Machine Learning

Assignment 4

Fatemeh Nadi 810101285

January 8, 2023

Problem 1

point	Z_1	Z_2	Label = T
X_1	-1	0	-1
X_2	0	1	-1
X_3	1	0	+1

Here is the mathematical expression for a hyperplane, where w_i are the coefficients and b is the arbitrary constant that determines the distance from the origin:

$$W^T Z_i + b_0 = 0$$

For the ith 2-dimensional point (Z_{i1}, Z_{i2}) the above expression is reduced to:

$$W_1 Z_{i1} + W_2 Zi2 + b = 0$$

Considering we are trying to maximize the margin between positive and negative data points, we would like the positive data points to satisfy the following constraint:

$$W^T Z_i^+ + b \geq +1 \rightarrow +1 \left(W^T Z_i^+ + b\right) \leq 0$$

Similarly, the negative data points should satisfy:

$$W^T Z_i^- + b \geq -1 \rightarrow -1 \left(W^T Z_i^- + b\right) \leq 0$$

To write a uniform equation for both sets of points, we can use $t_i \in \{-1, +1\}$ to denote the class label of data point Z_i:

$$t_i \left(W^T Z_i + b\right) \leq 0$$

As a result, the perpendicular distance d_i from the margin can be expressed as:

$$d_i = \frac{|w^T z_i + w_0|}{||w||}$$

We can maximize this distance by minimizing the square of the denominator, giving us a quadratic programming problem shown below:

$$\min \frac{1}{2}||w||^2 \text{ subject to } t_i(w^T z_i + w_0) \geq +1, \forall i$$

Lagrange multipliers can help us solve the above quadratic programming problem with inequality constraints.
The Lagrange function is therefore:

$$L(w, b, \lambda) = \frac{1}{2}||w||^2 + \sum_i \lambda_i g_i\,(Z)$$

$g_i\,(Z) = 1 - \big(t_i(w^T Z + b)\big)$

$$g_1\,(Z) = 1 - (t_1(w_1 z_1 + w_2 z_2 + b)) \xrightarrow[t_1=-1]{(-1,0)} 1 - w_1 + b$$

$$g_2\,(Z) = 1 - (t_2(w_1 z_1 + w_2 z_2 + b)) \xrightarrow[t_2=-1]{(0,1)} 1 + w_2 + b$$

$$g_3\,(Z) = 1 - (t_3(w_1 z_1 + w_2 z_2 + b)) \xrightarrow[t_3=+1]{(1,0)} 1 - w_1 - b$$

$$L(w, b, \lambda) = \frac{w_1^2 + w_2^2}{2} + \lambda_1\,(1 - w_1 + b) + \lambda_2\,(1 + w_2 + b\,) + \lambda_3\,(1 - w_1 - b)$$

To solve the above, we set the following:

$$\begin{cases} \frac{\partial L}{\partial w_1} = 0 \rightarrow w_1 - \lambda_1 - \lambda_3 = 0 \\ \frac{\partial L}{\partial w_2} = 0 \rightarrow w_2 + \lambda_2 = 0 \\ \frac{\partial L}{\partial b} = 0 \rightarrow \lambda_1 + \lambda_2 - \lambda_3 = 0 \end{cases}$$

Also, Karush-Kuhn-Tucker (KKT) conditions are satisfied by the above constrained optimization problem as given by:

$$\begin{cases} \lambda_i \geq 0 \\ \\ g_i(Z) \leq 0 \\ \\ \lambda_i \; g_i\,(Z) = 0 \Rightarrow \begin{cases} \lambda_1\,(1 - w_1 + b) = 0 \\ \lambda_2\,(1 + w_2 + b) = 0 \\ \lambda_3\,(1 - w_1 - b) = 0 \end{cases} \\ \end{cases}$$

As a result of solving these three equations, the weights should be as follows:
$w_1 = 1,\;\; w_2 = -1,\;\; b = 0$
It is now time to establish the first three equations:
$\lambda_1 = 0,\;\; \lambda_2 = 1,\;\; \lambda_3 = 1$

As these functions are convex, we can say that the obtained points are optimal, and we can define the separating line as follows:

$$z_1 - z_2 = 0$$

Computing support vectors:
A support is actually a vector whose λ is non-zero.

$$SV = \{X_2, X_3\}$$

$$W = \lambda_2 t_2 X_2 + \lambda_3 t_3 X_3 = 1 \times -1 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 \times 1 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Margins:

$$L_1 = z_1 - z_2 = 1$$
$$L_2 = z_1 - z_2 = -1$$

Problem 2

A.

$$k\left(\vec{x},\vec{z}\right)=\phi\left(\vec{x}\right)^{T}\phi\left(\vec{z}\right)=\phi\left(\vec{z}\right)^{T}\phi\left(\vec{x}\right)=k\left(\vec{z},\vec{x}\right)$$

the kernel function is symmetric.

B.

$$K\left(x_i,x_j\right)=exp\left(-\frac{1}{2}\left\|x_i-x_j\right\|^2\right)\geq 0$$

$$\begin{aligned}\left\|\phi\left(x_i\right)-\phi\left(x_j\right)\right\|^2&=\left\langle\phi\left(x_i\right),\phi\left(x_i\right)\right\rangle+\left\langle\phi\left(x_j\right),\phi\left(x_j\right)\right\rangle-2.\left\langle\phi\left(x_i\right),\phi\left(x_j\right)\right\rangle\\&=K\left(x_i,x_i\right)+K\left(x_j,x_j\right)-2K\left(x_i,x_j\right)\\&=1+1-2\exp\left(-\frac{1}{2}\left\|x_i-x_j\right\|^2\right)\leq 2\end{aligned}$$

C.

$$\begin{aligned}\left\langle\hat{W},\phi\left(x_{far}\right)\right\rangle+b=f\left(x_{far},\alpha,b\right)&=\sum_{i\in SV}y_i\alpha_iK\left(x_i,x_{far}\right)+b\\&=\sum_{i\in SV}y_i\alpha_i.\left\{\exp\left(-\frac{1}{2}\left\|x_i,x_{far}\right\|^2\right)\right\}+b\end{aligned}$$

$\left\|x_i,x_{far}\right\|^2\, is\ a\ large\ positive\ number$ so:

$\exp\left(-\frac{1}{2}\left\|x_i,x_{far}\right\|^2\right)\simeq 0$ therefore:

$\sum_{i\in SV}y_i\alpha_i.\left\{\exp\left(-\frac{1}{2}\left\|x_i,x_{far}\right\|^2\right)\right\}\simeq 0$

$$\rightarrow f\left(x_{far},\alpha,b\right)\simeq b$$

Problem 3

1. Linear soft margin with $C = 1$: figure 4.

2. Linear soft margin with $C = 10$: figure 2.

$\rightarrow$ In Figures 2 and 4, linear separators are used, and the larger the C, the wider the margin (the number of SVs will increase).

3. Polynomial hard margin with $degree = 2$: figure 5.

$\rightarrow$ Figure 3 shows a polynomial kernel with 3 degrees, while figure 2 shows a polynomial kernel with 2 degrees.

4. RBF hard margin with $Gamma = 10$: figure 1.

5. RBF hard margin with $Gamma = \frac{1}{10}$: figure 6.

$\rightarrow$ Gamma is a hyper parameter, and the higher Gamma, the more complex the model (smaller Gamma provides smooth boundaries).
A circle is usually the shape of the RBF decision boundary.

Problem 4

i	x	y	x_1_trans	x_2_trans	x_3_trans
1	0	-1	1	0	0
2	-1	1	1	-1	1
3	1	1	1	1	1

A.

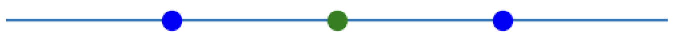
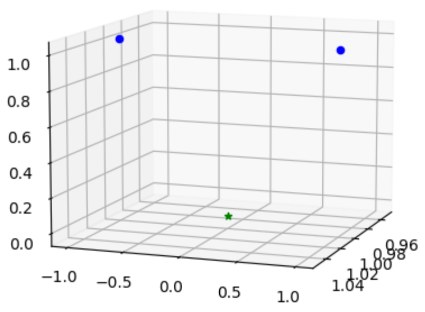
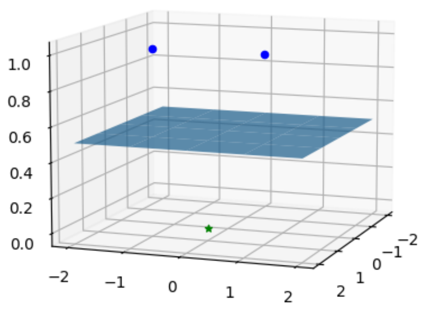


Figure 1

In the + class, the points on the axis Z have different values than in the - class, therefore. The two classes are separated by a separator along this axis.



(a) 3 points in new space



(b) Hyperplane separator

The points are mapped to $(1, 0, 0)$, $(1, -1, 1)$, $(1, 1, 1)$, in that order. It is now possible to separate points into three dimensions.

B.

Define a class variable $y_i \in \{-1, +1\}$ which denotes the class of x_i; and let $W = [w_1, w_2, w_3]^T$.
The max-margin SVM classifier solves the following problem:

$$\begin{aligned}
&\underset{w,b}{min} \frac{1}{2} \|w\|^2 \\
&s.t \;\; y_i \left(W^T \phi\left(x_i\right) + b\right) \geq 1 \quad i = 1, 2, 3
\end{aligned}$$

For optimization problems with inequality constraints, we should apply KKT conditions which is a generalization of Lagrange multipliers.

We have 3 constraints, and should have 3 Lagrange multipliers. We first form the Lagrangian function $L\left(W, \lambda\right)$ where $\lambda = \left(\lambda_1, \lambda_2, \lambda_3\right)$ as follows:

$$\begin{aligned}
L\left(W, \lambda\right) &= \frac{\|w\|^2}{2} + \sum_{i=1}^{3} \lambda_i \left(1 - y_i \left(W^T x_i + b\right)\right) \\
&= \frac{w_1^2 + w_2^2 + w_3^2}{2} + \sum_{i=1}^{3} \lambda_i g_i\left(x_i\right)
\end{aligned}$$

where,
$g_i\left(X\right) = 1 - \left(t_i(w^T X + b)\right)$

$$\begin{aligned}
g_1\left(X_1\right) &= 1 - \left(y_1(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)\right) \xrightarrow[y_1=-1]{(1,0,0)} 1 + w_1 + b \\
g_2\left(X_2\right) &= 1 - \left(y_2(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)\right) \xrightarrow[y_2=+1]{(1,-1,1)} 1 - w_1 + w_2 - w_3 - b \\
g_3\left(X_3\right) &= 1 - \left(y_3(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)\right) \xrightarrow[y_3=+1]{(1,1,1)} 1 - w_1 - w_2 - w_3 - b
\end{aligned}$$

and differentiate with respect to optimization variables w and b and equate to zero,

$$\begin{aligned}
\frac{\partial L}{\partial w} &= w - \sum_{i=1}^{3} \lambda_i y_i \phi\left(x_i\right) = 0 \\
\frac{\partial L}{\partial b} &= - \sum_{i=1}^{3} \lambda_i y_i = 0
\end{aligned}$$

so,

$$\begin{cases} \overset{w_1}{\rightarrow} w_1 + \lambda_1 - \lambda_2 - \lambda_3 = 0 \\ \overset{w_2}{\rightarrow} w_2 + \lambda_2 - \lambda_3 = 0 \\ \overset{w_3}{\rightarrow} w_3 - \lambda_2 - \lambda_3 = 0 \\ \lambda_1 - \lambda_2 - \lambda_3 = 0 \end{cases}$$

$\lambda_1 = \lambda_2 + \lambda_3 \rightarrow \boxed{w_1 = 0}$

Also, Karush-Kuhn-Tucker (KKT) conditions are satisfied by the above constrained optimization problem as given by:

$$\begin{cases} \lambda_i \geq 0 \\ g_i(X) \leq 0 \\ \lambda_i \, g_i(X) = 0 \Rightarrow \begin{cases} \lambda_1 (1 + w_1 + b) = 0 \overset{w_1=0}{\rightarrow} \boxed{b = -1} \\ \lambda_2 (1 - w_1 + w_2 - w_3 - b) = 0 \rightarrow w_3 - w_2 = 2 \\ \lambda_3 (1 - w_1 - w_2 - w_3 - b) = 0 \rightarrow w_3 + w_2 = 2 \end{cases} \end{cases}$$

$as\ a\ result, w_3 = 2, w_2 = 0$

$\lambda_1 = 2, \lambda_2 = 1, \lambda_3 = 1$

Therefore the optimal weights are $\hat{W} = (0, 0, 2)^T$ and b = -1. And the margin is 0.5.

$$Margin = \frac{2}{\|W\|^2} = \frac{2}{\begin{bmatrix} 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}} = \frac{2}{4} = \frac{1}{2}$$

Problem 5

The binomial distribution is the probability distribution of the number of successes in a sequence of N independent events.
The probability of k successes (e.g. predict correct) is:

$$\binom{N}{k} p^k (1-p)^{N-k}$$

Voting mechanism confirms that the ensemble classifier will make an error only when the majority of individual classifiers make an error if the individual classifiers are independent.

A.
$N = 5$ and $Accuracy = 0.51 \to Errorrate = 0.49$
A majority vote will determine the output class of the ensemble classifier from the five classifications.
The following formula can be used to calculate ensemble classifier accuracy: The probability that three of the five classifiers will make an error is:

$$\binom{5}{3} (.49)^3 (.51)^2 = 0.306005$$

The probability that four of the five classifiers will make an error is:

$$\binom{5}{4} (.49)^4 (.51)^1 = 0.1470024$$

The probability that three of the five classifiers will make an error is:

$$\binom{5}{5} (.49)^5 (.51)^1 = 0.02824752$$

$$Error\,(Ensemble) = 0.306005 + 0.1470024 + 0.02824752 = 0.481255$$

$$Accuracy = 1 - Error = 0.518745 > 0.51 \to increased$$

B.
$N = 9$

$$Error\left(Ensemble\right) = \binom{9}{5}\left(.49\right)^5\left(.51\right)^4 + \binom{9}{6}\left(.49\right)^6\left(.51\right)^3 + \binom{9}{7}\left(.49\right)^7\left(.51\right)^2 + \binom{9}{8}\left(.49\right)^8\left(.51\right)^1 + \binom{9}{9}\left(.49\right)^9\left(.51\right)^0 = 0.4754037$$

$$Accuracy = 1 - Error = 0.5245963 > 0.51 \rightarrow increased$$

C.
With N approaching infinity, the ensemble classifier's accuracy will approach 1. Each weak classifier has an error rate less than 50%. Therefore, when the number of weak classifiers increases, the product of their error rates will be close to 0, resulting in 1 accuracy.

The bagging method, theoretically, reduces overfitting and improves generalization for machine learning models by reducing overfitting.
Using the bagging method can also improve the accuracy of a model, but the improvement may not always be significant. Bagging can be effective depending on a number of factors, such as the quality of the individual classifiers, the type of data and the task at hand.

A bagging method may not be effective if the individual classifiers have high bias and low variance. However, if the individual classifiers have low bias and high variance, the bagging method may improve the model's accuracy more effectively.

D.

$$Error\left(Ensemble\right) = \binom{5}{3}\left(.5\right)^3\left(.5\right)^2 + \binom{5}{4}\left(.5\right)^4\left(.5\right)^1 + \binom{5}{5}\left(.5\right)^5\left(.5\right)^0 = 0.5$$

In this case, the accuracy of the ensemble classifier is fixed, so weak learners must outperform random models to use this method.

Problem 6

Please see this file: ”Q6.ipynb”
Code and explanation are provided.

A.

Linear: When data can be separated linearly, this kernel function is used. It creates a linear decision boundary by finding the hyperplane that maximally separates the two classes.

Polynomial: When the data cannot be linearly separated, this kernel function is used. It creates a non-linear decision boundary by mapping the data to a higher-dimensional space and finding the hyperplane that maximally separates the two classes.

RBF (Radial Basis Function): The kernel function is also used when data cannot be linearly separated. It creates a non-linear decision boundary by using a kernel function to measure the similarity between points in the data.

Sigmoid: Unlike RBF kernel functions, these kernel functions are specifically designed only for binary classification tasks. It creates a non-linear decision boundary by using a sigmoid function to measure the similarity between points in the data.

The choice of kernel function will depend on the nature of the data and the requirements of the classification task.
Generally, linear kernel functions are the fastest to compute and are suitable when data can be linearly separated.
It is more computationally expensive to use the polynomial and RBF kernel functions, but they can handle more complex data and may perform better in classification.
In binary classification tasks, sigmoid kernels are typically used but they are computationally expensive.

B.

Linear Kernel

It is the simplest type of kernel, usually one-dimensional. It is the most efficient function when there are a lot of features. As most classification problems can be linearly separated, the linear kernel is preferred for text classification problems.
The linear kernel functions are faster than other functions.

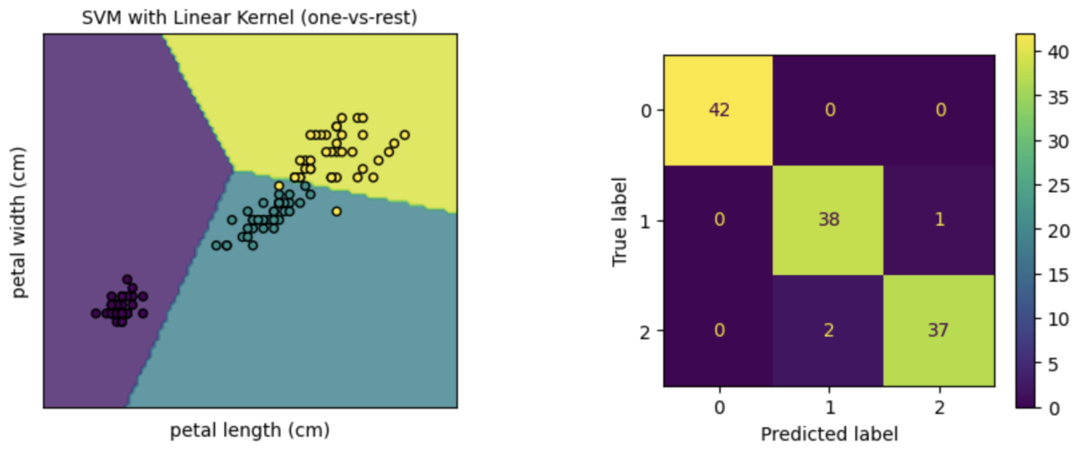


Figure 3: Decision boundary and Confusion matrix of SVM with Linear Kernel (one-vs-rest)

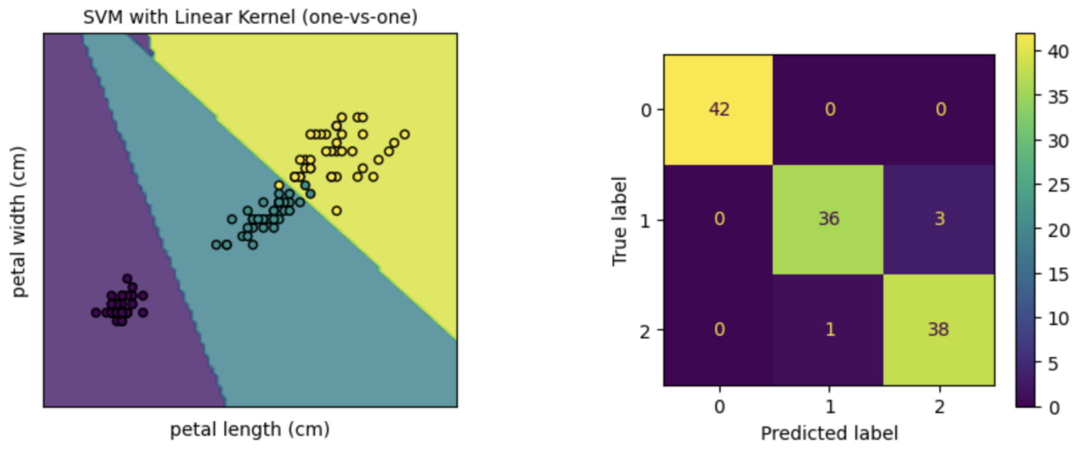


Figure 4: Decision boundary and Confusion matrix of SVM with Linear Kernel (one-vs-one)

RBF Kernel

It is one of the most popular and widely used kernel functions in SVM. Non-linear data is usually handled using this method. When there is no prior knowledge of data, it helps to make proper separations.

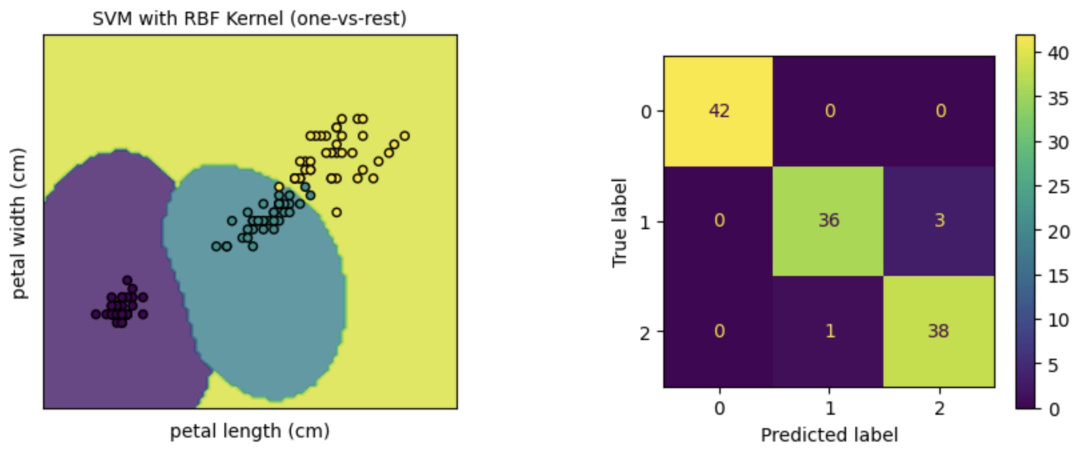


Figure 5: Decision boundary and Confusion matrix of SVM with RBF Kernel (one-vs-rest)

Polynomial Kernel

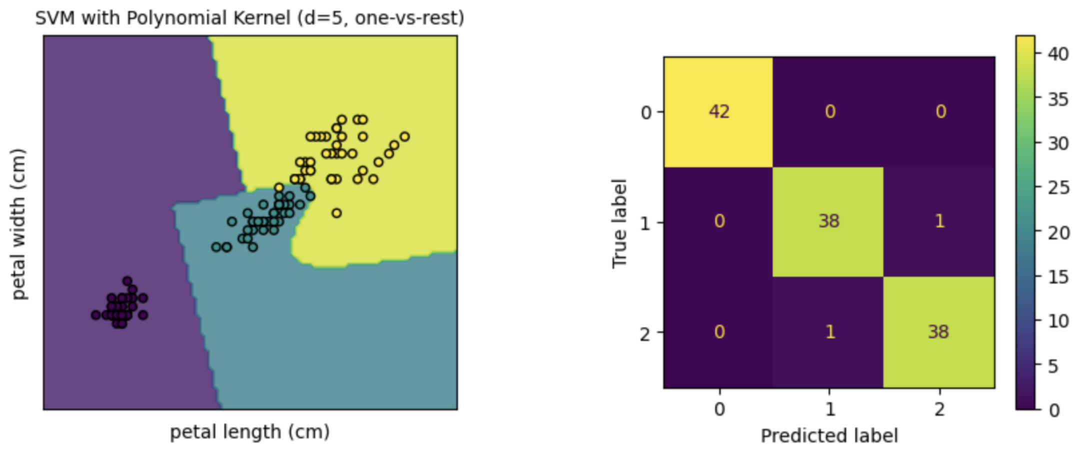
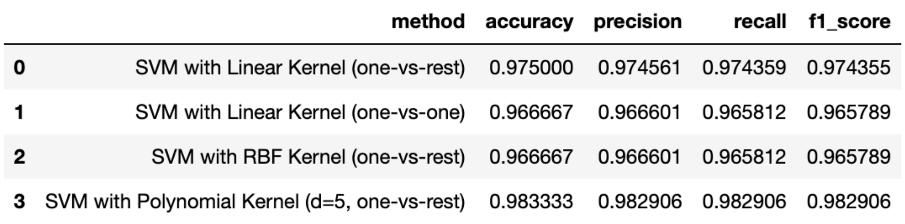


Figure 6: Decision boundary and Confusion matrix of SVM with Polynomial Kernel (d=5, one-vs-rest)

C.

Accuracy On Train Data:

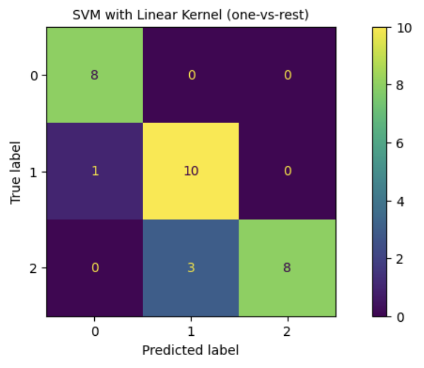


	method	accuracy	precision	recall	f1_score
0	SVM with Linear Kernel (one-vs-rest)	0.975000	0.974561	0.974359	0.974355
1	SVM with Linear Kernel (one-vs-one)	0.966667	0.966601	0.965812	0.965789
2	SVM with RBF Kernel (one-vs-rest)	0.966667	0.966601	0.965812	0.965789
3	SVM with Polynomial Kernel (d=5, one-vs-rest)	0.983333	0.982906	0.982906	0.982906

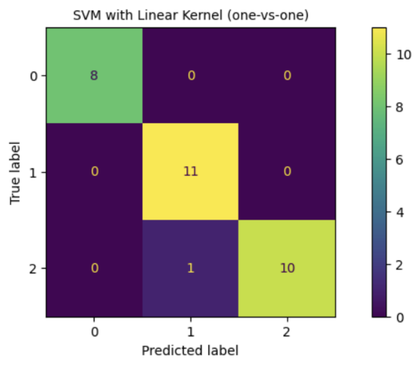
Figure 7: Models evaluations on train data

The simple, small, and separable classes in this dataset mean that different kernels perform similarly in the final classification, and accuracies are close to 95-97% on train data.

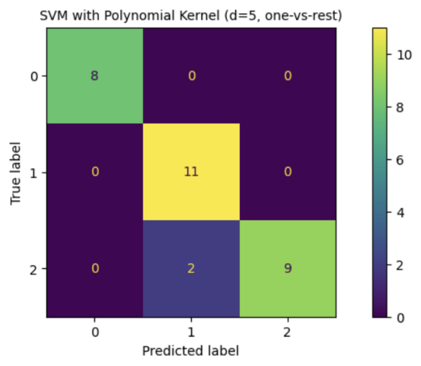
Confusion Matrix On Test Data



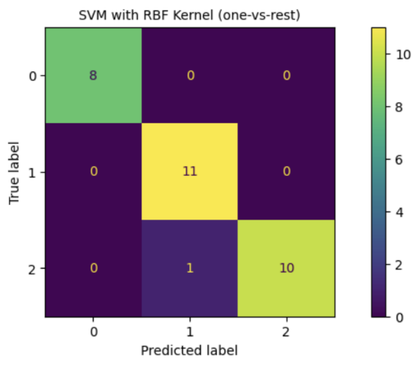
(a) Confusion matrix of Linear Kernel (one-vs-rest)



(b) Confusion matrix of Linear Kernel (one-vs-one)



(c) Confusion matrix of RBF Kernel (one-vs-rest)



(d) Confusion matrix of Polynomial Kernel (d=5, one-vs-rest)

Accuracy On Test Data:

	method	accuracy	precision	recall	f1_score
0	SVM with Linear Kernel (one-vs-rest)	0.866667	0.886040	0.878788	0.872205
1	SVM with Linear Kernel (one-vs-one)	0.966667	0.972222	0.969697	0.969634
2	SVM with RBF Kernel (one-vs-rest)	0.966667	0.972222	0.969697	0.969634
3	SVM with Polynomial Kernel (d=5, one-vs-rest)	0.933333	0.948718	0.939394	0.938889

Figure 9: Models evaluations on test data

Gamma and C, which are hyperparameters that should be determined by validation, play an essential role in determining the model's accuracy.

Gamma describes how far an individual training example's influence reaches, with low values indicating 'far' and high values indicating 'close'. Gamma parameters can be seen as the inverse of support vectors' radius of influence.
Models with small gamma cannot capture the complexity or "shape" of data when the gamma is too small.

The C parameter trades off the correct classification of training examples with maximization of the margin of the decision function. The margin for larger values of C will be smaller if the decision function correctly classifies all training points. When C is lower, there will be a larger margin, resulting in a simpler decision function at the expense of training accuracy. SVMs use C as a regularization parameter.

Conclusion:

SVM with Linear Kernel (one-vs-rest) has the least accuracy among other kernels. But this kernel has high accuracy in train; this model overfits to train data.
To compare two approaches, one vs one and one vs rest for different kernels. Generally, the one vs one method is better when the number of classes is small, and the number of samples for each class is almost the same. Still, the one vs rest approach is suitable for times when the number of classes is large, according to this description and the dataset in this example. We have. It seems that the one vs one approach is more suitable.

The polynomial kernel has high time complexity compared to other kernels of similar accuracy.

Problem 7

Please see this file: ”Q7.ipynb”
Code and explanation are provided.

A.
RBF kernel:
Best parameters: $C : 100, gamma : 1$

Linear kernel:
Best parameters: $C : 1e - 05$

Polynomial kernel:
Best parameters: $C : 1e - 20, degree : 3, gamma : 1e - 20$

B.

RBF kernel:

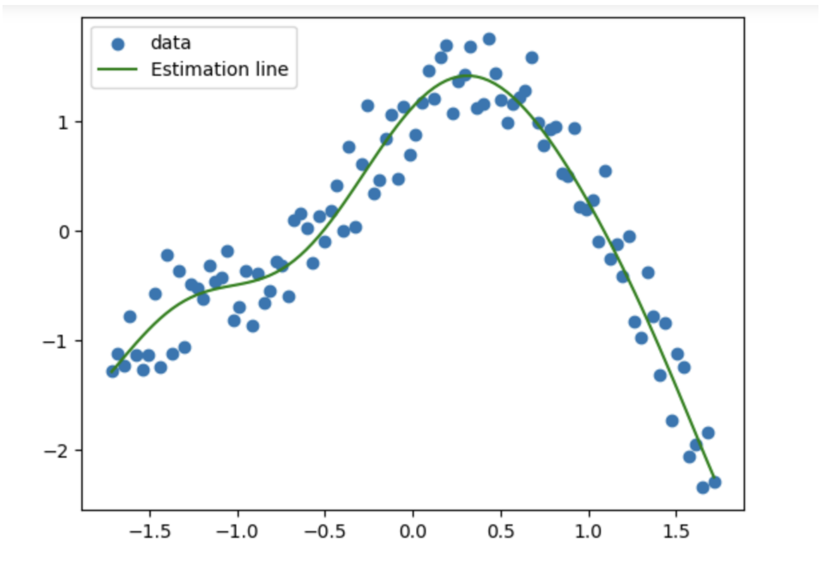


Figure 10: Estimation line with RBF kernel method

MSE = 0.071

Linear kernel:

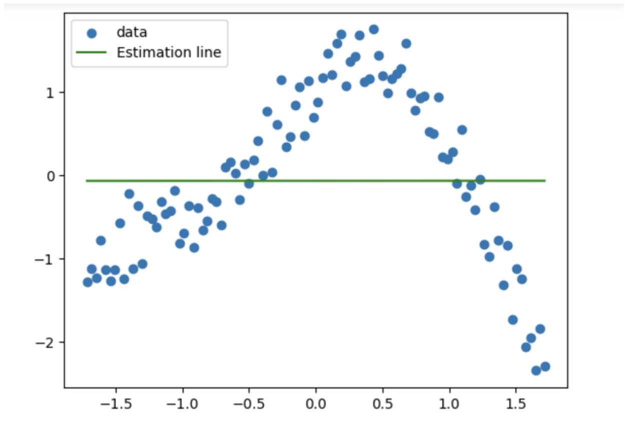


Figure 11: Estimation line with Linear kernel method

MSE = 1.004

Polynomial kernel:

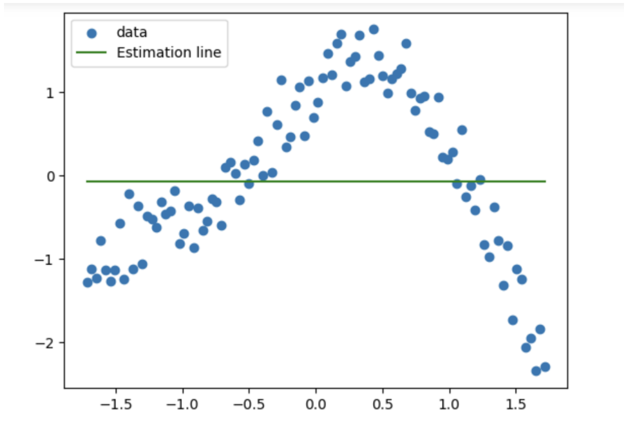


Figure 12: Estimation line with Polynomial kernel method

MSE = 1.004

Linear and polynomial kernels have high error rates; in problem 6, we explain that each kernel is appropriate for particular data. This data can't be modelled with a line or polynomial line, so with the correct parameters, these kernels didn't work.
But RBF is a complex kernel and can learn and fit the data.