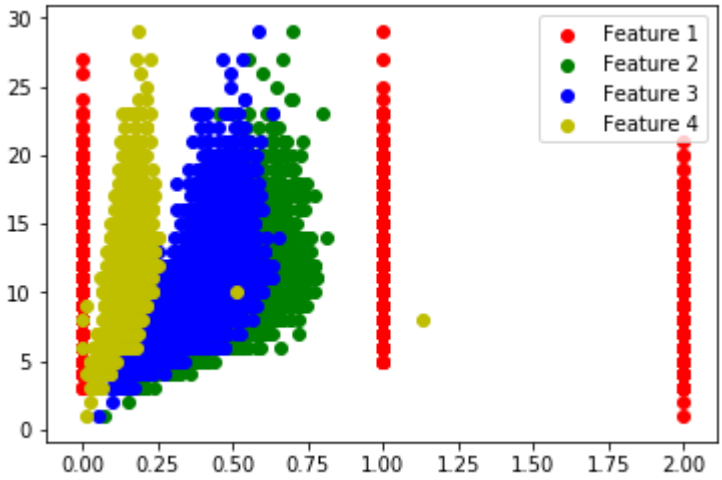
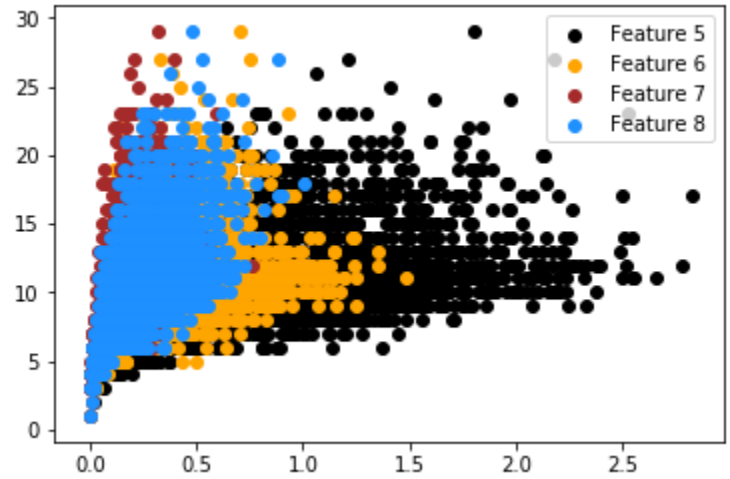
Details of Question 1

- Data- Dataset.data [rows – 4177, columns-9, final value ranges between 0 and 30]
- 5 Fold-indexes [[0, 834], [835, 1669], [1670, 2504], [2505, 3339], [3340, 4176]]
- Different plots showing relation between features and final output

  Plot-1 it's between first four features and output value



Plot-2 it's between last four features and output value



Not sure <<It is clear from the plot-1 that feature-1 will create noise in the learning process. So, we can remove it.>>

1(a) `.predict()' from scratch using the outcomes of `.fit()'.

1(b) [Implemented MSE function]

Observation: Table-I

Fold-Number	Training MSE	Validation MSE
1	3.774039336741574	9.880113
2	5.333943966901227	3.024387
3	4.638637144821676	5.8376906899
4	5.114791452508	3.754576202268
5	5.04191978	4.035545
Mean of all errors	4.78066633	5.306462

Same errors using MSE of sklearn- TableII

Fold-Number	Training MSE	Validation MSE
1	3.774039336741574	9.880113
2	5.333943966901227	3.024387
3	4.638637144821676	5.8376906899
4	5.114791452508	3.754576202268
5	5.04191978	4.035545
Mean of all errors	4.78066633	5.306462

Comparison between Table-I and Tabel-II: There is no difference between self-defined MSE function and sklearn.mse function. Both are giving the same values.

1(c)

Observation: Table-I

Fold-Number	Training MSE	Validation MSE
1	3.774039336741574	9.880113
2	5.333943966901227	3.024387
3	4.638637144821676	5.8376906899
4	5.114791452508	3.754576202268
5	5.04191978	4.035545
Mean of all errors	4.78066633	5.306462

Same errors using MSE of sklearn- TableII

Fold-Number	Training MSE	Validation MSE
1	3.774039336741574	9.880113
2	5.333943966901227	3.024387
3	4.638637144821676	5.8376906899
4	5.114791452508	3.754576202268
5	5.04191978	4.035545
Mean of all errors	4.78066633	5.306462

Same result as 1(b).

1(d)

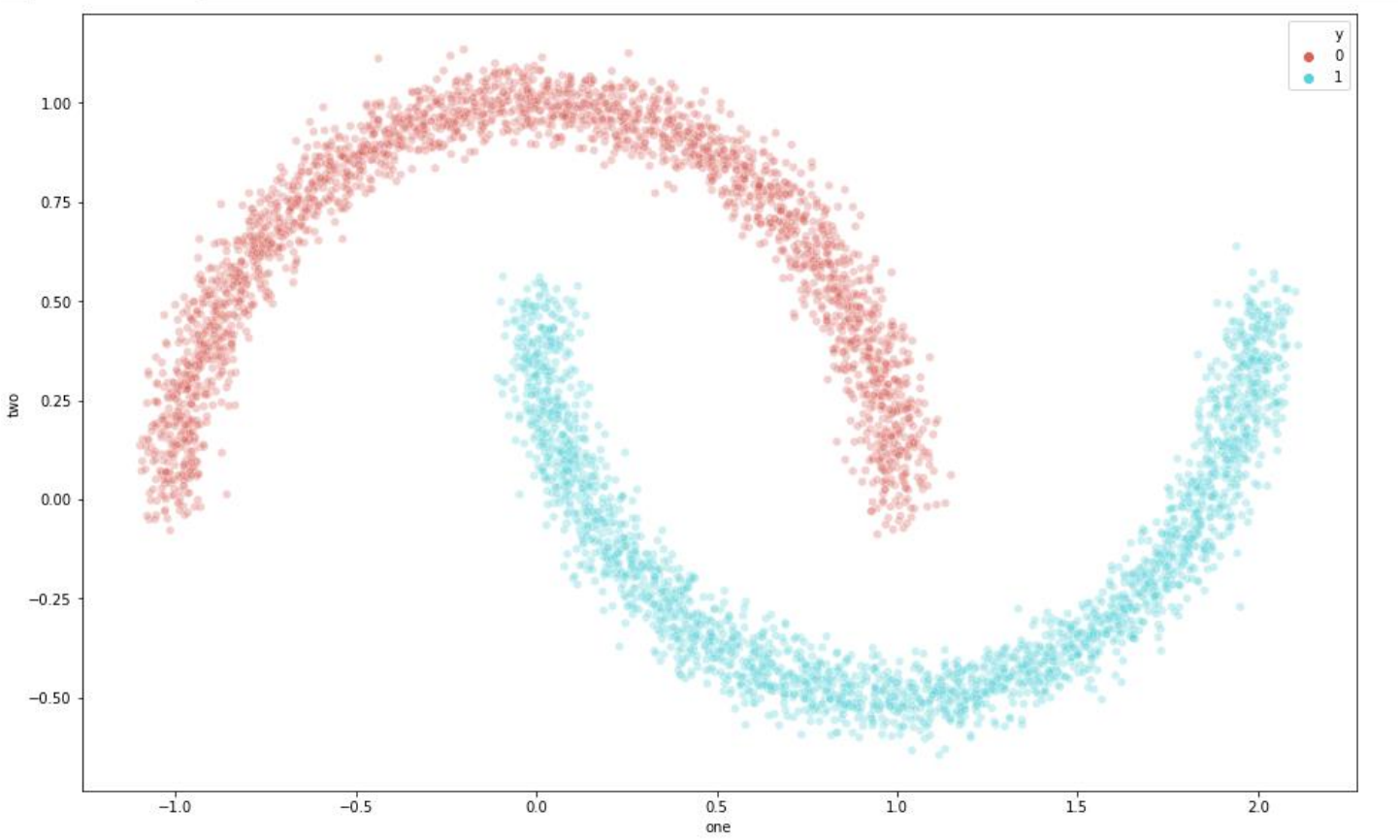
Fold-Number	Training MSE	Validation MSE
1	3.774039336741574	9.880113
2	5.333943966901227	3.024387
3	4.638637144821676	5.8376906899
4	5.114791452508	3.754576202268
5	5.04191978	4.035545
Mean of all errors	4.78066633	5.306462

Same results as in 1(b) and (c)

Details of Question 2

- **Dataset_1 dimensions – 5000 samples, 2 classes**
- **5-folds indexes**
- **[[0, 999], [1000, 1999], [2000, 2999], [3000, 3999], [4000, 4999]]**

2(a) Visualization of the 'dataset_1' in the form of scatter plot



2(c) Table representing train accuracy for each fold considered as validation set [Lr=0.5 gives optimal values]

Lr = 0.1 Table-III

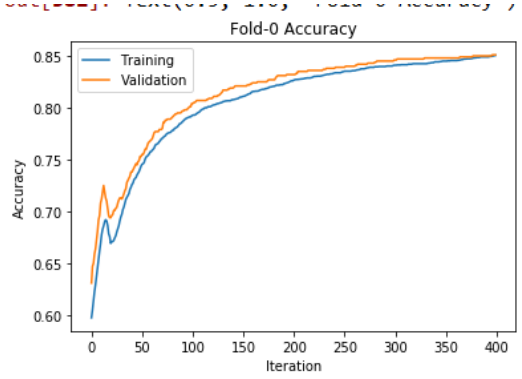
Fold Number	Train accuracy	Validation accuracy
1	0.8505	0.851
2	0.84925	0.844
3	0.8505	0.841
4	0.846	0.859
5	0.849	0.844

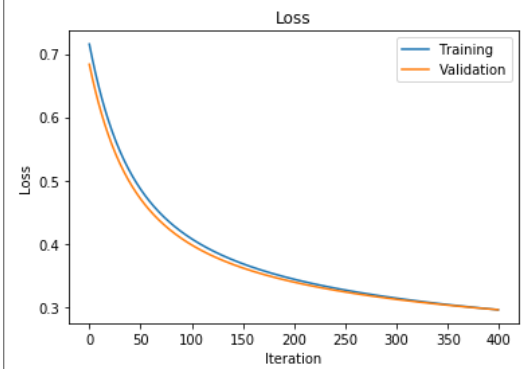
Table-IV

Fold-Number	Training MSE	Validation MSE
1	0.090	0.091
2	0.090	0.095
3	0.091	0.090
4	0.092	0.087
5	0.091	0.093
Mean of all errors	0.091	0.091

Plotting the curves-

Fold- 0

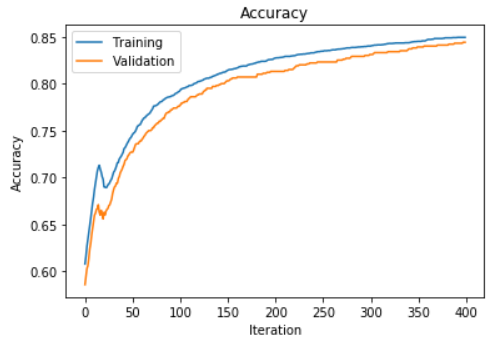


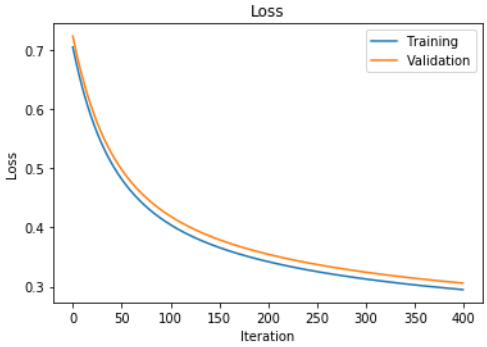


Fold-1

Fold- 1

Fold- 1

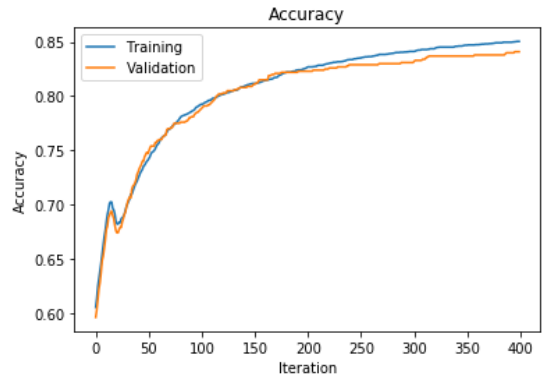


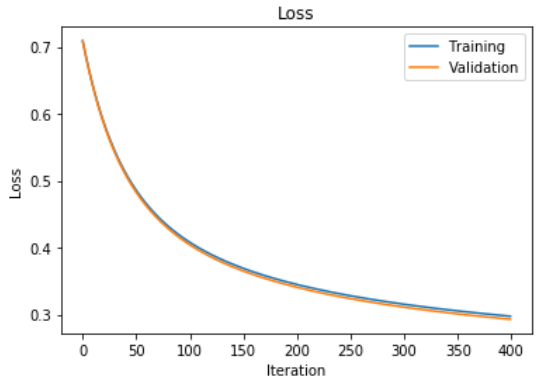


Fold-2

Fold- 2

Fold- 2

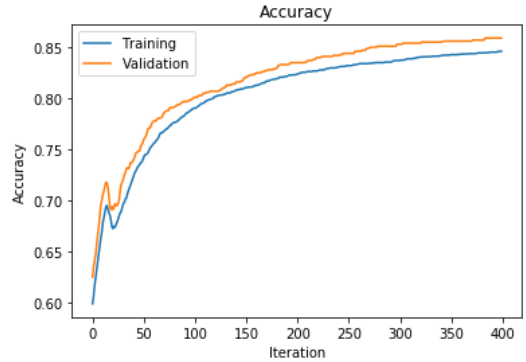


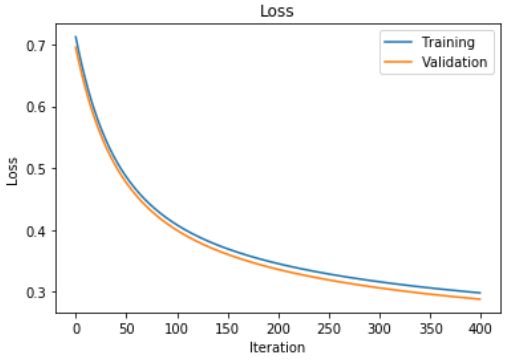


Fold-3

Fold- 3

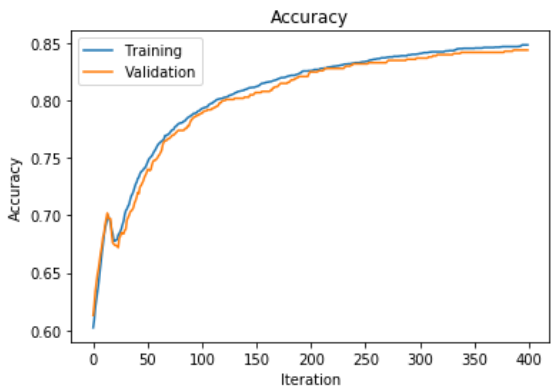
Fold- 3



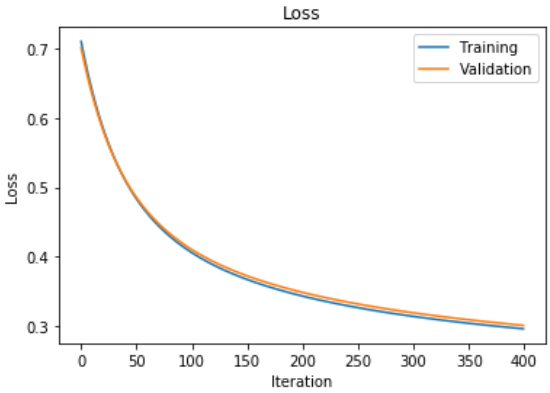


Fold—4

Fold- 4



Fold- 4



Lr=0.5

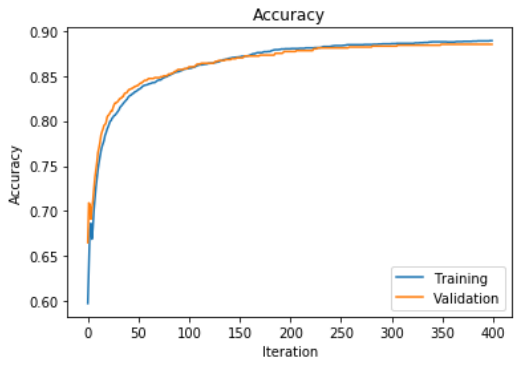
Table-III

Fold Number	Train accuracy	Validation accuracy
1	0.889	0.885
2	0.886	0.887
3	0.887	0.889
4	0.887	0.892
5	0.888	0.885

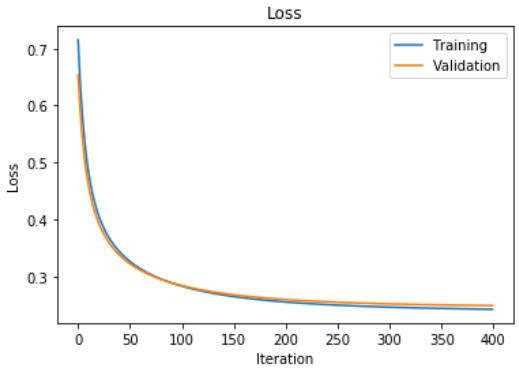
Table-IV

Fold-Number	Training MSE	Validation MSE
1	0.07	0.07
2	0.07	0.08
3	0.07	0.07
4	0.07	0.07
5	0.07	0.07
Mean of all errors	0.07	0.07

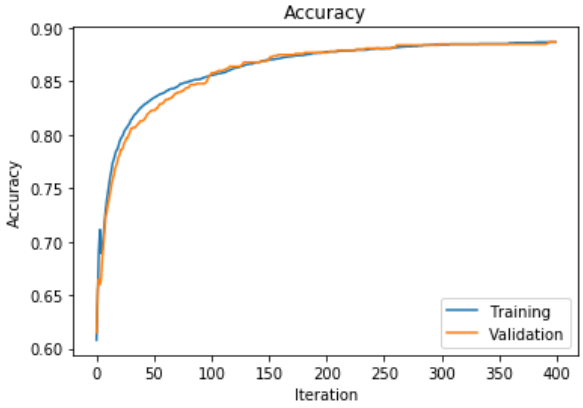
Fold- 0



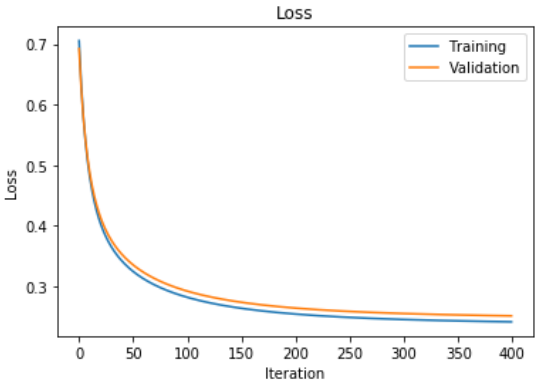
Fold- 0



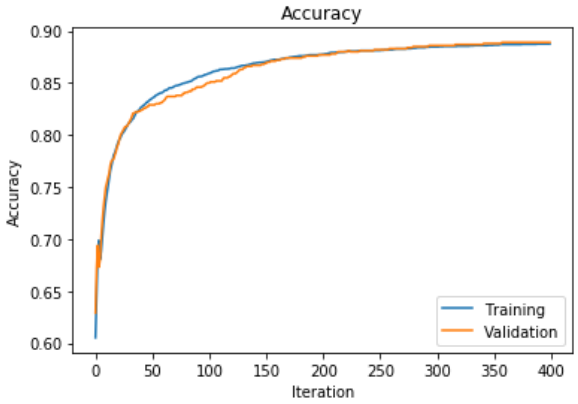
Fold- 1



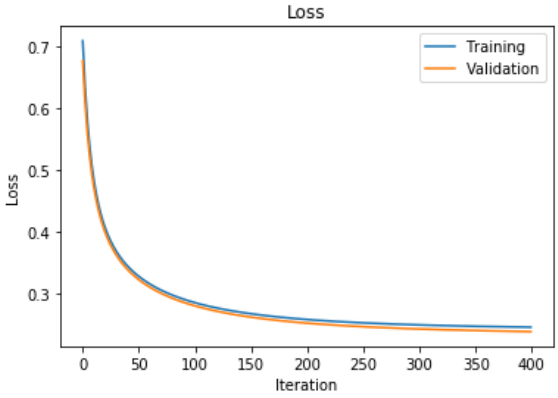
Fold- 1



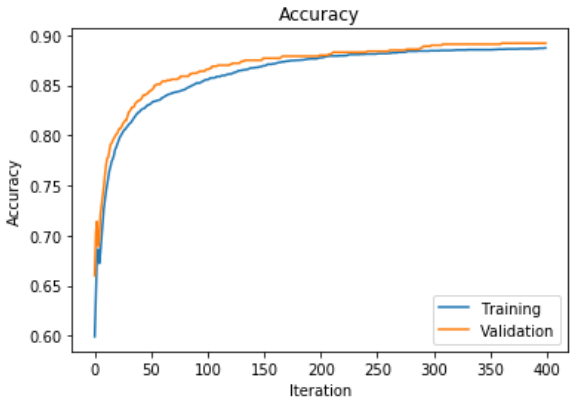
Fold- 2



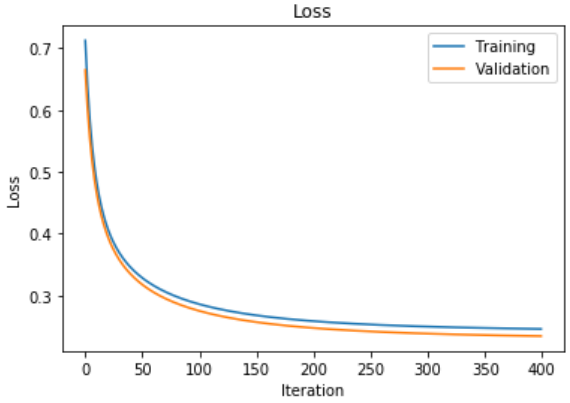
Fold- 2



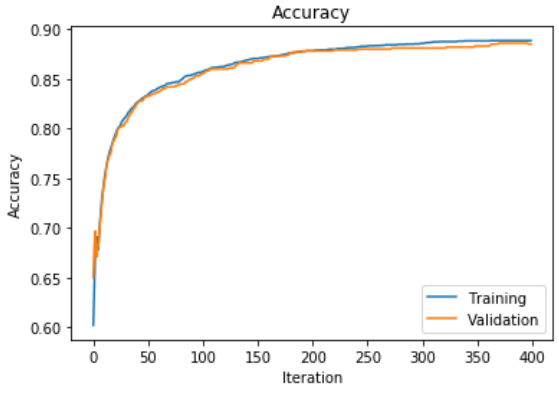
Fold- 3



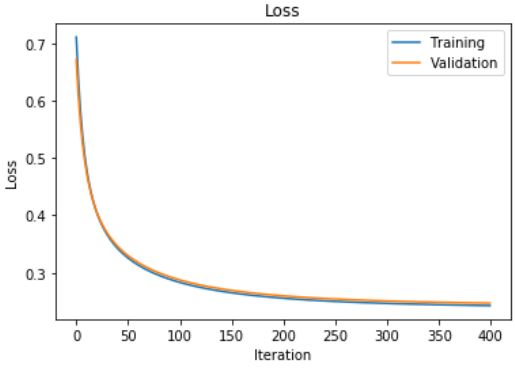
Fold- 3



Fold- 4



Fold- 4



2(d) L2-regularization

With the optimal value of **λ**, repeat the tables and curves of part (c) above.

Table-V

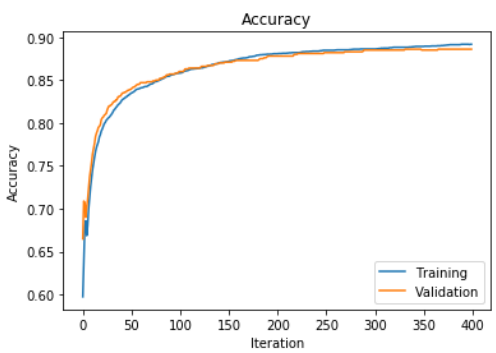
Fold Number	Train accuracy	Validation accuracy	Value of λ
1	0.892	0.886	0.001,0.5
2	0.88875	0.887	0.001,0.5
3	0.88825	0.89	0.001,0.5
4	0.8875	0.894	0.001,0.5
5	0.8895	0.886	0.001,0.5

Table-VI

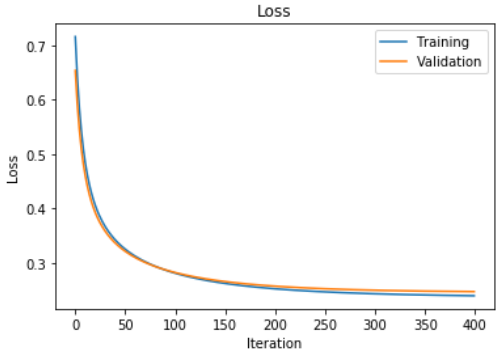
Fold-Number	Training MSE	Validation MSE
1	0.07	0.08
2	0.07	0.08
3	0.07	0.07
4	0.07	0.07
5	0.07	0.07
Mean of all errors	0.07	0.07

Plots

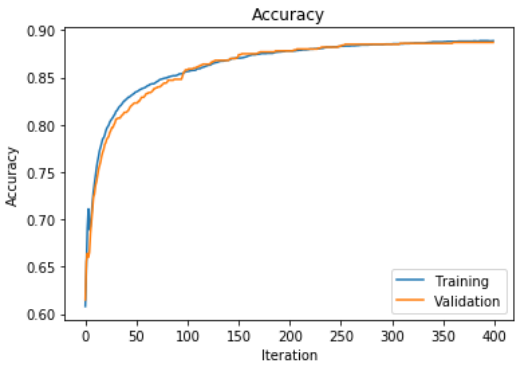
Fold- 0



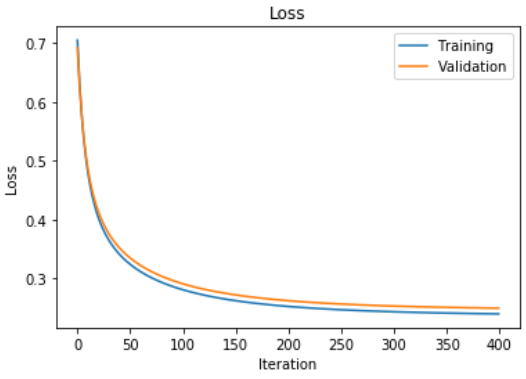
Fold- 0



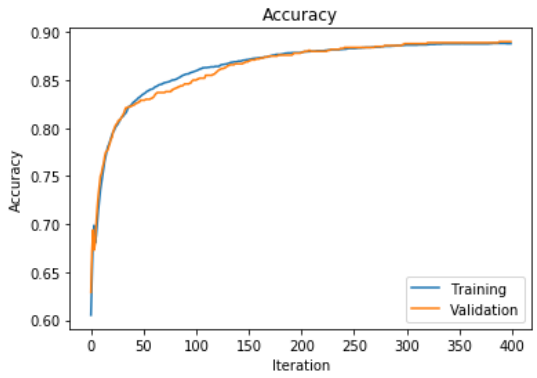
Fold- 1

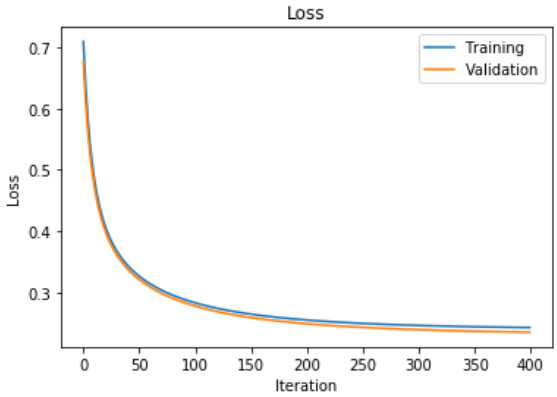


Fold- 1

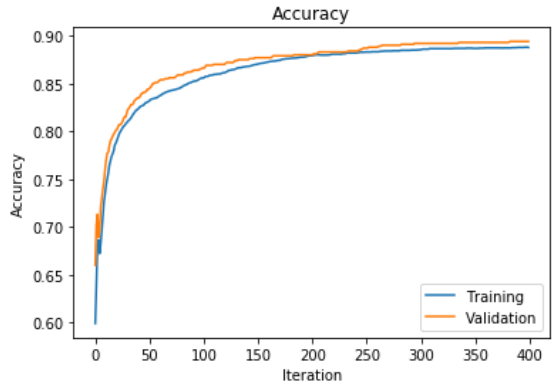


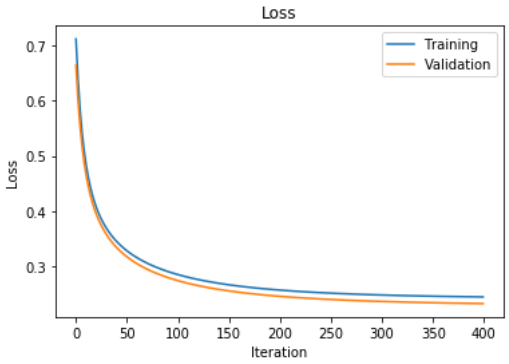
Fold- 2



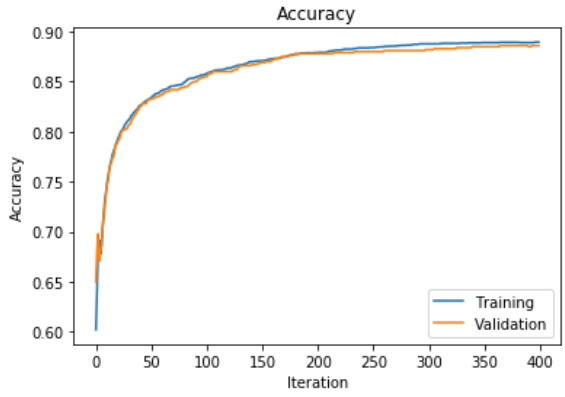


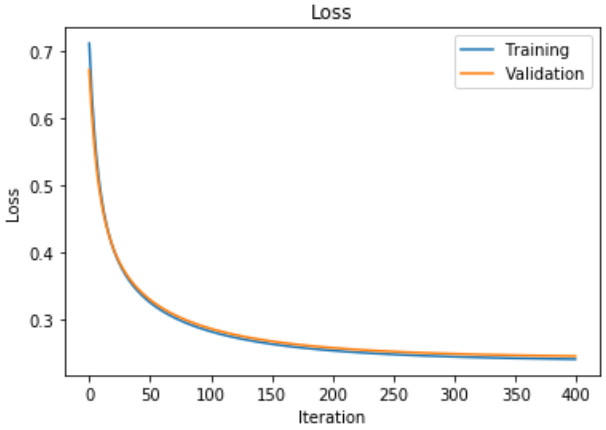
Fold- 3





Fold- 4





Observation- Validation accuracy increases by almost 1% in maximum cases as opposed to the 2(c) observations of validation score.

2(e) Repeat with sklearns' logistic regression part c and d

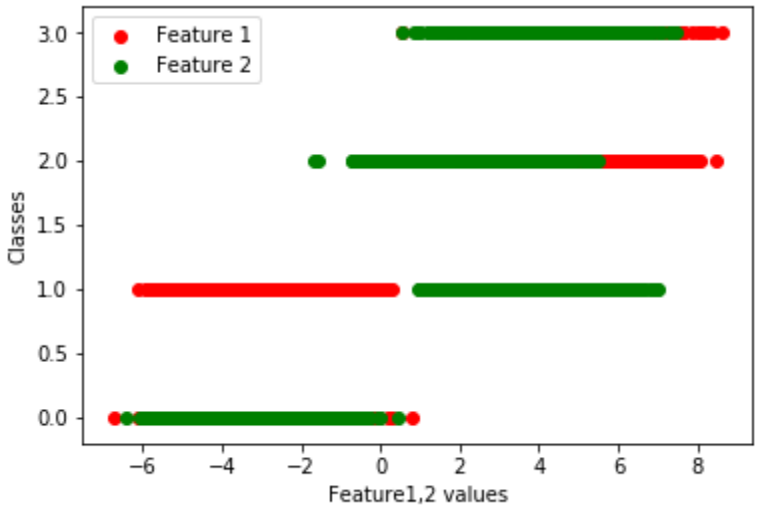
Table-VII

Fold Number	Train accuracy	Validation accuracy
1	0.892	0.887
2	0.89025	0.89
3	0.8895	0.893
4	0.889	0.893
5	0.892	0.888

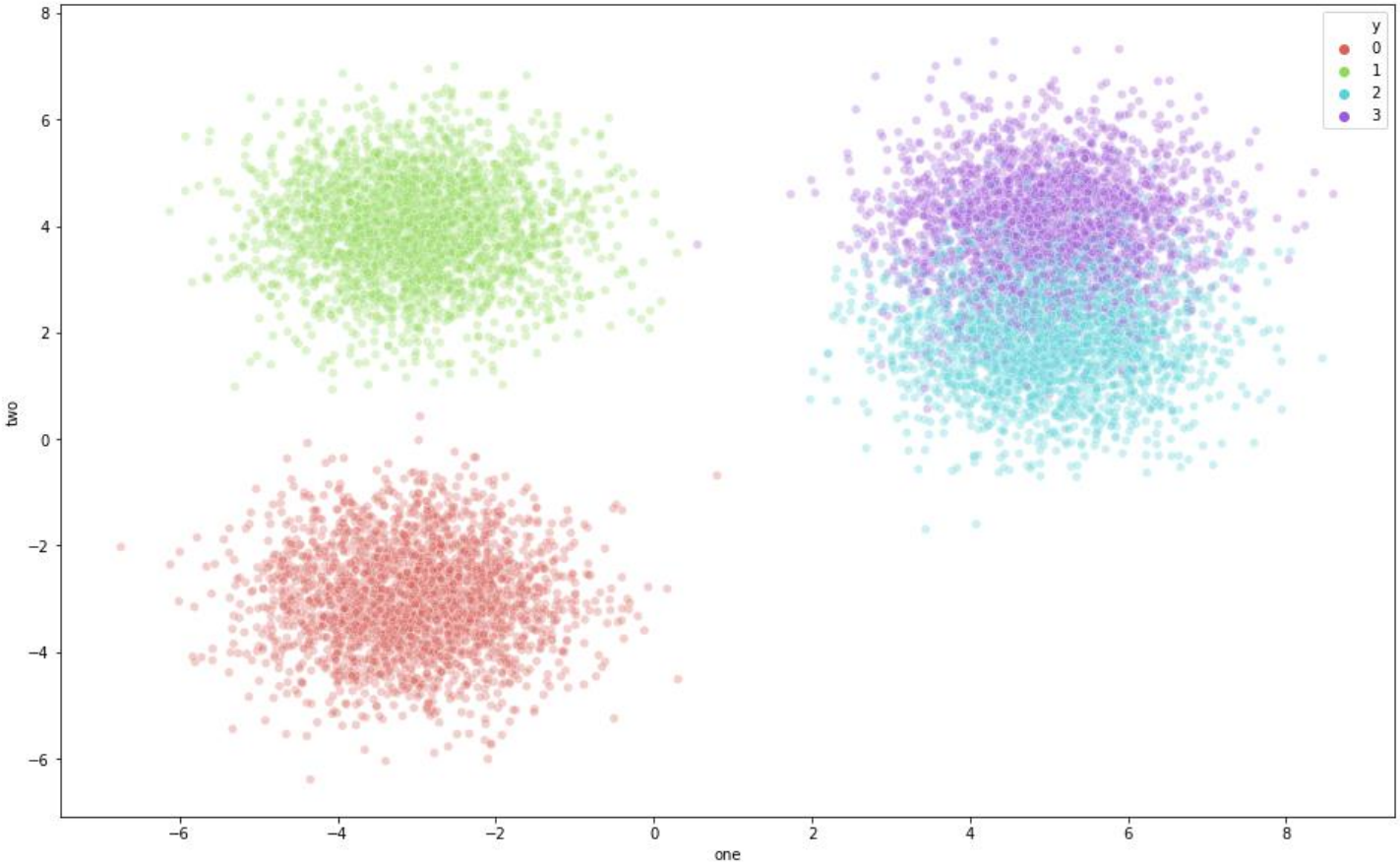
Observation: statistics are almost similar to observations made in previous sections, as we have performed grid search for optimal values in our self-defined fit functions.

Details of Question 3

- Dataset- dataset_2.mat [rows-10000, columns-2, classes-4]
- Fold-indexes: [[0, 1999], [2000, 3999], [4000, 5999], [6000, 7999], [8000, 9999]]



3(a) Visualization of the 'dataset_2' in the form of scatter plot



3(b) One-vs-One (OVO) **Table-VIII**

Fold Number	**Validation accuracy**	**Value of** λ, Lr
1	0.888	0.001,0.5
2	0.901	0.001,0.5
3	0.884	0.001,0.5
4	0.889	0.001,0.5
5	0.889	0.001,0.5

Class-wise accuracy for each fold : **Table-IX**

Fold Number	**Class-1 accuracy**	**Class-2 accuracy**	**Class-3 accuracy**	**Class-4 accuracy**
1	0.9516	1.0	0.772	0.8430
2	0.93032	1.0	0.7466	0.873
3	0.938461	1.0	0.7389	0.8605
4	0.9552	1.0	0.7587	0.8488
5	0.9302	1.0	0.756302	0.86590

3(c) One –vs- Rest(OVR) Table-X

Fold Number	**Validation accuracy**	**Value of** λ, Lr
1	0.9215	0.001,0.5
2	0.913	0.001,0.5
3	0.9285	0.001,0.5
4	0.92	0.001,0.5
5	0.9185	0.001,0.5

Class-wise accuracy for each fold : **Table-XI**

Fold Number	**Class-1 accuracy**	**Class-2 accuracy**	**Class-3 accuracy**	**Class-4 accuracy**
1	1.0	1.0	0.815	0.860
2	1.0	1.0	0.800	0.856
3	1.0	1.0	0.838	0.866
4	1.0	1.0	0.851	0.823
5	1.0	1.0	0.818	0.876

3(d) Repeat above results using sklearn' logistic regression- **Table-XII**

Fold Number	**Accuracy(One vs One)**	**Accuracy(One vs Rest)**
1	0.9215	0.921
2	0.9275	0.927
3	0.9365	0.935
4	0.921	0.9195
5	0.9185	0.9185

Observation: Accuracies are almost same in case of OVR, but in OVO it differs by almost 3%. It might because of taking another class as reference class. In my case, I have run the fit by taking class-4 as the reference class.