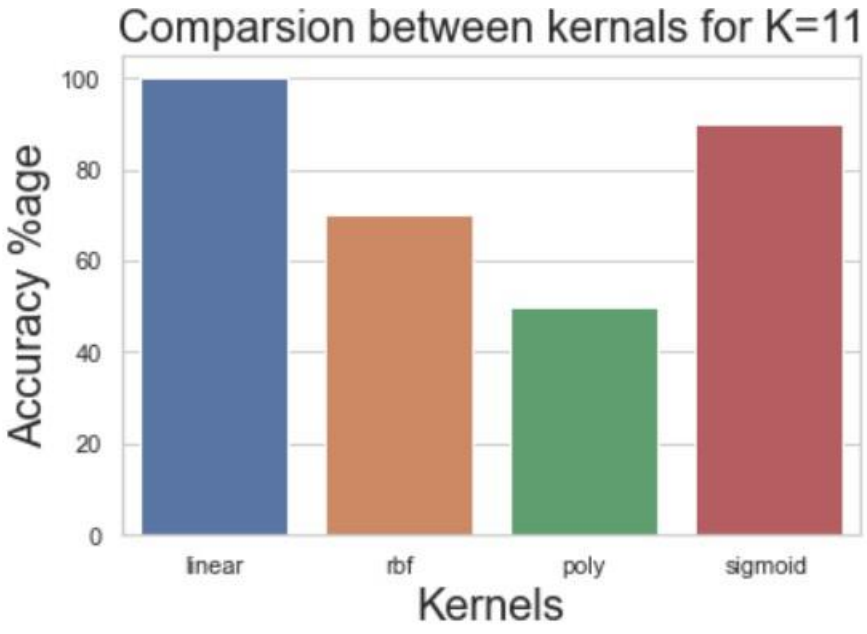
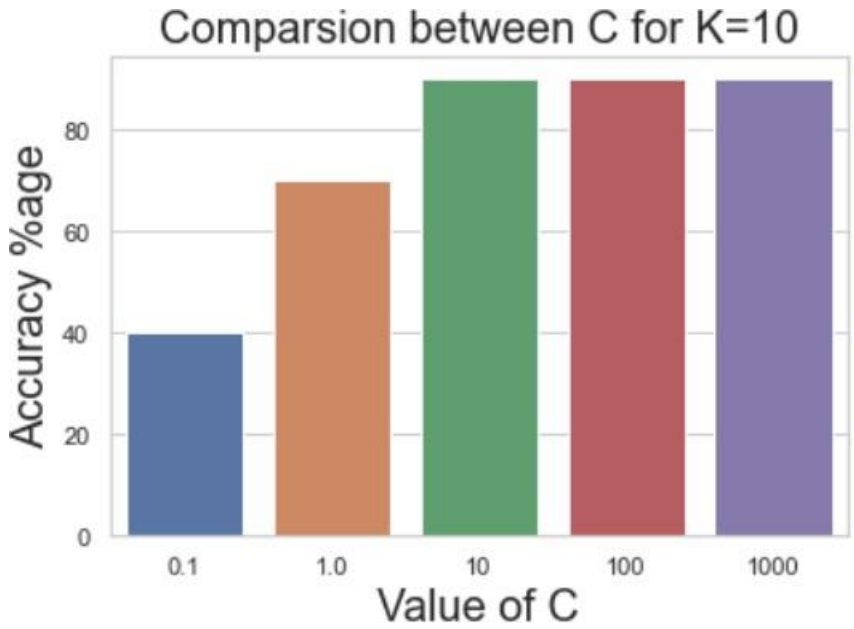
REPORT

Problem1- i) I have taken the value K=11 to compare them between the different kernel performance



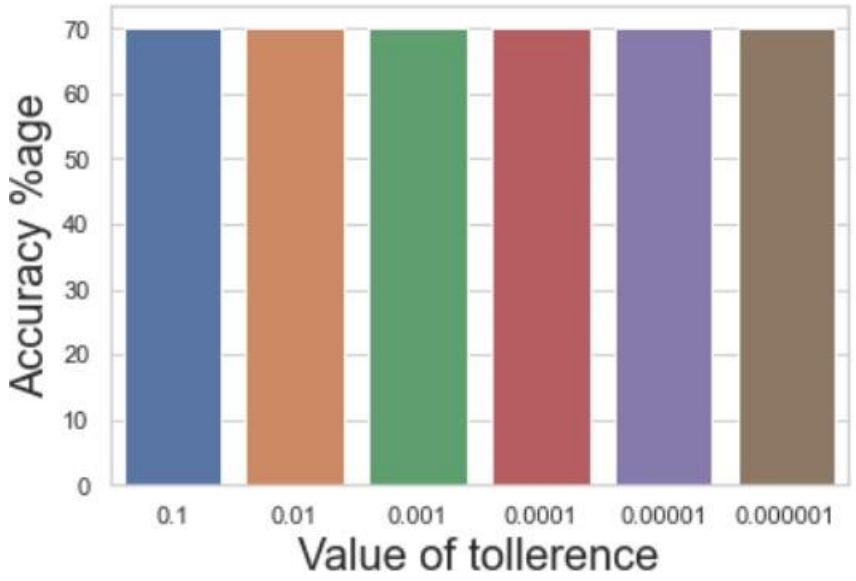
I got the best performance while using the linear function followed by the sigmoid and rbf but the polynomial function performed poorly.

To get the trends for different value of C , I have considered K=10



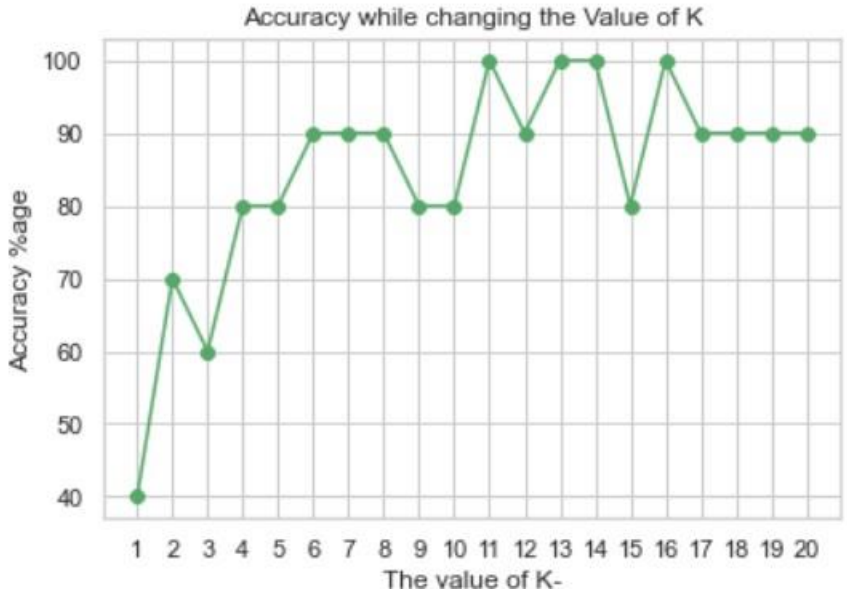
Increasing the value of C resulting in the higher accuracy , and it was obvious that what we discussed in class as increasing the C value fits the data more accurately but grater value of C may also results in overfitting.

Changing the value of Si(tollerence) didn't effect the performance much,still I have taken the value of K=12 to show the trends.



In most of the values of K it didn't affecting the accuracy.

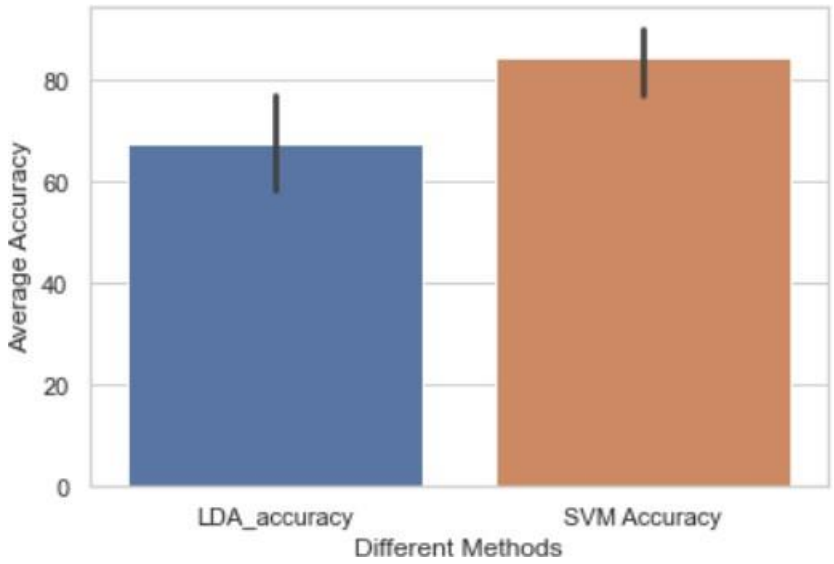
Now for showing the trend in value of K , I have taken the maximum accuracy in each K from 1 to 20

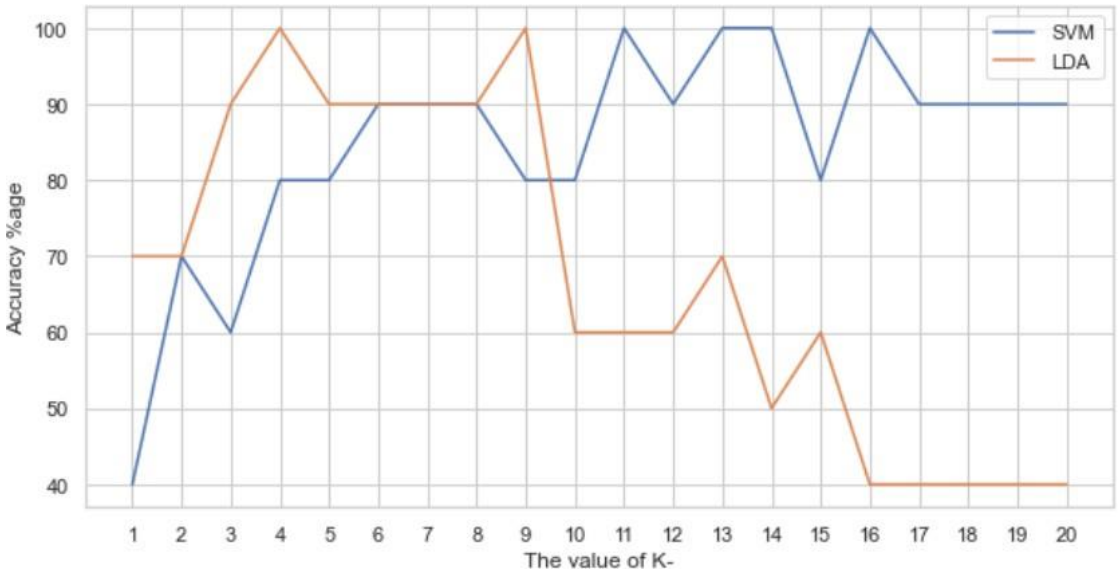


The accuracy is higher for higher value of K as compared to the lower values as for lower values we are just taking the few higher eigen values but as in the other case we are taking more eigen values but still we have seen the outlier in K=15.

ii:-

I have used my own code that I made in assignment 1 to compare between the LDA and SVM the results that I got are as below:





As we can see from the above plot that the average accuracy for LDA case is giving lower accuracy as compared to the SVM and from the second plot we can conclude LDA is performing better for lower values of K but it is performing poorly for the higher values of K as compared to the SVM. Mainly the difference between LDA and SVM is that SVM forces the points that are difficult to classify and LDA treats all the data points equal.

Problem2:-

Procedure-a.) First I splitted the data into training and test data as per the given question in the ratio of 75% to 25% by reading the first 750 documents into the different array and rest into the different.

b.) Then by the help of list I have collected the unique words in my whole 1000 documents as discussed in the class so that the test data wont experienced any new word , I made it by the help of an array of those unique words.

c) Now I calculated the TF of each documents of the training data and normalize it and also separated the labels of the training data in an array.

d) Than I calculated the IDF of those unique word by the help of training data.

e) Than I multiplied the both to get the TF-IDF of the training Data

f) Than I applied PCA on them to reduce them into the 10 dimension.

Test Data-g) I have calculated the TF of test data by the help of each of the documents in it and also separated the labels of the test data in an array.

h)For the IDF part I have used the IDF array of the train data and then multiplied both to get the TF-IDF of the test data.

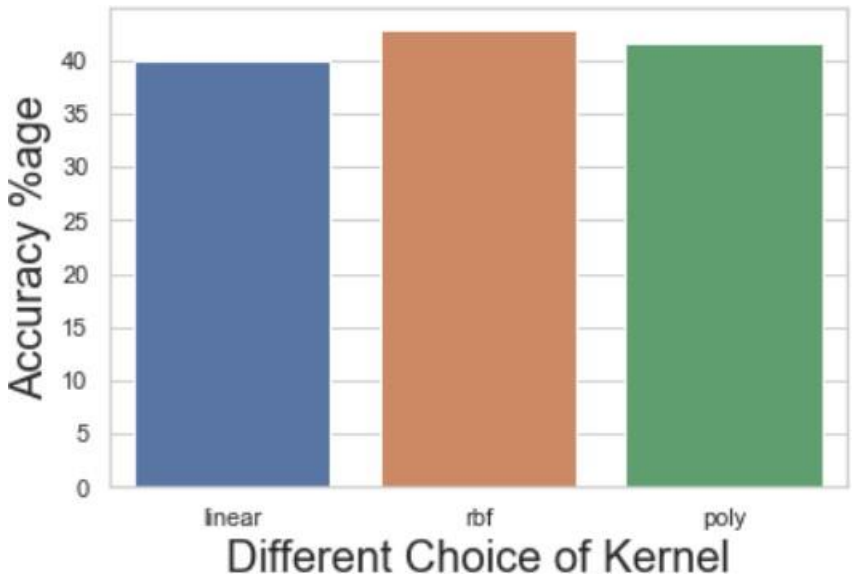
I)Now I have used the PCA of the train data to get the PCA of the test data.

j) Now I train the SVM model by the help of training data and its corresponding labels

k)Now I predicted the model by the help of test data by selecting the different kernel functions as said in the question.

l) Finally I calculated the accuracy score by the help of prediction and the true label of the test data.

OBSERVATION-




```
The accuracy score for linear as kernal function is  0.4
The value of Support vector for this case is 673
The accuracy score for rbf as kernal function is  0.428
The value of Support vector for this case is 640
The accuracy score for poly as kernal function is  0.416
The value of Support vector for this case is 662
```

In our given data set the rbf kernel performing best among the others and then the polynomial kernel and at the last linear kernel performs poorly. We get the more number of support vectors for which the accuracy is less and we are getting the high number of support vector since the Accuracy is not good.

Result- Data points are not seperable by the SVM model since we are getting poor accuracy so I think it may get the best accuracy if we apply ANN model in it.

Note- I got better accuracy while using the inbuilt library function for TF-IDF but as per their documentation what we have taught in class was a bit different so I stick with my own code.