HW1: LOGISTIC REGRESSION CLASSIFIER

* Name : B Sai Sahana Bhargavi
* Username on miner website: word2vec
* Rank and accuracy score: 0.59 <324>

**Introduction:**

Implemented Logistic regression classifier to predict sentiment for 18506 reviews for baby products provided in data set.

**Approach:**

1. **Cleaning Data and Preprocessing:**
2. **Read training document**: read the document containing reviews along with respective sentiments (+1 or -1) from training folder and stored all the training data into a “Traindata\_df” dataframe.
3. **Read testing document:** read the file that consists of reviews from testing folder line by line and stored the test data into a “Testdata\_df” dataframe.
4. **Text Preprocessing:**

* Foreach review read from the dataframes Traindata\_df and Testdata\_df, removed the words with numbers, special characters and converted into lowercase.

used following expressions

re.sub("\d+", "", str(text)) #for removal of numbers

re.sub('[^a-z,A-Z]', ' ', str(text)) #for removal of special characters

* **Removal of Stopwords and applying Lemmatization:**

For every word in the each review, applied lemmatization technique using WordNetLemmatizer from NLTK,if the word is not present in the stopwords list created from NLTK library. Lemmatization was chosen as it retains the meaning of the word instead of removing some characters at the end alike stemming.

All the cleaned and preprocessed reviews are appended into “preprocessed\_reviews” list

1. **Converting into Vectors:** Created a vector of TF-IDF features using TfidfVectorizer() of sklearn library with max\_features=1000 ,so that it considers top 5000 features ordered by term frequency across the whole set of preprocessed reviews and included ngram range (1,2) that consists of single and double words in order to improve the accuracy of the model, fit\_transform(preprocessed\_reviews) can learn vocabulary and IDF from training document and return a term-document matrix stored into final\_tf\_idf and the fit\_transform (preprocessed\_test\_reviews) allows to transform the testing document to term-document matrix and stored into final\_tf\_test\_idf.
2. **Logistic Regression:**

Created model using LogisticRegression() from sklearn.linear\_model with train data obtained after processing the above steps.

4. **Conclusion:**

The Logistic regression classifier was introduced in this assignment, which provided me a better view and concept on classification and data mining. I have learned how to deal with text document and create tf-idf. Further, there is a scope for improvement in terms of implementing Logistic Regression model using GridsearchCV and tuning hyperparameters.