FINAL PROJECT REPORT

APPLIED MACHINE LEARNING AND DATA SCIENCE-2020

COURSE CODE-002

NLP-TWEET SENTIMENT ANALYSIS

SUBMITTED BY-

SANSKRITI ANURAG SRIVASTAVA

1.ABSTRACT

This project addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral. Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users [24] - out of which 100 million are active users and half of them log on twitter on a daily basis - generating nearly 250 million tweets per day [20]. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analysing the sentiments expressed in the tweets. Analysing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

2:DATASET DESCRIPTION:

1. TRAIN DATASET:
   This dataset has the columns tweet_id, sentiment and tweet_text. It has 21465 data entries.
2. TEST DATASET:
   This dataset has the columns tweet_id and tweet_text. We will run our prediction on this dataset. It has 5398 data entries.

3.EXECUTIVE SUMMARY

1. Loading of libraries:
   Sentiment analysis involves natural language processing because it deals with human-written text. To process this and solve the problem we have to import some libraries to proceed with the project. Some libraries will be imported to process the datasets. Libraries like pandas,numpy, regex and nltk. The remaining libraries are machine learning libraries such as sklearn and its various functions such as LogisticRegression, MultinomialNB, accuracy_score.
2. Function to load the dataset and remove unwanted columns:
   The dataset has three columns 'tweet_id', 'tweet_text', 'sentiment', but we are only interested in 'tweet_text', 'sentiment'. Hence we will write functions to remove unnecessary columns.

3. Text cleaning:

NLP tasks require clean data in order to give results with higher accuracy. In this project, a function has been written to clean the text. Text cleaning removes all unwanted and unimportant parts of a text in order to give optimised results. The following steps have been taken to clean the text:-

- Removing all words beginning with “@” by using re.sub(). It will be used to replace multiple substrings with the same string.
- Removing any word that starts with “http” since hyperlinks are not required for this project. re.sub() function has been used for this task.
- Removing all symbols. For this, the “isalpha()” function will be used to remove all alphanumeric characters.
- Converting the tweet to all lower case.

4. Tokenization and removal of stopwords:
    - **Tokenization** is the process by which big quantity of text is divided into smaller parts called tokens. Here, the tokenize() method is used. The result is saved in a list. These tokens are very useful for finding such patterns as well as is considered as a base step for stemming and lemmatization.
    - **Removing stopwords-**The stopwords.words() method from nltk will be used. Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence.

5. Lemmatization:
   Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. The tokens that have been created have been lemmatized and saved as a list. wordnet from nltk is used for this purpose.

6. Data vectorization:
   This step, converts string features to numerical features. Here,TF-IDF vectorization is used.
    - **TF-IDF** stands for Term Frequency-Inverse Document Frequency which basically tells importance of the word in the corpus or dataset. TF-IDF contain two concept Term Frequency(TF) and Inverse Document Frequency(IDF).
    - **Term Frequency(TF):** The number of times a word appears in a document divded by the total number of words in the document. Every document has its own term frequency.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

- **Inverse Data frequency(IDF):** The log of the number of documents divided by the number of documents that contain the word w. Inverse data frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = log(\frac{N}{df_t})$$

- Finally, the TF-IDF is simply the TF multiplied by IDF.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

7. Naïve Bayes Model:
   multinomial Naive Bayes or multinomial NB model, a probabilistic learning method. The probability of a document d being in class c is computed as

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

   where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c. We interpret $P(t_k|c)$ as a measure of how much evidence t_k contributes that c is the correct class. $P(c)$ is the prior probability of a document occurring in class c.
   After executing this model, The resultant accuracy is `0.5918937805730259`

8. Logistic regression model:
   Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. logistic regression produces a logistic curve, which is limited to values between 0 and 1.
   Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the “odds” of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group which makes it an efficient classification algorithm.

$$0 \leq h_\theta(x) \leq 1$$

Logistic regression hypothesis expectation

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

The Hypothesis of logistic regression

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Cost function of logistic re

To minimize the cost function we have to run the gradient descent function on each parameter.

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Gradient descent

After running this model, the accuracy obtained is `0.6457023060796646.`

4.CONCLUSION

In this project, we built a model to analyse the sentiment behind a tweet. Sentiment analysis can be used for various purposes. 2 models were tried and finally it can be concluded that logistic regression provides a better accuracy.