MACHINE LEARNING - CS60050

Assignment - 2: Naive Bayes Classifier

Group Details:

Group : 17
Jatoth Charan Sai - 19CS10035
Rahul Eaga - 19CS10029

The task is to classify racist or sexist tweets from other tweets. Given a training sample of tweets and labels

Procedure:

a. The data is given in the form of a CSV file(trai.csv). We read the data using pandas.read_csv API.
b. The data consists of 7% of racist or sexist tweets and 93% positive tweets(two output classes).
c. Known short notations are converted to their full forms(by expand() function), and then stop words are removed by splitting the text by ‘re’ python package(by preprocess_tweet() function).
d. Vocabulary was created to store a set of words taken from the tweets.
e. Data of size 31962 and vocabulary of size 38961, which is large so feature matrix M is formed using scipy.space.csr_matrix.

We were randomly splitting the feature matrix into train split and test split with 70:30 ratio. By training the data with a Naive Bayes classifier on the train split, we know the accuracy.

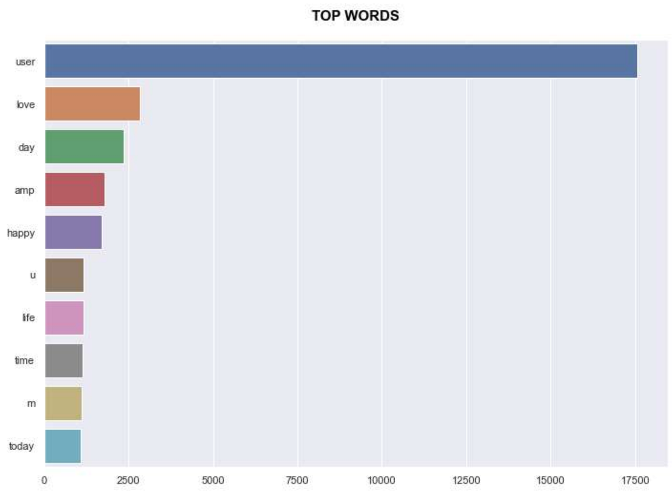
- x_train , y_train, x_test, y_test

Data Set Description:

Data consists of 2 output classes 1. Racist tweets, 2. Non-racist tweets
Number of attributes - 2 : 1. Tweet(sentence), 2. label(integer -- 1/0)(selector)

1. Total number of instances : 31962
2. Vocabulary : list of all words from tweets
3. VOC : list of set of words from list of vocabulary
4. VOC_ind : dictionary of words with its index in VOC as value.

Following graph show’s top 10 words from tweets.



5. Racist : non-Racist tweets -> 7.01 : 92.98

Implementation :

Class and Methods for **Naive Bayes Classifier** used in Python :

Class **NBC:**

Methods in the class

1. predict ()
2. fit ()
3. info ()
4. find ()

Other functions:

1. expand()
2. preprocess_tweet()

Methods in **NBC**:

1. `fit(self, X, Y):`
   - 1.1. This function trains the input data (X, Y).
   - 1.2. Train data consists of 2 classes(C_i) 1. Racist tweets(1) , 2. Non-racist tweets(0)
   - 1.3. Estimating $P(x_i|C_k)$ for discrete variables, a fraction of times the word(x_i) occurred in a class(C_k). (number of distinct words is the number of columns in the matrix).
   - 1.4. Maintains a list of estimates $P(x_i|C_k)$ for different classes (in this case, it's 2)
   - 1.5. Output classes that are classified into Racists and Not Racists. We define the estimates(probabilities) of the racists and non-racist as probability_0 and probabiituy_1, respectively.
2. `predict(self,x, LC = False):`
   - 2.1. This function returns the predicted output(y_pred) from the data set that 1 if the likelihood of the tweet being racist is more than the likelihood of the tweet being non-racist; otherwise 0.

3. `info(self, X, Y):`
   - 3.1. It prints out the accuracy, precision, tf-score, sensitivity and specificity with and without the Laplace correction.
   - 3.2. It also gives the values true positives, false positives, true negatives, false negatives.
4. `find(self,a,n):`
   - 4.1. Returns 95% confidence interval for value `a` on `n` size data.

Other **Functions**:

1. `expand(sent):`
   - 1.1. This function converts the known short forms present in sentence(`sent)` to full forms

1.1.1. Example: can't -> can not , *'t -> * not....

2. `preprocess_tweet(text)` :

2.1. This function takes takes pandas series data['tweet'] as `text` and split the text in and converts to lowercase alphabets and removes stopwords and joins the remaining words and returns the pandas series

Train the classifier by `NBC.fit(x_train, y_train)` function
Then test the test_data by `NBC.info(x_test, y_test, LC  = True/False)`
Above method prints the accuracy, precision, f-score, sensitivity, specificity, etc… with 95% confidence intervals.

- 95% confidence interval formula $\pm\ 1.96\sqrt{a(1\ -\ a)/n}$.

❖ **RESULTS and ANALYSIS**:

After testing the test data on classifier`NBC.info(x_test, y_test, LC  = True/False)`
It gives the following results:

without laplace correction

accuracy :
94.817 +/- 0.444
precision =
80.137 +/- 0.799
f-score =
48.497 +/- 1.000
sensitivity =
34.770 +/- 0.953
specificity =
99.349 +/- 0.161

true positives : 234
false positives : 58
true negatives : 8858
false negatives : 439

With laplace correction

accuracy :
94.848 +/- 0.442
precision =
65.674 +/- 0.950
f-score =
60.289 +/- 0.979
sensitivity =
55.721 +/- 0.994
specificity =
97.802 +/- 0.293

true positives : 375

false positives : 196

true negatives : 8720

false negatives : 298

Given data consists of 7% of racist tweets, most of the words in the vocabulary gives the probability of a word appearing in racist tweets is 0.

The above problem is solved by Laplace correction, So the true positives before and after are increased and false positives are decreased.

Before laplace correction accuracy = 94.848% , 95% confidence interval = 0.444%

After laplace correction accuracy = 94.848% , 95% confidence interval = 0.442%.