Assignment-2

Lalit Meena,2019CS50439

Q 2 Binary Classification:

We have expressed the SVM dual objective in the form of equation involving matrices (P,q,etc)in the respective part of assignment.

Digits-(4,0) as (+1,-1) classes

Size of preprocessed training set involving(d1,d2)=4000

Part- i,ii (linear kernel)

total SV- 1507

W= computed and can be printed

B= -0.14603501304812738

training time- 23.781121253967285secs

Accuracy- 70.8 %

Percentage of training samples constitute the support vectors =37.68%

iii)

#top 5 respective images get saved with w one also-

Here in images might be ,+1(deer)-4 and -1 (aeroplane)-0

Part b(i)

total SV- 1756

W- computed and can be printed

B= -1.8896381212773345

training time- 116.24798059463501 secs

Accuracy- 85.6 %

Matched support vectors with linear-1128

iii)

#top 5 respective images get saved

iv)

we can easily observe significant improvement in accuracy with gaussian kernel as linear one is underfitting it ,as gaussian one is more powerful,flexible,more support vectors.

Part c

[LibSVM]

Sv_idx-[1 2 4 ... 3994 3995 3999]

b-[0.04546923]

linear_libsvm -

total SV 1350

w computed and can be printed

training time 50.09999895095825

accuracy 74.75

gaussian_libsvm -

total SV 2647

w cannot be computed

b [-0.30639527]

training time 20.32749342918396

accuracy 86.5

Sv_idx-[0 1 2 ... 3994 3995 3996]

In sklearn,Matched support vectors with linear-1097

Observations(i,ii,iii,iv)explaining with above data-

1.LIBSVM package implementation is faster then manual training procedure by cvxopt in case of gaussian one but same around in linear one.

2.gaussian implementation takes more training time but also provide more accuracy as compare to linear kernel svm.

3.Total number of support vectors are more (roughly double times)in gaussian implementation.

Q 3 multi Classification:

(a)

Own multi model-

Ver1-

Test accuracy 57.12

train accuracy - 61.27

(b)

[LibSVM]classes-

[0 1 2 3 4]

fit_status 0

n_features_in_ 3072

gaussian_libsvm -

w cannot be computed

b [-0.45592676 0.17104177 -0.2066209   0.32432484 0.53957834 0.24597443

0.74029807 -0.42024561 0.22080463 0.58065379]

training time 242.59138083457947

test accuracy 61.7

train accuracy 95.37

nsv [1696 1947 1969 1971 1837]

predictions-

[0 0 0 ... 2 3 4]

Bii)Here with **LIBSVM** ,training time is low

Test accuracy is same around own one .

But train accuracy is more overfitted in sklearn and less in own one.

c)Confusion matrix, without normalization

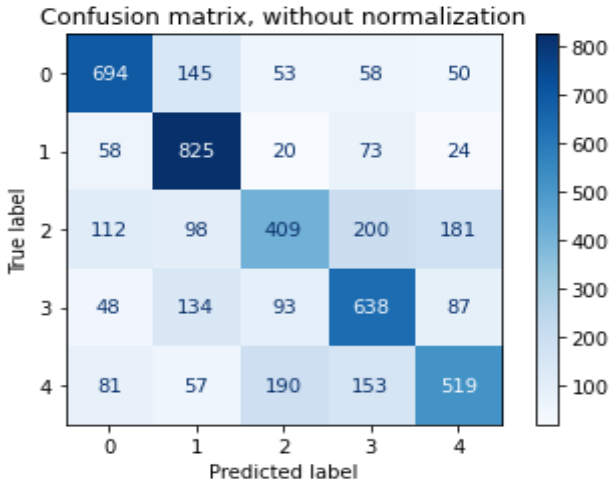
[[694 145 53 58 50]

 [58 825 20 73 24]

 [112 98 409 200 181]

 [48 134 93 638 87]

 [81 57 190 153 519]]



Normalized confusion matrix

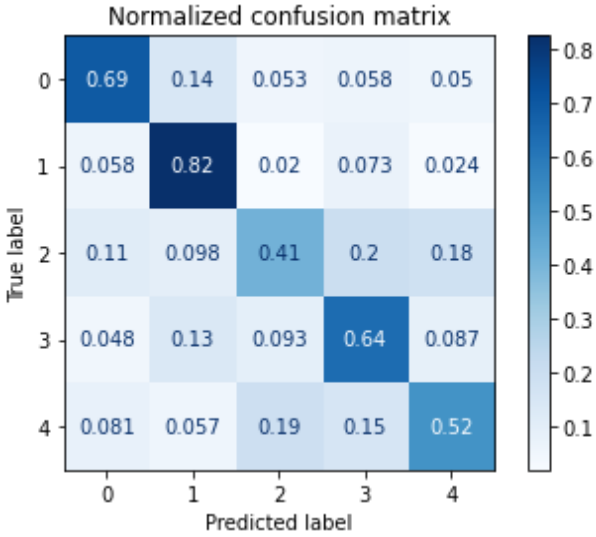
[[0.69 0.14 0.05 0.06 0.05]

 [0.06 0.82 0.02 0.07 0.02]

[0.11 0.1 0.41 0.2 0.18]

 [0.05 0.13 0.09 0.64 0.09]

 [0.08 0.06 0.19 0.15 0.52]]



own multi model confusion matrix-

Confusion matrix, without normalization

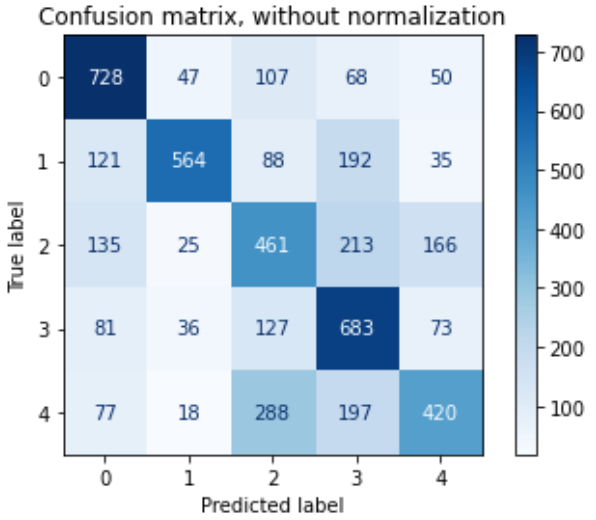
[[728 47 107 68 50]

 [121 564 88 192 35]

 [135 25 461 213 166]

 [81 36 127 683 73]

 [77 18 288 197 420]]



Normalized confusion matrix

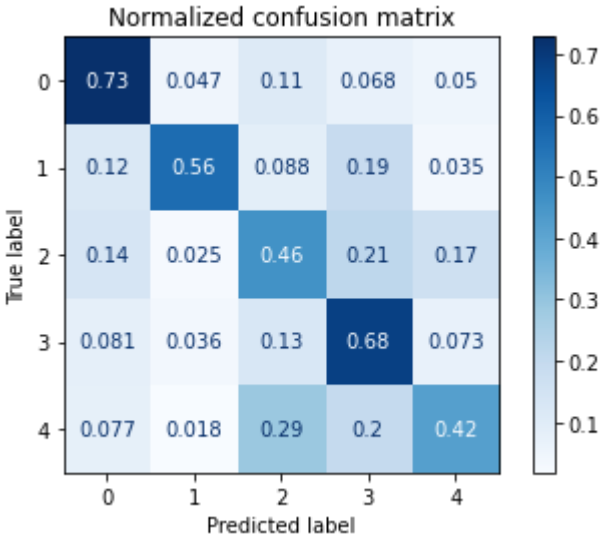
[[0.73 0.05 0.11 0.07 0.05]

 [0.12 0.56 0.09 0.19 0.04]

 [0.14 0.03 0.46 0.21 0.17]

 [0.08 0.04 0.13 0.68 0.07]

 [0.08 0.02 0.29 0.2 0.42]]



Observations from confusion matrix-

Confusion matrix of both models is matching same (density spreadness)-

Misclassified classes more often-

Among animals as mp 2,3,4(bird,cat,deer) ,as they have similar animal body structure,postures,etc.

After Visualize ,reporting 10 examples of mis-classified objects(yes as animals)-

For example mp 4-deer matches wrongly with bird-2,3-cat (easily seen from confusion matrix),

(d)

i)here the corresponding data to the cvals list-

Cvals-[1e-05, 0.001, 1, 5, 10]

test_acc=[0.2808, 0.2808, 0.617, 0.6244, 0.6222]

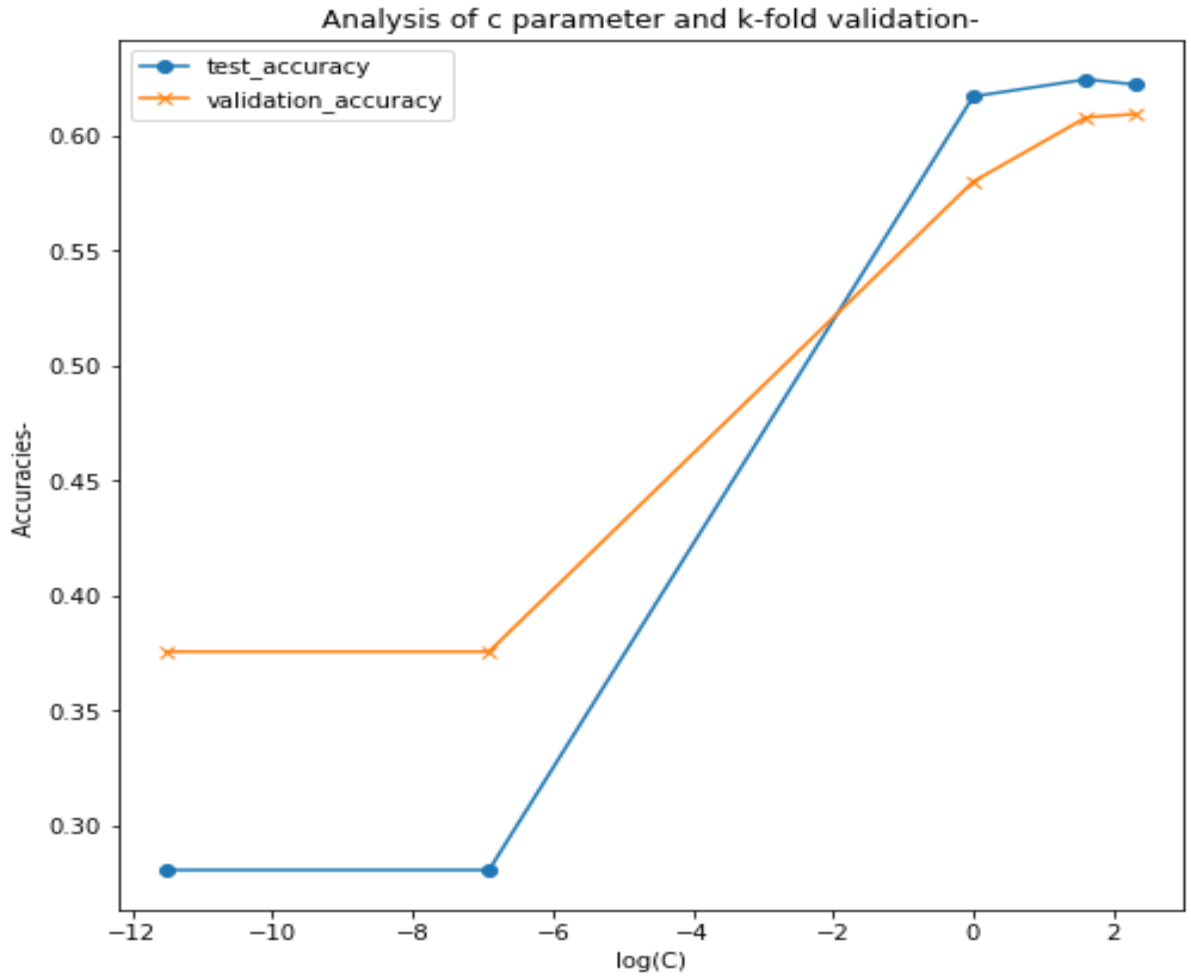
val_acc=[0.37570000000000003, 0.37570000000000003, 0.58, 0.6079, 0.6093]

ii)

10 give slightly best value of C (60.93 %).

No ,according to test accuracy 5 give slightly best value of C (62.44 %).

As test accuracy have more variance but ,k-fold have less variance due to averaging so we can easily see increasing trend in val_acc but some fluctuating in test_acc.



Q1 Text Classification

a)

i)

test accuracy-

80.14666666666666

Train accuracy -

93.06

Modal accuracies are around fine for Naïve Bays.

b)

test-

accuracy_r 50.03333333333333

accuracy_p 66.66666666666667

test accuracy org-

80.14666666666666

c)

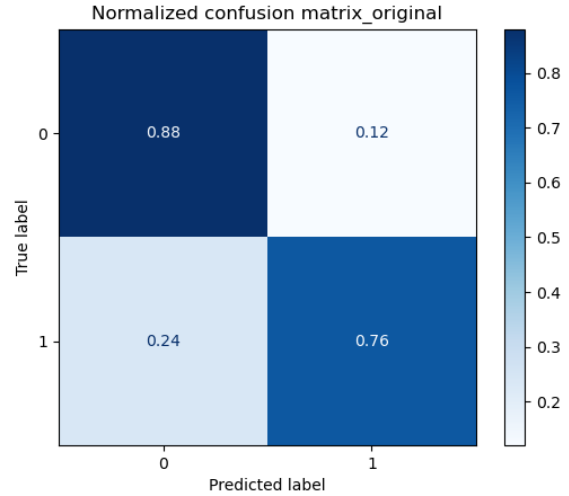
from below confusion matrices we can see our original model is best .

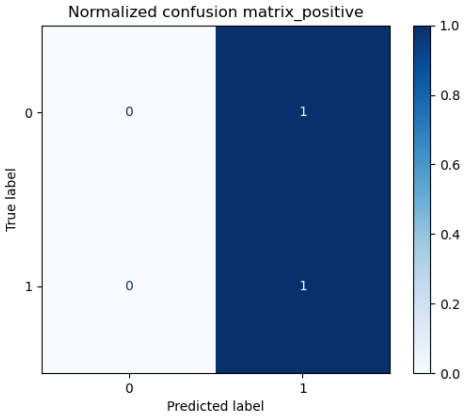
Also from above accuracies.

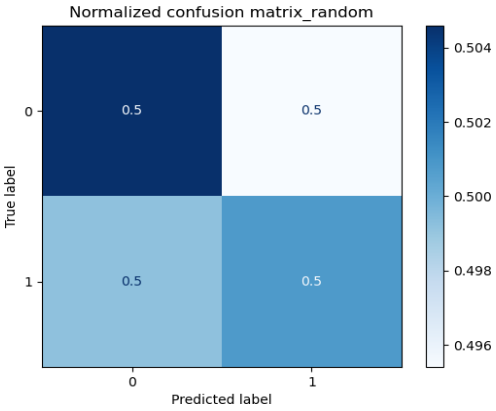
ii) original model have highest diagonal values.that means that it more accurately classifying each class to its correct label.

highest diagonal values->good model

iii)reasons can be explained with terms related to confusion matrix,difinition







d)

train-

50.096

test-

66.76

Word cloud are plotted.

Accuracies get lowered after stemming and stopword.

f) values printed along respective code files of models

after stemming and stopword-

	precision	recall	f1-score	support
0	0.79	0.00	0.01	5000
1	0.67	1.00	0.80	10000
accuracy			0.67	15000
macro avg	0.73	0.50	0.40	15000
weighted avg	0.71	0.67	0.54	15000

original model metrics-

	precision	recall	f1-score	support
0	0.65	0.88	0.75	5000
1	0.93	0.76	0.84	10000
accuracy			0.80	15000
macro avg	0.79	0.82	0.79	15000
weighted avg	0.83	0.80	0.81	15000

Best performing model is Biagram ,then original model.

As original model have 80%accuracy over 50%accuracy for the model(d)

Biagram model metrics-

	precision	recall	f1-score	support
0	0.66	0.91	0.77	5000
1	0.95	0.76	0.85	10000
accuracy			0.81	15000
macro avg	0.80	0.84	0.81	15000
weighted avg	0.85	0.81	0.82	15000

e)

train-

99.948

test-

81.36

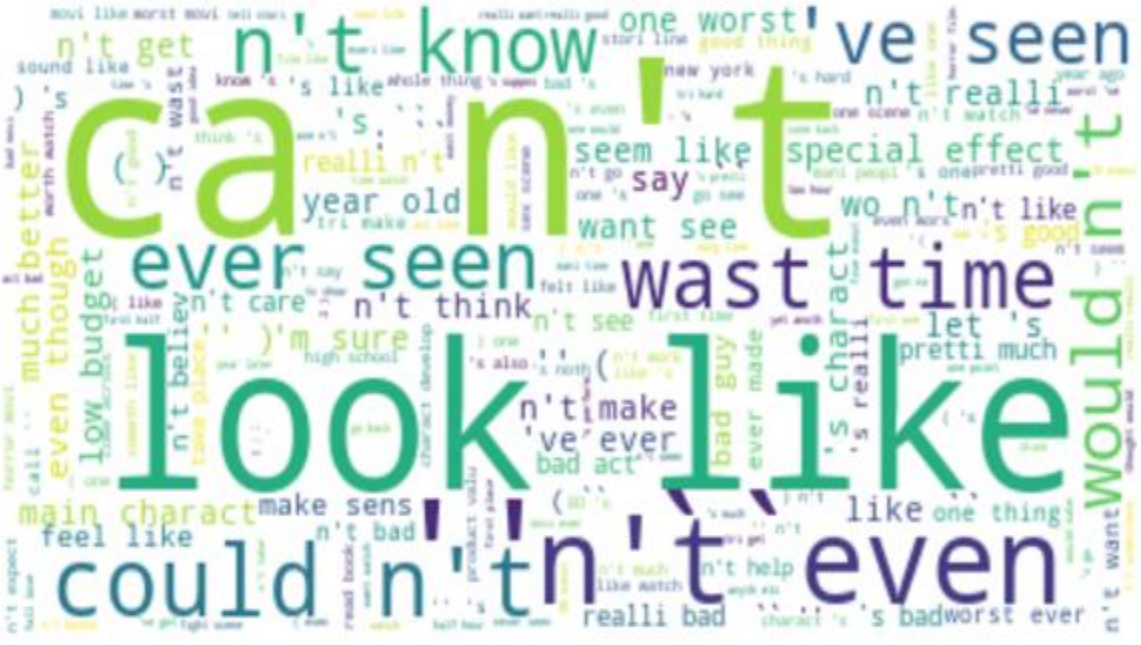
Test accuracy get slightly improved in Biagram but train accuracy improved mostly

Word cloud plotted in e part-

Word cloud plots after stemming and stopwords preprocessing-

Positive reviews-

Negative reviews-



Libraries used –

```
#libraries
import pandas as pnd
import numpy as nmp
from matplotlib import pyplot as pt
import cvxopt as cop
from time import time
from math import inf
import sys
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
q3-specific-
from sklearn.model_selection import cross_val_score
from sklearn.metrics import ConfusionMatrixDisplay
import pickle
q1-specific-
from collections import Counter
import re
import multidict as multidict
from wordcloud import WordCloud,STOPWORDS
import random
from sklearn.metrics import classification_report
```

q1-e specific-

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
# from nltk.corpus import stopwords
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from functools import lru_cache
```

Thank you!