COL 774 Assignment 2

Rishi Shah

2019CS10394

Q1. Text Classification

a) The model used for text classification was Naive Bayes bag of words model. Vocabulary of all the words in the model was created. Total words in the vocabulary are - . As per the model, parameters φ_x and φ_y were calculated. Then finally, for predictions the maximum probability class was considered. Logarithms were taken finally to avoid overflow.
Train Accuracy - 0.72086
Test Accuracy - 0.6656
Macro F1 Score - 0.1933

b) Randomly guessing any one label gives the accuray of 0.198 on test set. The maximum label is 5. And the accuray by giving the prediction as maximum labels comes out to be - 0.6608. Macro F1 score is 0.159. Thus our algorithm of naive bayes gives an improvement of 0.048 in accuracy and 0.034 in F1 Score.

c) Confusion Matrix :

	P	Labels				
A		1	2	3	4	5
Labels	1	2	0	0	21	205
	2	0	0	0	63	263
	3	2	0	1	231	852
	4	3	0	0	280	2825
	5	14	0	0	202	9036

The label 5 has the maximum diagonal entry.This means that maximum labels correctly labelled are 5. The algorithm predicts 5 in most cases as column 5 has large non-zero values, moreover it never predicts label 2. The prediction of 5 most time is justified as it occurs maximum number of times in the test data.

d) Model was trained after stop-word removal and stemming. NLTK library was used to do both the things.
Train Accuracy - 0.7243

Test Accuracy - 0.6665
Macro F1 Score - 0.2085
The model performs better than the original words model as F1 score and accuray both are higher. Hence stemming and stop words helps the model classify the reviews better.

e) Two features other than the word as features used are:
   i) Bi-Grams : Combining two consecutive words to make a new feature.
   For eg. a b c -> 'a b' , 'b c' as features. Two consecutive words form a small phrase in the sentence and gives more meaning to the features.
   Train Accuracy - 0.893
   Test Accuracy - 0.662
   Macro F1 Score - 0.169
   Thus the bi-grams model overfits the data(as train accuracy increase), which makes sense as the phrases can have high variance.
   ii) Lemmatizing the words. This was also done using the NTLK library. This changes the words as rocks-> rock, better -> good.
   Train Accuracy - 0.718
   Test Accuracy - 0.6655
   Macro F1 Score - 0.195

f) The F1 Score for the model in (d) is 0.2085. As the dataset is unbalanced , the F1 score shows that. However, the accuray is misleading as number of 5 labels dominate the test set. Hence F1 score is a better evaluation metric.

g) The words in the summary describe the whole review. So more weightage should be given to those words in review text which are present in summary test. The weightage given to those words are 10x. This number was decided by trial and error.
   Train Accuracy - 0.7208
   Test Accuracy - 0.6665
   Macro F1 Score - 0.1936
   This model performs slightly better than our original model as both test set accuracy and F1 score is higher.

Q2. MNIST Digit Classification

a) Binary Classification:
   i) According to the equation $\alpha^T P\alpha + q^T\alpha + c$, to express SVM dual objective. We compare it with the original equation :

   $min_\alpha 1/2 \sum_{i,j} y^i y^j \alpha_i \alpha_j x^{iT} x^j - \sum_i \alpha_i$, when $0 <= \alpha_i <= C$ and $\sum_i \alpha_i y^i = 0$.

$min_{\alpha} 1/2 x^T P x + q^T x$, $when\ Gx <= 0\ and\ Ax = b.$

We solve the equations using the CXOPT library and get the alphas. Then we use this alphas to calculate the w and b. And do the prediction using $w^T x + b$.

Train Accuracy - 1.0

Test Accuracy - 0.9866

ii) Here, we use the gaussian kernel instead of the linear kernel. Again we solve using CXOPT to get the alphas . Here to calculate w and b, we use the kernel formulas, described in the Andrew Ng notes.

Train Accuracy - 1.0

Test Accuracy - 0.9989

iii) Here , we train the SVM model using LIBSVM. Comparison of results :

CXOPT Linear :

Train Accuracy - 1.0

Test Accuracy - 0.9866

Running Time - 24.67 sec

SV - 150

CXOPT Gaussian :

Train Accuracy - 1.0

Test Accuracy - 0.9989

Running Time - 19.86 sec

SV - 1481

LIBSVM Linear :

Test Accuracy - 0.9866

Running Time - 1.476 sec

SV - 150

LIBSVM Gaussian :

Test Accuracy - 0.9989

Running Time - 7.1 sec

SV - 1459

Clearly, the using the LIBSVM library decreased the running time, however the accuracy is nearly equal . The reason is that matrix product computation in LIBSVM is more optized then our implementation of CXOPT.

b) Multi - Classification :

i) This is the kC2 model, where we use the implementation of CXOPT gaussian kernel. We do that for total of 45 times, and then finally get the results by maximum predictions.

Test Accuracy - 0.9722

Total Running Time - 17 min

Confusion Matrix -

	P	Labels									
A		0	1	2	3	4	5	6	7	8	9
Labels	0	969	0	1	0	0	3	4	1	2	0
	1	0	1122	3	2	0	2	2	0	3	1
	2	4	0	1000	4	2	0	1	6	15	0
	3	0	0	8	984	0	4	0	6	5	3
	4	0	0	4	0	962	0	6	0	2	8
	5	2	0	3	6	1	866	7	1	5	1
	6	6	3	0	0	4	4	939	0	2	0
	7	1	4	19	2	4	0	0	986	2	9
	8	4	0	3	10	3	5	1	3	942	3
	9	5	4	3	8	13	3	0	9	12	952

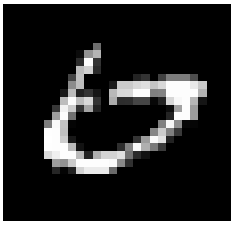
ii) Here, we do the multi-class prediction using LIBSVM.
Test Accuracy - 0.9724
Total Running Time - 7 min
Confusion Matrix -

	P	Labels									
A		0	1	2	3	4	5	6	7	8	9
Labe	0	969	0	1	0	0	3	4	1	2	0
	1	0	1121	3	2	1	2	2	0	3	1

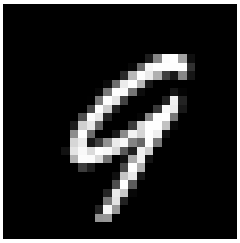
l s	2	4	0	1000	4	2	0	1	6	15	0
	3	0	0	8	985	0	4	0	6	5	2
	4	0	0	4	0	962	0	6	0	2	8
	5	2	0	3	6	1	866	7	1	5	1
	6	6	3	0	0	4	4	939	0	2	0
	7	1	4	19	2	4	0	0	986	2	9
	8	4	0	3	10	1	5	3	3	942	3
	9	4	4	3	8	13	4	0	7	12	954

iii) The miss-classification is mainly due to 7 and 2, as observed from the confusion matrix. Label 7 are missclassified as 2, just because there is a difference of single line in both the strokes of 7 and 2. Similarly, labels 0 and 6 are interchanged, as both have them have similar structure. Some examples are even difficult to read by human eye, so our model is performing close to human level. PIL was used to plot images from numpy array.
Eg :

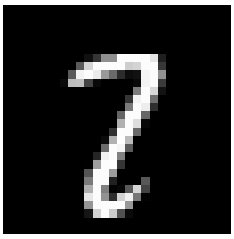
Predicted : 0 , Actual : 6



Predicted : 4, Actual : 9



Predicted 7, Actual 2



iv) The total time required for doing the 5-fold validation is around 2 hours . This is because we run the LIBSVM optimisation 5*5 times as for each ‘c’ , we run 5 times.

The best parameter ‘c’ observed was - 1. This is because for lower values of ‘c’, the model is underfitting as validation accuracy is 11.74%, and at higher values of ‘c’ the model is overfitting as difference between validation(97.3%) and test accuracy(97.1%) is increasing.