Machine Learning
Assignment #2

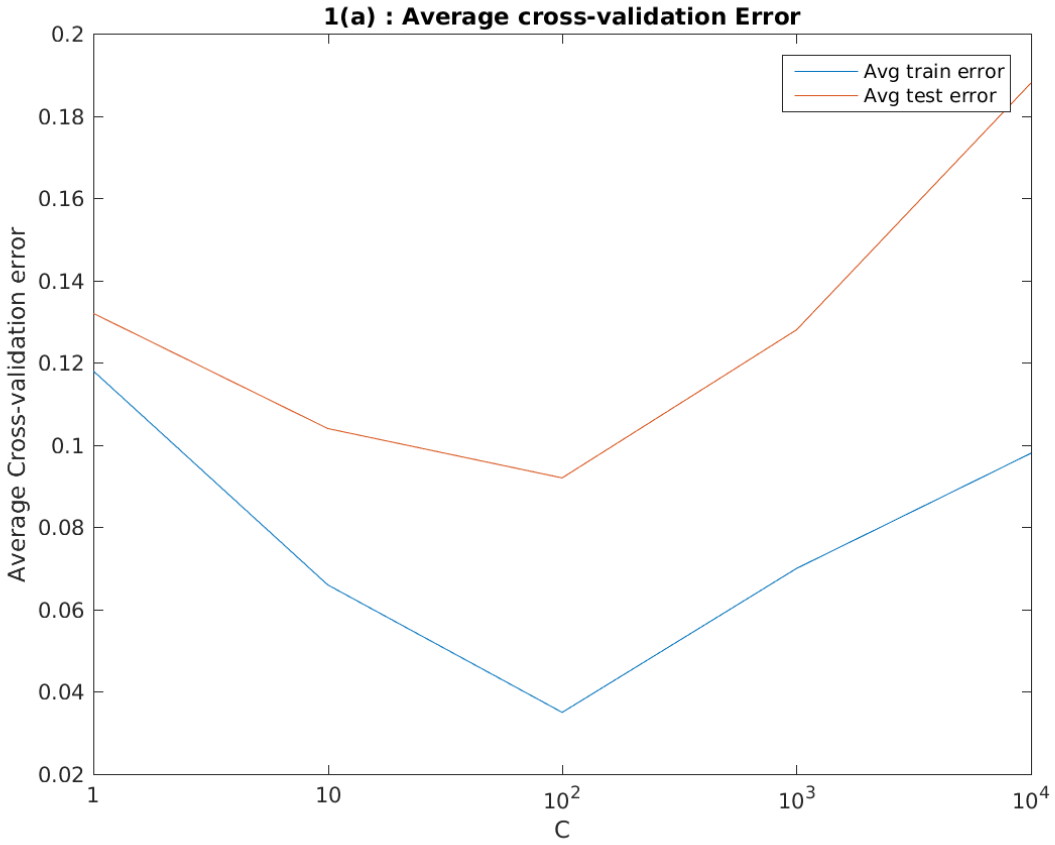
Shikhar Vashishth

M.Tech CSA - 13374

Problem 1

Part (a):

NOTE: I have used cvpartition method for creating cross-validation set, so my results may not match with the standard solution key.

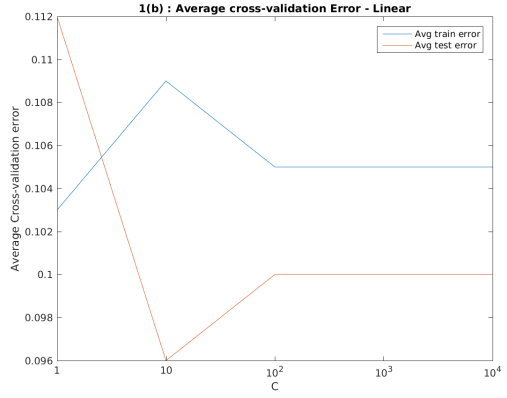


C	Avg Training Error	Avg Test Error
1	0.1180	0.1320
10	0.0660	0.1040
100	**0.0350**	**0.0920**
1000	0.0700	0.1280
10000	0.0980	0.1880

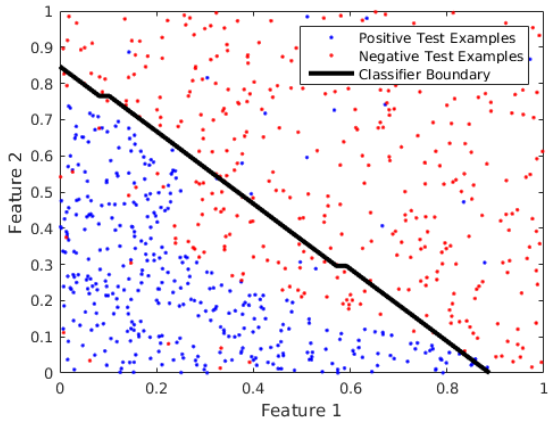
We can see from the above results that the best performance is obtained at $C = 100$.

Part (b):

Using Linear kernel



(a) 5-fold cross validation error using linear kernel

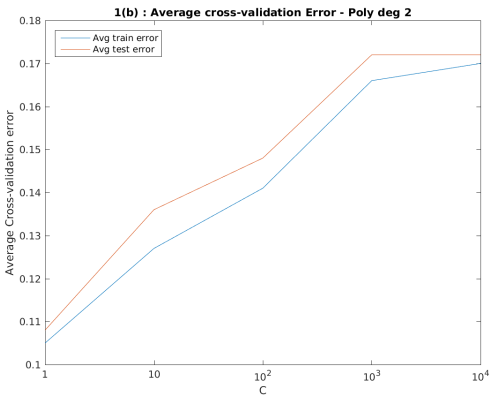


(b) Performance of linear kernel with C = 10

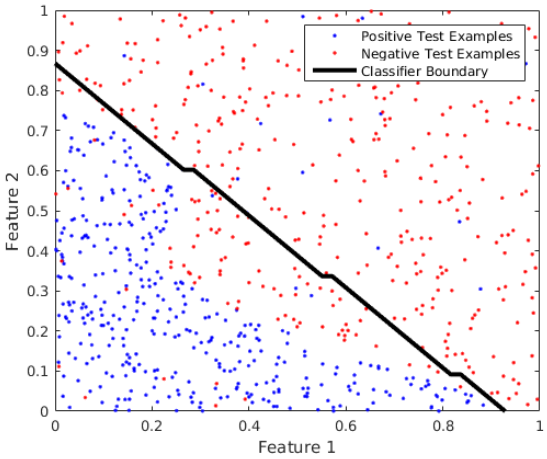
C	Avg Training Error	Avg Test Error
1	0.1030	0.1120
10	**0.1090**	**0.0960**
100	0.1050	0.1000
1000	0.1050	0.1000
10000	0.1050	0.1000

Best C: **10**
Training error on full dataset: **0.1040**
Test error on full dataset: **0.1320**

Using degree 2 Polynomial kernel



(a) 5-fold cross validation error using degree 2 polynomial

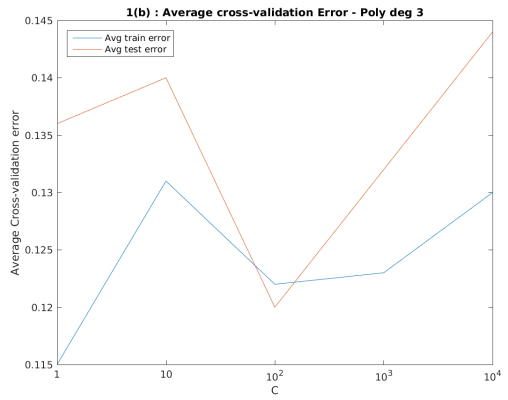


(b) Performance with polynomial degree 2 kernel with C = 1

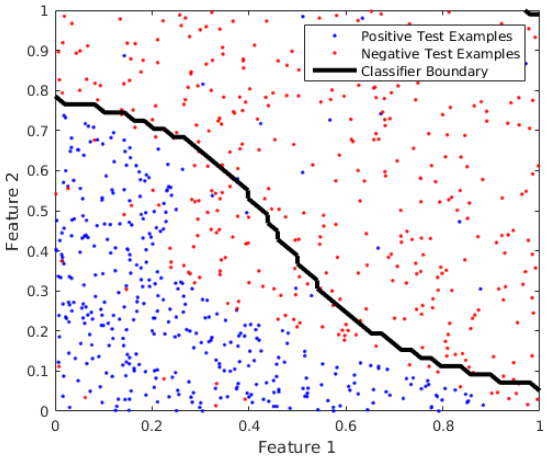
C	Avg Training Error	Avg Test Error
1	**0.1050**	**0.1080**
10	0.1270	0.1360
100	0.1410	0.1480
1000	0.1660	0.1720
10000	0.1700	0.1720

Best C: **1**
Training error on full dataset: **0.1080**
Test error on full dataset: **0.1360**

Using Polynomial degree 3 Kernel



(a) Cross validation error using degree 3 polynomial kernel

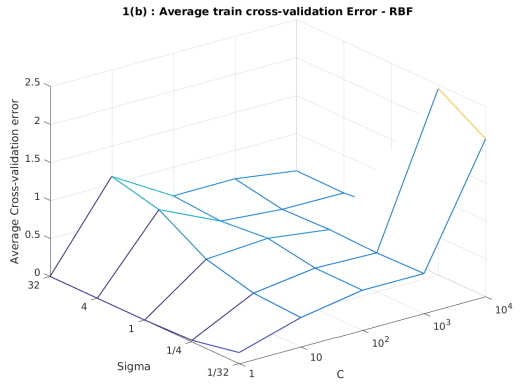


(b) Performance with polynomial degree 3 kernel with C = 100

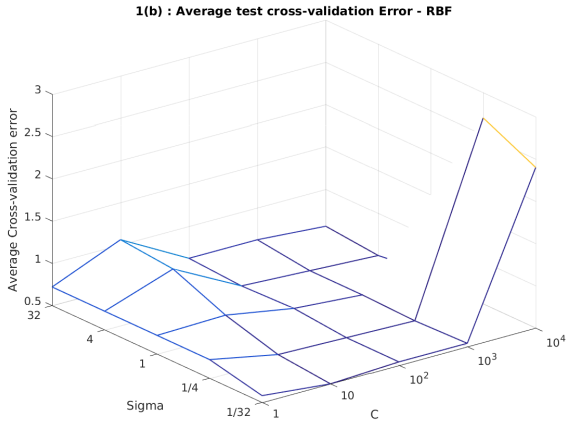
C	Avg Training Error	Avg Test Error
1	0.1150	0.1360
10	0.1310	0.1400
100	**0.1220**	**0.1200**
1000	0.1230	0.1320
10000	0.1300	0.1440

Best C: **100**
Training error on full dataset: **0.1080**
Test error on full dataset: **0.1440**

Using RBF Kernel



(a) Cross validation error on training data using RBF kernel



(b) cross validation error on test data using RBF kernel

Training Error:

C/sigma	1/32	1/4	1	4	32
1	0.1550	0.4150	0.5300	0.5250	2.0600
10	0.0200	0.3950	0.5300	0.5150	2.4500
100	0	0.6750	0.6300	0.5150	0.5750
1000	0	0.8700	0.5700	0.5250	0.5300
10000	0	0.8650	0.6350	0.6150	0.5350

Test Error:

C/sigma	1/32	**1/4**	1	4	32
1	0.6400	0.5400	0.5800	0.6600	2.2800
10	0.7400	**0.4800**	0.5800	0.5600	2.7000
100	0.7600	0.6200	0.6200	0.5400	0.5800
1000	0.7600	0.7800	0.6000	0.5800	0.6000
10000	0.7600	0.8600	0.6600	0.6400	0.5000

Best C: **10**
Best Sigms: **1/4**
Training error: **0.0880**
Test error: **0.1387**

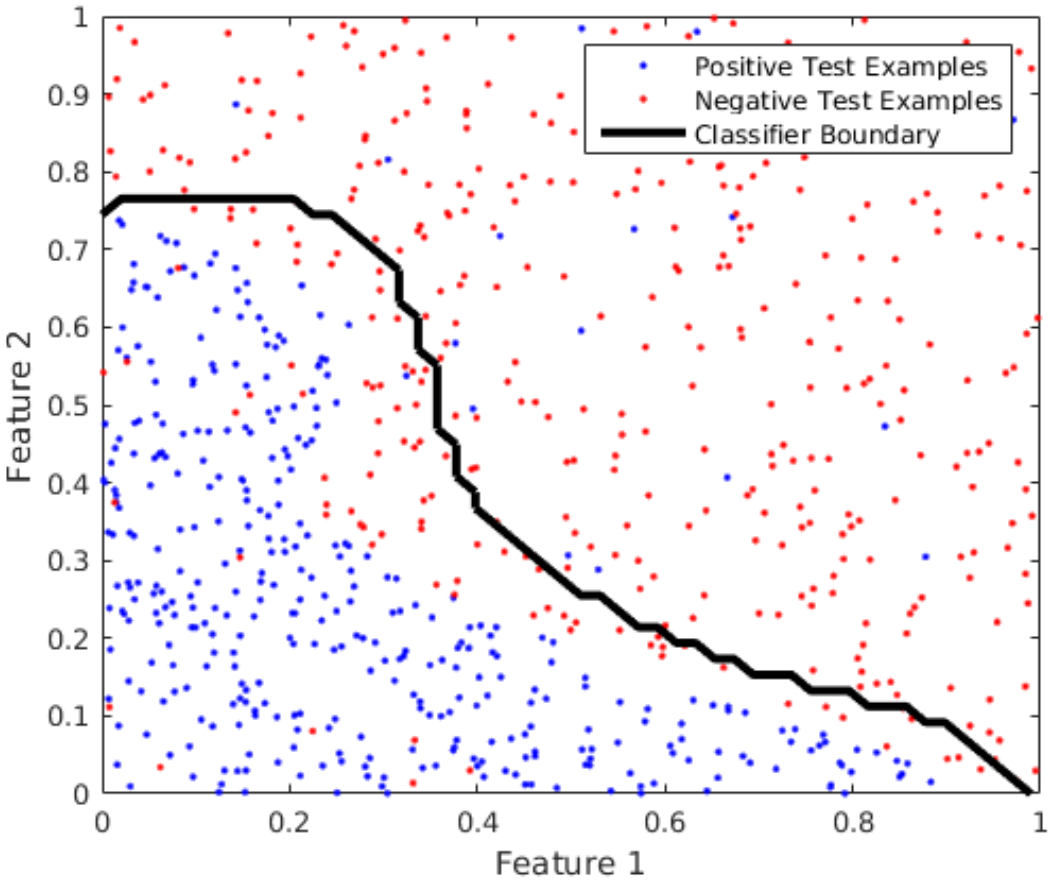


Figure 5: Performance with RBF kernel with C = 10 and sigma = 1/4

Part (c):

Error on the test set in case of SVM is **0.1168**, whereas in the case of logistic regression error is **0.1191**. We can clearly see that the performance of SVM is better than performance of logistic regression. The performance of SVM is better, because it fits a maximal margin linear classifier which gives us better generalization any than other linear classifiers.

Problem 2

Part (c):

Training Error: **101.3872**
Testing Error: **26.8586**

Part (d):

Error on Training folds:

λ/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	105.3216	98.2991	100.0530	101.8518	101.2350
0.1	105.3216	98.2991	100.0530	101.8518	101.2350
1	105.3245	98.3021	100.0561	101.8551	101.2378
10	105.5627	98.5515	100.3096	102.1285	101.4685
100	112.5263	105.6139	107.3352	109.6258	108.1266

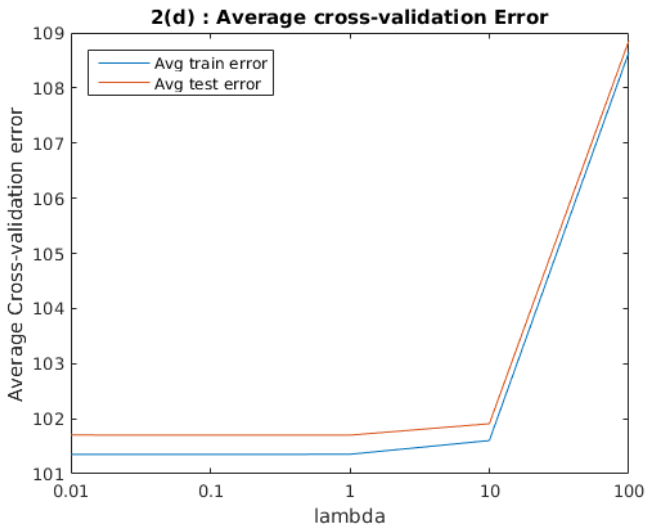
Error on test folds

λ/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	85.7615	113.9051	106.7523	99.7846	102.3173
0.1	85.7628	113.9046	106.7523	99.7746	102.3238
1	85.7780	113.9033	106.7560	99.6801	102.3902
10	86.0910	114.1163	107.0459	99.1158	103.1771
100	92.4212	121.1080	115.0829	103.6544	112.0226

Average cross-validation training and test error

C	Training Error	Test Error
0.01	101.3521	101.7041
0.1	101.3521	101.7036
1	**101.3551**	**101.7015**
10	101.6042	101.9092
100	108.6456	108.8578

Cross-Validation results The average error for first four λ values $(.01, 0.1, 1, 10)$ are almost the same but we can see from the above table (test error) that with $\lambda = 1$ the test error is slightly lesser than with other λ values.



C	Training Error	Test Error
0.01	101.3872	26.8575
0.1	101.3872	26.8476
1	101.3891	26.7510
10	**101.5543**	**26.0140**
100	107.1628	26.8059

Table 1: Training and test error with complete data set

Results with complete data sets We can see from the above results that the best test error is obtained with $\lambda = 10$ which doesn't match with the lambda value obtained from the cross-validation method.

Comparison with linear least square regression Although, as compared to linear least squares the training set error has slightly increased in the case of ridge regression, but the test error has reduced from **26.8586 to 26.0140**, which shows that ridge regression gave us a more generalized classifier.

Problem: 2(e)

Original form of Ridge regression

$$\min \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2$$

$$\text{s.t } X \in \mathbb{R}^{n \times d},\ y \in \mathbb{R}^n,\ w \in \mathbb{R}^d$$

Let $z = Xw - y$

Objective function $\min_{w,z} \frac{1}{2} z^T z + \frac{\lambda}{2}\|w\|^2$

$$\text{s.t } z = Xw - y$$

Lagrangian

$$L(z, w, \alpha) = \frac{1}{2} z^T z + \frac{\lambda}{2} w^T w + \alpha^T (z - Xw + y) \quad ; \alpha \in \mathbb{R}^n$$

Dual

$$\min_{z,w} L(z, w, \alpha)$$

$$\nabla_z L = z + (\alpha^T(-1))^T = 0$$

$$\boxed{z = -\alpha}$$

$$\nabla_w L = \lambda w - X^T \alpha = 0$$

$$\boxed{w = \frac{1}{\lambda} X^T \alpha}$$

Dual Optimization problem

$$\max_{\alpha} L\left(-\alpha, \frac{1}{\lambda} X^T \alpha, \alpha\right)$$

$$\max_{\alpha} \frac{1}{2}\alpha^T\alpha + \frac{\lambda}{2}\left(\frac{1}{\lambda}X^T\alpha\right)^T\left(\frac{1}{\lambda}X^T\alpha\right) + \alpha^T\left(-\alpha - X\left(\frac{1}{\lambda}X^T\alpha\right) + y\right)$$

$$\max_{\alpha} \frac{1}{2}\alpha^T\alpha + \frac{1}{2\lambda}\alpha^T XX^T\alpha - \alpha^T\alpha - \frac{1}{\lambda}\alpha^T XX^T\alpha + \alpha^T y$$

$$\max_{\alpha} -\frac{1}{2}\alpha^T\alpha - \frac{1}{2\lambda}\alpha^T XX^T\alpha + \alpha^T y$$

Dual objective function $= g(\alpha)$

Dual optimization problem

$$\max_{\alpha} \; -\frac{1}{2}\alpha^T\alpha - \frac{1}{2\lambda}\alpha^T XX^T\alpha + \alpha^T y$$

⇒ This is an unconstrained problem in α

$$\nabla_\alpha g = 0$$

$$-\alpha - \frac{1}{2\lambda}(XX^T + XX^T)\alpha + y = 0$$

$$-\alpha - \frac{1}{\lambda}(XX^T)\alpha + y = 0$$

$$\alpha\left(I + \frac{1}{\lambda}XX^T\right) = y$$

$$\boxed{\alpha^* = \left(I + \frac{1}{\lambda}XX^T\right)^{-1} y}$$

⇒ closed form soln for α

$$w = \frac{1}{\lambda}X^T\alpha$$

Let $\beta = \frac{\alpha}{\lambda}$, $\therefore \beta \in \mathbb{R}^n = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$

$$w = X^T\beta$$

$$\boxed{w = \sum_{i=1}^{n} \beta_i x_i}$$

$$\boxed{\beta^* = \frac{\alpha^*}{\lambda} = \frac{1}{\lambda}\left(I + \frac{1}{\lambda}XX^T\right)^{-1} y}$$

As we can see that the formulation of Ridge regression interms of β variable is equivalent to its formulation in terms of w.

If suppose β^* is the solution of above formulation then using relation $w = \sum_{i=1}^{n} \beta_i x_i$ we can get w which will satisfy original formulation of Ridge regression.

Part (f):

Error on training folds

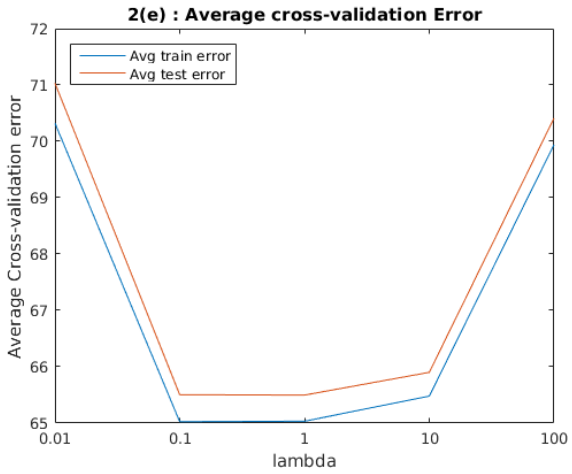
λ/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	68.1371	62.3615	86.5153	67.1764	67.4253
0.1	68.0578	62.2227	63.6664	66.0992	65.0810
1	68.0666	62.2314	63.6733	66.1081	65.0863
10	68.5379	62.7093	64.0655	66.5871	65.4939
100	73.0739	67.1732	68.1676	71.1525	70.1366

Error on test folds

λ/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	53.4618	76.8461	95.0592	61.7699	68.0263
0.1	53.3219	76.6653	71.0157	61.0810	65.4354
1	53.2795	76.6249	71.1206	61.0230	65.4367
10	53.4503	76.9216	72.2954	61.1128	65.7226
100	57.5729	82.2489	78.7476	65.3079	68.1862

Average cross-validation training and test error

C	Training Error	Test Error
0.01	70.3231	71.0327
0.1	65.0254	65.5039
1	**65.0332**	**65.4969**
10	65.4787	65.9005
100	69.9408	70.4127



Cross Validation results We can see from the results that $\lambda = 1$ looks like the best choice for the given setting. Although, $\lambda = 0.1$ is also quite close in performance to 1 but on average results with $\lambda = 1$ are better.

C	Training Error	Test Error
0.01	66.9187	23.7157
0.1	65.0736	16.0687
1	65.0775	16.0159
10	65.3960	14.5744
100	**69.2362**	**12.8448**

Table 2: Training and test error with complete data set

Results with complete data sets We can see from the graph that the best test error with full data set is obtained with $\lambda = 100$ which doesn't match with the results which we obtained from the cross-validation method.

Comparison with linear ridge regression As compared to the linear ridge regression the performance in this case is much better. Both training error and test errors have reduced which shows that the actual distribution of the data can be better modelled using a polynomial of degree 3 rather than a linear model.

Problem 3

Part (b):

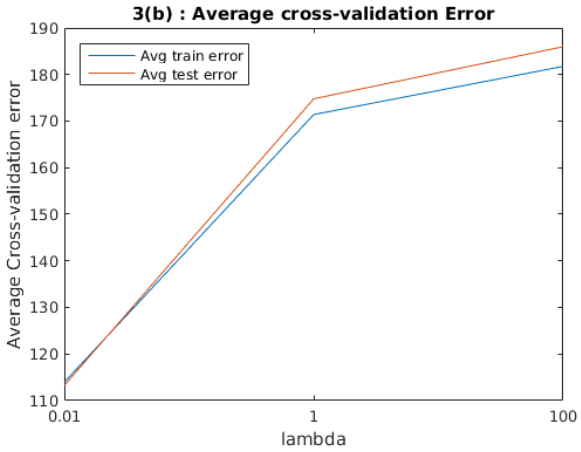
Cross-Validation

Error on training folds

C/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	113.9167	106.8238	123.0163	111.0709	114.9589
1	118.7971	120.3559	113.3968	134.9623	369.5602
100	106.9600	157.2253	101.7997	103.9910	438.7858

Error on test folds

C/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	93.1396	125.8512	129.2377	104.1163	113.9241
1	94.8845	128.5385	122.5024	128.4787	399.5460
100	87.6967	160.2333	109.3543	100.2524	472.3654



Average cross-validation training and test error

C	Training Error	Test Error
0.01	**113.9573**	**113.2538**
1	171.4144	174.7900
100	181.7524	185.9804

Cross Validation results Cross-Validation method tells us that the best value of C is 0.01, because with this value, we are getting the least average test and training error.

Training and test error on full training and test set (*1.0e+03)

C	Training Error	Test Error
0.01	107.7798	20.6209
1	**104.8866**	**21.2221**
100	104.1523	21.5284

Results with complete data sets From the above results we can see that the best value of C is 1, because it is giving us least test error. Hence, in this age cross-validation didn't give us the right estimate of C.

Comparing with linear least squares and linear ridge regression We can see from the results on entire training and test set that the performance of support vector regression is better compared to linear least squares and linear ridge regression algorithms.

Part (c):

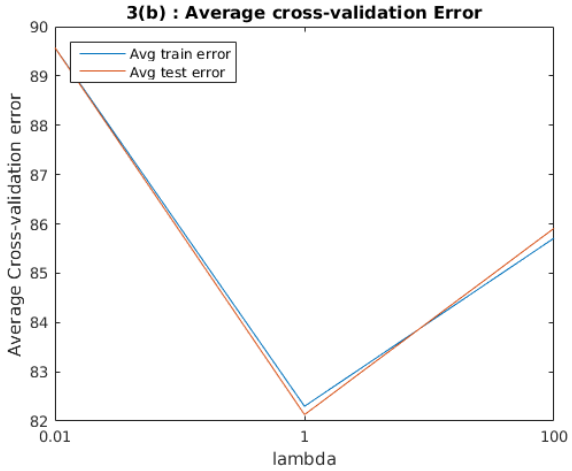
Cross-Validation

Error on training folds

C/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	94.5786	83.2528	86.0032	91.7747	92.2636
1	87.8022	76.3866	81.9559	85.6317	79.7258
100	97.4177	80.1165	86.5986	80.7769	83.6444

Error on test folds

C/Folds	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.01	74.4860	101.4484	97.0814	84.9846	89.8629
1	68.4400	93.0966	91.9062	77.4420	79.7727
100	77.2873	98.2538	97.0021	73.8498	83.1568



Average cross-validation training and test error

C	Training Error	Test Error
0.01	89.5746	89.5727
1	**82.3005**	**82.1315**
100	85.7108	85.9100

Cross Validation results Cross-Validation method tells us that the best value of C is 1, because with this value, we are getting the least average test and training error.

Training and test error on full training and test set

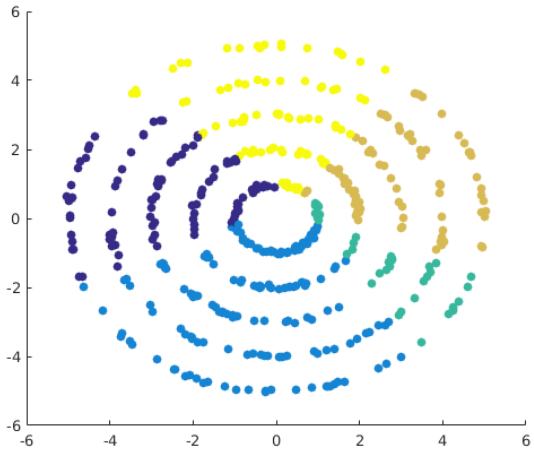
C	Training Error	Test Error
0.01	86.6784	8.9359
1	85.0728	0.8421
100	**83.5751**	**0.7109**

Results with complete data sets From the above results we can see that the best value of C is 100, because it is giving us least test error. Hence, in this age cross-validation didn't give us the right estimate of C.

Comparing with linear SVR As compared to linear SVR results in this case are much better. We can see that the test error has fallen from 21.2 to 0.7109, which is about 30 times improvement.

Problem 4

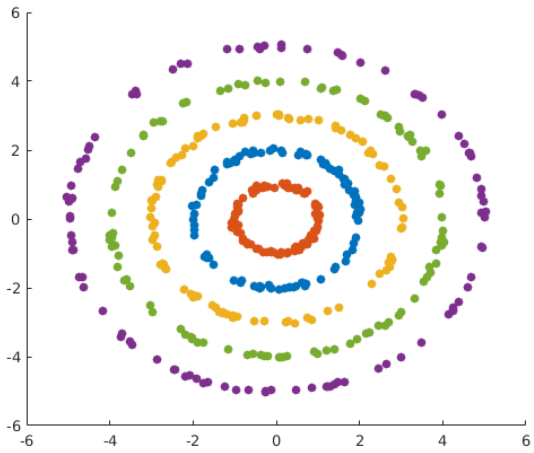
Part (a):



Part (b):

No, the results are not satisfactory, because the basic k-means algorithm assumes that the variance of the distribution of each cluster is spherical, which is not the case here (clusters are in form of rings). K-means also assumes that all the clusters have same variance which is also not true for the dataset.

Part (c):



Part (d):

RAND score K-means: **0.6409**
RAND score Kernel K-means: **1.0**