Lab Five and Six: Radial Basis Functions and MLPs

name: Huizhi Yu
number: ********
email: ********

1 Radial Basis Functions

In order to achieve and improve the Gaussian Radial Basis Function (RBF) model, compare its performance with the linear regression model, and use ten-fold cross-validation to evaluate the stability of the models, the following improvements were implemented:

1. **Normalization of Input Data:** Each feature of the input data was normalized to have a mean of 0 and a standard deviation of 1. This ensures consistent scaling across all features and improves the stability of the model training process.
2. **Adjustment of Basis Function Width (σ):** The width parameter σ was modified from being the distance between two randomly chosen data points to the average of all pairwise distances between data points. This adjustment ensures a more robust and representative parameter value for the basis functions.
3. **Selection of Basis Function Centers:** Instead of selecting the centers of the basis functions (m_j) randomly, the K-means clustering algorithm was employed, with the number of clusters (K) set equal to the number of basis functions (M). This ensures that the basis functions are positioned more effectively in the input space.

Ten-fold cross-validation was conducted by dividing the data into ten equal parts, with nine parts used for training and one for testing in each iteration. This process was repeated ten times, recording the mean squared error (MSE) for each fold. The RBF model's performance was compared with a linear regression model on the same dataset using the MSE metric, and the results were visualized with a boxplot, which is shown in **Figure 1**, to illustrate the distribution of MSE values across the folds, highlighting the stability and average performance of both models.

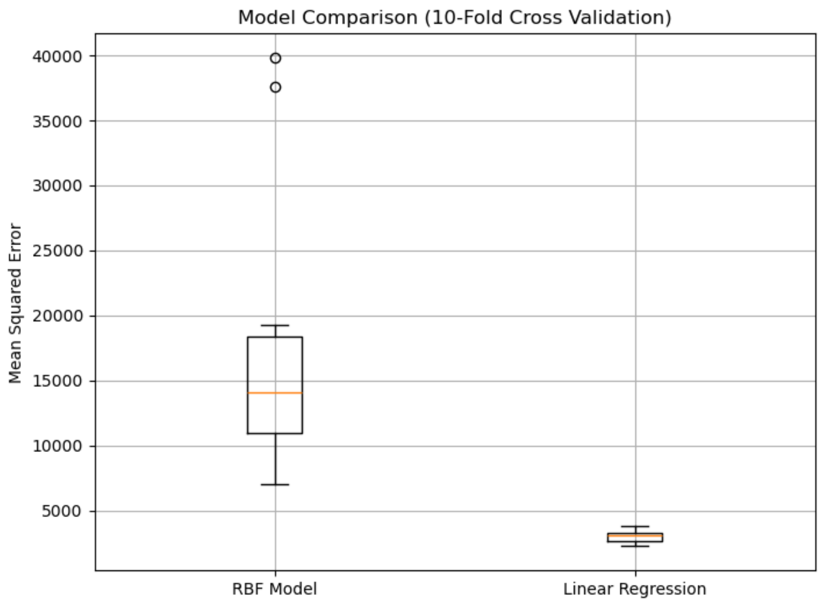

Figure 1: 10-Fold Cross-Validation MSE Comparison of RBF and Linear Regression Models

1.1 Results

The RBF model's performance was compared to the linear regression model using the MSE metric. The average MSE values across the ten folds were as follows: RBF Model: 17744.63 , Linear Regression Model: 2985.24

The distribution of MSE values across the ten folds for both models is visualized in Figure 1. This boxplot shows that the RBF model exhibited higher variability in MSE values, with several outliers indicating inconsistent performance. Moreover, the linear regression model demonstrated more stable performance, with lower and more consistent MSE values.

1.2 Conclusions

The RBF model struggled to achieve stable performance due to its sensitivity to hyperparameters such as the number of basis functions (M) and the width parameter (σ). While the use of K-means clustering improved the placement of basis function centers, the model still exhibited high mean squared error (MSE) values compared to the linear regression model. This indicates that the RBF model requires further tuning and optimization to achieve competitive results.

The linear regression model performed significantly better, with consistently lower MSE values and minimal variability across the ten folds. Its simplicity and reliance on linear relationships made it a more suitable choice for the given dataset, highlighting its ability to effectively model the problem with fewer parameters and less complexity.

2 Multi-Layer Perceptron (MLP)

2.1 Complex Classification Problem Analysis

The data distribution for the complex classification problem exhibits significant overlap between classes, which is shown in **Figure 2**, particularly between Class 1 and Class 2, as well as Class 2 and Class 3. This overlapping introduces ambiguity in defining clear decision boundaries, making the classification task more challenging. The testing dataset mirrors the characteristics of the training dataset, with similarly unclear class boundaries, further complicating the ability of models to generalize effectively to unseen data.

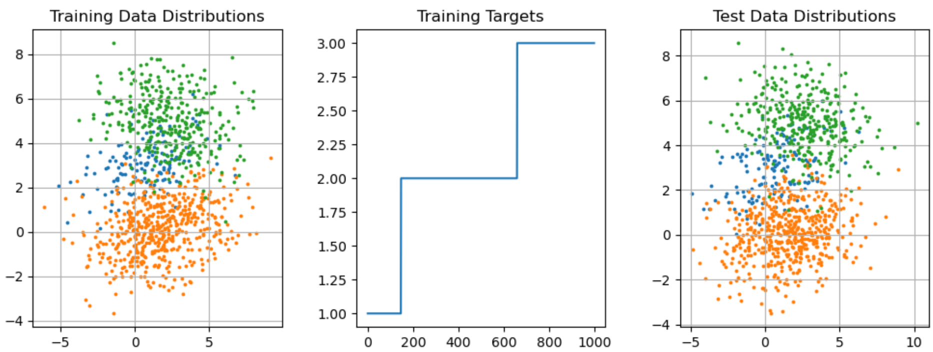

Figure 2: Complex Classification Problem Analysis

2.1.1 Performance of the Classifiers

The Multilayer Perceptron (MLP) classifier achieved a training accuracy of 86.19%. The confusion matrix indicates that Class 1 and Class 3 exhibit higher misclassification rates compared to Class 2, where the classifier demonstrates better performance with a higher proportion of correctly classified samples:

$$\begin{bmatrix} 85 & 20 & 42 \\ 25 & 484 & 3 \\ 27 & 8 & 305 \end{bmatrix}$$

The cross-validation results show an average accuracy of 87.48% with a standard deviation of ±4.05%, suggesting moderate stability in the classifier's performance across different data splits.

The Bayesian classifier, based on Gaussian Naive Bayes, achieved a slightly higher training accuracy of 88.09%. Similar to the MLP, the confusion matrix shows higher misclassification rates for Class 1 and Class 3, with Class 2 being the most accurately classified:

$$\begin{bmatrix} 73 & 28 & 46 \\ 14 & 491 & 7 \\ 17 & 7 & 316 \end{bmatrix}$$

The cross-validation results yielded an average accuracy of 87.48% with a standard deviation of ±4.15%, closely matching the performance of the MLP classifier. However, the slightly larger standard deviation suggests that the Bayesian classifier may be less stable in comparison.

2.1.2 Conclusions

Both classifiers exhibited comparable performance on the complex classification problem, achieving average accuracies around 87%. The significant overlap between classes poses challenges for both models, resulting in noticeable misclassifications, particularly for Class 1 and Class 3. The MLP classifier demonstrated slightly higher stability, as evidenced by its lower cross-validation standard deviation, while the Bayesian classifier marginally outperformed in training accuracy. These results suggest that both models are viable but face limitations in handling the inherent complexity of the data distribution.

2.2 Simple Classification Problem Analysis

The data distribution for the simple classification problem shows clearly separated classes, which is shown in **Figure 3**. The boundaries between Class 1, Class 2, and Class 3 are well-defined, making the classification task straightforward. The testing dataset aligns closely with the training dataset, with minimal overlap between classes, allowing the models to generalize effectively.

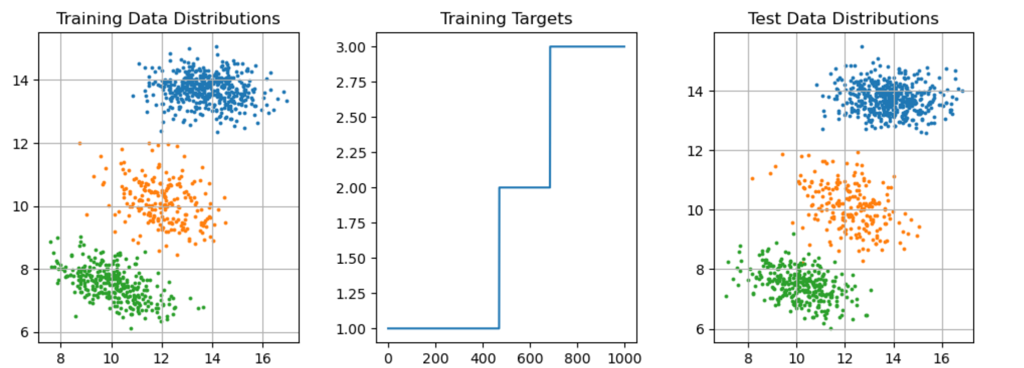

Figure 3: Simple Classification Problem Analysis

2.2.1 Performance of the Classifiers

The Multilayer Perceptron (MLP) classifier achieved a training accuracy of 98.00%. The confusion matrix highlights the effectiveness of the model, with only minor misclassifications observed for Class 2:

$$\begin{bmatrix} 470 & 0 & 0 \\ 20 & 195 & 0 \\ 0 & 0 & 314 \end{bmatrix}$$

The model's performance on the test dataset resulted in an accuracy of 97.60%, confirming its strong generalization ability on well-separated classes.

The Bayesian classifier, based on Gaussian Naive Bayes, achieved a perfect training accuracy of 100.00%, as shown in the confusion matrix:

$$\begin{bmatrix} 470 & 0 & 0 \\ 0 & 215 & 0 \\ 0 & 0 & 314 \end{bmatrix}$$

The test accuracy reached 99.60%, slightly surpassing the MLP classifier, demonstrating the Bayesian classifier's ability to effectively model clearly separated classes with its probabilistic approach.

2.2.2 Conclusions

In the simple classification problem, both classifiers performed exceptionally well, with minimal misclassification. The Bayesian classifier exhibited perfect accuracy on the training data and slightly outperformed the MLP classifier on the test data. These results indicate that the Bayesian classifier is well-suited for problems with well-separated classes due to its reliance on probabilistic assumptions. Meanwhile, the MLP classifier also demonstrated strong performance, with its slight inaccuracies likely attributable to its reliance on iterative optimization and parameter tuning.