Artificial Intelligence & Machine Learning Lab (PCCCS495)

Detection of Hostile/bully tendencies using Twitter Dataset

Souvik Sarkar

12021002004012

2nd Year

Dept. of IT

|  |  |
| --- | --- |
| Index | |
|  | Page |
| Abstract & Acknowledgement | 3 |
| Problem Definition | 4 |
| Introduction | 5 |
| Method | 6 |
| Code | 7 |
| Result | 8 |
| Conclusion | 9 |
| Reference | 10 |

**ABSTRACT**

Artificial Intelligence as well as Machine Learning are two of the biggest technologies in modern day. To develop a strong grab in this field, project is the best way to start on. I built a classification model that recognises hostile/bully tweets by using Natural Language Processing. A sample of a huge Twitter dataset retrieved from Kaggle is used to train the model and its performance is quite good.

Acknowledgement

I express great gratitude to the teachers of Artificial Intelligence & Machine Learning Lab for introducing such an interesting field to work on. All the concepts taught are magnificent. Without the early support I could not make it this far. I also want to thank the whole department and the institute itself for this wonderful subject.

Sign Here

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Souvik Sarkar

12021002004012

A5

2nd year 4th semester

Information Technology

**Problem Definition:**

Nowadays, it is an absolute requirement for applications to find contents violating their terms & conditions and dismiss them for good. When text content is to be analysed, natural language processing stands out in the field of machine learning. I found a dataset on Kaggle containing 1.6M Twitter messages with corresponding label that represents its status of negativity. There are messages of two types – negative tweet represented by 0 and positive tweet represented by 4. Now, I will make a machine learning model that can recognise negative or positive tweets by analysing the text information inside the message. I want to train the model by typical Twitter messages that has certain decisive words to predict unseen future messages as negative or positive. The text content must be cleaned, organised, and digitised for numerical analysis to further provide a probabilistic guarantee of the corresponding label.

**Introduction:**

Since humans express their thoughts and feelings more openly than ever before, sentiment analysis is fast becoming an essential tool to monitor and understand sentiment in all types of data. Here, the data is either positive or negative that are to be classified using machine learning. Bag of Words is well renowned method of natural language processing. After cleaning the data, it is structured within a sparse matrix. The probabilities play the main role to the algorithm that is calculated by the naive bayes classifier.

**Method:**

1. As the dataset is too large to calculate for every record, it is randomly sampled to get a smaller view of the data for latency optimization in training the model.
2. Firstly, words containing usernames and links are removed from every message.
3. The content of each message is tokenized as list of words.
4. Punctuations are removed from the lists.
5. Each word in a list is lemmatized to a base form from other tense of the same root.
6. After that the words are stemmed to a shorter unique expression to reduce the number of characters for processing optimization.
7. The whole data is split into training set and testing set.
8. A sparse matrix is created based on the training data where the message indices are along the rows and the vocabulary is along the columns containing the count of each word in a message as data.
9. The word counts are incremented by 1 in every entry to avoid potential misclassification.

1. Now, a naïve bayes classifier is created to calculate probabilities of each word given a negative and positive tweet.
2. For a new tweet, probabilities of the word elements are multiplied with in case of negative tweet and positive tweet. Whichever is higher, the message is labelled as the same.
3. Thus, the model is evaluated for testing set.

**Code:**

# Data Analysis

import pandas as pd

df = pd.read\_csv('training.1600000.processed.noemoticon.csv',

names=['target','id','date','flag','user','text'])

df.head()

df.info()

# Data Visualization

import seaborn as sns

sns.countplot(x='target',data=df)

df = df.sample(frac=0.01)

sns.countplot(x='target',data=df)

# Natural Language Processing

from nltk.tokenize import word\_tokenize

from nltk.stem import WordNetLemmatizer,PorterStemmer

from nltk.corpus import stopwords

from string import punctuation

def text\_process(tweet):

    words = tweet.split()

    no\_mention = [w for w in words if '@' not in w]

    no\_link = [w for w in no\_mention if 'http' not in w]

    tokens = word\_tokenize(' '.join(no\_link))

    no\_punc = [w for w in tokens if w not in punctuation]

    lemmer = WordNetLemmatizer()

    lemmed = [lemmer.lemmatize(w) for w in no\_punc]

    no\_stop = [w for w in lemmed if w not in stopwords.words('english')]

    stemmer = PorterStemmer()

    stemmed = [stemmer.stem(word) for word in no\_stop]

    return stemmed

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'],df['target'],

test\_size=0.3)

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.pipeline import Pipeline

pipe = Pipeline([

    ('Bag of words',CountVectorizer(analyzer=text\_process)),

    ('Classifier',MultinomialNB())

])

pipe.fit(X\_train,y\_train)

predictions = pipe.predict(X\_test)

# Model Evaluation

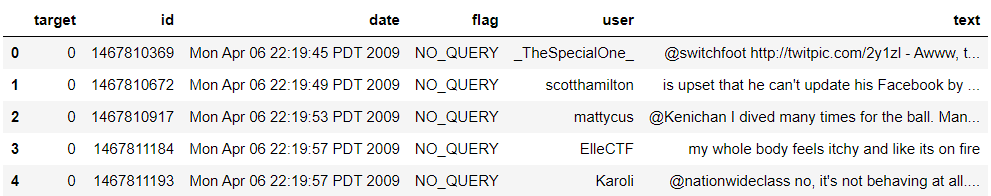
from sklearn.metrics import confusion\_matrix,classification\_report

print(confusion\_matrix(y\_test,predictions))

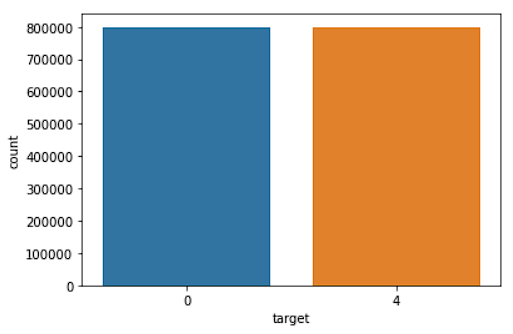
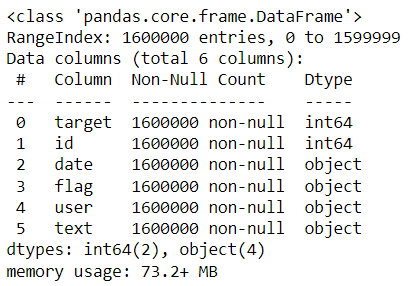
print(classification\_report(y\_test,predictions))

**Result:**

The structure of the dataset to be analysed –

****

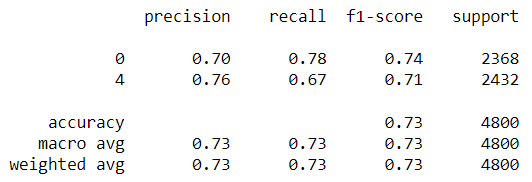
Information about all the features of the dataset and count-plot visualizing the target category –

****

|  |  |  |
| --- | --- | --- |
| Predicted Result | Negative Tweet (0) | Positive Tweet (4) |
| Actual Result |  |  |
| Negative Tweet (0) | 1854 | 514 |
| Positive Tweet (4) | 798 | 1634 |

Confusion Matrix –

Classification Report –



**Conclusion:**

The model has F1 score of 0.73 that can be considered as good. In the context of the given data, more adjustments could be done to result into a better model. The model has a downfall of the recall score for positive messages that can be an issue. It can classify the negative messages with higher efficiency. Overall, the machine learning model performs well enough on typical real-world data.

**Reference:**

Twitter Dataset –

<https://www.kaggle.com/datasets/kazanova/sentiment140>

Machine Learning Tutorial –

<https://www.udemy.com/share/101Wjc3@9GHPhv1Wg0heXTk4SWFRxn1041Xl-ResZBDKFuPDS8W5_xEfqDN4YqGwIsXGlQQ9kw==>

Bag of Words approach –

<https://towardsdatascience.com/basics-of-countvectorizer-e26677900f9c#:~:text=Countvectorizer%20is%20a%20method%20to,sparse%20matrix%20as%20shown%20below>

Naïve Bayes classifier –

<https://www.javatpoint.com/machine-learning-naive-bayes-classifier#:~:text=Na%C3%AFve%20Bayes%20algorithm%20is%20a,a%20high%2Ddimensional%20training%20dataset>