CS558, HW1

name: Negar Nejatishahidin
Gnum : 01207447

Approach and Methodology

Read the data:

I first read all the data from the both test and train file in two different para meters, then shuffle the train file to be sure that in the cross validation part I will not face face accuracies. After that I builds my labels array and combine both test and train in to one parameter.

Preprocessing:

As a preprocessing I did three main tasks to clean my data :

- Remove all the stop words, punctuations and numbers form the data.
- Tokenize the data(it is not different for me to use split() or tokenize, because the data is cleaned and split() function is faster).
- Reduced the world to the root(delete ing, plural form and …) with NLTK function (PorterStemmer).
- I also keep the words which has more that 4 letter.
  As an example this is the final cleaned tokenized data:
  [‘love', 'crib', 'babi', 'sleep', 'pretti', 'much', 'sinc', ‘brought’]

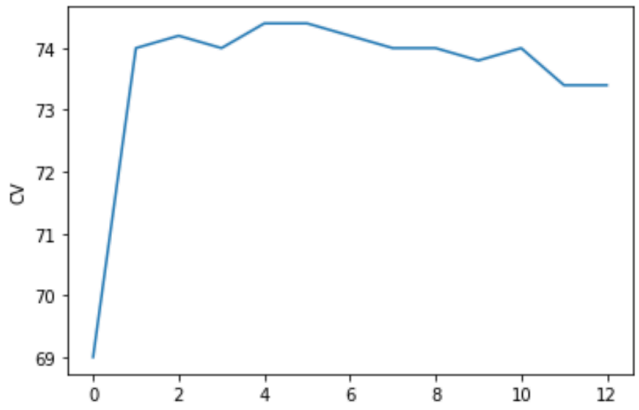
Fined the features / Dimensionality reduction:

First I got the frequency of each word in the train and test data separately. Then I got the 20% of most frequent ones of test and train separately. Then I fined the intersection of these two sets which reduced the features from 26967 to 4611 (I also tried 30 and 40 but they get the same accuracy for TFIDF. Thus it was not computational beneficial.

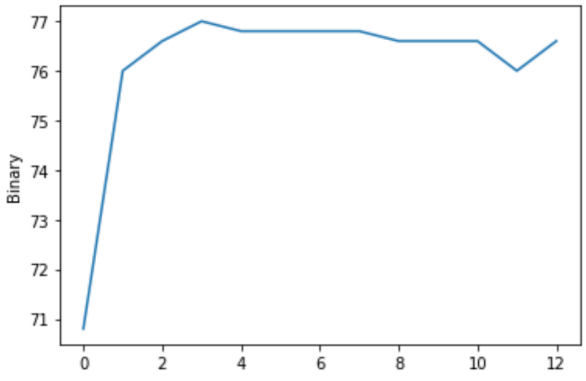
Feature vector methods:

I tried three ways to make my feature vectors

1. **CountVectorizer:** I used sklearn function with my own feature list which described in previous section (I did nit use separate function for training and testing vectorization because each count vectorizer vector is independent to the whole other dataset.)For K=33 best result of cross validation.



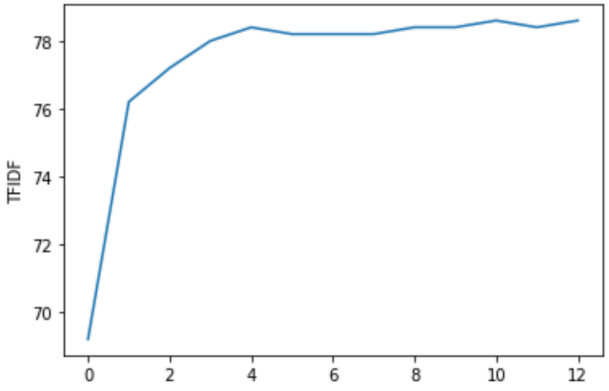
2.Binary:I used sklearn function with my own feature list which described in previous section. (I did nit use separate function for training and testing vectorization because each binary vector is independent to the whole other dataset.). For K=25 best result of cross validation.



3.TFIDF: I made this matrix with the use of sparse matrix of sklearn. It has been created of three main part: (for this part I separately filed the TFIDF matrix of the train and test sets, the result is different with the cross validation.)

- TF function which create the TF part.
- IDF function which multiply TF and IDF.
- The normalization part, which normalize with L2 Normalizer.

For K=81 best result of cross validation.



The KNN Model:

My KNN is stored in models/KNN. It consist of three main part:

- The train part(which for KNN is only storing the data)
- The Distance function: it first normalize the inputs and then compute the dot product which is the cosine similarity function.
- The prediction part which get the distance matrix and sort the whole matrix, get the insect of the K greater point. Then find the mode among them and store the label.

Results:

In the submitted accuracy my best results was for binary and then count vectorizer and after them TFIDF. I am not sure for K=55 for all of them.

I Also implement the k=5 fold crossValidation to fined the best K but because of the constrain I could not submit to fined the best results.

I had a problem Which I could not fix :D,
I have to run two times my writing on the list part to get the wight number of lines in the file.