Report on coding problems

4a. No of samples: 0 no of support vectors: (782,) accuracy score: 0.9834905660377359

4b.

No of samples: 50

no of support vectors: (6,)

accuracy score: 0.9811320754716981

no of support vectors: (20,)

accuracy score:0.9811320754716981

No of samples: 200

no of support vectors: (60,)

accuracyscore:0.9811320754716981

No of samples: 800

no of support vectors: (320,)

accuracy score:0.9811320754716981

4c.

1. False

Q= 2 C= 0.0001 Training error: 0.029468289557975647

Q= 5 C= 0.001 Training error: 0.00512491992312624

2. True

Q= 2 C= 0.001 Support vectors: (190,190)

Q= 5 C= 0.001 Support vectors: (32,32)

3. False

Q= 2 C= 0.01 Training error: 0.004484304932735439

Q= 5 C= 0.01 Training error: 0.004484304932735439

4. True

Q= 2 C= 1 test error: 0.018867924528301883

Q= 5 C= 1 test error: 0.01650943396226412

Increasing C results in increased penalty coming up with a maximum margin as a bonus. On the other hand, if C is small that say's that we are allowing smaller margin than maximum margin ensuring penalties become too much.

5. Results of all errors:

C value: 0.01 train error: 0.00832799487508007 Test error: 0.04481132075471694

C value: 1 train error: 0.004484304932735439 Test error: 0.021226415094339646

C value: 100 train error: 0.004484304932735439 Test error: 0.021226415094339646

C value: 10000 train error: 0.004484304932735439 Test_error:0.018867924528301883

C value: 1000000 train error: 0.004484304932735439 Test error: 0.021226415094339646

Test error was low when C = 10000, while training error is same& low when the C values are 1,100,10000,1000000

5.a *Linear Kernel*:

Train error=0, Test error=0.024

No of support vectors= 1084

5.b *Polynomial kernel*:

Train error=0.0005, Test error=0.02

No of support vectors= 1332

Gaussian kernel:

Train error=0, Test error=0.5

No of support vectors= 6000

Training error is 0 for Gaussian kernel