**Assignment 2 Machine Learning COL774**

* **How to run run.sh file?**

**Q1🡺 run.sh 1 <training file path> <testing file path> {a/b/d/e/f/g}**

**Q2🡺 run.sh 2 <training file path> <test file path> 0/1 {a/b/c} / {a/b/c/d}**

**Q1.**

Implemented Naïve Bayes model for the text classification (Sentiment Classification) of the movie reviews based on the ‘Review’ given by the user, which is in text format and the metric to analyse how the movie is, given by the ‘Rating’ (out of 5).

1. Used Naïve Bayes model

P(Y|X)=P(X|Y)P(Y)/P(X)

Here,

Y is number of classes i.e the ratings 1 to 5

X is the Sentence/Review (text)

So, to solve this we will use Bag of Words model.

In bag of words model, we consider the whole sentence as just bunch of words in such a manner that the position of any word does not matter or affect the outcome of its rating.

Each review X will become bunch of words,

X={x1,x2,x3,………………….xn}

So, P(Y=1|x1,x2,x3,………….xn)=P(x1,x2,x3…………..,xn|Y) P(Y)/P(x1,x2………xn)

In naïve bayes we assume the P(X) are independent so which makes,

P(Y=1|x1,x2………..xn)=P(x1|Y)P(x2|Y)……P(xn|Y)P(Y) +1(for Laplace smoothing)

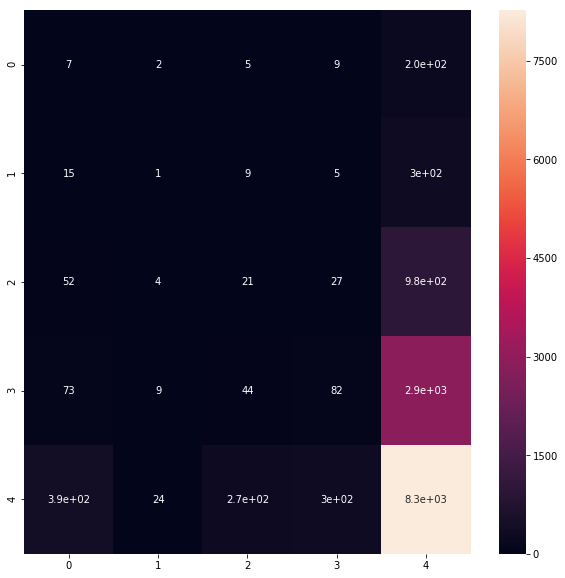
Here we get accuracy as **59.84%**

1. By randomly guessing any of the class we get the test set accuracy of 1/5 = 0.2 =**20%**

By always guessing the class which occurs most frequently we get for class 5 which is 25932/50109 = 0.5175 = **51.75%**

the algorithm implemented in part a) gives **39.84%** more accuracy as compared to random baseline and **8.09%** more accuracy as compared to majority baseline

c)



Here the element 4x4 has higher concentration of predictions, which means that most of the classes which the model predicted were belonging to class 5.

1. Here we have pre-processed the data by removing the ‘Stopwords’, special characters also.

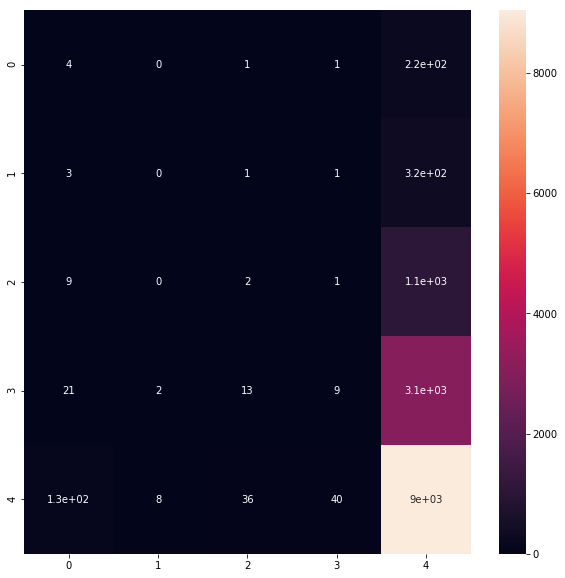
Also, we have done lemmetization of the words, e.g. ‘games’ 🡺’game’.

Using this pre-processed vocabulary we will estimate the NB on the same data set.

(Instead of stemming I have used Lemmetization here, Reason because using stemming my accuracy was decreasing very sharply)

The accuracy we get here is **64.66%.**

Accuracy has increased as compared to the part ‘i’ of the same question.

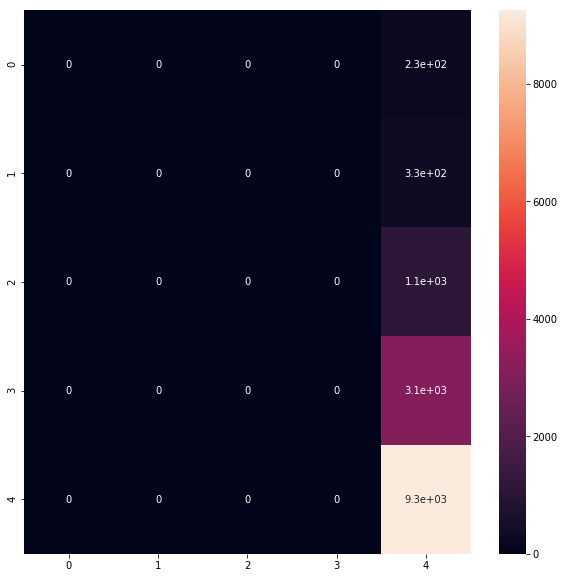


1. Feature engineering is very helpful when we want to improve the accuracy and get the exact content of the model. In our problem we are taking into account only single word for the prediction/classification of the

Text. If we go for Bigrams, Trigrams or even further to N-grams, we actually get the context of the sentence which helps model to improve it’s accuracy in prediction the correct output class.

Here we are getting Accuracy as **66.08%** by implementing Bi-gram model.

1. From the above models implemented the model from part ‘e’ is giving the best accuracy.



The above confusion matrix gives us the Recall, Precision of each of the classes.

Recall = TP/(TP+FN)

Precision = TP/(TP+FP)

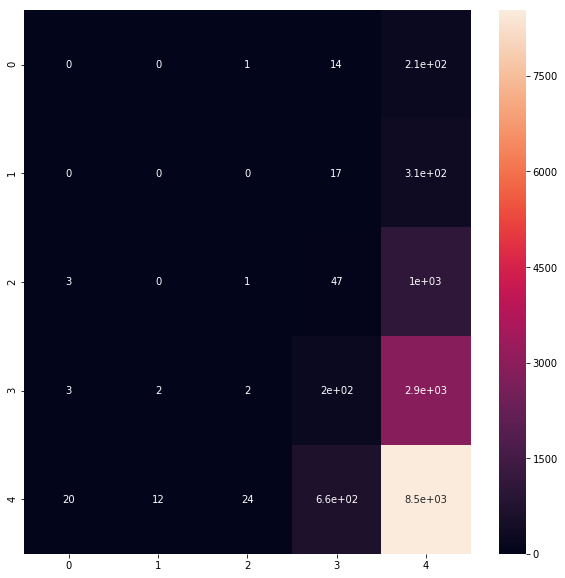
F1 measure is given as:

F1 = 2\*Recall\*Precision /( Recall + Precision )

F1 for given CM is:

|  |  |
| --- | --- |
| Class 1 | **0** |
| Class 2 | **0** |
| Class 3 | **0** |
| Class 4 | **0** |
| Class 5 | **1** |

Macro F1 is **0.2**



Here we are calculation the sentiment from ‘summary’ field along with the reviews,

And the combined NB probability is used to predict the class.

Accuracy we get here is **62.37%**

Q2.

1. **Binary Classification**

In this question we will implement a classifier based on the MNIST data set, which is a number (from 0 to 9) image dataset. We will use SVM to implement it. Especially in this part a we will be implementing binary classifier to classify just two numbers.

Here we are using last digit of our Entry number as one digit ‘d’ and other number (d+1)mod10

So, for me d=8, since my entry number is 2021SIY7558

So, d1 = (8+1) mod 10=9

1. In this subpart we are using CVXOPT library.

We will be expressing the SVM dual problem in terms of the library’s Solver which is ‘Quadratic Programming’ solver. Which takes input as Q, p, A, b, G, h as per

Max 0.5\*xTPx +qTx

Subject to , Gx<=h and Ax=b

So, as per our dual problem of SVM(note here x is ‘alpha’)

P =sum(xiyi)(xjyj)

q= column vector of -1 values

A = output Y matrix (y train converted to matrix)

b= zero

G=Identity matrices vertically stacked

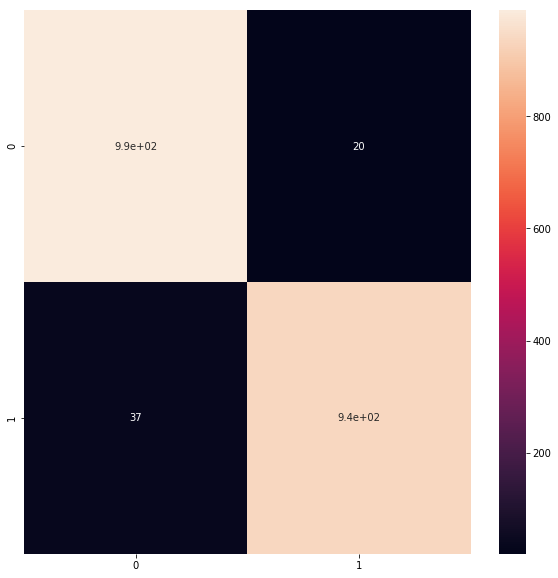
h=matrix holds the penalty for the soft svms

now we put these computed matrices to solver.qp() and we get the solution.

Accuracy is **97.12**

b= **1.057**

number of linear support vectors = **218**



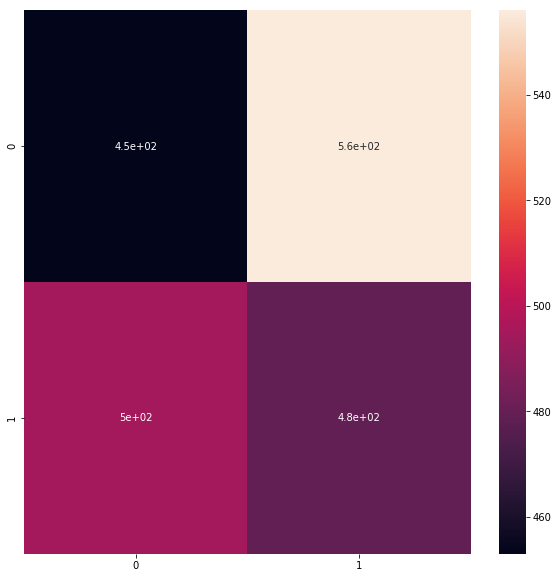
1. In this subpart we will use the gaussian kernel to compute the same as subpart i.

Number of gaussian support vectors = **1509**

b = **0**

accuracy =**49.99**

**Here, FP are more**

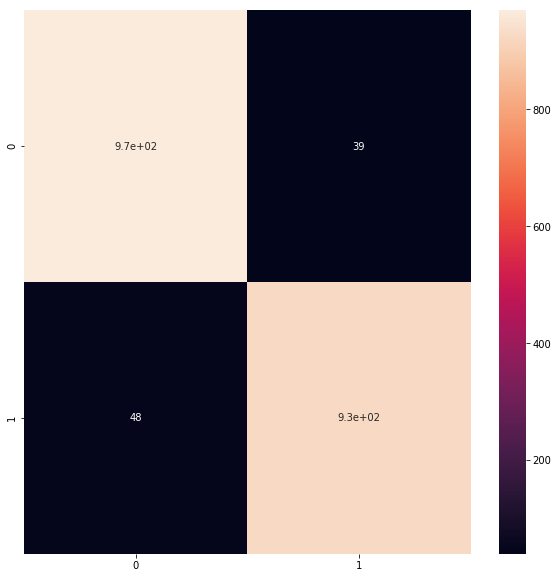


1. In this subpart we will use the different library LIBSVM and implement both the above subparts which implements the binary classification with Linear and Gaussian Kernals Respectively.
   1. For linear we will use '-t 0 -c 1' setting that is C-SVC type of SVM, Linear kernal and C =1.0

Accuracy is **95.61%**

Number of support vectors are **217**

**CM:**

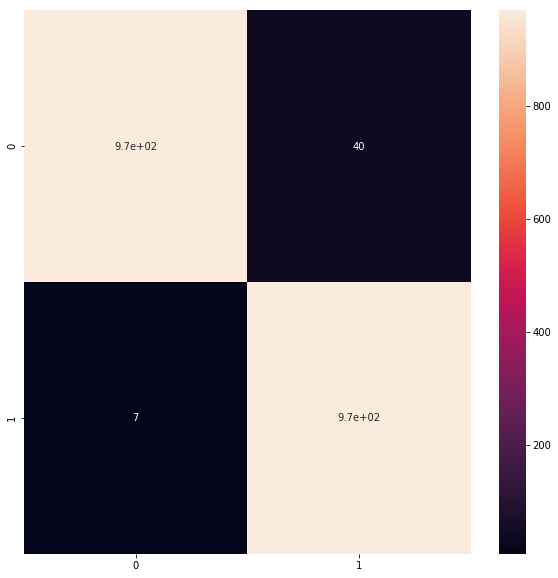


* 1. And For gaussian we will use '-t 2 -g 0.05 -c 1.0' setting that is C-SVC type of SVM, RBF kernel , C=1.0 and Gamma = 0.05.

Accuracy is **97.62%**

Number of Support Vectors are **1463**

**CM:**



For calculating the Gaussian kernel time taken is so much more as compared to linear kernel.

Also, performance of LIBSVM is faster and Accuracy is greater than CVXOPT.

1. **Multiclass Classification**

In this problem we will use same MNIST dataset and use it to do Multiclass classification.

Which is to classify the images of number in 10 different categories.

1. We will use the same CVXOPT library and implement the One vs One method to do the multiclass classification.

In One vs One multiclass classification,

We take all the pairs of the classes and train the separate classifier for it.

Here we have k=10 classes, so we need to find KC2 = 10C2 = 45 different classifiers.

Kernel used here is Gaussian kernel.

So, during the prediction time we run our image of number through all the classifiers and assign the output class which is being predicted by the most number of classifiers.

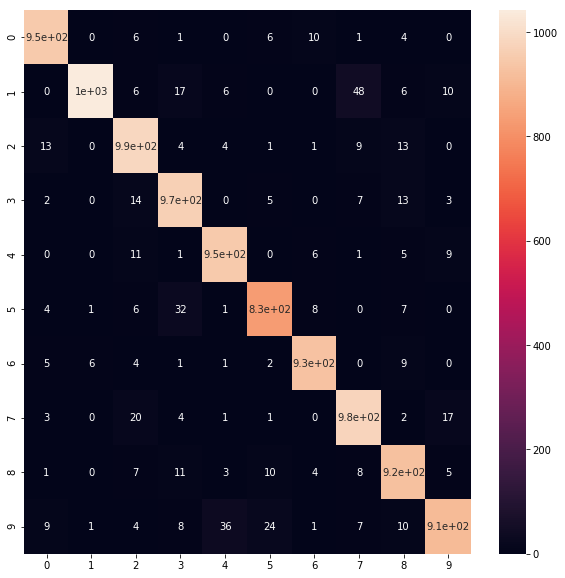
1. In this subpart we will use the LIBSVM to do the multiclass classification of the whole data set in 10 different classes.

For this we will use '-t 2 -g 0.05 -c 1.0' setting in which we use C-SVC type of SVM, Gaussian kernel, gamma = 0.05 and C= 1.0.

The accuracy we get is **94.72%**

The number of Support vectors here we get are **10492i**

1. CM:



Observations:

class 1 are the most miss-classified as the class 7, which is 48 miss classifications

'2' is miss-classified as '0' 13 time and as a '8' also 13 times

'9' is miss-classified as '4' 36, as '5' 24 and as '8' 10 times

'5' is miss-classified as '3' 32, as '6' 8 and as '8' 7 times

'8' is miss-classified as '3' 11, as '5' 10 and as '2' 7 times

yes, result does make sense from our observation of similarities between the shapes of the numbers.

1. In this subpart we will use LIBSVM to do the K fold cross validation.

In this we will train our model with k-1 parts of the training data and then dev test on the kth remaining part.

Here we will use '-t 2 -v 5 -g 0.05 -c ' this setting where we use 5 fold cross validation, which means we will train on 4 parts of training data and test on 1/5th part of it, We use the Gaussian kernel and gamma 0.05 also C will be varying for each iteration as per [1e-5, 1e-3, 1, 5, 10].

Cross validation accuracy we got is **67.59%**