- Multi layer with 1 hidden layer and tanh as the activation function

configuration:
  - 20, hidden layers
  - early_stopping True
  - max_iter 200
  - activation tanh
score = 0.9164

	precision	recall	f1-score	support
0	0.94	0.98	0.96	460
1	0.96	0.98	0.97	571
2	0.90	0.92	0.91	530
3	0.89	0.91	0.90	500
4	0.90	0.94	0.92	500
5	0.93	0.84	0.89	456
6	0.93	0.93	0.93	462
7	0.93	0.88	0.91	512
8	0.89	0.89	0.89	489
9	0.90	0.88	0.89	520
avg / total	0.92	0.92	0.92	5000

configuration:
  - 50, hidden layers
  - early_stopping True
  - max_iter 200
  - activation tanh
score = 0.9368

	precision	recall	f1-score	support
0	0.94	0.98	0.96	460
1	0.96	0.98	0.97	571
2	0.93	0.95	0.94	530
3	0.92	0.94	0.93	500
4	0.93	0.94	0.94	500
5	0.93	0.91	0.92	456
6	0.95	0.94	0.94	462
7	0.95	0.90	0.92	512
8	0.94	0.91	0.93	489
9	0.91	0.91	0.91	520

avg / total 0.94 0.94 0.94 5000

configuration:
- 150, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.9446

	precision	recall	f1-score	support
0	0.95	0.98	0.97	460
1	0.97	0.98	0.97	571
2	0.95	0.95	0.95	530
3	0.92	0.96	0.94	500
4	0.91	0.96	0.93	500
5	0.96	0.93	0.94	456
6	0.97	0.93	0.95	462
7	0.94	0.93	0.93	512
8	0.94	0.93	0.93	489
9	0.95	0.90	0.92	520
avg / total	0.95	0.94	0.94	5000

configuration:
- 300, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.9504

	precision	recall	f1-score	support
0	0.96	0.98	0.97	460
1	0.97	0.98	0.98	571
2	0.94	0.96	0.95	530
3	0.94	0.97	0.95	500
4	0.93	0.97	0.95	500
5	0.96	0.93	0.95	456
6	0.96	0.94	0.95	462
7	0.95	0.91	0.93	512
8	0.94	0.94	0.94	489
9	0.96	0.91	0.93	520
avg / total	0.95	0.95	0.95	5000

configuration:
- 400, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.9482

	precision	recall	f1-score	support
0	0.95	0.97	0.96	460
1	0.97	0.98	0.97	571
2	0.95	0.95	0.95	530
3	0.95	0.94	0.95	500
4	0.93	0.97	0.95	500
5	0.94	0.95	0.94	456
6	0.96	0.92	0.94	462
7	0.94	0.95	0.94	512
8	0.93	0.95	0.94	489
9	0.97	0.90	0.93	520
avg / total	0.95	0.95	0.95	5000

- Multi layer with 2 hidden layers and tanh as the activation function

fixing the number of hidden nodes in the first layer to 300 and increasing number of hidden nodes in sec layer starting from 40 till 300

configuration:
- 300,40, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.949

configuration:
- 300,60, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.9532

configuration:

- 300,100, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.9526

configuration:
- 300,200, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.948

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.954

	precision	recall	f1-score	support
0	0.96	0.99	0.97	460
1	0.96	0.98	0.97	571
2	0.96	0.95	0.95	530
3	0.94	0.96	0.95	500
4	0.94	0.97	0.95	500
5	0.95	0.95	0.95	456
6	0.96	0.95	0.95	462
7	0.94	0.94	0.94	512
8	0.96	0.95	0.96	489
9	0.96	0.91	0.93	520
avg / total	0.95	0.95	0.95	5000

- Multi layer with 3 hidden layers and tanh as the activation function

configuration:
- 100,50,25 hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.948

	precision	recall	f1-score	support
0	0.96	0.98	0.97	460
1	0.97	0.98	0.98	571
2	0.95	0.95	0.95	530
3	0.91	0.97	0.94	500
4	0.95	0.95	0.95	500
5	0.96	0.92	0.94	456
6	0.95	0.95	0.95	462
7	0.94	0.92	0.93	512
8	0.94	0.94	0.94	489
9	0.94	0.92	0.93	520
avg / total	0.95	0.95	0.95	5000

configuration:
- 100,100,100 hidden layers
- early_stopping True
- max_iter 200
- activation tanh

score = 0.9438

	precision	recall	f1-score	support
0	0.95	0.98	0.96	460
1	0.97	0.98	0.98	571
2	0.94	0.95	0.95	530
3	0.91	0.97	0.94	500
4	0.95	0.94	0.94	500
5	0.95	0.91	0.93	456
6	0.95	0.93	0.94	462
7	0.94	0.94	0.94	512
8	0.93	0.94	0.93	489
9	0.95	0.90	0.92	520
avg / total	0.94	0.94	0.94	5000

CONCLUSION

- its enough to have two hidden layers in fact increasing the number of hidden layers above that doesn't make any difference in the final error

- the best result was with accuracy = 95.4% seen when using 2 hidden layers with sizes 300 hidden nodes in the first layer and 300 hidden nodes in the second layer

Now we will try changing the activation function in the best configuration from the previous results and see how it will affect the accuracy:

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation identity f(x) = x

score = 0.8926

	precision	recall	f1-score	support
0	0.93	0.96	0.95	460
1	0.96	0.97	0.97	571
2	0.88	0.91	0.89	530
3	0.91	0.80	0.86	500
4	0.88	0.90	0.89	500
5	0.80	0.86	0.83	456
6	0.93	0.91	0.92	462
7	0.91	0.87	0.89	512
8	0.87	0.87	0.87	489
9	0.86	0.87	0.86	520
avg / total	0.89	0.89	0.89	5000

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation logistic f(x) = 1 / (1 + exp(-x))

score = 0.9424

	precision	recall	f1-score	support
0	0.96	0.97	0.97	460
1	0.96	0.98	0.97	571
2	0.95	0.94	0.94	530
3	0.91	0.95	0.93	500
4	0.95	0.94	0.94	500
5	0.91	0.94	0.92	456
6	0.95	0.94	0.95	462
7	0.96	0.91	0.93	512
8	0.95	0.93	0.94	489
9	0.93	0.91	0.92	520

avg / total 0.94 0.94 0.94 5000

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)

score = 0.964

	precision	recall	f1-score	support
0	0.96	0.99	0.97	460
1	0.96	0.99	0.98	571
2	0.98	0.95	0.96	530
3	0.98	0.96	0.97	500
4	0.96	0.97	0.97	500
5	0.95	0.96	0.96	456
6	0.97	0.96	0.97	462
7	0.95	0.96	0.96	512
8	0.96	0.95	0.96	489
9	0.97	0.94	0.95	520

avg / total 0.96 0.96 0.96 5000

CONCLUSION:

- the best accuracy was 96.4% using the relu function f(x) = max(0, x) as the activation function

in all the previous trainings the default learning rate 'constant' with value = 0.001 was used
Now we will try changing this learning rate in the configuration that gave us lowest error from the previous results and see how changing the learning rate will affect the accuracy:

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- learning_rate 0.01

score = 0.9466

	precision	recall	f1-score	support
0	0.94	0.99	0.96	460
1	0.98	0.98	0.98	571

2	0.94	0.96	0.95	530
3	0.97	0.94	0.95	500
4	0.97	0.90	0.94	500
5	0.95	0.95	0.95	456
6	0.96	0.95	0.95	462
7	0.95	0.95	0.95	512
8	0.88	0.95	0.91	489
9	0.93	0.91	0.92	520
avg / total	0.95	0.95	0.95	5000

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- learning_rate 0.1

score = 0.2882

	precision	recall	f1-score	support
0	0.00	0.00	0.00	460
1	0.17	0.95	0.29	571
2	0.00	0.00	0.00	530
3	0.14	0.05	0.07	500
4	0.51	0.49	0.50	500
5	0.00	0.00	0.00	456
6	0.00	0.00	0.00	462
7	0.63	0.75	0.68	512
8	0.00	0.00	0.00	489
9	0.45	0.47	0.46	520
avg / total	0.20	0.29	0.21	5000

- as we saw in the results increasing the learning_rate decreased the accuracy

Now we will try changing the momentum and see how it will affect accuracy

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- solver sgd

- momentum 0.5

score = 0.882

	precision	recall	f1-score	support
0	0.92	0.97	0.94	460
1	0.93	0.98	0.95	571
2	0.90	0.85	0.88	530
3	0.84	0.87	0.85	500
4	0.86	0.88	0.87	500
5	0.85	0.83	0.84	456
6	0.89	0.90	0.90	462
7	0.90	0.84	0.87	512
8	0.86	0.84	0.85	489
9	0.85	0.85	0.85	520
avg / total	0.88	0.88	0.88	5000

configuration:

- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- solver sgd
- momentum 0.1

score = 0.8542

	precision	recall	f1-score	support
0	0.90	0.97	0.93	460
1	0.89	0.96	0.93	571
2	0.89	0.82	0.85	530
3	0.79	0.85	0.82	500
4	0.85	0.86	0.86	500
5	0.81	0.76	0.79	456
6	0.87	0.86	0.86	462
7	0.88	0.82	0.85	512
8	0.84	0.80	0.82	489
9	0.82	0.82	0.82	520
avg / total	0.85	0.85	0.85	5000

As we saw from the previous results decreasing the momentum decreased the accuracy and increased the error

The batch size in the previous tests was min(200, n_samples) and since the number of samples is 20000 the batch size was 200 so the training style was mini batches

Now we will try changing the training style to batch by setting the batch size to the number of training samples = 20000 and see how it will affect the accuracy
configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- batch_size 20000 (batch)

score = 0.93959999999999999

	precision	recall	f1-score	support
0	0.96	0.98	0.97	460
1	0.96	0.98	0.97	571
2	0.94	0.94	0.94	530
3	0.90	0.94	0.92	500
4	0.93	0.95	0.94	500
5	0.94	0.92	0.93	456
6	0.95	0.94	0.94	462
7	0.94	0.92	0.93	512
8	0.94	0.92	0.93	489
9	0.93	0.91	0.92	520
avg / total	0.94	0.94	0.94	5000

batch training style produced (93.9%) less accuracy than the mini batches training style

configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- batch_size 1 (batch)

score = 0.114312

	precision	recall	f1-score	support
0	0.00	0.00	0.00	460
1	0.12	0.11	0.12	571
2	0.14	0.14	0.13	530
3	0.10	0.10	0.14	500

4	0.13	0.13	0.11	500
5	0.11	0.11	0.11	456
6	0.11	0.11	0.11	462
7	0.12	0.12	0.12	512
8	0.11	0.13	0.12	489
9	0.12	0.14	0.13	520
avg / total	0.11	0.11	0.11	5000

sequential training style produced (11.43%) much less accuracy than the mini batches and batches training style

now trying something in between 200 and 20000 to test mini batches again but with higher batch size
configuration:
- 300,300, hidden layers
- early_stopping True
- max_iter 200
- activation relu f(x) = max(0, x)
- batch_size 500 (batch)

score = 0.9604

	precision	recall	f1-score	support
0	0.96	0.99	0.98	460
1	0.97	0.98	0.98	571
2	0.96	0.97	0.96	530
3	0.95	0.98	0.97	500
4	0.96	0.96	0.96	500
5	0.96	0.96	0.96	456
6	0.98	0.94	0.96	462
7	0.96	0.95	0.95	512
8	0.94	0.96	0.95	489
9	0.96	0.93	0.94	520
avg / total	0.96	0.96	0.96	5000

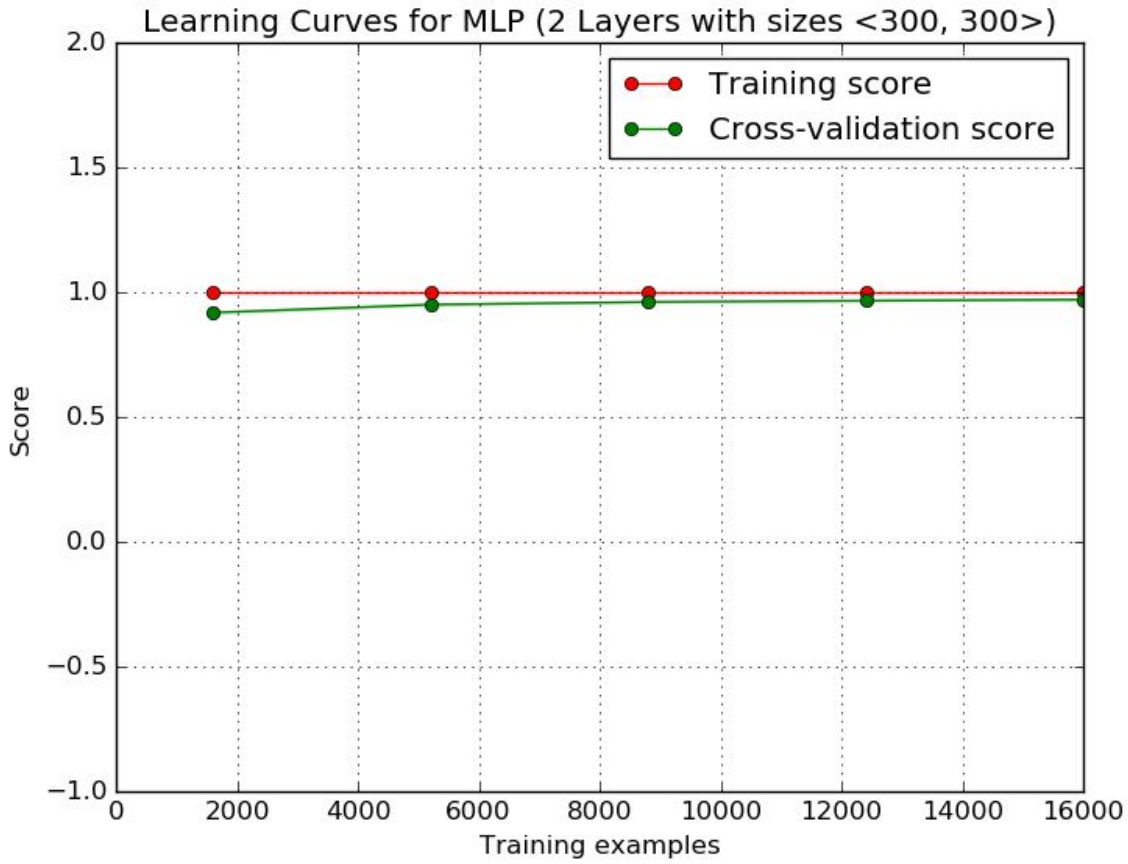
In all the previous tests the early stopping technique was used we will now disable it and see how the accuracy will be affected
configuration:
- 300,300, hidden layers
- early_stopping False
- max_iter 200
- activation relu f(x) = max(0, x)

score = 0.9658

	precision	recall	f1-score	support
0	0.97	0.98	0.97	460
1	0.97	0.99	0.98	571
2	0.95	0.97	0.96	530
3	0.95	0.98	0.97	500
4	0.97	0.96	0.96	500
5	0.98	0.96	0.97	456
6	0.97	0.96	0.96	462
7	0.97	0.95	0.96	512
8	0.96	0.95	0.96	489
9	0.97	0.96	0.96	520
avg / total	0.97	0.97	0.97	5000

Without the early stopping the accuracy of the MLP increased to 96.58% higher than 96.4% with early stopping

This is the plotting for the validation and training errors when using MLP with 2 layers <300, 300> and without early stopping



As we can see the error on of the validation starts to increase at some point while the learning error isn't changed at this moment its better to stop the training

==========

- SVM with linear kernel function and changing the C value

configuration:
  - kernel linear
  - C = 1
  - max_iter 200

score = 0.8854

	precision	recall	f1-score	support
0	0.92	0.96	0.94	460
1	0.94	0.98	0.96	571
2	0.90	0.86	0.88	530
3	0.85	0.87	0.86	500
4	0.88	0.91	0.90	500

5	0.83	0.82	0.82	456
6	0.92	0.91	0.91	462
7	0.89	0.88	0.88	512
8	0.83	0.80	0.82	489
9	0.88	0.85	0.86	520
avg / total	0.88	0.89	0.88	5000

configuration:
- kernel linear
- C = 0.5
- max_iter 200

score = 0.886

	precision	recall	f1-score	support
0	0.93	0.97	0.95	460
1	0.94	0.98	0.96	571
2	0.90	0.86	0.88	530
3	0.86	0.88	0.87	500
4	0.88	0.91	0.89	500
5	0.83	0.82	0.83	456
6	0.92	0.91	0.91	462
7	0.89	0.88	0.88	512
8	0.83	0.80	0.82	489
9	0.87	0.85	0.86	520
avg / total	0.89	0.89	0.89	5000

configuration:
- kernel linear
- C = 0.1
- max_iter 200

score = 0.8898

	precision	recall	f1-score	support
0	0.91	0.98	0.94	460
1	0.94	0.98	0.96	571
2	0.91	0.88	0.89	530
3	0.88	0.88	0.88	500
4	0.87	0.90	0.89	500
5	0.84	0.84	0.84	456
6	0.92	0.90	0.91	462
7	0.90	0.88	0.89	512

8	0.85	0.80	0.82	489
9	0.87	0.84	0.85	520
avg / total	0.89	0.89	0.89	5000

configuration:
- kernel linear
- C = 0.01
- max_iter 200

score = 0.883

	precision	recall	f1-score	support
0	0.90	0.98	0.94	460
1	0.92	0.98	0.95	571
2	0.91	0.87	0.89	530
3	0.87	0.88	0.88	500
4	0.85	0.90	0.88	500
5	0.85	0.81	0.83	456
6	0.90	0.91	0.91	462
7	0.88	0.87	0.87	512
8	0.87	0.79	0.83	489
9	0.86	0.83	0.85	520
avg / total	0.88	0.88	0.88	5000

- As we saw the best accuracy was reached when the value of C was set to 0.1

- Now we will change the kernel function and see how it will affect the accuracy

configuration:
- kernel poly
- C = 0.1
- max_iter 200

score = 0.838

	precision	recall	f1-score	support
0	0.72	0.97	0.83	460
1	0.81	0.99	0.89	571
2	0.89	0.82	0.85	530
3	0.79	0.90	0.84	500
4	0.94	0.81	0.87	500
5	0.95	0.61	0.74	456
6	0.78	0.89	0.83	462

7	0.82	0.84	0.83	512
8	0.90	0.75	0.82	489
9	0.91	0.79	0.85	520
avg / total	0.85	0.84	0.84	5000

configuration:
- kernel rbf
- C = 0.1
- max_iter 200

score = 0.8188

	precision	recall	f1-score	support
0	0.85	0.96	0.91	460
1	0.76	0.98	0.85	571
2	0.89	0.77	0.83	530
3	0.74	0.87	0.80	500
4	0.81	0.88	0.84	500
5	0.88	0.55	0.67	456
6	0.80	0.90	0.85	462
7	0.78	0.85	0.81	512
8	0.86	0.68	0.76	489
9	0.91	0.72	0.81	520
avg / total	0.83	0.82	0.81	5000

configuration:
- kernel sigmoid
- C = 0.1
- max_iter 200

score = 0.775

	precision	recall	f1-score	support
0	0.81	0.96	0.88	460
1	0.67	0.98	0.80	571
2	0.89	0.71	0.79	530
3	0.69	0.85	0.76	500
4	0.82	0.85	0.83	500
5	0.87	0.43	0.57	456
6	0.74	0.88	0.81	462
7	0.70	0.84	0.77	512
8	0.88	0.59	0.71	489
9	0.92	0.64	0.75	520

avg / total 0.80 0.78 0.77 5000

As we can see the best accuracy 88.98% was reached when using the linear kernel function with C = 0.1

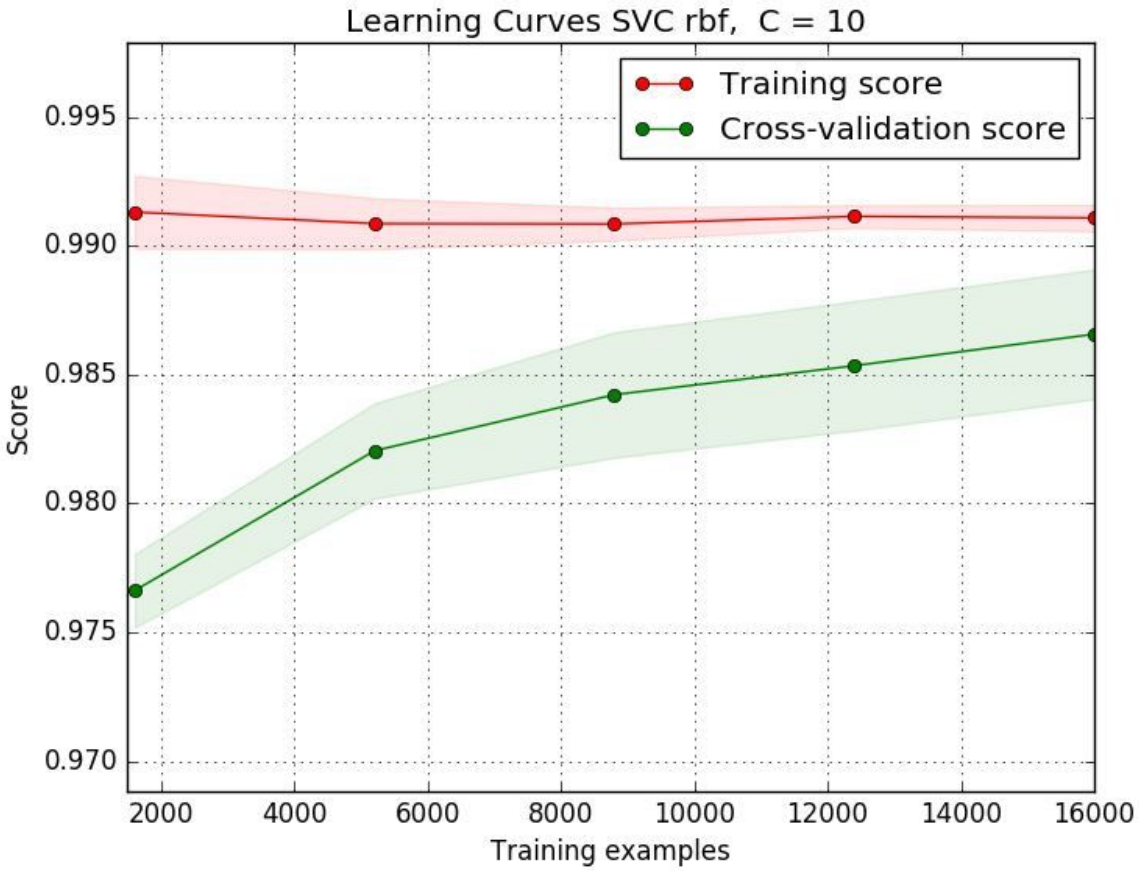
Now since the best kernel was the linear we will try it with a higher value C = 10

configuration:
- kernel rbf
- C = 10
- max_iter 200

score = 0.8768

	precision	recall	f1-score	support
0	0.91	0.94	0.93	460
1	0.94	0.98	0.96	571
2	0.89	0.84	0.86	530
3	0.82	0.86	0.84	500
4	0.89	0.89	0.89	500
5	0.84	0.82	0.83	456
6	0.92	0.90	0.91	462
7	0.89	0.87	0.88	512
8	0.81	0.79	0.80	489
9	0.87	0.85	0.86	520
avg / total	0.88	0.88	0.88	5000

This is the plotting for the validation and training errors when using SVC with rbf kernel and C = 10



==========

As a comparison between the 2 obviously MLP brought better results

MLP With accuracy = 96.58% and error = 3.42%

SVG With accuracy = 88.3% and error = 11.7%