COL774 Assignment 2

Harshil Vagadia - 2019CS10356

1. Text Classification

A. Naive Bayes classification was implemented by using the bag of words model. Each word in the review text was encoded to a number. Thus, each review can be represented by an array of numbers. The total words in vocabulary were 490370. The parameters ϕ_x and ϕ_y were calculated. The probability of a test review being in each class was calculated, and the maximum probability was considered the class of the test review. Logarithms were used to avoid underflows.
The training accuracy was **0.72086** and the test accuracy was **0.66564**.

B. Randomly guessing a label from 1 to 5 gives an accuracy of **0.19746** on the test set while guessing the maximum occurring label gives an accuracy of **0.51864**. Thus, the naive bayes gave an improvement of 0.147 in accuracy.

C. Given below is the confusion matrix:
```
[   2.    0.    0.   21.  205.]
[   0.    0.    0.   63.  263.]
[   2.    0.    1.  231.  852.]
[   3.    0.    0.  280. 2825.]
[  14.    0.    0.  202. 9036.]
```
The rows represent actual labels, while the columns represent predicted labels. The maximum diagonal entry is for label 5. This means maximum reviews of label 5 are classified correctly as 5. Note that most of the non-zero values are concentrated in column 5, which means the algorithm predicts 5 in most of the cases. In fact, it never predicts 2. As 5 is the majority label in the dataset, this results in a larger accuracy, but it is not much better than max classifier in part B.

D. A new model was trained after removing stop words and performing stemming. This was done by using the NLTK library. The training accuracy was **0.72394** and the test accuracy was **0.66685** which is slightly greater than part A. Thus stemming and removing stop words helps the model better classify the reviews.

E. Here are two alternative features apart from the standard words as features:
Firstly, use a combination of three consecutive words as a single feature. Since, three words commonly form a phrase, this will help the model classify on the basis of certain phrases. The training accuracy came out to be **0.99596** while the test accuracy was **0.66457**. The large difference between training and test accuracies suggests that the model is overfitting. The test accuracy is also slightly lower than our baseline model (part A). Hence trigram is not a good feature for review prediction.
Second approach was to lemmatize all the words in the review text (along with stemming). Lemmatization groups various forms of inflected words like rocks -> rock,

been -> be, etc. The training accuracy was **0.72388** and the test accuracy was **0.66692**. This is again a slight improvement over part B.

F. The F1 score for the model described in part E2 is **[0.00873 0. 0. 0.15057 0.80552]** and the macro-F1 score is **0.19296**. As discussed earlier, the dataset is dominated by class 5, and hence is unbalanced. So here accuracy can be misleading, as a model predicting only 5 also performs well. The F1 score better captures this imbalance and hence is a better metric.

G. One option is to simply add the summary words to the review words. However,the summary field provides the crux of the review text and hence the words in summary should be given more weightage. I appended each word of summary k times into the review text, where k was chosen 5 by experimentation. This model gives a training accuracy of **0.6858** and test accuracy of **0.67092**. This model generalises well due to the small difference in training and test accuracy. It also has slightly better test accuracy than the model in part E2.

2. MNIST Digit Classification

A. Binary Classification

a. The SVM dual equation can be represented as:

$min_{\alpha}\, 1/2 \sum_{i,j=1}^{m} y^i y^j \alpha_i \alpha_j x^{iT} x^j \; - \; \sum_{i=1}^{m} \alpha_i$ subject to $C \geq \alpha_i \geq 0$ and $\sum_{i=1}^{m} \alpha_i y^i \; = \; 0$

This needs to be reported in the form:

$min_{\alpha} 1/2 \alpha^T P \alpha \; + \; q^T x$ subject to $Gx \; \leq \; 0$ and $Ax \; = \; b$

This formulation can be done by assigning the parameters as follows:

```
P = matrix(np.dot(y * X , (y * X).T), tc = 'd')
q = matrix(-np.ones((m, 1)), tc = 'd')
G = matrix(np.vstack((-np.eye(m), np.eye(m))), tc = 'd')
h = matrix(np.hstack((np.zeros(m), np.ones(m) * C)), tc = 'd')
A = matrix(y.T, tc = 'd')
b = matrix(np.zeros(1), tc = 'd')
```
for linear kernel

and

```
P = matrix(np.outer(y, y) * K, tc = 'd')
q = matrix(-np.ones((m, 1)), tc = 'd')
G = matrix(np.vstack((-np.eye(m), np.eye(m))), tc = 'd')
h = matrix(np.hstack((np.zeros(m), np.ones(m) * C)), tc = 'd')
A = matrix(y.T, tc = 'd')
b = matrix(np.zeros(1), tc = 'd')
```
for non-linear kernel

The alphas can be calculated using the CVXOPT library. The parameters w and b can be calculated from alphas. Finally the prediction is the sign of $w^T x \; + \; b$. Using this model, the training accuracy was **1.0** and test accuracy was **0.99395**.

b. In this part, the RBF kernel was used instead of the linear kernel. The alphas can be calculated using CVXOPT as described above. This alphas can be directly used to make predictions using the formulas described in the Sept 28 lecture. The training accuracy is **1.0** and test accuracy is **0.99949** which is slightly greater than using a linear kernel.

c. SVM models were trained using the LIBSVM model. The corresponding outputs for all for models are given below

LIBSVM Linear Kernel
nSV = 94
Accuracy = 0.99395
Time Taken: 1.12950 sec

LIBSVM RBF Kernel
nSV = 1094
Accuracy = 0.99949
Time Taken: 4.99658 sec

CVXOPT Linear Kernel
nSV: 94
Accuracy: 0.99395
Time Taken: 25.059758 sec

CVXOPT RBF Kernel
nSV: 1098
Accuracy: 0.99949
Time Taken: 28.45499 sec

The accuracies and number of support vectors are almost equal with both the implementations. Further, the value of w and b in case of linear kernel are also almost equal (using L2 distance). However, the LIBSVM implementation is much more efficient and takes much less time.

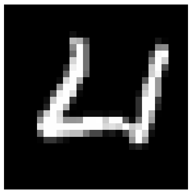
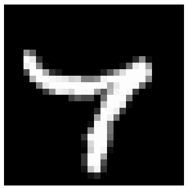
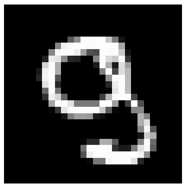
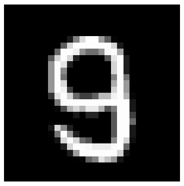
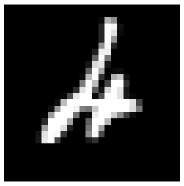
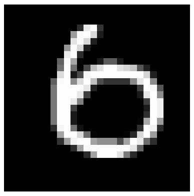
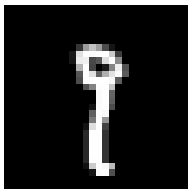
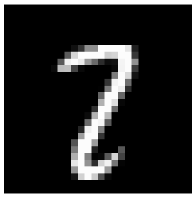
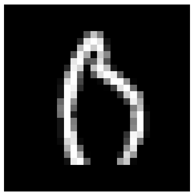
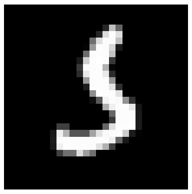
B. Multi-Class Classification

a. 45 binary classifiers (part B) were run on all possible pairings. The class predicted the most was taken as the final prediction. The test accuracy was **0.96709**. ~**35 min** were taken for complete execution. The confusion matrix is given below:
[969 0 2 0 0 3 4 1 1 0]
[0 1122 3 2 0 2 2 0 3 1]
[40 1002 4 2 0 1 7 12 0]
[0 0 9 991 0 1 0 6 1 2]
[0 0 4 0 962 0 6 0 2 8]
[2 0 4 31 1 842 7 1 3 1]

[6 3 0 0 4 4 939 0 2 0]
[1 4 19 3 4 0 0 987 0 9]
[4 0 0 11 41 4 5 1 3 902 3]
[6 4 5 12 13 3 0 8 4 954]

b. The entire dataset was trained with LIBSVM. The test accuracy was **0.97249** which is quite close to part A accuracy. The total time taken was ~**5 min**, which is much less than CVXOPT time. The confusion matrix is given below:
[969 0 1 0 0 0 3 4 1 2 0]
[0 1121 3 2 1 2 2 0 3 1]
[4 0 1000 4 2 0 1 6 15 0]
[0 0 8 985 0 4 0 6 5 2]
[0 0 4 0 962 0 6 0 2 8]
[2 0 3 6 1 866 7 1 5 1]
[6 3 0 0 4 4 939 0 2 0]
[1 4 19 2 4 0 0 986 2 9]
[4 0 3 10 1 5 3 3 942 3]
[4 4 3 8 13 4 0 7 12 954]

c. Looking at the non diagonal elements of the above confusion matrix, we find that the most misclassification occurs on 7 while is wrongly classified to 2. This makes sense as both are only differentiated by a small horizontal line. Given below are some examples of misclassified images:


6

4

8

8

2

0

7

7

5

3

The misclassified images are somewhat unconventional written digits, and some of them are even hard for humans to classify. For example, the last third digit looks much like 7, which is predicted by our model.