Assignment - 2

Aditya
2020csb1065

Task 1: Classification

2.

Series M is classified correctly most of the time because it can be easily distinguished from the other two classes in a straight line. However, Series N and Series L are not as easy to separate in this way. Despite this challenge, the model still achieves a good level of accuracy, showing that it can effectively differentiate between the classes using linear methods.

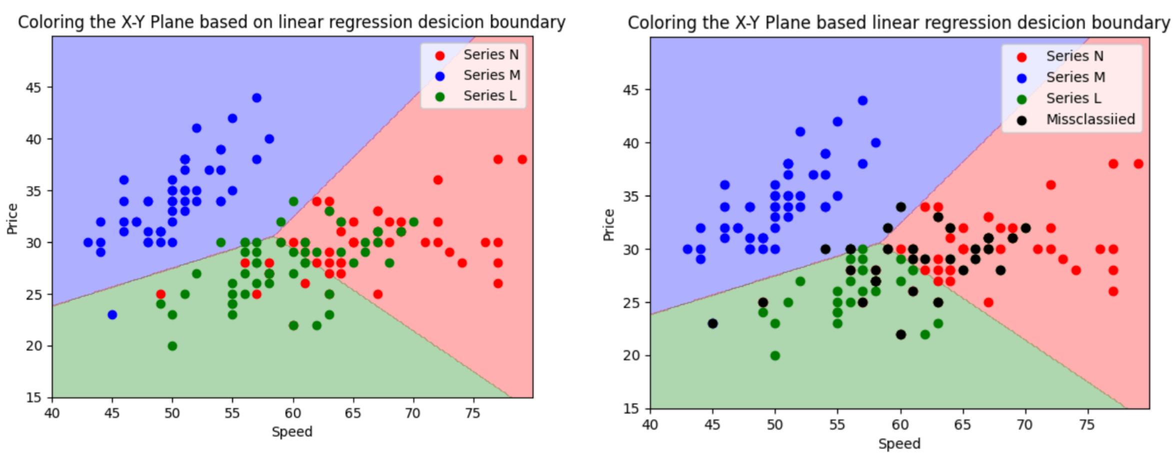

Fig. 1: Least square decision boundary without and with misclassified points marked with black

3A.

The highest testing and training accuracy is achieved when $\lambda = 0.01$. ($\beta = 0.01$, N= 100000)

For λ : 0.01
Training Accuracy mean: 0.677
Training Accuracy std: 0.055
Testing Accuracy mean: 0.643
Testing Accuracy std: 0.129

For λ : 0.1
Training Accuracy mean: 0.651
Training Accuracy std: 0.039
Testing Accuracy mean: 0.617
Testing Accuracy std: 0.073

For λ : 1
Training Accuracy mean: 0.522
Training Accuracy std: 0.105
Testing Accuracy mean: 0.493
Testing Accuracy std: 0.174

For λ : 10
Training Accuracy mean: 0.432
Training Accuracy std: 0.102
Testing Accuracy mean: 0.38
Testing Accuracy std: 0.133

For λ : 100
Training Accuracy mean: 0.529
Training Accuracy std: 0.128
Testing Accuracy mean: 0.427
Testing Accuracy std: 0.174

3B.

The optimal accuracy is attained when $\beta = 0.001$. ($\lambda = 0.1$, N= 100000)

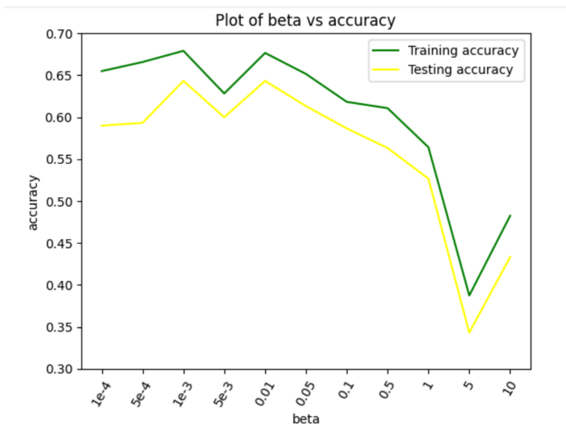

Fig. 2: Training and testing accuracy of l2 regularized logistic regression ($\lambda = 0.1$, N= 100000)

3C.

No, there is no improvement in performance. ($\lambda = 0.1$, $\beta = 0.01$, N= 100000)

The training and testing accuracy for the least square discriminant model were both 0.793.

For the l2 regularized logistic regression model, the training accuracy was 0.679, and the testing accuracy was 0.643.

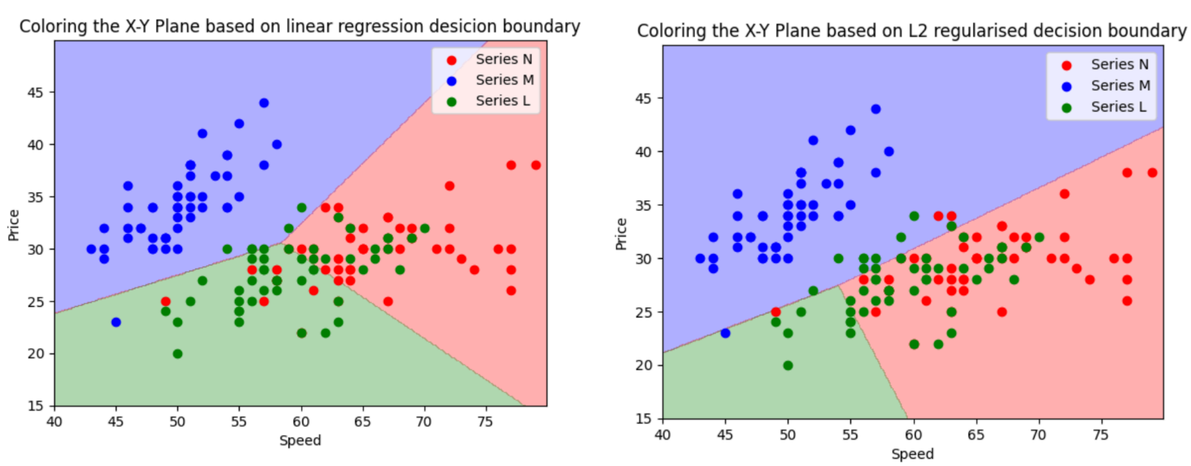

Fig. 3: Decision boundary of least square and L2 regularized logistic regression ($\lambda = 0.1$, $\beta = 0.01$, N= 100000)

3D.

The highest testing accuracy of 0.633 is achieved when the value of α is set to 0.2. (λ = 0.1, β = 0.01, N= 1000)

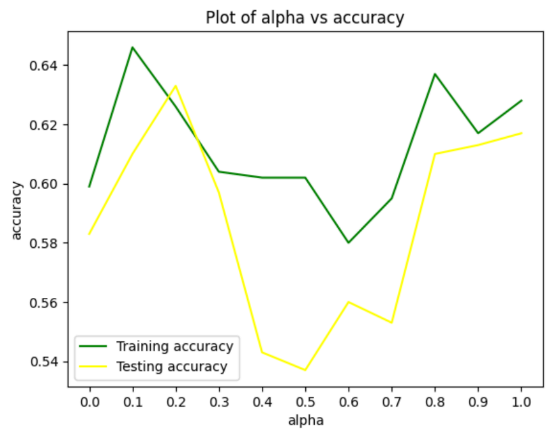

Fig. 4: Training and testing accuracy of net elastic logistic regression (λ = 0.1, β = 0.01, N= 1000)

4.

For λ = 0.01, β = 0.001, N= 100000

Training and testing accuracy of vanilla logistic regression is 0.758 and 0.733.

Training and testing accuracy of l2 regularized logistic regression is 0.679 and 0.643.

Accuracy of vanilla logistic regression is more, after regularization the accuracy decreased.

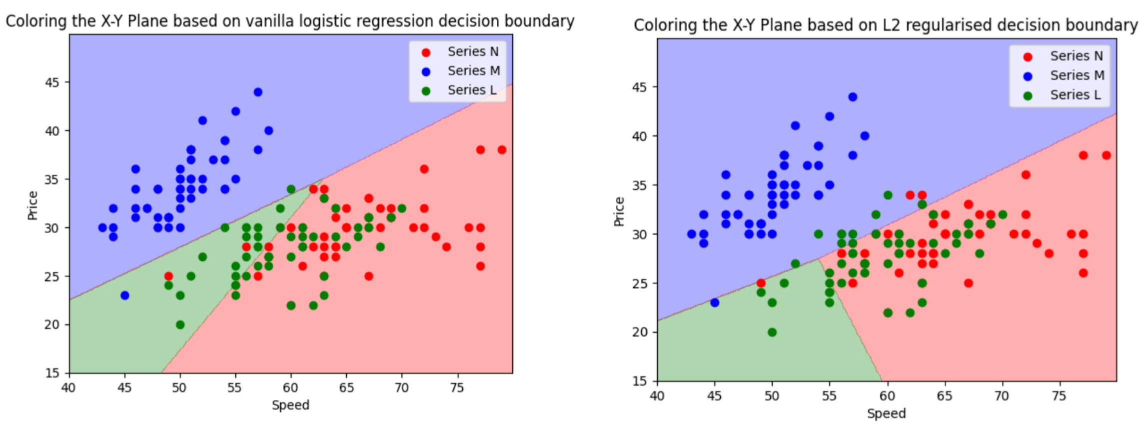

Fig. 5: Decision boundary of vanilla and l2 regularized logistic regression (λ=0.01, β= 0.001, N=100000)

Task 2: SVM

A : Linear Kernel

Optimal penalty parameter (C) = 0.0005

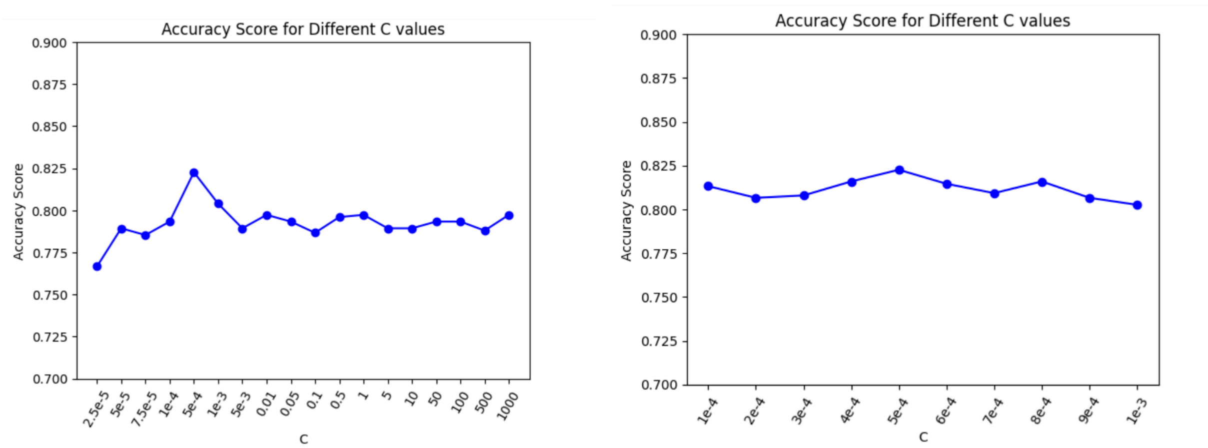

Fig. 6: accuracy score with different value of C

Plot of number of support vector and misclassified points with C

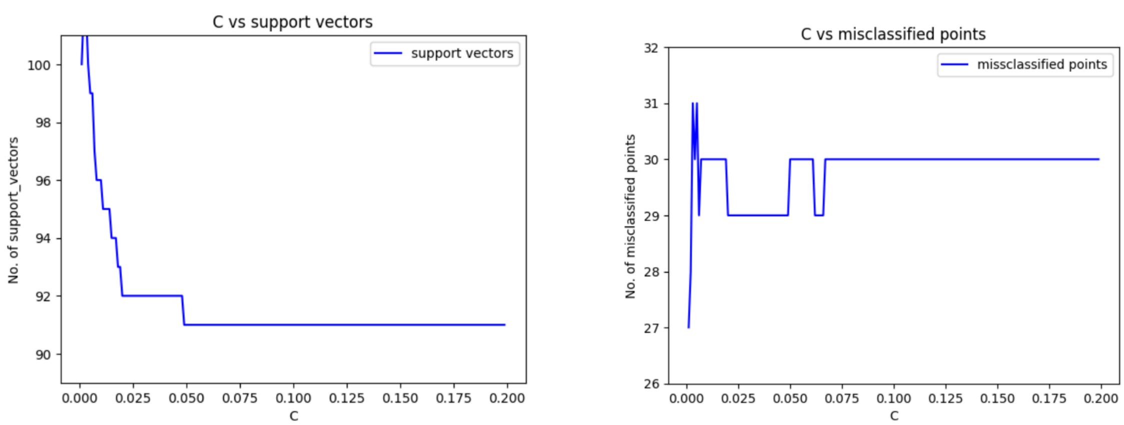

Fig. 7: Number of support vectors and misclassified points with different C

Observations:
As the penalty parameter C increases, the number of support vectors decreases and becomes constant. Conversely, as C increases, the number of misclassified points also increases and later becomes constant.

These observations align with the theory, as higher penalty parameters in Support Vector Machines (SVMs) prioritize maximizing the margin between classes over correct classification. This shift is a result of how the penalty parameter impacts the optimization objective of the SVM. With higher values of C, the decision boundary becomes smoother and better generalizes to unseen data, thereby requiring fewer support vectors to define the margin.

Optimal decision boundary of Linear Kernel SVM (C = 0.0005)

Here series M is correctly classified as it linearly separable from the rest of the two classes.
Series N and series L are not perfectly classified but good enough accuracy is achieved.

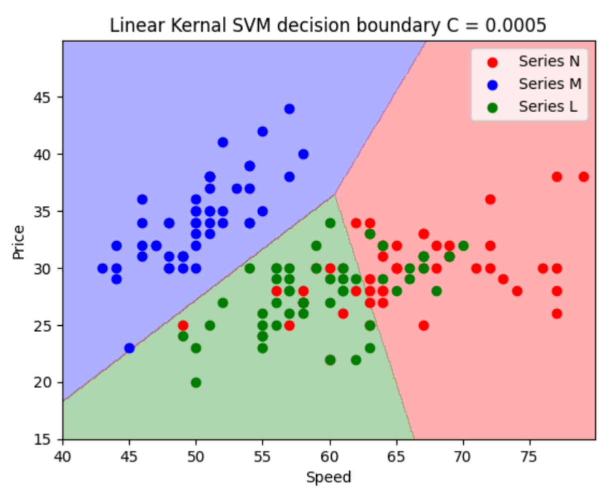

Fig. 8: Decision boundary of linear kernel SVM (C = 0.0005)

B. Polynomial and RBF kernel

Optimal penalty parameter for polynomial kernel, C = 0.0015 (degree = 3)

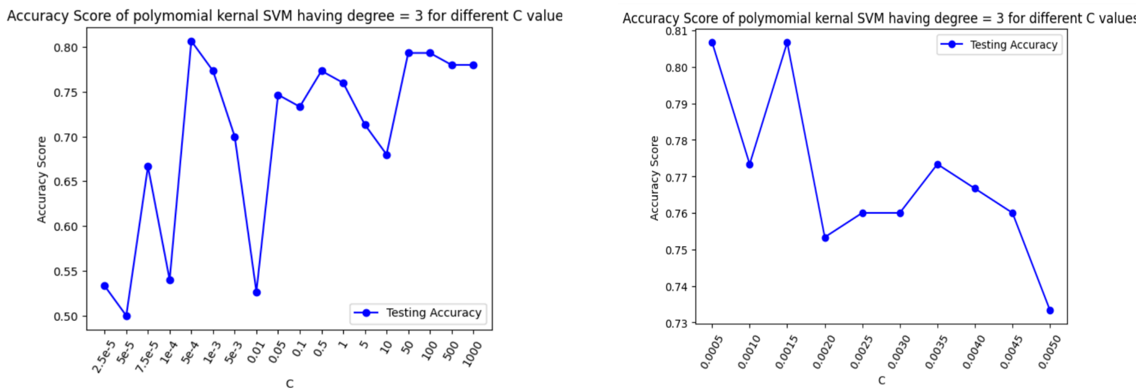

Fig. 9: Polynomial kernel accuracy with varied Penalty Parameter (γ = 3)

Optimal penalty parameter for RBF kernel, C = 1300 (gamma = 0.05)

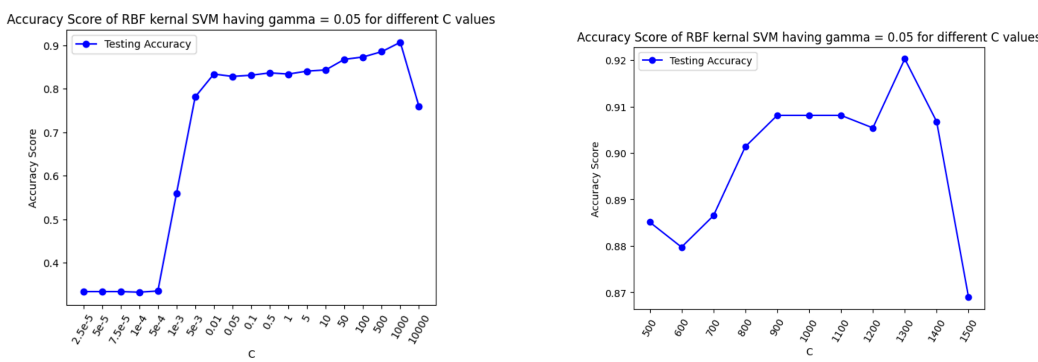

Fig. 10: RBF kernel accuracy with varied Penalty Parameter (γ = 0.05)

Decision boundary of optimal classifier for polynomial kernel and RBF kernel

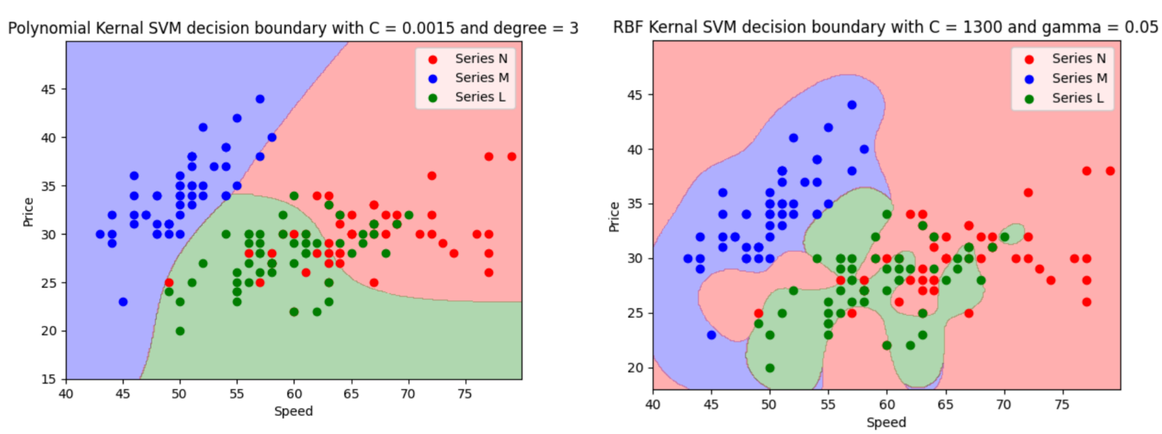

Fig 11: Polynomial kernel (C = 0.0015 and degree = 3), RBF kernel (C = 1300 and γ = 0.05)

Comparing Polynomial and RBF kernel:

At low values of gamma, the polynomial kernel outperforms the RBF kernel.

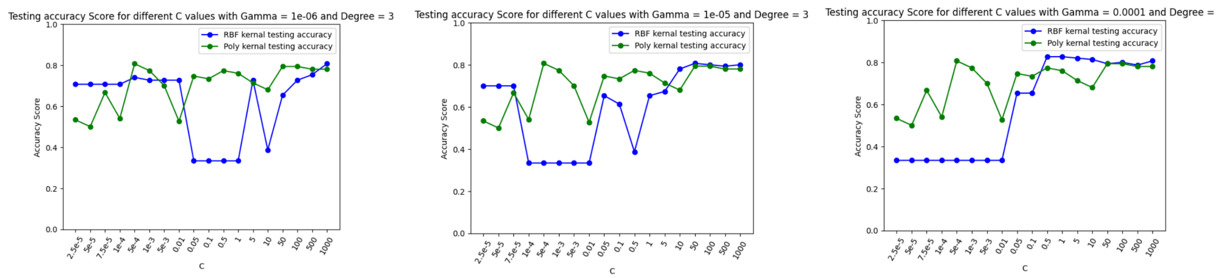

Fig 12: Testing error when γ is 1e-6, 1e-5 and 1e-4 respectively.

At mid values of gamma, the polynomial kernel performs better at lower values of C, while the RBF kernel performs better at higher values of C.

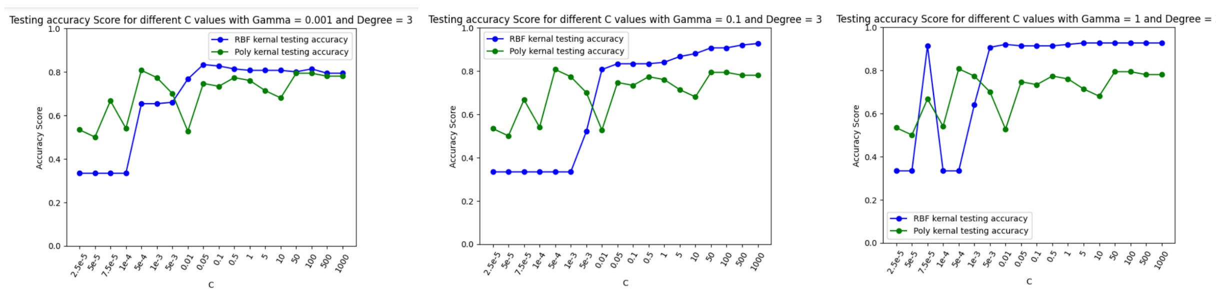

Fig 13: Testing error when γ is 0.01, 0.1 and 1 respectively.

At high values of gamma, the RBF kernel outperforms the polynomial kernel.

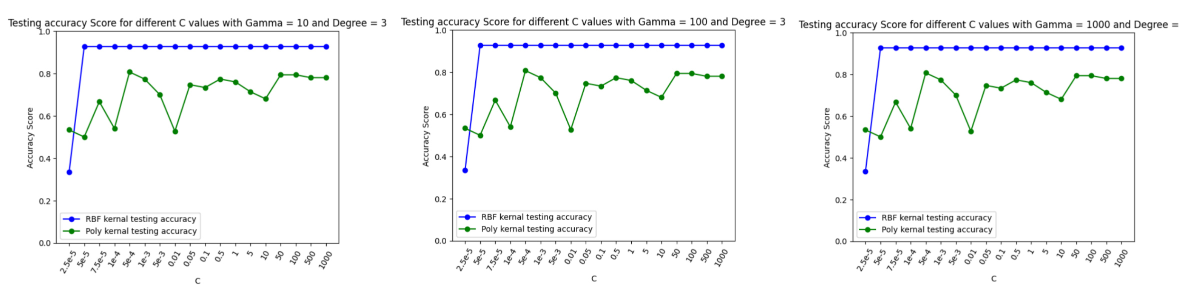

Fig 14: Testing error when γ is 10, 100 and 1000 respectively.

Does the RBF kernel overfit?

No, it doesn't overfit. There isn't a significant difference between training and testing accuracy. Although the RBF kernel is susceptible to overfitting, it's not the case here.

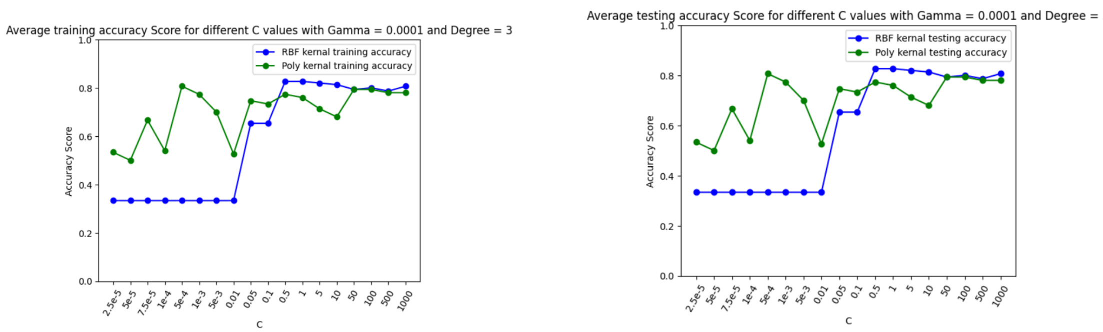

Fig. 12: Training and testing error of RBF kernel SVM (γ = 0.0001)

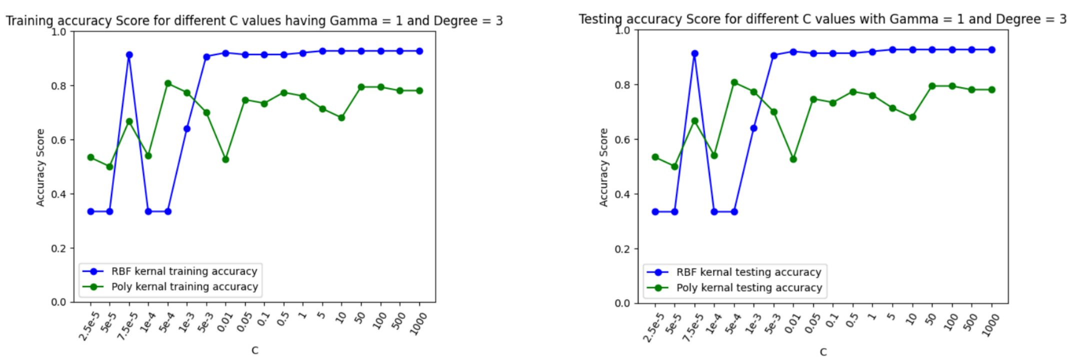

Fig. 12: Training and testing error of RBF kernel SVM (γ = 1)

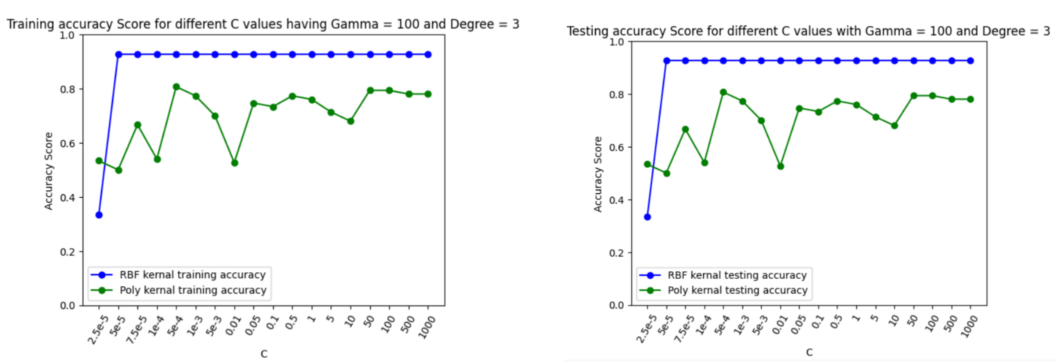

Fig. 13: Training and testing error of RBF kernel SVM (γ = 100)

3.

Precision

Precision per Class of Linear kernel- Series M : Series L: Series N :: 1: 0.7 : 0.73
Precision per Class of Poly. kernel- Series M : Series L: Series N :: 0.96: 0.73: 0.60
Precision per Class of RBF kernel- Series M : Series L: Series N :: 1: 0.86: 0.89
Macro Precision of Classifier- Linear Kernel : Polynomial Kernel: RBF Kernel :: 0.82: 0.76: 0.92
Micro Precision of Classifier- Linear Kernel : Polynomial Kernel: RBF Kernel :: 0.82: 0.76: 0.92

Recall

Recall per Class of Linear kernel- Series M : Series L: Series N :: 1: 0.72: 0.76
Recall per Class of Poly. kernel- Series M : Series L: Series N :: 1: 0.44: 0.82
Recall per Class of RBF kernel- Series M : Series L: Series N :: 1: 0.9: 0.86
Macro Recall of Classifier- Linear Kernel : Polynomial Kernel: RBF Kernel :: 0.82: 0.75: 0.92
Micro Recall of Classifier- Linear Kernel : Polynomial Kernel: RBF Kernel :: 0.82: 0.7: 0.92

F1-Score

F1-score per Class of Linear kernel- Series M : Series L: Series N :: 1: 0.73: 0.74
F1-score per Class of Poly. kernel- Series M : Series L: Series N :: 0.98: 0.55: 0.69
F1-score per Class of RBF kernel- Series M : Series L: Series N :: 1: 0.88: 0.87
Macro F1-score of Classifier- Linear Kernel : Polynomial Kernel: RBF Kernel :: 0.83: 0.74: 0.92
Micro F1-score of Classifier- Linear Kernel : Polynomial Kernel: RBF Kernel :: 0.83: 0.74: 0.92

Observations:

The RBF kernel outperforms other kernels across all benchmarks.
Series M exhibits higher precision, recall, and F1 score compared to the other classes due to its separability from the rest.
Both macro and micro metrics produced identical results, as each class has an equal number of instances.

Task 3: Gradient Boosted Trees

1. Plot of training error and testing error against the height in the decision tree.

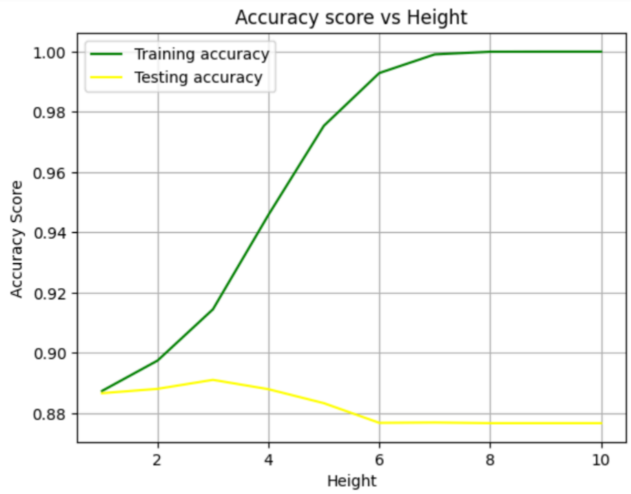

Fig. 14: Training and testing accuracy with maximum height of decision tree

Observations:

As the maximum height of a tree grows, it tends to overfit the data. This is apparent when observing the rise in training accuracy alongside the decline in testing accuracy.

2A. Create 10 weak learners using code for the decision tree.

Observations:

Leaning rate : 0.1

Max Depth : 4

Training Accuracy = 0.954

Testing Accuracy = 0.868

2B. Find the optimal learning rate and number of weak learners.

Observations:
The highest testing accuracy attained when β = 0.4, with 5 weak learners and a maximum height of 3.

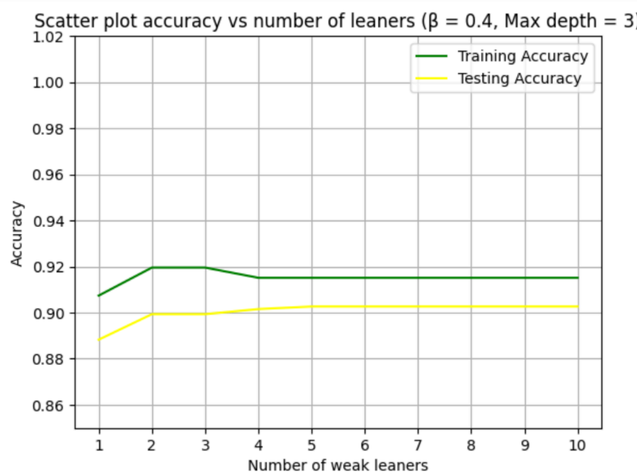

Fig. 15: Training and testing accuracy with number of weak learners (β = 0.4 and height = 3)

3. Report the error on final prediction by using the formula given above. Did the performance improve? Report your findings.

The testing accuracy is 0.904, and the training accuracy is 0.915. Therefore, the testing error is 0.096, and the training error is 0.085.

While there was an improvement in performance, it was not substantial. Without boosting, the testing accuracy remains lower than the training accuracy, as depicted in Fig. 14 and Fig. 15.

Various possible combinations of maximum height, learning rate, and the number of weak learners were plotted in the notebook.

Observations:

1. With a low learning rate, a higher number of learners is needed. There exists a tradeoff.
2. Up to 10 weak learners and with low learning rates, no difference was observed between the accuracy of training and testing data.
3. The gradient boosting technique boosts training accuracy, but while testing accuracy increased, it did not increase significantly.
4. More number of weak learners leads to overfitting.