

Training samples: (10000, 784),  
Testing samples: (10000, 784)

=== Kernel=linear, C=0.1 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9061  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9094  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9031  
Completed → mean = 0.9062, std = 0.0032

=== Kernel=linear, C=1 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9061  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9094  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9031  
Completed → mean = 0.9062, std = 0.0032

=== Kernel=linear, C=10 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9061  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9094  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9031  
Completed → mean = 0.9062, std = 0.0032

=== Kernel=poly, C=0.1, degree=2 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9088  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9148  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9118  
Completed → mean = 0.9118, std = 0.0030

=== Kernel=poly, C=0.1, degree=3 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.8857  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.8881  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.8863  
Completed → mean = 0.8867, std = 0.0012

=== Kernel=poly, C=0.1, degree=4 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.8281  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.8299  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.8257  
Completed → mean = 0.8279, std = 0.0021

=== Kernel=poly, C=1, degree=2 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9481  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9523  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9529  
Completed → mean = 0.9511, std = 0.0026

=== Kernel=poly, C=1, degree=3 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9451  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9460  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9469  
Completed → mean = 0.9460, std = 0.0009

=== Kernel=poly, C=1, degree=4 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9178  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9229  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9220  
Completed → mean = 0.9209, std = 0.0027

=== Kernel=poly, C=10, degree=2 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9505  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9607  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9565  
Completed → mean = 0.9559, std = 0.0051

=== Kernel=poly, C=10, degree=3 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9463  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9496  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9508  
Completed → mean = 0.9489, std = 0.0023

=== Kernel=poly, C=10, degree=4 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9349  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9331  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9385  
Completed → mean = 0.9355, std = 0.0027

=== Kernel=rbf, C=0.1, gamma=scale ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9160  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9223  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9169  
Completed → mean = 0.9184, std = 0.0034

=== Kernel=rbf, C=0.1, gamma=0.01 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.1125  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.1125  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.1125  
Completed → mean = 0.1125, std = 0.0000

=== Kernel=rbf, C=0.1, gamma=0.001 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.1125  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.1125  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.1125  
Completed → mean = 0.1125, std = 0.0000

=== Kernel=rbf, C=1, gamma=scale ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9538  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9580  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9571  
Completed → mean = 0.9563, std = 0.0022

=== Kernel=rbf, C=1, gamma=0.01 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.1125  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.1125  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.1125  
Completed → mean = 0.1125, std = 0.0000

=== Kernel=rbf, C=1, gamma=0.001 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.1125  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.1125  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.1125  
Completed → mean = 0.1125, std = 0.0000

=== Kernel=rbf, C=10,  
gamma=scale ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.9592  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.9655  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.9631  
Completed → mean = 0.9626, std =  
0.0032

=== Kernel=rbf, C=10,  
gamma=0.01 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.1125  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.1125  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.1125  
Completed → mean = 0.1125, std =  
0.0000

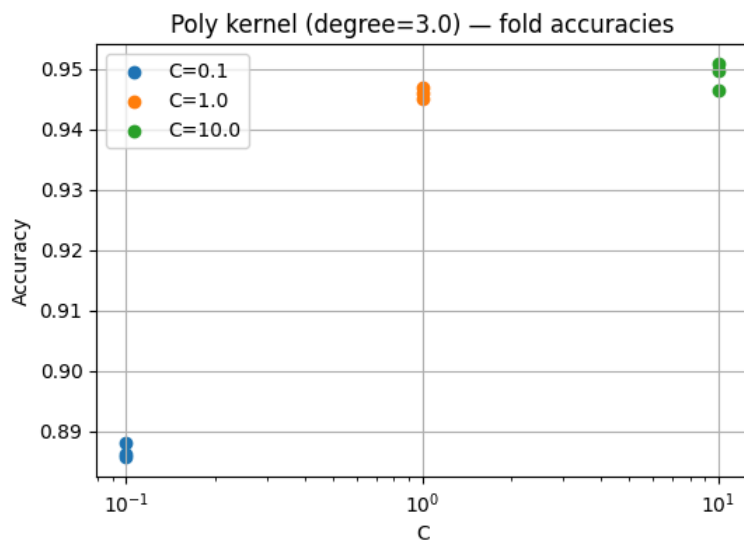
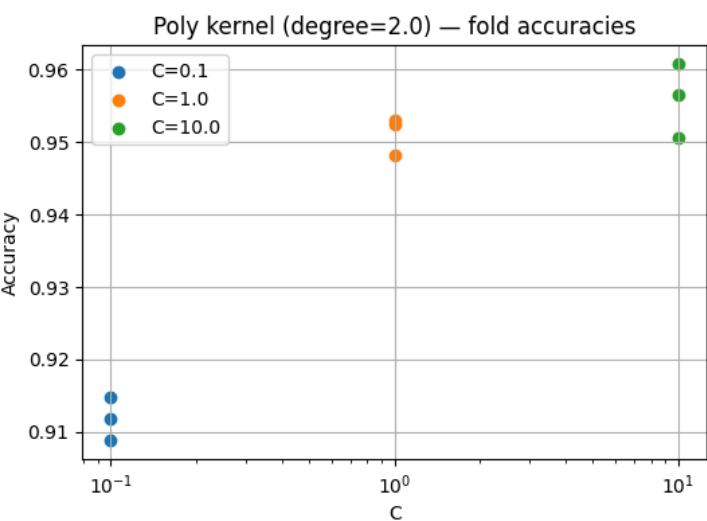
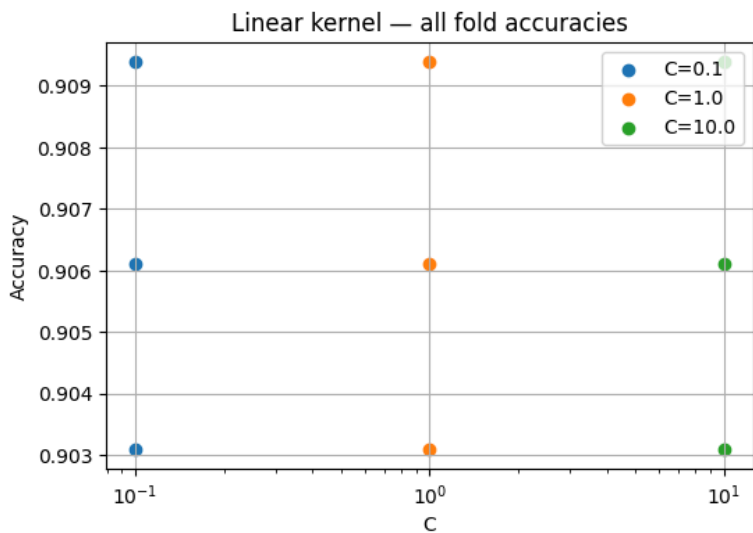
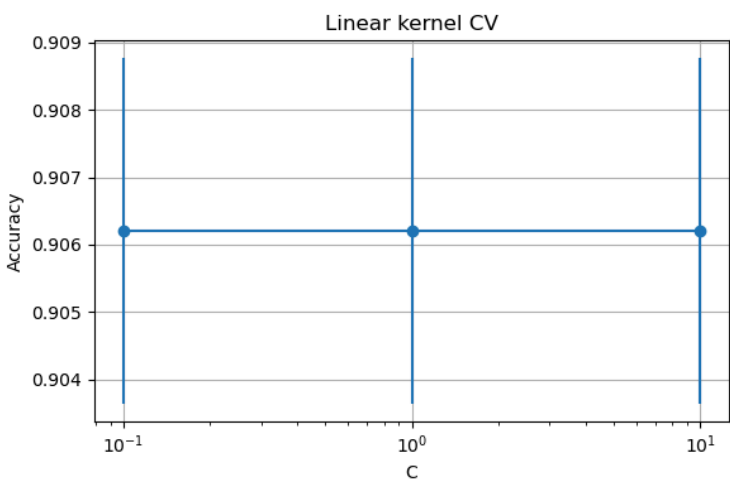
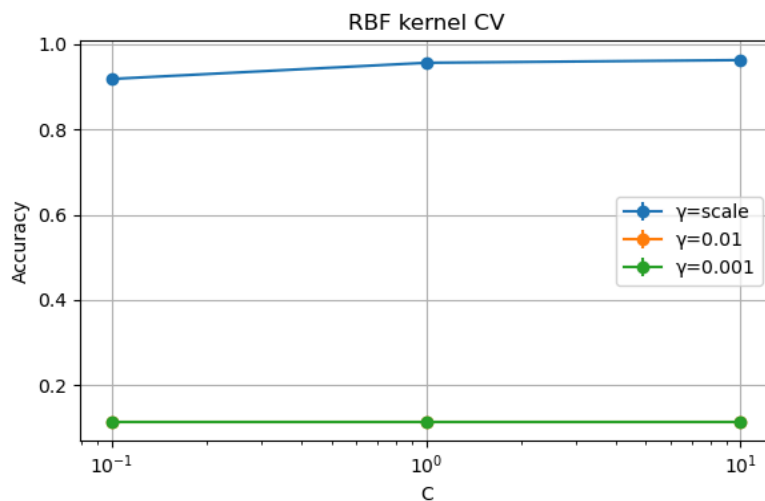
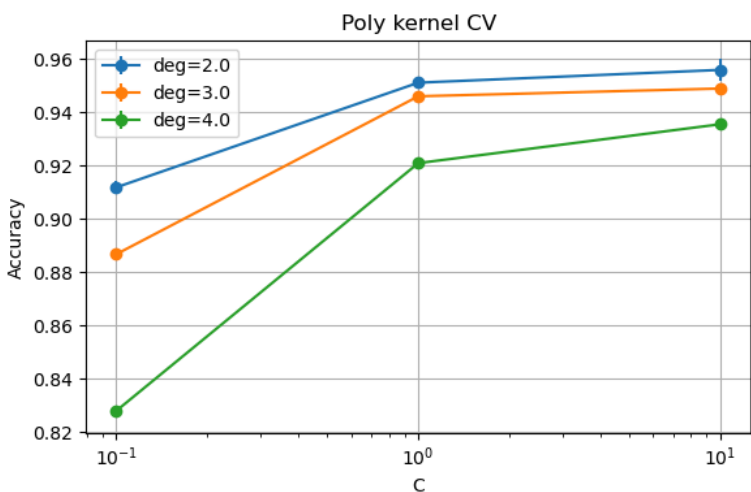
=== Kernel=rbf, C=10,  
gamma=0.001 ===  
Fold 1: train=6666, val=3334  
→ Fold 1 acc = 0.1125  
Fold 2: train=6667, val=3333  
→ Fold 2 acc = 0.1125  
Fold 3: train=6667, val=3333  
→ Fold 3 acc = 0.1125  
Completed → mean = 0.1125, std =  
0.0000

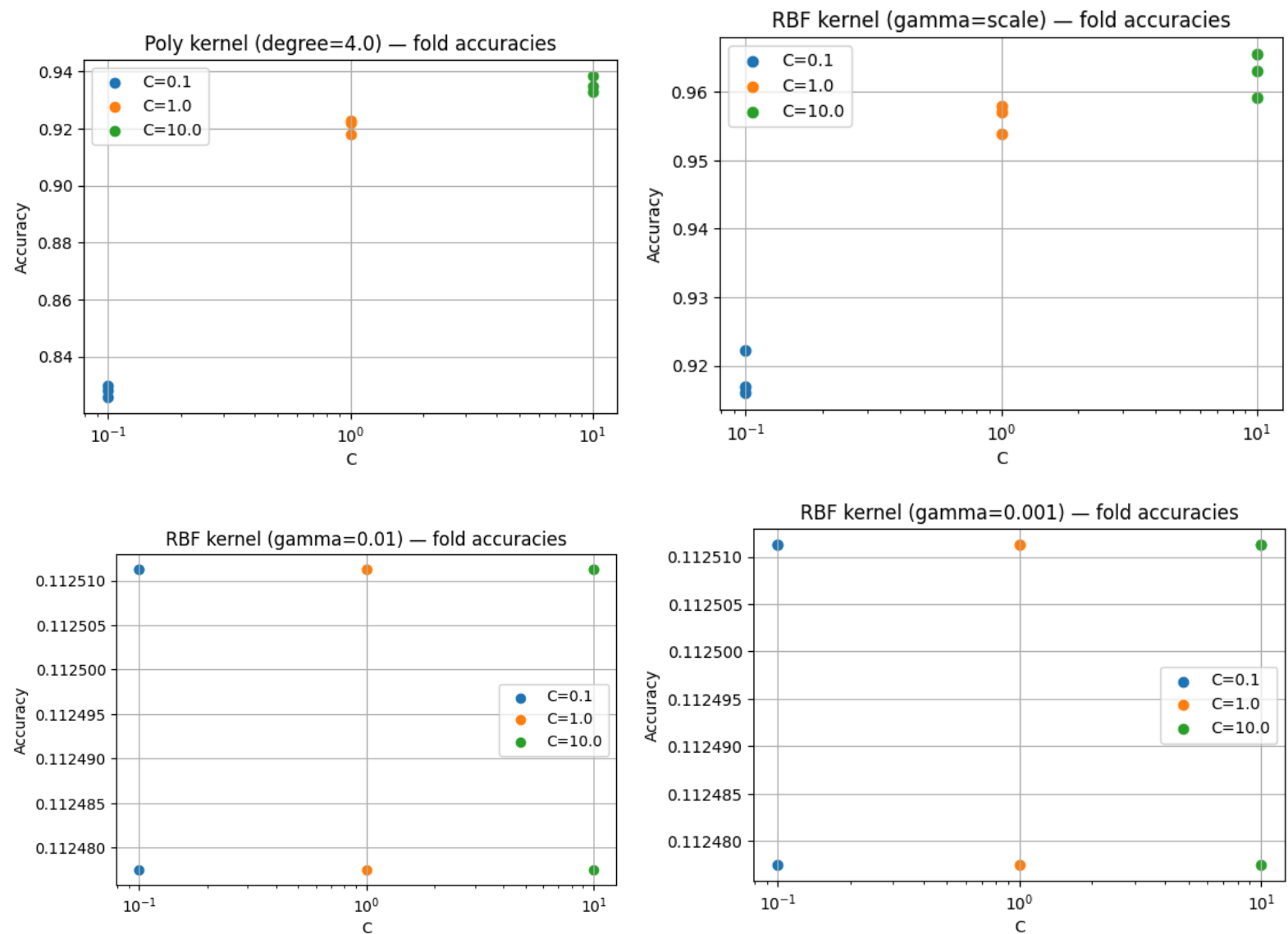
=== All fold results ===  
kernel C degree gamma fold  
accuracy  
linear 0.1 - None 1  
0.906119  
linear 0.1 - None 2  
0.909391  
linear 0.1 - None 3  
0.903090  
linear 1.0 - None 1  
0.906119  
linear 1.0 - None 2  
0.909391  
linear 1.0 - None 3  
0.903090  
linear 10.0 - None 1  
0.906119  
linear 10.0 - None 2  
0.909391

linear 10.0 - None 3  
0.903090  
poly 0.1 2.0 None 1  
0.908818  
poly 0.1 2.0 None 2  
0.914791  
poly 0.1 2.0 None 3  
0.911791  
poly 0.1 3.0 None 1  
0.885723  
poly 0.1 3.0 None 2  
0.888089  
poly 0.1 3.0 None 3  
0.886289  
poly 0.1 4.0 None 1  
0.828134  
poly 0.1 4.0 None 2  
0.829883  
poly 0.1 4.0 None 3  
0.825683  
poly 1.0 2.0 None 1  
0.948110  
poly 1.0 2.0 None 2  
0.952295  
poly 1.0 2.0 None 3  
0.952895  
poly 1.0 3.0 None 1  
0.945111  
poly 1.0 3.0 None 2  
0.945995  
poly 1.0 3.0 None 3  
0.946895  
poly 1.0 4.0 None 1  
0.917816  
poly 1.0 4.0 None 2  
0.922892  
poly 1.0 4.0 None 3  
0.921992  
poly 10.0 2.0 None 1  
0.950510  
poly 10.0 2.0 None 2  
0.960696  
poly 10.0 2.0 None 3  
0.956496  
poly 10.0 3.0 None 1  
0.946311  
poly 10.0 3.0 None 2  
0.949595  
poly 10.0 3.0 None 3  
0.950795  
poly 10.0 4.0 None 1  
0.934913

poly 10.0 4.0 None 2  
0.933093  
poly 10.0 4.0 None 3  
0.938494  
rbf 0.1 - scale 1 0.916017  
rbf 0.1 - scale 2 0.922292  
rbf 0.1 - scale 3 0.916892  
rbf 0.1 - 0.01 1  
0.112478  
rbf 0.1 - 0.01 2  
0.112511  
rbf 0.1 - 0.01 3  
0.112511  
rbf 0.1 - 0.001 1  
0.112478  
rbf 0.1 - 0.001 2  
0.112511  
rbf 0.1 - 0.001 3  
0.112511  
rbf 1.0 - scale 1 0.953809  
rbf 1.0 - scale 2 0.957996  
rbf 1.0 - scale 3 0.957096  
rbf 1.0 - 0.01 1  
0.112478  
rbf 1.0 - 0.01 2  
0.112511  
rbf 1.0 - 0.01 3  
0.112511  
rbf 1.0 - 0.001 1  
0.112478  
rbf 1.0 - 0.001 2  
0.112511  
rbf 1.0 - 0.001 3  
0.112511  
rbf 10.0 - scale 1  
0.959208  
rbf 10.0 - scale 2  
0.965497  
rbf 10.0 - scale 3  
0.963096  
rbf 10.0 - 0.01 1  
0.112478  
rbf 10.0 - 0.01 2  
0.112511  
rbf 10.0 - 0.01 3  
0.112511  
rbf 10.0 - 0.001 1  
0.112478  
rbf 10.0 - 0.001 2  
0.112511  
rbf 10.0 - 0.001 3  
0.112511

param_kernel	param_C	param_degree	param_gamma	mean_test_score	std_test_score
linear	0.1			0.9062	0.002573
linear	1			0.9062	0.002573
linear	10			0.9062	0.002573
poly	0.1	2		0.9118	0.002439
poly	0.1	3		0.8867	0.001009
poly	0.1	4		0.8279	0.001723
poly	1	2		0.9511	0.002128
poly	1	3		0.946	0.000728
poly	1	4		0.9209	0.002211
poly	10	2		0.955901	0.00418
poly	10	3		0.9489	0.001895
poly	10	4		0.9355	0.002243
rbf	0.1		scale	0.9184	0.002775
rbf	0.1		0.01	0.1125	1.59E-05
rbf	0.1		0.001	0.1125	1.59E-05
rbf	1		scale	0.9563	0.001799
rbf	1		0.01	0.1125	1.59E-05
rbf	1		0.001	0.1125	1.59E-05
rbf	10		scale	0.9626	0.002591
rbf	10		0.01	0.1125	1.59E-05
rbf	10		0.001	0.1125	1.59E-05





## Report:

Best CV model: SVC(kernel='rbf', C=1, gamma='scale') Test accuracy: 0.9654

RBF outperformed the linear and poly kernels by about 4 percent points on average, capture the non linear structure of the data

C-Sensitivity: Linear had small gains beyond C = 1, for RBF C = 1 was optimal, if it got too large it tended to overfit

Scale was about in the 0.01–0.001 range. Too small  $\gamma$  underfits, too large overfits.

Degree 2 seems to nearly match the linear data, degree 3–4 degraded too much because of the high-dim overfitting it was doing.

RBF local kernels were able to make decision boundaries on clusters of similar shapes