Lab Report: SVM Implementation and Analysis

Question 1 – Binary SVM with Non-linear Kernel

I maximized the dual objective (7.32) subject to constraints (7.33–7.34).

I adopt the RBF kernel: np.exp(-gamma *K)

$$k(x, x') = \exp\left(-\gamma \|x - x'\|^2\right), \qquad \gamma = 0.1.$$

Rewriting in matrix form gives the QP matrices P, q, G, h, A, b required by cvxopt.

Question 2 – Prediction & Bias

Using (7.13) I compute the decision function, np.dot(K_test,self.sv_a*self.sv_y)+self.b

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b.$$

The bias term is obtained via (7.37) by averaging over margin Support Vectors.

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

Question 3 – Multi-class Results

I employed random balanced subsampling (γ=0.1, C=10): sample_ratio=0.3 to improve the computing efficiency. OvO achieves marginally better accuracy (+0.2%)

Method	**Test Accuracy**
One-vs-Rest (OvR)	**0.876**
One-vs-One (OvO)	**0.878**

Question 4 – Hyper-parameter C

The training process is time consuming. To save computing resources, 20 % of the training set was randomly chosen, this subset was further split 80 % and 20 % for training / validation.

Candidate C values were generated by np.logspace(1, 3, 5), and the best C: 10.0000, Validation Accuracy: 0.8250

Question 5 – Confusion Matrices

Each row represents the true label, and each column shows the predicted label.

An ideal classifier would have all nonzero values along the diagonal, indicating all predictions match the true class. Off-diagonal entries indicate misclassifications.

Both methods achieve near-perfect accuracy on easily distinguishable classes like Class 0 and Class 1 show minimal confusion. OvO improves accuracy for Class 5(OvR: 74/92 -> OvO: 81/92) and Class 7 (OvR: 81/99 -> OvO: 89/99), but misclassification for Class 9 does not improve much. This aligns with the visual similarity of handwritten 4s and 9s.

The model could be further improved by increasing the training set size (currently only 20% is used due to subsampling), although this is limited by computational cost.