LAB 10 REPORT

[redacted]

Q1)

For the first question, first I pre-processed the data. There were no NaN values and all of the features were continuous, so there was no need of encoding. Then I used train test split to split the data in a 70:30 ratio of train to test. There were 57 features in total in the dataset and 0,1 are the two class labels.
I imported SVC (Support Vector Classifier) from to use SVM.sklearn.svm. I created three models using the following kernels: linear kernel, quadratic kernel, andKernel with RBF.
So, by using the Linear kernel, I fitted the train data and predicted test data labels using default values of parameter C(=1). When it came to accuracy on test data, it scored 92 percent. When the value of C is changed from 1 to 10, the accuracy drops from 92 percent to 91.1 percent. I only used 10 values in the linear kernel because it was taking so long to run. As a result, I took less C values, and accuracy improved. As a result, C = 0.4 was determined to be the best value.
I chose kernel = poly and degree = 2 for the Quadratic kernel, and the default values of parameter C are always 1 while the default values of degree are 3. The train data was fitted, and the test data labels were predicted. When testing accuracy on test data, it came up with a score of 66.9%. The accuracy of C increases from 66.9% to 70.6 percent when the value of C is changed from 1 to 100. I used 100 values in the quadratic kernel because it was taking a long time to run. And it appears that the accuracy increases as the value of C increases.
I forecasted test data labels using the RBF kernel. When it came to accuracy on test data, it came up with a score of 70.8 percent. When the value of C is changed from 1 to 100, the accuracy rises from 70.8 percent to 90.3 percent. I used 1000 values in the RBF kernel because it was quick to execute. And it appears that the accuracy increases as the value of C increases.