text mining techniques on social networking sites can further reveal results related to human thinking patterns, group identification and recommendation, and opinion about any specific topics of interests.

In this study, we are extracting the tweets from twitter and classifying them into four different them emotions using Multinomial Naïve Bayes and Support Vector Machine algorithms. In this process, we are learning a set of rules from the set of examples in training data and examining the errors. The process of text classification is divided into three phases to extract the tweets from twitter and process these texts into an understandable format. The three phases include Pre-Processing, text mining and result comparison of different models to discover groups and structures in the dataset.

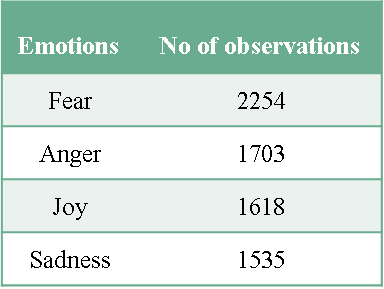
* 1. **Dataset Description**

The data is collected from [Codalab](https://competitions.codalab.org/competitions/17751) and consists of 8711 tweets. It consists of four columns i.e., ID, Tweet, Affect Dimension, and Intensity Score. The ID column is the unique column in the dataset and each tweet in the tweet column corresponds to unique ID. Affect Dimension is the target variable which has four different categories i.e., Fear, Joy, Sadness and Anger. The last column in the dataset is the Intensity score, this score measures the intensity of each emotion from 0 to 1. The higher the score, the higher the emotion in the tweet. Fig(i) is the snippet of our data we are using in our further analysis.



**Fig (i) Twitter Dataset**

The dataset is imbalanced with more number of tweets related to “Fear” and least number of tweets for “Sadness”. Fig (ii) shows the distribution of emotions in the dataset



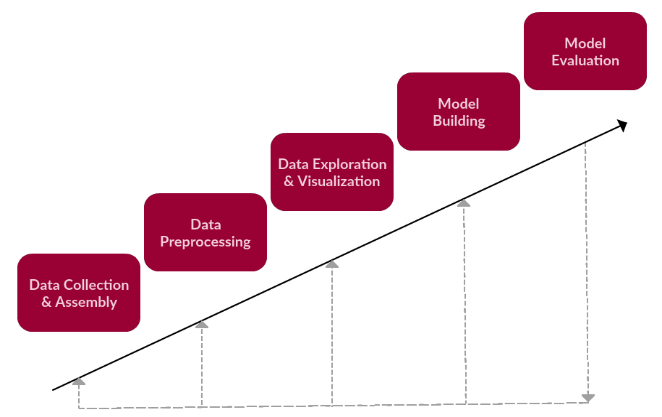
**Fig (ii) Distribution of Emotions**

**2. Methodology**

Fig (iii) shows the different phases we followed for classifying the tweets based on emotion. Initially, we are extracting the tweets from the tweets column in the dataset. Converting the unstructured data to structured format by performing various preprocessing steps. Various vectorization methods are used to covert the text into vectors and machine learning algorithms are applied on it.

Top words such are unigrams, bigrams and trigrams are extracted for each emotion using different vectorization techniques.Data Visualization is used to view the most common words and hastags for each emotion. Count Vectorization and TF-IDF vectorization with different term document frequencies are applied on each model. The hashtags for each emotion are extracted and we try to show the top hastags that are used in tweets for each emotion.

Implemented Multinomial Naïve Bayes, and Support Vector Machine Algorithms. Evaluated these models by comparing the Accuracy, Precision, Recall measures of each model with different vectorization and varying document frequency.



**Fig (iii) Phases in Text Mining**

**2.1 Data Collection and Assembly**

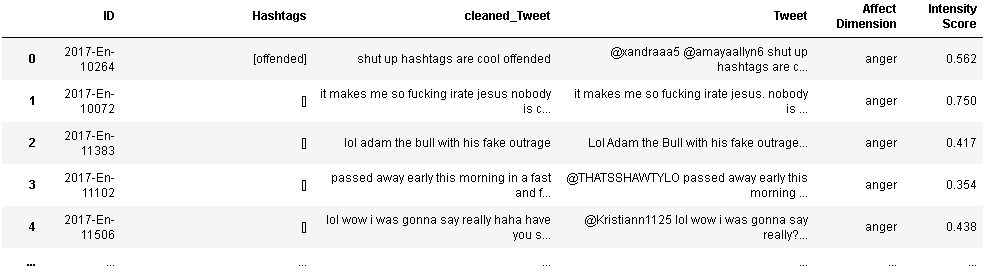
The source location has four different text files with emotion and their tweets in each file. All these files with different lengths are read using pandas and are stored in four different data frames for each emotion. Concatenation function in pandas is used to combine all the data frames created above and the result is stored in a data frame which is used in further analysis.

**2.2 Data Preprocessing**

Twitter provides application interface to pull data for further analysis. Here we have the data of 8711 tweets expressing four different kinds of emotions. The entire analysis revolves around this data making the preprocessing an important step.

After careful analysis we can see the data has information like hashtags which can be considered as a valuable resource. At the same time, these tweets also have numbers, links and white spaces which may not weigh much for the analysis. First, we converted all the tweets to lower case to remove the redundancy of words. We removed special characters, white spaces, numbers, and links from the data to ease the analysis. Each tweet also contains stop words which can be cited as not much useful for analysis hence we cleaned the data for stop words. Coming to hashtags, they can be termed as very important factors for analyze emotion. We created a separate column by isolating hashtags identified by each row. With this the data is ready for classification analysis.

We also observed that there are mentions in each tweet. As part of preprocessing, we eliminated all the mentions and created a new column and named it as “Cleaned\_Tweet” and these tweets are used in our further analysis. Fig(ii) shows the snippet of dataset with additional column “Hashtags” and “Cleaned\_Tweet” that are created after the preprocessing.



**Fig (iv) Twitter Dataset with additional columns**

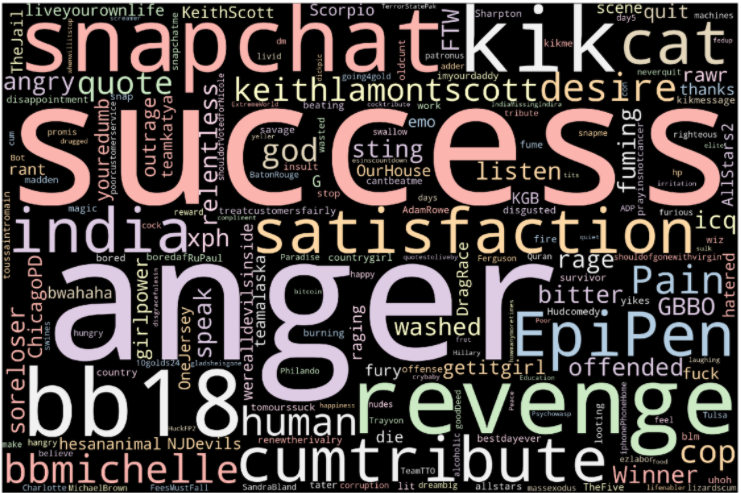
**2.3 Word Clouds**

Most common words in all the tweets are plotted. From the fig(iv), it is observed that “Sad”,“fear”,“Happy”,“Anxiety”,“depression” are most frequently used.



**Fig (iv) Most common words in all the tweets**

Hashtags are extracted from the tweets during preprocessing and most frequently used hashtags are plotted below using Word Cloud.



**Fig (iv) Most common hashtags in all the tweets**

**3. Model Building**

**3.1 Multinomial Naïve Bayes**

The multinomial nave Bayes (MNB) learning algorithm is commonly used in a variety of fields, including text classification. However, when it is used in domains where its naive assumption is violated or where the training set is too small to find reliable probabilistic estimates, its accuracy decreases.

Naïve Bayes classifier Algorithm is a family of probabilistic algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of feature.  
Bayes theorem calculates probability P(c|x) where c is the class of the possible outcomes and x is the given instance which has to be classified, representing some certain features.

**P(c|x) = P(x|c) \* P(c) / P(x)**

Naive Bayes are mostly used in natural language processing (NLP) problems. Naive Bayes predict the tag of a text. They calculate the probability of each tag for a given text and then output the tag with the highest one.

**3.1.1 Pre-Processing**

Imported the required libraries. Prepared the training and testing dataset using the train\_test\_split() method of the sklearn library. Divided the training and test data in 70:30 ratio. Converted the tweets into vector using count vectorizer and TFIDF vector with minimum document frequency as “3” and n\_grams ranging from 1 to 3 (i.e., for unigrams, bigrams and trigrams)

**3.1.2 Results**

The table (i) has the accuracies on the emotion classification dataset using Multinomial Naïve Bayes algorithm. From the table it is observed that increasing n-grams for count vectorizer decreased the accuracy. Hence, it is clear from the results that trigram TFIDF vectorizer has the maximum accuracy.



**Table (i)**

The classification report in Table (ii) is the result obtained from trigram count vectorizer for TFIDF Vectorizer using Multinomial Naïve Bayes Algorithm. From the Precision column it is observed that joy is properly classified for this model, whereas, fear is incorrectly classified

**Table (ii)**

**3.1.3 Top features**

**Top 10 sad words:**

[(-3.06315808361909, 'depression'), (-2.9779006235290986, 'blues'), (-2.8192775810515958, 'depressing'), (-2.716483313393928, 'sadness'), (-2.642976702968986, 'sober'), (-2.5381182656416668, 'grim'), (-2.4003190416201647, 'stayed'), (-2.380562114527775, 'sink'), (-2.3363745024679723, 'dull'), (-2.3075329073810753, 'sadly')]

**Top 10 anger words:**

[(2.4003146143238077, 'insult'), (2.4053577270642093, 'offend'), (2.455009639462162, 'bitter'), (2.4931857999672484, 'outrage'), (2.4948699545225885, 'burning'), (2.5512259174413945, 'angry'), (2.7023521461920694, 'rage'), (2.738964329590294, 'snap'), (2.792642605504893, 'revenge'), (2.8215154240069893, 'fuming')]

**Top 10 Joy words:**

[(-2.918225676651635, 'optimism'), (-2.8195917240108344, 'cheer'), (-2.7781654049275666, 'glee'), (-2.7461346028533367, 'smiling'), (-2.630526793195875, 'lively'), (-2.5664140640216537, 'hilarious'), (-2.467191800093202, 'pleasing'), (-2.447824066884831, 'cheering'), (-2.4452221858245515, 'breezy'), (-2.4047965266756623, 'musically')]

**Top 10 Fear words:**

[(2.4081137680261264, 'alarm'), (2.4615491941169347, 'terrible'), (2.5447723153927475, 'nervous'), (2.5504115211685683, 'horrible'), (2.578457620208244, 'terror'), (2.591630552801604, 'bully'), (2.6316631189008843, 'panic'), (2.6578083531981544, 'terrorism'), (2.69707126217736, 'shocking'), (2.8730523362888265, 'nightmare')]

**3.1.4 Error analysis**

Accuracy yielded for current models range from 80% to 89%. Though this is a better than base line accuracy it is important to understand why there is a loss in prediction. The machine learning algorithms may fail because of the generalization of weights attached to each word during classification. Let us look at the multinomial naïve Bayes classifier where joy is classified as sadness. The tweet we are looking at is “*this tweet is dedicated to my back pain which i do not understand because i am youthful and spry full of life vivacious*”. If we examine closely for the words used there are top features from both the emotions, taking sadness there are words like “pain”, “understand” and “back” while looking at words which can examine joy there are words like “youthful”, “vivacious”, and “spry full”. Our understanding is that MNB classified the word which are expressing sadness with more weight making a wrongful classification. Let us look at one more example “*favourite quote of the year so far is by i swore at a parsnip*.”, this tweet is classified as joy while the original emotion it express is Anger. Here words like favorite can classify the tweet with Joy but there is a sarcasm which went undetected.

**3.1.5 Cross Validation**

Performed 5-fold cross validation test on Multinomial Naïve bayes using Count Vectorizer with minimum document frequency as “2”. It is found that the accuracy before removing stop words is “82.4” and is “83” after stop words removal.

**3.2 Linear Support Vector Machine**

SVM is the most widely used ML technique-based pattern classification technique available nowadays. SVM classifies the given data into separate categories. This process is done by taking a linear line across the data and calculating the distance(margin) of closest objects (support vectors) fromall the categories to the line. The line with highest margin to support vectors on each side is the ideally classified model for the data. This is the process in which SVM works in a two-dimensional plane. If we have multiple dimensions, we take the line as a hyperplane separating various categories of data.

**3.2.1 Pre-Processing**

Imported the required libraries. Prepare the training and testing dataset using the train\_test\_split() method of the sklearn library. Used different vectorization techniques, with minimum document frequency set to “2”. Also removed stop words during vectorization.

**3.2.2 Results**

Implemented linear Support Vector Classifier using both count Vectorizer and TFIDF Vectorizer with minimum document frequency as “2” and by removing stop words. It is observed from the results that the accuracy is high for Count Vectorizer with trigrams. However, the accuracy decreased for TFIDF.



**Table (iii)**

Table (iv) shows the classification report of Linear SVC using trigram countVectorizer. It is observed from the classification report that the emotion “Joy” is more highly classified, whereas fear and Joy are less correctly classified.



**Table (iv)**

**3.2.3 Top features**

**Top 10 sad words:**

[(-2.705871434283337, 'blues'), (-2.704620233315381, 'depression'), (-2.6319005153337782, 'sad'), (-2.5880774469421386, 'sober'), (-2.541579207803034, 'sadness'), (-2.429218160573786, 'depressing'), (-2.3059674545870417, 'dark'), (-2.2959927165034326, 'grim'), (-2.1749071072459176, 'lost'), (-2.105364141005753, 'unhappy')]

**Top 10 anger words:**

[(2.0419578407814587, 'snap'), (2.092328442471524, 'offended'), (2.0955202702875484, 'raging'), (2.114549892171484, 'outrage'), (2.3101945021834216, 'bitter'), (2.342515332344053, 'angry'), (2.3538223560069875, 'fuming'), (2.380939474153317, 'revenge'), (2.3820565015966855, 'anger'), (2.480642223450739, 'rage')]

**Top 10 joy words:**

[(-3.1165329454088537, 'hilarious'), (-2.8497192707628045, 'optimism'), (-2.6421222173607655, 'laughter'), (-2.6392968937477885, 'smile'), (-2.546948800024391, 'happy'), (-2.3017129819498447, 'cheer'), (-2.2905878128614767, 'glee'), (-2.266358903817528, 'smiling'), (-2.1632434418121704, 'animated'), (-2.0396501970673646, 'hilarity')]

**Top 10 fear words:**

[(2.0953815385491303, 'shocking'), (2.1426811631642373, 'start'), (2.199182133013344, 'terrible'), (2.2129941834210887, 'awful'), (2.2164230717463425, 'horror'), (2.2194012605920754, 'panic'), (2.254860152527843, 'nervous'), (2.259528750121179, 'terrorism'), (2.3263812768255763, 'nightmare'), (2.3728161636501004, 'fear')]

**3.2.4 Error Analysis**

Looking at SVM where accuracy is little higher than MNB there are couple of errors where fear is classified as joy. One example would be “*happy th may u keep haunting us for many years writing*”. Here joy can be classified with the words “happy” while fear can be classified by the use of word “haunting”. Word “Happy” holds a higher ground while classifying joy while “haunting” can express fear it is not as frequent as happy making SVM assign more weight for happy eventually classifying the entire tweet expressing joy as emotion. Another example is “*we need bust up the elites in dc we need jobs all we need to clean up the blithe in inner cities rebuild housing* *cont.”* While this tweet is classified as anger by the initial use of word“bust” there is a negation in the sentence pooling it to Joy this went undetected. From this we could say classifying the words are important but the context these words and negations used matters for perfect classification.Some areas of improvement would be to run ML models on more data to get generalized ranking for each word.

**3.2.5 Cross Validation**

Performed 5-fold cross validation test on Linear SVC using Count Vectorizer with minimum document frequency as “2”. It is found that the accuracy before removing stop words is “86.9” and is “87.3” after stop words removal.

**4. Model Evaluation**

By looking at the classification models ran using various vectorizations and their accuracies we can say SVC has best prediction rates. Let us dig deep into the test statistics for a moment, MNB classifier which takes word count as the primary metric to classify yielded an accuracy of around 82% when combined with both Trigram and TF-IDF vectorizers. The tweets expressing joy are classified better while tweets termed as expressing sadness has less test scores. One insight we can think of is it is harder to classifying sadness over text. The drop in accuracy can be the direct result of weighing the words based on their usage. Coming to the SVM model we used unigrams, bigrams, trigrams and TF-IDF vectorizer. SVC uses grouping of words as the classification metric hence the better test statistics with bigrams and trigrams. In par with MNB classifier SVC yielded better statistics for tweets expressing joy. One part SVC did better is when classifying sadness. As discussed, sadness is a complex emotion, SVC’s method of grouping combined with bigrams and trigrams yielded better prediction rates. Coming to the overall test statistics SVC when modeled with Bigrams yielded an accuracy of 8% and has best test statistics among two models with various combination of vectorizers.

**5. Conclusion and Future Work**

On tweets extracted from Twitter, the Multinomial Nave Bayes and Linear Support Vector Machine algorithms were used to train text-based emotion recognition. This paper aims to investigate machine learning algorithms in depth to provide the best model of text-based emotions on tweets. Anger and Joy consistently had the best results in all models, while Fear and Anger had the worst

**6. References**

* <https://ieeexplore.ieee.org/abstract/document/7449218>
* <https://www.researchgate.net/publication/301443639_A_text_mining_application_of_emotion_classifications_of_Twitter's_users_using_Naive_Bayes_method>
* <https://www.sciencedirect.com/science/article/pii/S1877050921001320>
* <https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/>