Cmpe 462 Machine Learning
Project 2 Report

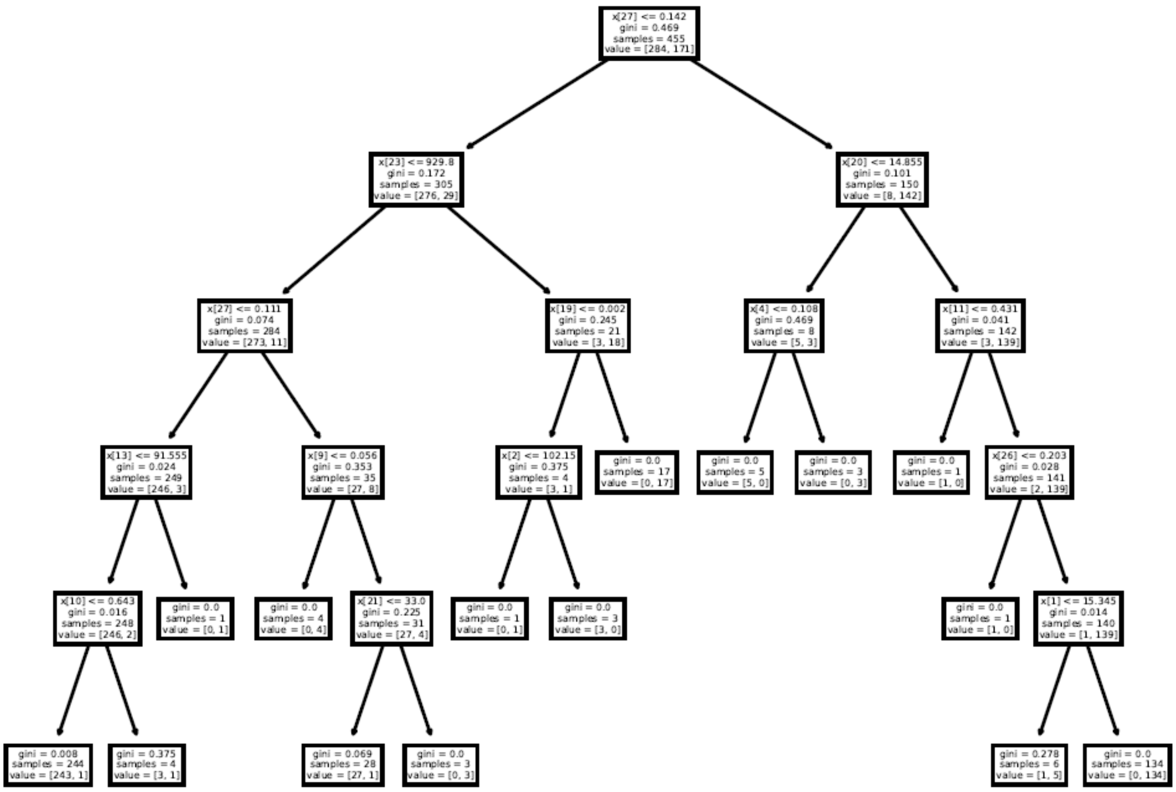
2020400147 – Muhammet Ali Topcu
2019400309 – Halil İbrahim Gürbüz
2021400171 – Mücahit Erdoğan Ünlü

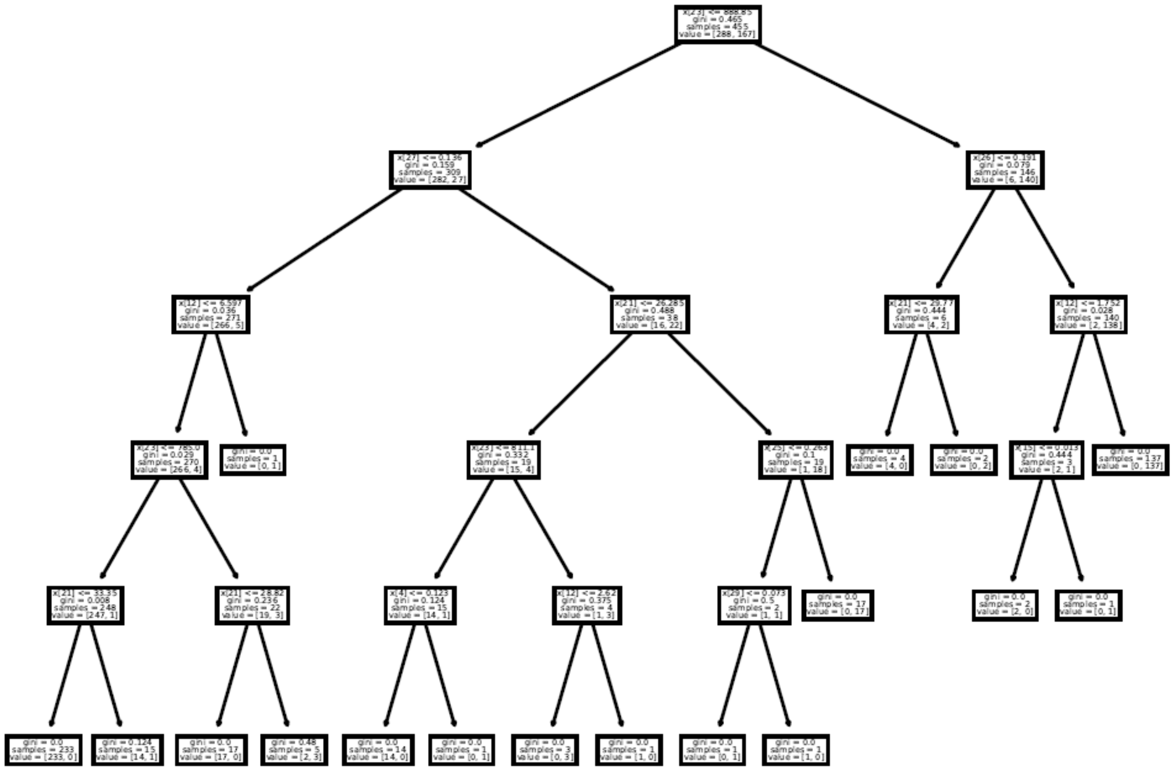
Link to Our Code

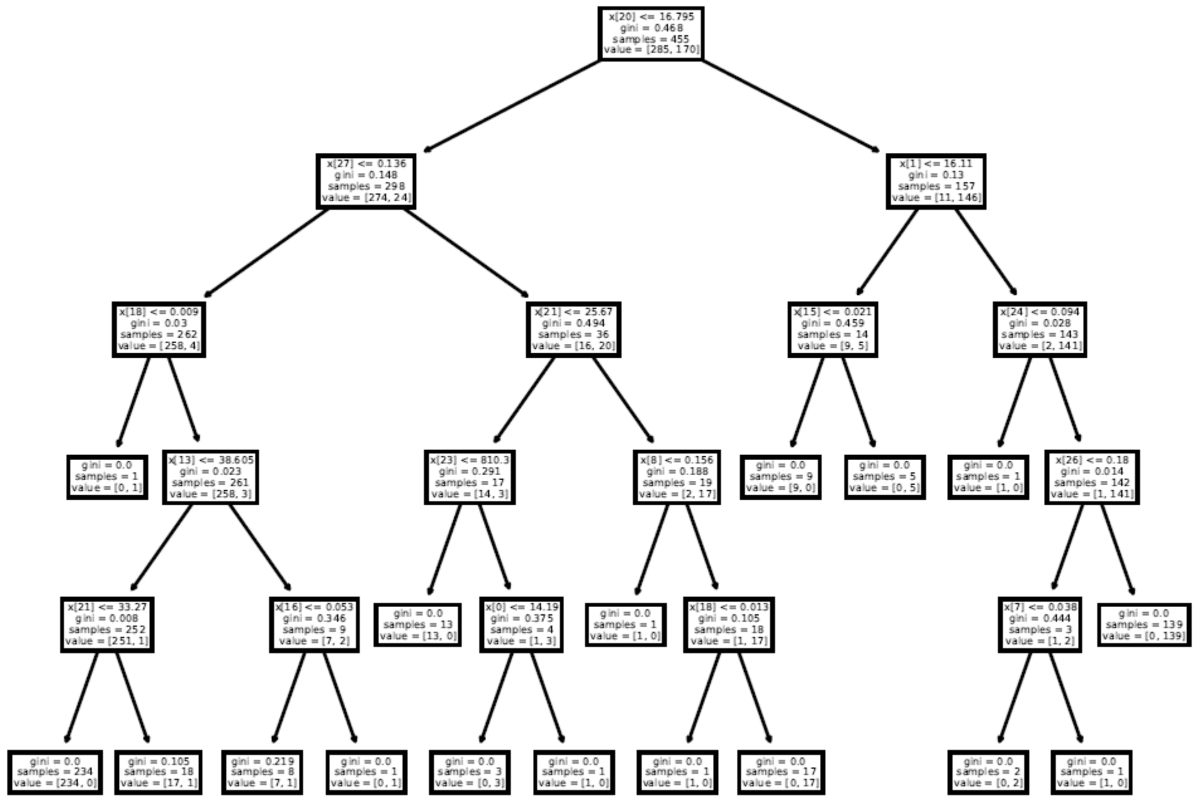
https://github.com/alitpc25/cmpe462projects/tree/main/HW2

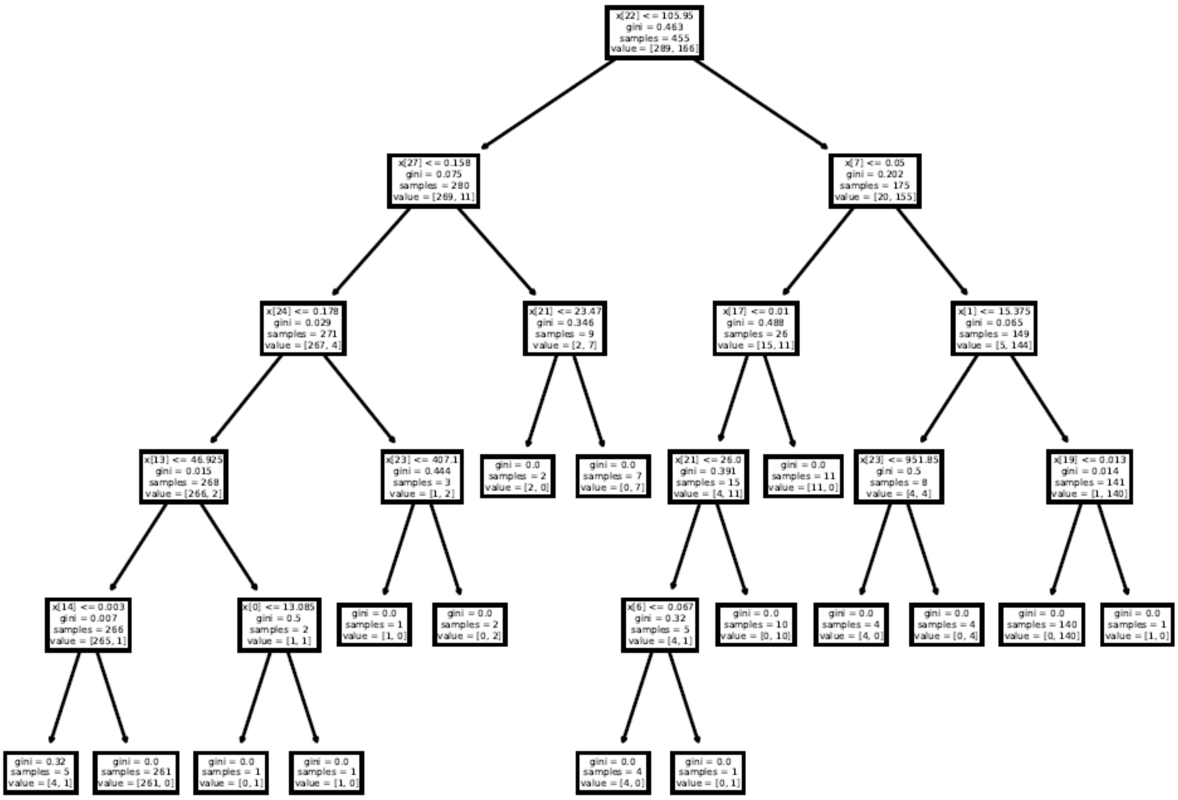
PART 1: Decision Trees

In the first part of the project, we trained decision trees using scikit-learn's function. Maximum depth of the tree is set to 5. Below are the images for decision trees created with 5-fold cross validation.

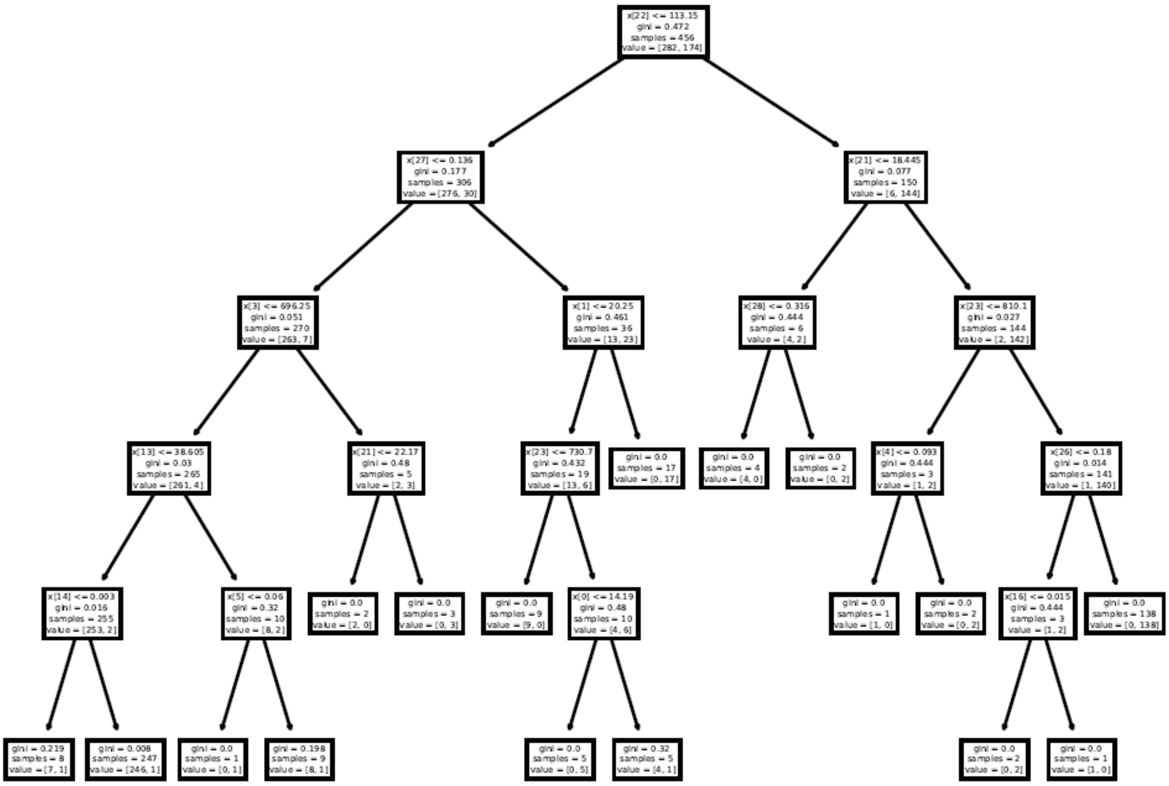








x[22] <= 105.95
gini = 0.463
samples = 455
value = [289, 166]
x[27] <= 0.158
gini = 0.075
samples = 280
value = [269, 11]
x[7] <= 0.05
gini = 0.202
samples = 175
value = [20, 155]
x[24] <= 0.178
gini = 0.029
samples = 271
value = [267, 4]
x[21] <= 23.47
gini = 0.346
samples = 9
value = [2, 7]
x[17] <= 0.01
gini = 0.488
samples = 26
value = [15, 11]
x[1] <= 15.375
gini = 0.065
samples = 149
value = [5, 144]
x[13] <= 46.925
gini = 0.015
samples = 268
value = [266, 2]
x[23] <= 407.1
gini = 0.444
samples = 3
value = [1, 2]
gini = 0.0
samples = 2
value = [2, 0]
gini = 0.0
samples = 7
value = [0, 7]
x[21] <= 26.0
gini = 0.391
samples = 15
value = [4, 11]
gini = 0.0
samples = 11
value = [11, 0]
x[23] <= 951.85
gini = 0.5
samples = 8
value = [4, 4]
x[19] <= 0.013
gini = 0.014
samples = 141
value = [1, 140]
x[14] <= 0.003
gini = 0.007
samples = 266
value = [265, 1]
x[0] <= 13.085
gini = 0.5
samples = 2
value = [1, 1]
gini = 0.0
samples = 1
value = [1, 0]
gini = 0.0
samples = 2
value = [0, 2]
x[6] <= 0.067
gini = 0.32
samples = 5
value = [4, 1]
gini = 0.0
samples = 10
value = [0, 10]
gini = 0.0
samples = 4
value = [4, 0]
gini = 0.0
samples = 4
value = [0, 4]
gini = 0.0
samples = 140
value = [0, 140]
gini = 0.0
samples = 1
value = [1, 0]
gini = 0.32
samples = 5
value = [4, 1]
gini = 0.0
samples = 261
value = [261, 0]
gini = 0.0
samples = 1
value = [0, 1]
gini = 0.0
samples = 1
value = [1, 0]
gini = 0.0
samples = 4
value = [4, 0]
gini = 0.0
samples = 1
value = [0, 1]



x[22] <= 113.15
gini = 0.472
samples = 456
value = [282, 174]
x[27] <= 0.136
gini = 0.177
samples = 306
value = [276, 30]
x[21] <= 18.445
gini = 0.077
samples = 150
value = [6, 144]
x[3] <= 696.25
gini = 0.051
samples = 270
value = [263, 7]
x[1] <= 20.25
gini = 0.461
samples = 36
value = [13, 23]
x[28] <= 0.316
gini = 0.444
samples = 6
value = [4, 2]
x[23] <= 810.1
gini = 0.027
samples = 144
value = [2, 142]
x[13] <= 38.605
gini = 0.03
samples = 265
value = [261, 4]
x[21] <= 22.17
gini = 0.48
samples = 5
value = [2, 3]
x[23] <= 730.7
gini = 0.432
samples = 19
value = [13, 6]
gini = 0.0
samples = 17
value = [0, 17]
gini = 0.0
samples = 4
value = [4, 0]
gini = 0.0
samples = 2
value = [0, 2]
x[4] <= 0.093
gini = 0.444
samples = 3
value = [1, 2]
x[26] <= 0.18
gini = 0.014
samples = 141
value = [1, 140]
x[14] <= 0.003
gini = 0.016
samples = 255
value = [253, 2]
x[5] <= 0.06
gini = 0.32
samples = 10
value = [8, 2]
gini = 0.0
samples = 2
value = [2, 0]
gini = 0.0
samples = 3
value = [0, 3]
gini = 0.0
samples = 9
value = [9, 0]
x[0] <= 14.19
gini = 0.48
samples = 10
value = [4, 6]
gini = 0.0
samples = 1
value = [1, 0]
gini = 0.0
samples = 2
value = [0, 2]
x[16] <= 0.015
gini = 0.444
samples = 3
value = [1, 2]
gini = 0.0
samples = 138
value = [0, 138]
gini = 0.219
samples = 8
value = [7, 1]
gini = 0.008
samples = 247
value = [246, 1]
gini = 0.0
samples = 1
value = [0, 1]
gini = 0.198
samples = 9
value = [8, 1]
gini = 0.0
samples = 5
value = [0, 5]
gini = 0.32
samples = 5
value = [4, 1]
gini = 0.0
samples = 2
value = [0, 2]
gini = 0.0
samples = 1
value = [1, 0]

In the first project, our naive bayes models accuracies were as following:

naive bayes test accuracy: 0.9314392175128084
naive bayes training accuracy: 0.939365721997301

Decision tree's accuracies are as following:

Average train accuracy: 0.9938500096394833
Average test accuracy: 0.9279614966620089

As it could be seen, training accuracy of the decision tree is higher than naive bayes, but test accuracy is lower. It could be argued that decision tree is overfitted.

Below are the features sorted based on their importances:

	importance
24	0.291229
29	0.214623
25	0.186607
22	0.147115
23	0.043244
3	0.021535
9	0.013766
28	0.007876
17	0.007429
6	0.006718
19	0.006523
14	0.005835
2	0.005464
21	0.005428
11	0.005209
26	0.004335
15	0.004308
20	0.003665
5	0.003227
18	0.002582
30	0.002563
13	0.001870
10	0.001615
8	0.001529
4	0.001454
7	0.001367
31	0.000966

27   0.000864
16   0.000593
12   0.000461

And followings are the perceptron accuracies for the selected features:
Perceptron accuracy on top 5 features:
0.852336593696631
Perceptron accuracy on top 10 features:
0.836655798789008
Perceptron accuracy on top 15 features:
0.8488278217668064
Perceptron accuracy on top 20 features:
0.7858407079646017

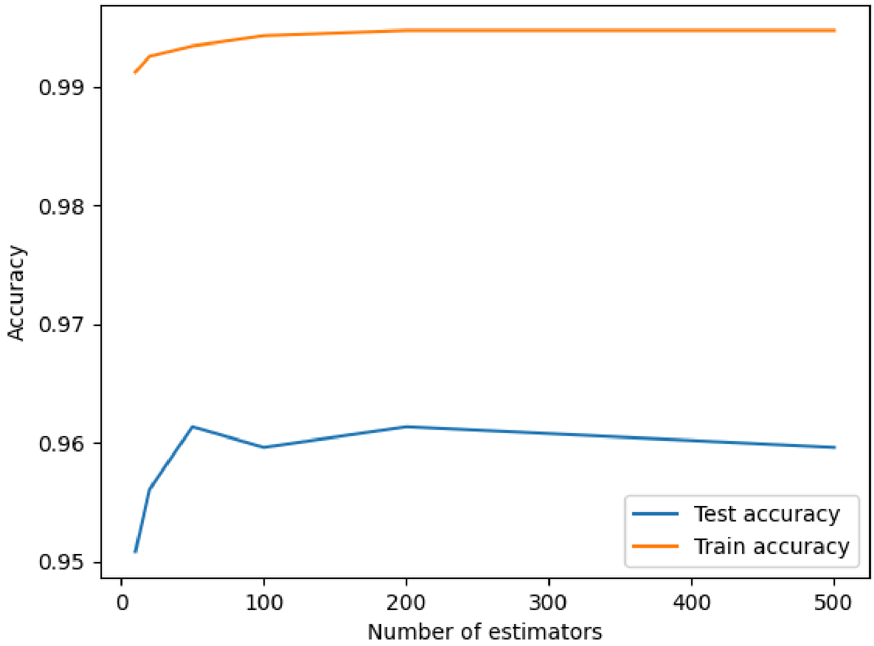
Feature selection using decision tree gives advantage to perceptron since the features are selected based on their importance and perceptron gives weights to the features based on their importance and uses them to classify.

In general, fewer features give higher accuracies. This may be because fewer features can reduce overfitting and improve generalization to unseen data. But, on the other hand, it may be expected that more features can capture more complex relationships in the data, potentially increasing accuracy. However, our observations show that fewer features give higher accuracies.

Random forests test accuracy with 10 estimators:
0.9508150908244062
Random forests train accuracy with 10 estimators:
0.9912126470021206
Random forests test accuracy with 20 estimators:
0.9560627231796305
Random forests train accuracy with 20 estimators:
0.9925322922691343
Random forests test accuracy with 50 estimators:
0.9613414066138798
Random forests train accuracy with 50 estimators:
0.9934114131482552
Random forests test accuracy with 100 estimators:
0.9595870206489675
Random forests train accuracy with 100 estimators:
0.9942886061307113
Random forests test accuracy with 200 estimators:

0.9613414066138798
Random forests train accuracy with 200 estimators:
0.9947281665702719
Random forests test accuracy with 500 estimators:
0.9595870206489675
Random forests train accuracy with 500 estimators:
0.9947281665702719

Below is the plot for random forests test and training accuracies:



In general, increasing the number of estimators increases training accuracy. In terms of test accuracy, it increases to some extent as the number of estimators increases. However, after some point, test accuracy decreases slightly. This may be again because of the overfitting.

PART 2: Support Vector Machines

1.(a) In this part, primal formulation of SVM is implemented from scratch and solved using a quadratic programming solver. Below are the matrices and vectors fed to the solver, and u solution vector returned from the solver:

u: u is the solution vector of the optimization problem.
u[0:d] = w
u[d] = b
u[d+1:] = epsilon (margin violations)

Q: Q is the matrix representing the quadratic coefficients in the objective function. It is created as a sparse matrix so that it keeps less memory space since it contains mostly 0 values. Q is a matrix of size (N+d+1, N+d+1). First d+1 diagonal elements are equal to 1.

p: p is the vector representing the linear coefficients in the objective function. It is a vector of size (N+d+1) with p[d+1:] = C, and all others are set to 0.

A: A is the matrix representing the coefficients of the inequality constraints. It is a matrix of size (2*N, N+d+1). It is designed as following:
First N rows (r), first d columns (c) = -y[r]*x_train[r, c]
First N rows (r), dth column (c) = -y[r]
First N rows (r), last N columns (c) = -(identity matrix of size N)
Second N rows (r), d columns (c) = 0 matrix
Second N rows (r), dth column (c) = 0 vector
Second N rows (r), last N columns (c) = -(identity matrix of size N)

c: c is the vector representing the right-hand side of the inequality constraints. It has size 2*N with c[:N] = -1 and all others set to 0.

In addition, to convert the problem into a one-vs-all problem, y values are changed so that the cells that has the target are set to 1, and others set to -1.

Output of this part are as following:

```
     pcost       dcost       gap    pres   dres
 0:  3.4597e+00  2.2759e+03  2e+05  3e+00  2e+05
 1:  6.7398e+00 -1.0036e+04  1e+04  1e-01  9e+03
 2:  5.9669e+00 -5.6845e+02  6e+02  8e-03  5e+02
 3:  5.6047e+00 -3.5793e+01  4e+01  5e-04  3e+01
 4:  4.2226e+00 -2.4493e+00  7e+00  5e-05  3e+00
 5:  1.2597e+00  3.5318e-01  9e-01  3e-06  2e-01
 6:  8.2235e-01  4.9818e-01  3e-01  7e-07  4e-02
 7:  7.1848e-01  5.4734e-01  2e-01  3e-07  2e-02
 8:  6.6502e-01  5.7392e-01  9e-02  1e-07  8e-03
 9:  6.3778e-01  5.8847e-01  5e-02  7e-08  4e-03
10:  6.2564e-01  5.9525e-01  3e-02  4e-08  2e-03
```

```
11:  6.1490e-01  6.0166e-01  1e-02  1e-08  8e-04
12:  6.1155e-01  6.0368e-01  8e-03  7e-09  4e-04
13:  6.0935e-01  6.0513e-01  4e-03  3e-09  2e-04
14:  6.0799e-01  6.0608e-01  2e-03  1e-09  7e-05
15:  6.0725e-01  6.0663e-01  6e-04  3e-10  2e-05
16:  6.0694e-01  6.0688e-01  5e-05  3e-12  2e-07
17:  6.0691e-01  6.0691e-01  2e-06  9e-14  6e-09
18:  6.0691e-01  6.0691e-01  5e-08  3e-15  2e-10
Optimal solution found.

Training accuracy for class 2 : 0.9527816149692327
Test accuracy for class 2 : 0.9552795031055901

     pcost       dcost       gap    pres   dres
 0:  2.9123e+00  2.8700e+03  3e+05  4e+00  2e+05
 1:  7.3053e+00 -1.0019e+04  1e+04  1e-01  7e+03
 2:  6.6855e+00 -3.6635e+02  4e+02  4e-03  3e+02
 3:  6.1951e+00 -1.5955e+01  2e+01  2e-04  1e+01
 4:  3.6203e+00 -3.2887e-01  4e+00  1e-05  9e-01
 5:  1.1832e+00  4.9107e-01  7e-01  3e-06  1e-01
 6:  8.9380e-01  6.1380e-01  3e-01  7e-07  4e-02
 7:  8.0057e-01  6.5873e-01  1e-01  3e-07  2e-02
 8:  7.6200e-01  6.7837e-01  8e-02  1e-07  8e-03
 9:  7.3226e-01  6.9447e-01  4e-02  5e-08  3e-03
10:  7.1889e-01  7.0231e-01  2e-02  2e-08  1e-03
11:  7.1280e-01  7.0616e-01  7e-03  7e-09  4e-04
12:  7.1048e-01  7.0770e-01  3e-03  2e-09  1e-04
13:  7.0934e-01  7.0852e-01  8e-04  6e-10  3e-05
14:  7.0894e-01  7.0882e-01  1e-04  4e-11  2e-06
15:  7.0888e-01  7.0886e-01  2e-05  6e-12  3e-07
16:  7.0887e-01  7.0887e-01  8e-07  2e-13  9e-09
17:  7.0887e-01  7.0887e-01  3e-08  5e-15  3e-10
Optimal solution found.

Training accuracy for class 3 : 0.9368747122106409
Test accuracy for class 3 : 0.9460869565217391

     pcost       dcost       gap    pres   dres
 0:  3.6419e+00  2.9473e+03  3e+05  3e+00  2e+05
 1:  7.4640e+00 -1.4351e+04  2e+04  2e-01  1e+04
 2:  6.8563e+00 -5.3862e+02  5e+02  7e-03  4e+02
 3:  6.1745e+00 -3.7236e+01  4e+01  5e-04  3e+01
```

```
 4:  4.7909e+00 -1.4944e+00  6e+00  3e-05  2e+00
 5:  1.5346e+00  6.5946e-01  9e-01  3e-06  2e-01
 6:  1.1468e+00  8.0492e-01  3e-01  1e-06  5e-02
 7:  1.0501e+00  8.5257e-01  2e-01  4e-07  2e-02
 8:  9.7625e-01  8.8891e-01  9e-02  2e-07  9e-03
 9:  9.4299e-01  9.0733e-01  4e-02  6e-08  3e-03
10:  9.3139e-01  9.1396e-01  2e-02  2e-08  1e-03
11:  9.2501e-01  9.1827e-01  7e-03  6e-09  3e-04
12:  9.2290e-01  9.1976e-01  3e-03  2e-09  1e-04
13:  9.2153e-01  9.2079e-01  7e-04  4e-10  2e-05
14:  9.2122e-01  9.2104e-01  2e-04  8e-11  4e-06
15:  9.2114e-01  9.2112e-01  2e-05  5e-12  3e-07
16:  9.2113e-01  9.2113e-01  8e-07  2e-13  1e-08
Optimal solution found.

Training accuracy for class 8 : 0.9127213361798318
Test accuracy for class 8 : 0.9122981366459627

     pcost       dcost       gap    pres   dres
 0:  3.6446e+00  1.9853e+03  3e+05  4e+00  2e+05
 1:  7.7695e+00 -1.3431e+04  1e+04  2e-01  1e+04
 2:  6.9928e+00 -5.6036e+02  6e+02  6e-03  4e+02
 3:  6.4555e+00 -2.5196e+01  3e+01  3e-04  2e+01
 4:  4.2873e+00 -1.6418e+00  6e+00  3e-05  2e+00
 5:  1.3405e+00  2.3894e-01  1e+00  5e-06  3e-01
 6:  7.6943e-01  4.3233e-01  3e-01  9e-07  6e-02
 7:  7.0087e-01  4.6751e-01  2e-01  5e-07  3e-02
 8:  6.2969e-01  5.0134e-01  1e-01  3e-07  2e-02
 9:  5.9783e-01  5.1756e-01  8e-02  1e-07  9e-03
10:  5.7132e-01  5.3181e-01  4e-02  6e-08  4e-03
11:  5.6089e-01  5.3766e-01  2e-02  3e-08  2e-03
12:  5.5405e-01  5.4175e-01  1e-02  1e-08  1e-03
13:  5.4986e-01  5.4433e-01  6e-03  5e-09  3e-04
14:  5.4834e-01  5.4534e-01  3e-03  2e-09  2e-04
15:  5.4703e-01  5.4630e-01  7e-04  5e-10  3e-05
16:  5.4667e-01  5.4657e-01  1e-04  1e-15  3e-14
17:  5.4662e-01  5.4662e-01  7e-06  1e-15  4e-14
18:  5.4662e-01  5.4662e-01  3e-07  1e-15  3e-13
Optimal solution found.

Training accuracy for class 9 : 0.9695257231361715
Test accuracy for class 9 : 0.9714285714285714
```

Overall test accuracy: [0.9552795031055901, 0.9460869565217391, 0.9122981366459627, 0.9714285714285714]
Overall training accuracy: [0.9527816149692327, 0.9368747122106409, 0.9127213361798318, 0.9695257231361715]
Average test accuracy: 0.946273291925465825
Average training accuracy: 0.942975846623969225
Time: 40577.62305521965 seconds

1.(b) This part involves the training a 4-class SVM using the scikit-learn's soft margin primal SVM function with linear kernel. Outputs of this part are as following:

Test accuracy with C = 0.0001 : 0.9207453416149068
Time: 209.34830045700073 seconds
Training accuracy with C = 0.0001 : 0.9128117457763475

Test accuracy with C = 0.01 : 0.9590062111801242
Time: 47.410584449768066 seconds
Training accuracy with C = 0.01 : 0.9598250201126307

Test accuracy with C = 0.1 : 0.9587577639751553
Time: 37.700422048568726 seconds
Training accuracy with C = 0.1 : 0.9691271118262269

Test accuracy with C = 10.0 : 0.9448447204968944
Time: 83.30861759185791 seconds
Training accuracy with C = 10.0 : 0.9838093322606597

As C increases, we care more about violating the margin, which gets us closer to the hard-margin SVM and our test & training accuracies increases in general. Choosing C = 0.1 may be the best option since it avoids a decision boundary that overfits the training noise and allows small violation of the margins or even some classification errors.

1.(c) In this part, dual formulation of non-linear SVM with Gaussian RBF kernel is implemented from scratch and solved using a quadratic programming solver. Below are the matrices and vectors fed to the solver, and u(alpha) vector returned from the solver:

u: u is the vector that maximizes the lagrangian (which is returned from the qpsolver). w and b then could be found using u vector.

Q: Q is the matrix representing the quadratic coefficients in the objective function. It is a symmetric matrix of size (N, N) with elements Q(r,c) = y[r]*y[c]*rbf_kernel(X[r], X[c], gamma) where y = y_train, X = x_train.

p: p is the vector representing the linear coefficients in the objective function. It is a vector of size (N) with all elements set to -1.

A: A is the matrix representing the coefficients of the inequality constraints. It is a sparse matrix of size (N, N) with diagonal elements set to -1 (-(identity matrix)).

c: c is the vector representing the right-hand side of the inequality constraints. It has size N with all elements set to 0.

G: G is the matrix representing the coefficients of the equality constraints. It is a sparse matrix of size (1,N) where G[0, c] = y[c], c for column.

b_vec: b_vec is the vector representing the right-hand side of the equality constraints. It is in essence a vector of size 1, which is equal to 0.

After feeding the qpsolver and getting u, w and b could be figured out using the necessary formulas.

Output of this part is as following:

```
Running part 1c
     pcost       dcost       gap    pres   dres
 0: -8.8412e+02 -2.4963e+03  2e+03  2e-12  1e+00
 1: -8.9825e+02 -9.2376e+02  3e+01  2e-13  5e-02
 2: -8.9972e+02 -9.0002e+02  3e-01  4e-14  6e-04
 3: -8.9972e+02 -8.9973e+02  3e-03  1e-13  6e-06
 4: -8.9972e+02 -8.9972e+02  3e-05  2e-13  6e-08
Optimal solution found.
Training accuracy for class 2 : 0.7493966210780371
Test accuracy for class 2 : 0.7436024844720497
     pcost       dcost       gap    pres   dres
 0: -7.9100e+02 -2.2174e+03  1e+03  1e-12  1e+00
 1: -8.0311e+02 -8.2684e+02  2e+01  1e-13  5e-02
 2: -8.0446e+02 -8.0473e+02  3e-01  2e-14  5e-04
 3: -8.0446e+02 -8.0447e+02  3e-03  3e-14  5e-06
 4: -8.0446e+02 -8.0446e+02  3e-05  2e-14  5e-08
Optimal solution found.
Training accuracy for class 3 : 0.7433125502815768
```

Test accuracy for class 3 : 0.7490683229813665

```
     pcost       dcost       gap    pres   dres
 0: -1.4520e+02 -4.3149e+02  3e+02  5e-12  1e+00
 1: -1.4528e+02 -1.4907e+02  4e+00  9e-14  2e-02
 2: -1.4530e+02 -1.4536e+02  6e-02  4e-15  3e-04
 3: -1.4530e+02 -1.4530e+02  6e-04  6e-15  3e-06
 4: -1.4530e+02 -1.4530e+02  6e-06  3e-14  3e-08
```

Optimal solution found.
Training accuracy for class 8 : 0.7567880128720836
Test accuracy for class 8 : 0.7580124223602485

```
     pcost       dcost       gap    pres   dres
 0: -1.1862e+02 -3.5312e+02  2e+02  8e-12  1e+00
 1: -1.1865e+02 -1.2165e+02  3e+00  1e-13  2e-02
 2: -1.1867e+02 -1.1872e+02  5e-02  2e-15  3e-04
 3: -1.1867e+02 -1.1867e+02  5e-04  4e-15  3e-06
 4: -1.1867e+02 -1.1867e+02  5e-06  1e-14  3e-08
```

Optimal solution found.
Training accuracy for class 9 : 0.7505028157683025
Test accuracy for class 9 : 0.7493167701863354
Overall test accuracy: [0.7436024844720497, 0.7490683229813665, 0.7580124223602485, 0.7493167701863354]
Overall training accuracy: [0.7493966210780371, 0.7433125502815768, 0.7567880128720836, 0.7505028157683025]
Time: 23146.245436906815 seconds

1.(d) This part involves the training a 4-class SVM using the scikit-learn's soft margin dual SVM function with polynomial kernel. Outputs of this part are as following:

Time: 430.56470489501953 seconds
Test accuracy with gamma = 0.001 : 0.6444720496894409
Training accuracy with gamma = 0.001 : 0.6357099758648431

Time: 54.64802694320679 seconds
Test accuracy with gamma = 0.01 : 0.9858385093167702
Training accuracy with gamma = 0.01 : 0.9904465004022526

Time: 30.954280376434326 seconds
Test accuracy with gamma = 0.1 : 0.9878260869565217
Training accuracy with gamma = 0.1 : 1.0

As gamma increases, test & training accuracies both increases.

2. This part involves feature extraction using PCA. PCA chosen since it is a feature reduction technique that uses linear combinations of all original features. PCA retains most of the sample's variances.

2. Part 1.(a)

```
Number of components: 10
     pcost       dcost       gap    pres   dres
 0:  7.3587e+01  7.8922e+03  1e+05  2e+00  1e+04
 1:  4.1693e+02 -1.0871e+03  2e+03  3e-02  1e+02
 2:  2.8428e+02  8.5602e+01  2e+02  5e-04  3e+00
 3:  1.0279e+02  1.0009e+02  3e+00  4e-06  2e-02
 4:  1.0021e+02  1.0018e+02  3e-02  4e-08  2e-04
 5:  1.0018e+02  1.0018e+02  3e-04  4e-10  2e-06
 6:  1.0018e+02  1.0018e+02  3e-06  4e-12  2e-08
Optimal solution found.
Training accuracy for class 2 : 0.7493966210780371
Test accuracy for class 2 : 0.7436024844720497
     pcost       dcost       gap    pres   dres
 0:  7.4785e+01  8.0114e+03  1e+05  2e+00  1e+04
 1:  4.1697e+02 -1.0699e+03  2e+03  3e-02  1e+02
 2:  2.8510e+02  8.9248e+01  2e+02  5e-04  2e+00
 3:  1.0503e+02  1.0251e+02  3e+00  4e-06  2e-02
 4:  1.0262e+02  1.0260e+02  3e-02  4e-08  2e-04
 5:  1.0260e+02  1.0260e+02  3e-04  4e-10  2e-06
 6:  1.0260e+02  1.0260e+02  3e-06  4e-12  2e-08
Optimal solution found.
Training accuracy for class 3 : 0.7433125502815768
Test accuracy for class 3 : 0.7490683229813665
     pcost       dcost       gap    pres   dres
 0:  7.2092e+01  7.7434e+03  1e+05  2e+00  1e+04
 1:  4.1685e+02 -1.1080e+03  2e+03  3e-02  1e+02
 2:  2.8321e+02  8.1151e+01  2e+02  6e-04  3e+00
 3:  1.0009e+02  9.7142e+01  3e+00  5e-06  2e-02
 4:  9.7268e+01  9.7239e+01  3e-02  5e-08  2e-04
 5:  9.7240e+01  9.7240e+01  3e-04  5e-10  2e-06
 6:  9.7240e+01  9.7240e+01  3e-06  5e-12  2e-08
Optimal solution found.
Training accuracy for class 8 : 0.7567880128720836
Test accuracy for class 8 : 0.7580124223602485
```

```
    pcost       dcost       gap    pres   dres
 0:  7.3366e+01  7.8702e+03  1e+05  2e+00  1e+04
 1:  4.1692e+02 -1.0902e+03  2e+03  3e-02  1e+02
 2:  2.8412e+02  8.4938e+01  2e+02  5e-04  3e+00
 3:  1.0238e+02  9.9646e+01  3e+00  4e-06  2e-02
 4:  9.9766e+01  9.9739e+01  3e-02  4e-08  2e-04
 5:  9.9740e+01  9.9740e+01  3e-04  4e-10  2e-06
 6:  9.9740e+01  9.9740e+01  3e-06  4e-12  2e-08
Optimal solution found.
Training accuracy for class 9 : 0.7505028157683025
Test accuracy for class 9 : 0.7493167701863354
Overall test accuracy: [0.7436024844720497, 0.7490683229813665,
0.7580124223602485, 0.7493167701863354]
Overall training accuracy: [0.7493966210780371, 0.7433125502815768,
0.7567880128720836, 0.7505028157683025]
Average test accuracy: 0.75
Average training accuracy: 0.75
Time: 137.53975129127502 seconds
------------------------------
Number of components: 20
    pcost       dcost       gap    pres   dres
 0:  7.3722e+01  7.8916e+03  1e+05  2e+00  1e+04
 1:  4.0328e+02 -5.4203e+03  6e+03  1e-01  6e+02
 2:  3.8054e+02  7.1276e+00  4e+02  2e-03  1e+01
 3:  1.1770e+02  9.9527e+01  2e+01  2e-16  3e-12
 4:  1.0036e+02  1.0017e+02  2e-01  9e-17  4e-14
 5:  1.0018e+02  1.0018e+02  2e-03  5e-16  8e-12
 6:  1.0018e+02  1.0018e+02  2e-05  3e-16  1e-11
Optimal solution found.
Training accuracy for class 2 : 0.7493966210780371
Test accuracy for class 2 : 0.7436024844720497
    pcost       dcost       gap    pres   dres
 0:  7.4801e+01  8.0114e+03  1e+05  2e+00  1e+04
 1:  4.1330e+02 -2.2304e+03  3e+03  5e-02  3e+02
 2:  3.3731e+02  6.1683e+01  3e+02  1e-03  6e+00
 3:  1.1327e+02  1.0227e+02  1e+01  3e-05  1e-01
 4:  1.0376e+02  1.0257e+02  1e+00  3e-06  1e-02
 5:  1.0264e+02  1.0259e+02  4e-02  7e-18  3e-11
 6:  1.0260e+02  1.0260e+02  1e-03  3e-16  6e-12
 7:  1.0260e+02  1.0260e+02  3e-05  3e-16  3e-11
Optimal solution found.
Training accuracy for class 3 : 0.7433125502815768
```

```
Test accuracy for class 3 : 0.7490683229813665
     pcost       dcost       gap    pres   dres
 0:  7.2106e+01  7.7433e+03  1e+05  2e+00  1e+04
 1:  4.1353e+02 -2.1700e+03  3e+03  5e-02  3e+02
 2:  3.3221e+02  5.3255e+01  3e+02  1e-03  7e+00
 3:  1.0953e+02  9.6891e+01  1e+01  3e-05  2e-01
 4:  9.8358e+01  9.7212e+01  1e+00  3e-06  2e-02
 5:  9.7270e+01  9.7236e+01  3e-02  3e-16  1e-11
 6:  9.7241e+01  9.7240e+01  1e-03  2e-16  5e-12
 7:  9.7240e+01  9.7240e+01  2e-05  3e-16  2e-11
Optimal solution found.
Training accuracy for class 8 : 0.7567880128720836
Test accuracy for class 8 : 0.7580124223602485
     pcost       dcost       gap    pres   dres
 0:  7.3381e+01  7.8702e+03  1e+05  2e+00  1e+04
 1:  4.1344e+02 -2.1984e+03  3e+03  5e-02  3e+02
 2:  3.3465e+02  5.7133e+01  3e+02  1e-03  6e+00
 3:  1.1128e+02  9.9402e+01  1e+01  3e-05  2e-01
 4:  1.0088e+02  9.9709e+01  1e+00  3e-06  1e-02
 5:  9.9774e+01  9.9735e+01  4e-02  3e-16  4e-11
 6:  9.9741e+01  9.9740e+01  1e-03  2e-16  7e-13
 7:  9.9740e+01  9.9740e+01  2e-05  3e-17  7e-12
Optimal solution found.
Training accuracy for class 9 : 0.7505028157683025
Test accuracy for class 9 : 0.7493167701863354
Overall test accuracy: [0.7436024844720497, 0.7490683229813665,
0.7580124223602485, 0.7493167701863354]
Overall training accuracy: [0.7493966210780371, 0.7433125502815768,
0.7567880128720836, 0.7505028157683025]
Average test accuracy: 0.75
Average training accuracy: 0.75
Time: 287.4265511035919 seconds
------------------------------
Number of components: 50
     pcost       dcost       gap    pres   dres
 0:  7.5178e+01  7.8767e+03  1e+05  2e+00  1e+04
 1:  3.9144e+02 -8.8781e+03  1e+04  2e-01  1e+03
 2:  4.0333e+02  6.2853e+01  3e+02  7e-04  3e+00
 3:  1.4513e+02  9.7303e+01  5e+01  9e-05  4e-01
 4:  1.0094e+02  1.0013e+02  8e-01  1e-06  6e-03
 5:  1.0018e+02  1.0017e+02  8e-03  1e-08  6e-05
 6:  1.0017e+02  1.0017e+02  8e-05  1e-10  6e-07
```

```
 7:  1.0017e+02  1.0017e+02  8e-07  1e-12  6e-09
Optimal solution found.
Training accuracy for class 2 : 0.7493966210780371
Test accuracy for class 2 : 0.7436024844720497
     pcost       dcost       gap    pres   dres
 0:  7.4972e+01  8.0096e+03  1e+05  2e+00  1e+04
 1:  4.0918e+02 -3.5017e+03  4e+03  8e-02  4e+02
 2:  3.6230e+02  3.8519e+01  3e+02  2e-03  8e+00
 3:  1.2377e+02  1.0180e+02  2e+01  7e-05  4e-01
 4:  1.0370e+02  1.0247e+02  1e+00  6e-07  3e-03
 5:  1.0267e+02  1.0259e+02  8e-02  2e-08  1e-04
 6:  1.0260e+02  1.0260e+02  2e-03  6e-10  3e-06
 7:  1.0260e+02  1.0260e+02  5e-05  1e-11  6e-08
Optimal solution found.
Training accuracy for class 3 : 0.7433125502815768
Test accuracy for class 3 : 0.7490683229813665
     pcost       dcost       gap    pres   dres
 0:  7.2260e+01  7.7417e+03  1e+05  2e+00  1e+04
 1:  4.0961e+02 -3.3979e+03  4e+03  8e-02  4e+02
 2:  3.5823e+02  2.8792e+01  3e+02  2e-03  9e+00
 3:  1.2084e+02  9.6359e+01  2e+01  9e-05  4e-01
 4:  9.8548e+01  9.7133e+01  1e+00  2e-06  1e-02
 5:  9.7322e+01  9.7231e+01  9e-02  9e-08  4e-04
 6:  9.7242e+01  9.7240e+01  2e-03  2e-09  1e-05
 7:  9.7240e+01  9.7240e+01  6e-05  4e-11  2e-07
 8:  9.7240e+01  9.7240e+01  2e-06  5e-13  3e-09
Optimal solution found.
Training accuracy for class 8 : 0.7567880128720836
Test accuracy for class 8 : 0.7580124223602485
     pcost       dcost       gap    pres   dres
 0:  7.3543e+01  7.8685e+03  1e+05  2e+00  1e+04
 1:  4.0943e+02 -3.4468e+03  4e+03  8e-02  4e+02
 2:  3.6019e+02  3.3277e+01  3e+02  2e-03  9e+00
 3:  1.2215e+02  9.8903e+01  2e+01  8e-05  4e-01
 4:  1.0093e+02  9.9621e+01  1e+00  1e-06  7e-03
 5:  9.9818e+01  9.9731e+01  9e-02  5e-08  3e-04
 6:  9.9742e+01  9.9740e+01  2e-03  1e-09  6e-06
 7:  9.9740e+01  9.9740e+01  6e-05  3e-11  1e-07
 8:  9.9740e+01  9.9740e+01  2e-06  3e-13  2e-09
Optimal solution found.
Training accuracy for class 9 : 0.7505028157683025
Test accuracy for class 9 : 0.7493167701863354
```

Overall test accuracy: [0.7436024844720497, 0.7490683229813665, 0.7580124223602485, 0.7493167701863354]
Overall training accuracy: [0.7493966210780371, 0.7433125502815768, 0.7567880128720836, 0.7505028157683025]
Average test accuracy: 0.75
Average training accuracy: 0.75
Time: 764.6435108184814 seconds
----

2. Part 1.(b)
Number of components: 10
Test accuracy with C = 0.01 : 0.9108074534161491
Time: 3.0747487545013428 seconds
Training accuracy with C = 0.01 : 0.9005430410297667

Number of components: 20
Test accuracy with C = 0.01 : 0.9319254658385093
Time: 3.056218385696411 seconds
Training accuracy with C = 0.01 : 0.9249296057924377

Number of components: 50
Test accuracy with C = 0.01 : 0.9542857142857143
Time: 8.29580020904541 seconds
Training accuracy with C = 0.01 : 0.950020112630732

----

2. Part 1.(c)
Number of components: 10

```
     pcost       dcost       gap    pres   dres
 0: -8.8389e+02 -2.4957e+03  2e+03  5e-13  1e+00
 1: -8.9802e+02 -9.2354e+02  3e+01  2e-14  5e-02
 2: -8.9949e+02 -8.9979e+02  3e-01  1e-14  6e-04
 3: -8.9949e+02 -8.9950e+02  3e-03  1e-14  6e-06
 4: -8.9949e+02 -8.9949e+02  3e-05  2e-14  6e-08
Optimal solution found.
```

Training accuracy for class 2 : 0.7493966210780371
Test accuracy for class 2 : 0.7436024844720497

```
     pcost       dcost       gap    pres   dres
 0: -7.9084e+02 -2.2170e+03  1e+03  2e-13  1e+00
 1: -8.0295e+02 -8.2669e+02  2e+01  2e-14  5e-02
 2: -8.0431e+02 -8.0459e+02  3e-01  3e-15  5e-04
 3: -8.0431e+02 -8.0431e+02  3e-03  8e-15  5e-06
 4: -8.0431e+02 -8.0431e+02  3e-05  4e-14  5e-08
Optimal solution found.
```

Training accuracy for class 3 : 0.7433125502815768
Test accuracy for class 3 : 0.7490683229813665

```
     pcost       dcost       gap    pres   dres
 0: -1.4518e+02 -4.3141e+02  3e+02  2e-12  1e+00
 1: -1.4525e+02 -1.4904e+02  4e+00  3e-14  2e-02
 2: -1.4527e+02 -1.4533e+02  6e-02  5e-16  3e-04
 3: -1.4527e+02 -1.4527e+02  6e-04  2e-15  3e-06
 4: -1.4527e+02 -1.4527e+02  6e-06  2e-15  3e-08
Optimal solution found.
```

Training accuracy for class 8 : 0.7567880128720836
Test accuracy for class 8 : 0.7580124223602485

```
     pcost       dcost       gap    pres   dres
 0: -1.1842e+02 -3.5256e+02  2e+02  2e-12  1e+00
 1: -1.1846e+02 -1.2145e+02  3e+00  3e-14  2e-02
 2: -1.1848e+02 -1.1853e+02  5e-02  4e-15  3e-04
 3: -1.1848e+02 -1.1848e+02  5e-04  6e-16  3e-06
 4: -1.1848e+02 -1.1848e+02  5e-06  1e-15  3e-08
Optimal solution found.
```

Training accuracy for class 9 : 0.7505028157683025
Test accuracy for class 9 : 0.7493167701863354
Overall test accuracy: [0.7436024844720497, 0.7490683229813665, 0.7580124223602485, 0.7493167701863354]
Overall training accuracy: [0.7493966210780371, 0.7433125502815768, 0.7567880128720836, 0.7505028157683025]
Time: 19827.0478618145

Number of components: 20

```
     pcost       dcost       gap    pres   dres
 0: -8.8412e+02 -2.4963e+03  2e+03  2e-12  1e+00
 1: -8.9825e+02 -9.2376e+02  3e+01  1e-13  5e-02
 2: -8.9972e+02 -9.0002e+02  3e-01  5e-14  6e-04
 3: -8.9972e+02 -8.9973e+02  3e-03  2e-14  6e-06
 4: -8.9972e+02 -8.9972e+02  3e-05  1e-13  6e-08
Optimal solution found.
```

Training accuracy for class 2 : 0.7493966210780371

Test accuracy for class 2 : 0.7436024844720497

```
     pcost       dcost       gap    pres   dres
 0: -7.9100e+02 -2.2174e+03  1e+03  9e-13  1e+00
 1: -8.0311e+02 -8.2684e+02  2e+01  4e-14  5e-02
 2: -8.0446e+02 -8.0473e+02  3e-01  3e-13  5e-04
 3: -8.0446e+02 -8.0447e+02  3e-03  4e-14  5e-06
 4: -8.0446e+02 -8.0446e+02  3e-05  4e-14  5e-08
Optimal solution found.
```

Training accuracy for class 3 : 0.7433125502815768

Test accuracy for class 3 : 0.7490683229813665

```
     pcost       dcost       gap    pres   dres
 0: -1.4520e+02 -4.3149e+02  3e+02  5e-12  1e+00
 1: -1.4528e+02 -1.4907e+02  4e+00  9e-14  2e-02
 2: -1.4530e+02 -1.4536e+02  6e-02  4e-14  3e-04
 3: -1.4530e+02 -1.4530e+02  6e-04  1e-14  3e-06
 4: -1.4530e+02 -1.4530e+02  6e-06  5e-15  3e-08
Optimal solution found.
```

Training accuracy for class 8 : 0.7567880128720836

Test accuracy for class 8 : 0.7580124223602485

```
     pcost       dcost       gap    pres   dres
 0: -1.1862e+02 -3.5312e+02  2e+02  8e-12  1e+00
 1: -1.1865e+02 -1.2165e+02  3e+00  1e-13  2e-02
 2: -1.1867e+02 -1.1872e+02  5e-02  3e-15  3e-04
 3: -1.1867e+02 -1.1867e+02  5e-04  5e-15  3e-06
 4: -1.1867e+02 -1.1867e+02  5e-06  1e-14  3e-08
```

Optimal solution found.

Training accuracy for class 9 : 0.7505028157683025
Test accuracy for class 9 : 0.7493167701863354
Overall test accuracy: [0.7436024844720497, 0.7490683229813665, 0.7580124223602485, 0.7493167701863354]
Overall training accuracy: [0.7493966210780371, 0.7433125502815768, 0.7567880128720836, 0.7505028157683025]
Time: 19957.692722558975

2. Part 1.(d)
Number of components: 10
Time: 6.727479457855225 seconds
Test accuracy with gamma = 0.01 : 0.9239751552795031
Training accuracy with gamma = 0.01 : 0.9131637168141593

Time: 6.771031379699707 seconds
Test accuracy with gamma = 0.01 : 0.9239751552795031
Training accuracy with gamma = 0.01 : 0.9131637168141593

Time: 2.523374080657959 seconds
Test accuracy with gamma = 0.1 : 0.9560248447204969
Training accuracy with gamma = 0.1 : 0.9604283990345938

Number of components: 20
Time: 8.73709225654602 seconds
Test accuracy with gamma = 0.01 : 0.9562732919254658
Training accuracy with gamma = 0.01 : 0.9529364440868866

Time: 9.698148727416992 seconds
Test accuracy with gamma = 0.01 : 0.9562732919254658
Training accuracy with gamma = 0.01 : 0.9529364440868866

Time: 2.3066234588623047 seconds
Test accuracy with gamma = 0.1 : 0.9803726708074534
Training accuracy with gamma = 0.1 : 0.9997988736926791

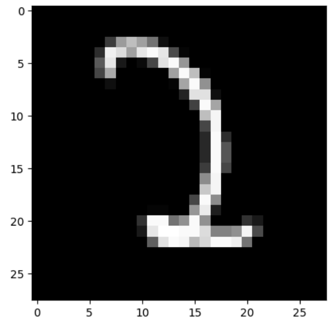
Number of components: 50

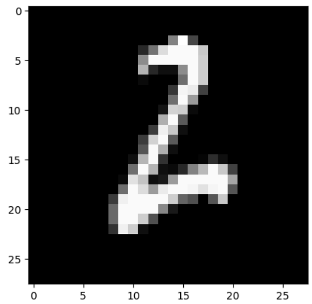
Time: 19.61922287940979 seconds
Test accuracy with gamma = 0.01 : 0.9719254658385094
Training accuracy with gamma = 0.01 : 0.9735518905872889

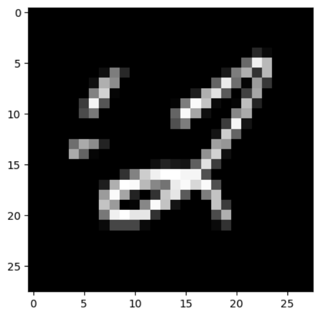
Time: 19.882307052612305 seconds
Test accuracy with gamma = 0.01 : 0.9719254658385094
Training accuracy with gamma = 0.01 : 0.9735518905872889

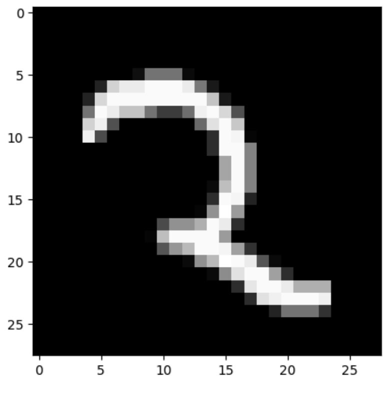
Time: 6.909613609313965 seconds
Test accuracy with gamma = 0.1 : 0.9880745341614907
Training accuracy with gamma = 0.1 : 1.0
----

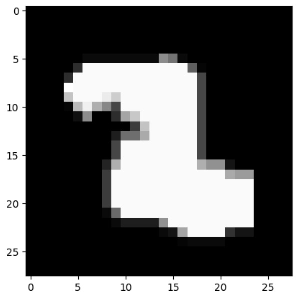
3. Images of the support vectors are as follows:



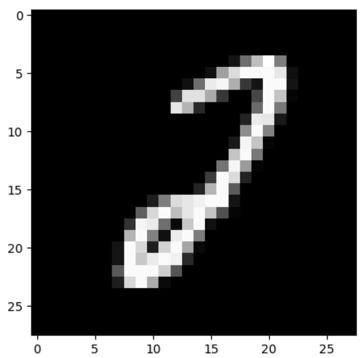


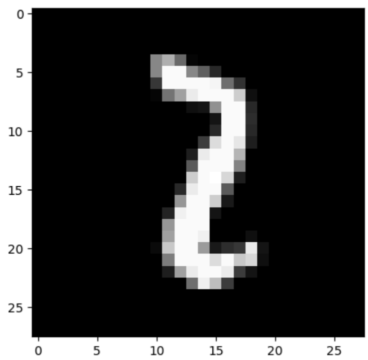


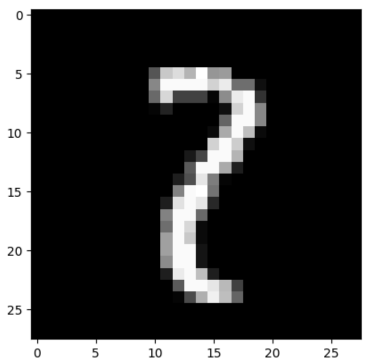


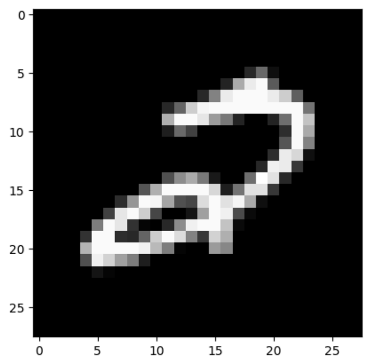


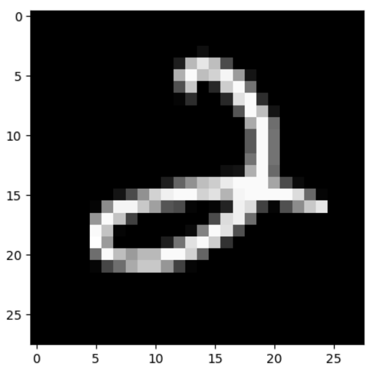
Images of the non-support vectors are as follows:







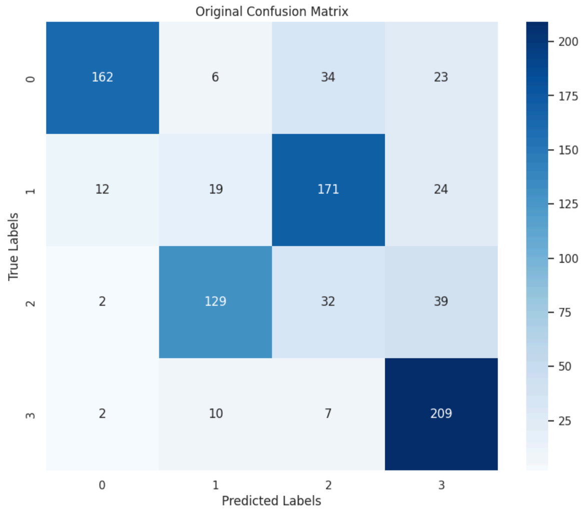


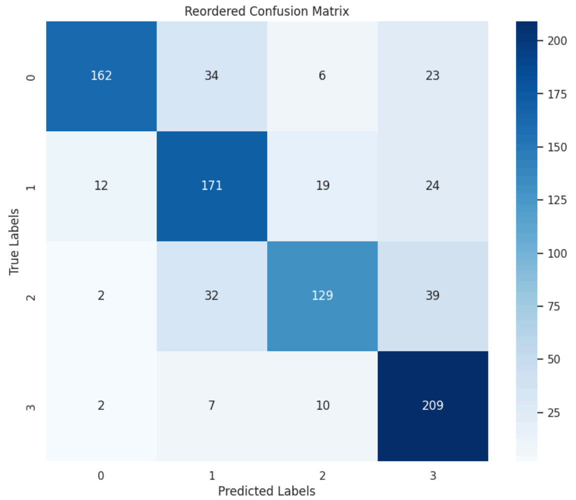


PART 3: Clustering

3.1 Normalizing data points before running k-means is very important. K-means clustering relies on distance measures to assign data points to clusters. If the data features have different scales, the features with larger scales will dominate the distance calculation. This leads to biased results. In addition, normalization can provide more efficient calculations and speed up the process. [1]

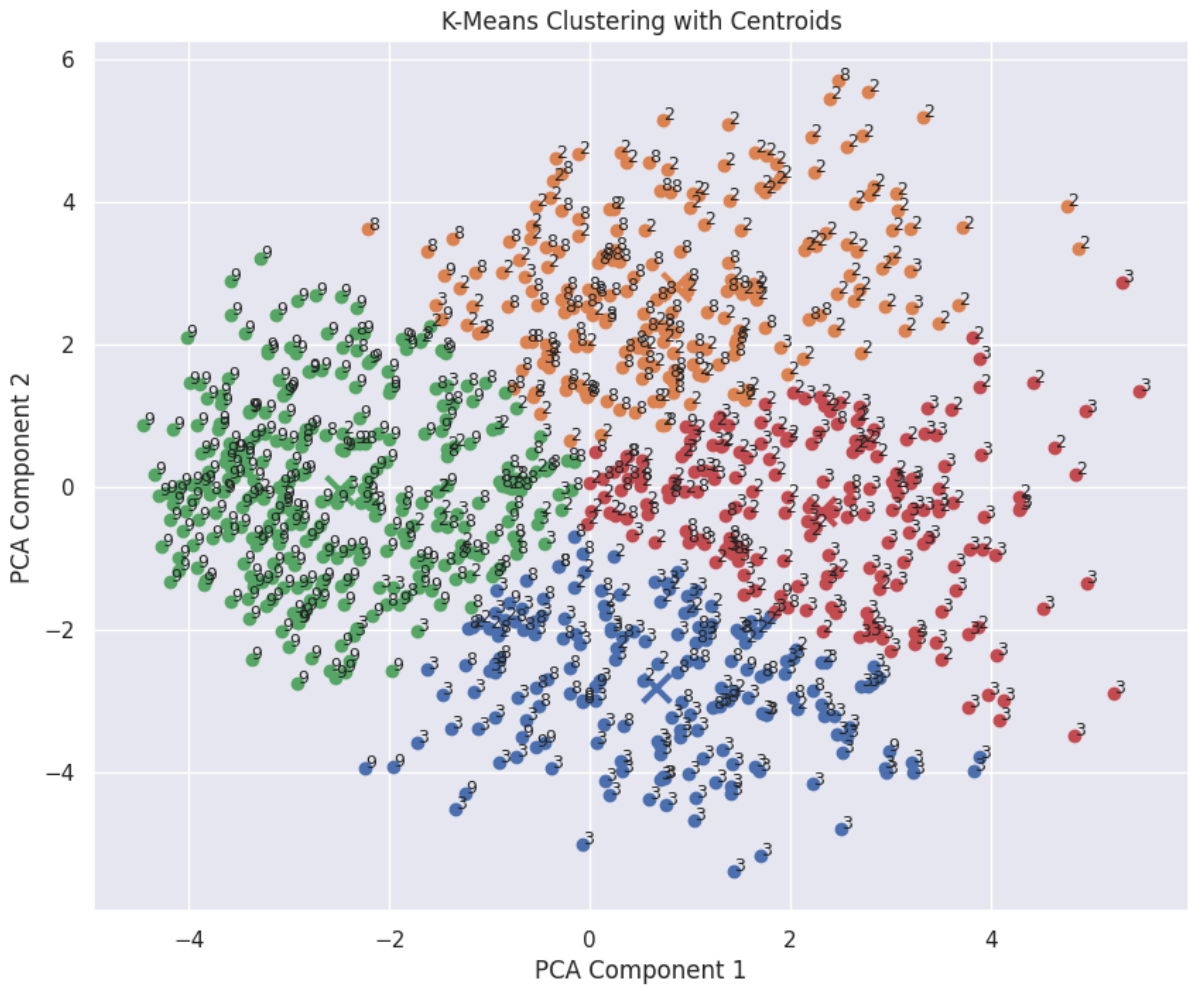
3.2 Since clustering algorithms are unsupervised and not intended for classification, the accuracy calculation is not as meaningful as supervised ones. However, we have labeled data and we can somehow calculate the clustering accuracy. We used a confusion matrix and permuted to maximize the sum of the diagonal elements. We then calculated the accuracy by dividing the sum of the diagonal elements by the number of samples. [2]





We used 10 PCA components and took the average of 50 iterations. PCA slightly increased the accuracy but significantly improved the SSE value.

Average Clustering Accuracy (Euclidean) : 0.7399772985244042
Average Clustering Accuracy (Euclidean_PCA) : 0.7520090805902382
Average SSE Score (Euclidean) : 37510.38144626843
Average SSE Score (Euclidean_PCA) : 15464.784415431119



Note: For visualization, data is reduced to have 2 PCA components

3.3 We used again 10 PCA components and took the average of 50 iterations. Again, PCA slightly increased the accuracy but significantly improved the SSE value.

Average Clustering Accuracy (Cosine)	: 0.7145516458569807
Average Clustering Accuracy (Cosine_PCA)	: 0.7409988649262202
Average SSE Score (Cosine)	: 37673.15406891733
Average SSE Score (Cosine_PCA)	: 15541.066397754716

If we compare Euclidean and Cosine, we can say that the SSE values are almost the same. We see that the accuracy value of Euclidean is slightly higher (74% vs 71%). We can say that there is no significant difference between Euclidean and Cosine similarity measures.

References:

[1] https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35
[2] https://smorbieu.gitlab.io/accuracy-from-classification-to-clustering-evaluation/