Twitter-Covid19 Sentiment Analysis

Abstract:

Sentiment analysis is the process of finding and categorizing views or sentiments conveyed in a text source. In this task, we are trying to predict the emotions of a twitter user with respect to COVID-19 depending the tweets and other features extracted from the tweets. In this task, we are extracting features from the tweet text such as hashtag, word count, sentence count and so forth, which can contribute directly or indirectly to the opinion prediction. Furthermore, tweets have been processed and converted to vectors using custom Gensim word2vec model. These features are then sampled and used for training a voting classifier model to predict the sentiment for unseen data. The performance of the model is evaluated by calculating precision, recall, f1 score and accuracy. With the mention approach the model is able to predict up to 85% of the unseen data successfully.

Data description:

1. The dataset comprises of information of 100 tweets where each entry has information on tweet text, tweet timestamp and sentiments (Angry, Fear, Joy, Neutral, Sadness, Surprise and Unrelated) associated with each tweet.
2. Distribution of data in the dataset is as follows:
   Neutral: 48 entries
   Fear: 33 entries
   Surprise: 7 entries
   Sadness: 4 entries
   Unrelated: 3 entries
   Joy: 3 entries
   Anger: 2 entries
   By the above summary of data distribution, it is evident that the dataset is highly imbalanced.
3. Datatype associated with each feature in the dataset is as follows:
   Tweet_id: int64
   Created_at: object
   Full_text: object
   Sentiment: object

Implementation:

Below are the different methodologies carried out on the dataset for the given task.

Data Cleaning:

1. Dataset is checked for null values, where presence of null values can affect the performance and accuracy of algorithms. Since the dataset doesn't contain any null, imputation is not required.
2. Eliminating columns that are of less important such as created_at and tweet_id.

Generation a feature set:

Using the tweet text from the dataset, below features are extracted.

1. ***Number of capital characters, Number of capital words and Number of Hashtag:*** These features can be used to express emotions.
2. ***Number of Sentences:*** Sentence count can be a useful feature as it can usually corelate with fear or enthusiasm. Example: According to our dataset, people who are tense tend to tweet more sentences.
3. ***Number of characters and words:*** Word and character count can be useful in understanding the density of words and characters used in tweets based on different sentiment.
4. ***Number of URL:*** URL count is useful information extracted from the user tweet, which provide some insight on user opinion.
5. ***Number of Mentions:*** Most of the time people reply or mention someone in their tweet, counting the number of mentions can also be treated as a feature
6. ***Average sentences, words count and unique words count.***

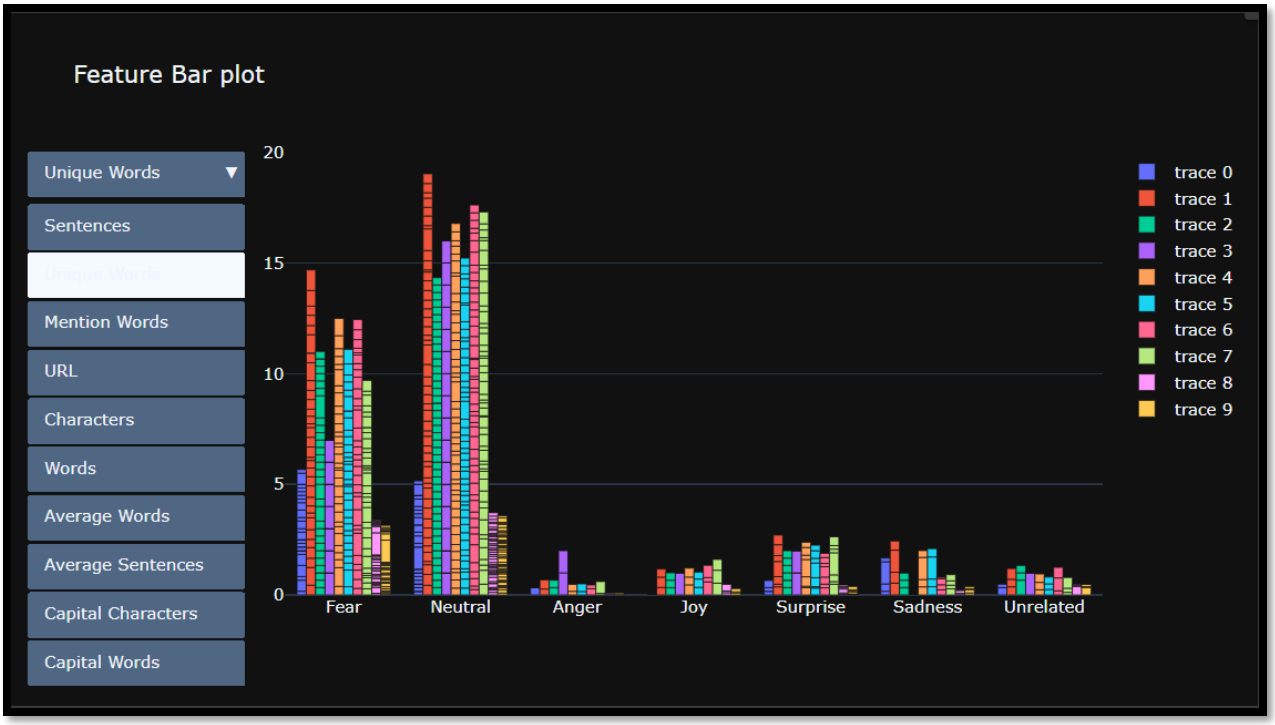
Tweets Preprocessing:

1. Remove the mentions, as tweets used to train the model should be generalized.
2. Remove the hash tag sign (#) but not the actual tag as this may contain information.
3. Replace 2 or more dots with spaces.
4. Remove RT "retweet".
5. Set all words to lowercase.
6. Replace Multiple spaces into single space
7. Remove all punctuations, including the question and exclamation marks.
8. Remove the URLs as they do not contain useful information. But We did notice a difference in the number of URLs used between the sentiment classes, which is extracted into separate feature.
9. Convert the emojis into one word as it is a way to express user emotions.
10. Remove stop words.
11. Apply the Lemmatization to get the tweet words in its root format while maintaining the context.

Data Visualization:

Data visualization provides a quick and clear idea about what the information is through graphical representation.

As part of data visualization different features are analyzed with respect to sentiment using graphs. Also, word cloud provides a greater way to visualize the unstructured text and frequency of words. Tweets with respect to each sentiment has been visualized using word cloud.



Extracted feature Scaling:

Features extracted from tweets have different range of values. As machine learning algorithms can only understand and work on numbers, if the dataset contains wider range of number from ten to thousands, then algorithm gives priority to the highest value. In such situations features with higher values turns out to be dominant which affects the model efficiency.

Therefore, scaling is done using MinMaxScaler () where all the extracted features are scaled between 0 and 1 due to which the model efficiency increases.

Target label encoding:

As we understand from the dataset info the target column in categorical in nature. Since the machine learning algorithms can only understand numeric values, target label in the dataset is converted to numeric values using label encoding.

Generate word embeddings:

Since all the values in dataset has to be numeric in nature, processed tweets should also be converted to numeric values without losing the context. This can be achieved using word embedding.

In this task word embedding is performed using Gensim, wherein it provides the Word2Vec class for working with a Word2Vec model.

Initially each tweet is divided into small unit called tokens, which will be fed as input to the word2Vec class to generate vocabulary set. The trained model will be accessible using wv.

The parameters used building word2vec model is as follows:

1. Vector_size: Number of dimensions of embedding.

2. Window: Maximum distance between the target word to the other words around the target.
3. Min_count: Minimum number of words to be considered while training, words with less than min_count will be eliminated from training.
4. sg: Algorithm to be used for training the model. Skip gram
5. hs: Negative sampling will be used.
6. Seed: Random number generator.

Below are the parameters used for training the model:

1. vector_size=100
2. window=7
3. min_count=2
4. sg = 1
5. hs = 0
6. seed = 34

Tweets to be vectorized will be sent to the trained word2vec model, to obtain vector representation of given tweet. Each token in a tweet will have 100 vectors associated with it, so average of vectors will be considered to represent one vector per token.

Generating training and testing data:

As the dataset is highly imbalanced, combination of random over sampling and under sampling will be applied on the dataset to achieve uniform data distribution.

Random over sampling and under sampling:

Combination of both random sampling will increase the performance compared to isolated methods, where modest amount of oversampling is applied on to the minority class which increases the bias of minority class similarly modest amount of under sampling is applied on to the majority class which decreases the bias of majority class.

Modeling:

Ensemble technique is employed for modeling, where we stack up the base model to provide the optimal prediction results. The prediction from each model will be fed as an input to the voting classifier and average of those predictions will be obtained as output.

Hyperparameters for each model will be tuned using GridSearchCV which increases the performance and accuracy of the model.

Result Evaluation:

As a first step of result evaluation, confusion matrix will be used which provides the summary of prediction results.

Furthermore, model performance can be evaluated using the following metrics:

Precision: Model precision score represents the model's ability to correctly predict the true positives out of all the positive predictions it made. The precision score is a useful measure of the success of prediction when the classes are very imbalanced.

Recall: Model recall score represents the model's ability to correctly predict the positives out of actual positives. This is unlike precision which measures how many predictions made by models are actually positive out of all positive predictions made.

Accuracy: Accuracy is a machine learning model performance metric that is defined as the ratio of true positives and true negatives to all positive and negative observations. The current model can predict 85% of the given labels correctly.

F1 Score: F1 score represents the model score as a function of precision and recall score. F-score is a machine learning model performance metric that gives equal weight to both the Precision and Recall for measuring its performance in terms of accuracy.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	0.69	0.82	13
2	0.90	1.00	0.95	9
3	0.80	0.89	0.84	9
4	0.88	1.00	0.93	7
5	1.00	1.00	1.00	9
6	0.92	1.00	0.96	11
accuracy			0.93	68
macro avg	0.93	0.94	0.93	68
weighted avg	0.93	0.93	0.92	68

Labels decoded: 0: Anger, 1: Fear, 2: Joy, 3: Neutral, 4: Sadness, 5: Surprise, 6: Unrelated

Colab link:
https://colab.research.google.com/github/poojashreeNS/RA_Twitter_Sentiment_Analysis/blob/main/RA_Twitter_Sentiment_Analysis.ipynb

GitHub Link:
https://github.com/poojashreeNS/RA_Twitter_Sentiment_Analysis