EAST WEST UNIVERSITY

Lab Report

on

"Support Vector Machines (SVM) for Classification and Regression."

Course Title: Machine Learning.
Course Code: CSE475
Section: 02

Submitted by:
Md Tariqul Islam
ID:2021-1-60-054

Submitted to:
Dr.Raihan ul Islam(DRUI)
Associate Professor
Department of Computer Science and Engineering
East West University

Submission Date: 15/7/2024

Lab Title

Support Vector Machines (SVM) for Classification and Regression.

Introduction

In this lab, we will dive into the versatility of Support Vector Machines (SVM) by applying them to two distinct machine learning tasks. First, we'll tackle classification by working with the MNIST dataset to classify handwritten digits. Next, we'll shift gears to regression, using the Boston Housing dataset to predict housing prices. Throughout the lab, we'll gain hands-on experience in essential processes such as data preprocessing, model training, hyperparameter tuning, and performance evaluation. This journey will not only enhance our understanding of SVM but also equip us with practical skills applicable to real-world machine learning challenges.

Part 1: SVM Classification for MNIST Handwritten Digits

Learning Objectives

1. Understand SVM Classification Principles: Gain insights into the fundamentals of Support Vector Machines, including how they work and their application in classification tasks.

2. Prepare and Preprocess Image Data for SVM: Learn the necessary steps to prepare image data for SVM classification, including flattening and scaling pixel values.

3. Train SVM Models with Different Kernels: Explore the implementation of SVM models using various kernels—linear, Radial Basis Function (RBF), and polynomial.

4. Evaluate Classification Performance: Assess the performance of each model using metrics such as accuracy and confusion matrices, which provide insights into classification results.

5. Optimize Hyperparameters for Improved Classification: Discover techniques for optimizing model performance through hyperparameter tuning, particularly focusing on the RBF kernel's parameters (C and gamma).

Lab Steps:

1. Load the MNIST Dataset:

- Utilize the load_digits function from scikit-learn to access the dataset. Split the data into training (80%) and testing (20%) sets to facilitate model training and evaluation.

2. Data Preprocessing:

- Flatten the 28x28 pixel images into 784-dimensional vectors to prepare them for SVM processing. Normalize the pixel values to the range [0, 1] to enhance model performance.

3. SVM Model Training:

Construct three distinct SVM models utilizing different kernels:

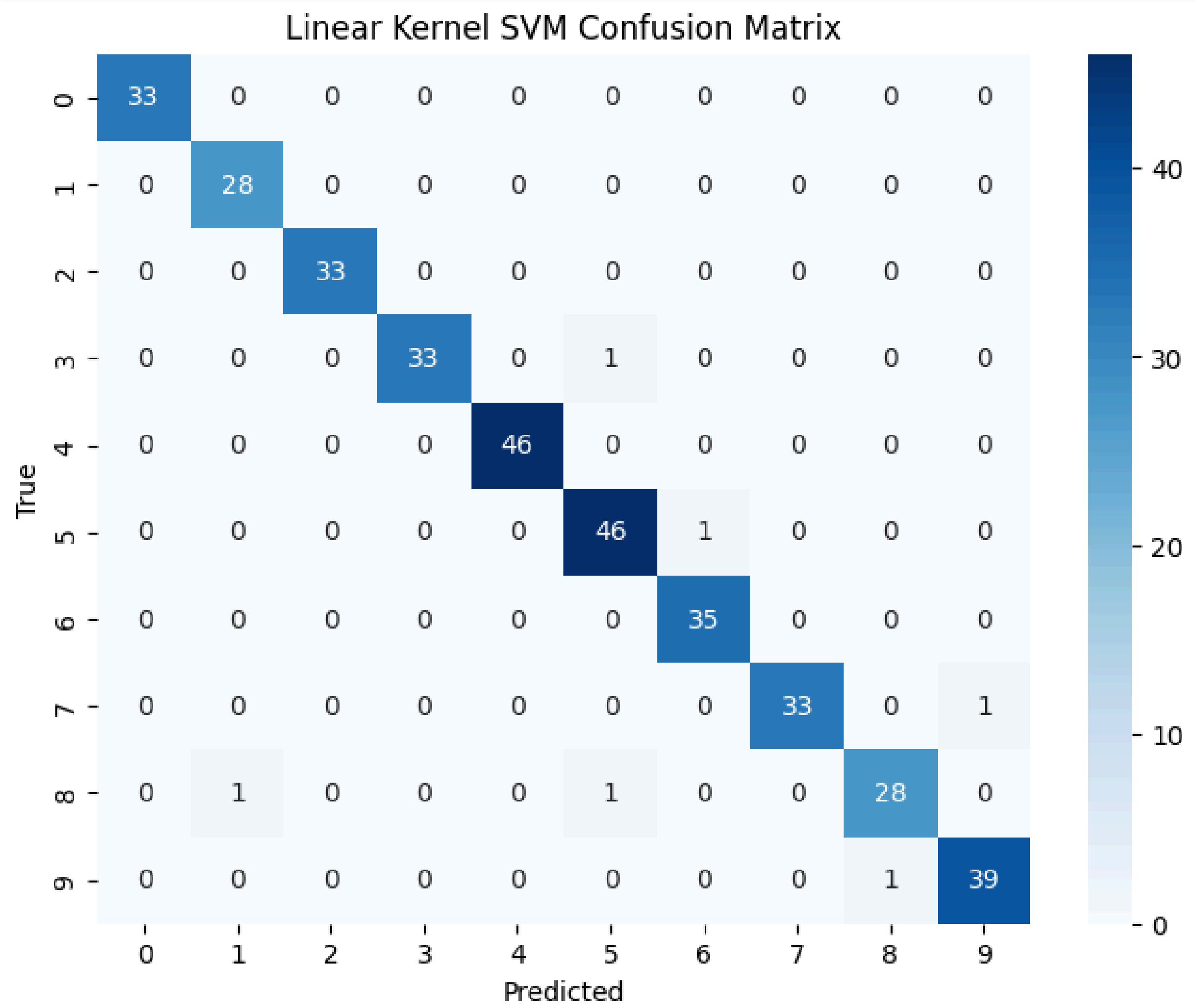
- Linear kernel (kernel='linear')
- Radial Basis Function (RBF) kernel (kernel='rbf')
- Polynomial kernel (kernel='poly', degree=3) Train each model on the prepared training data.

4. Model Evaluation:

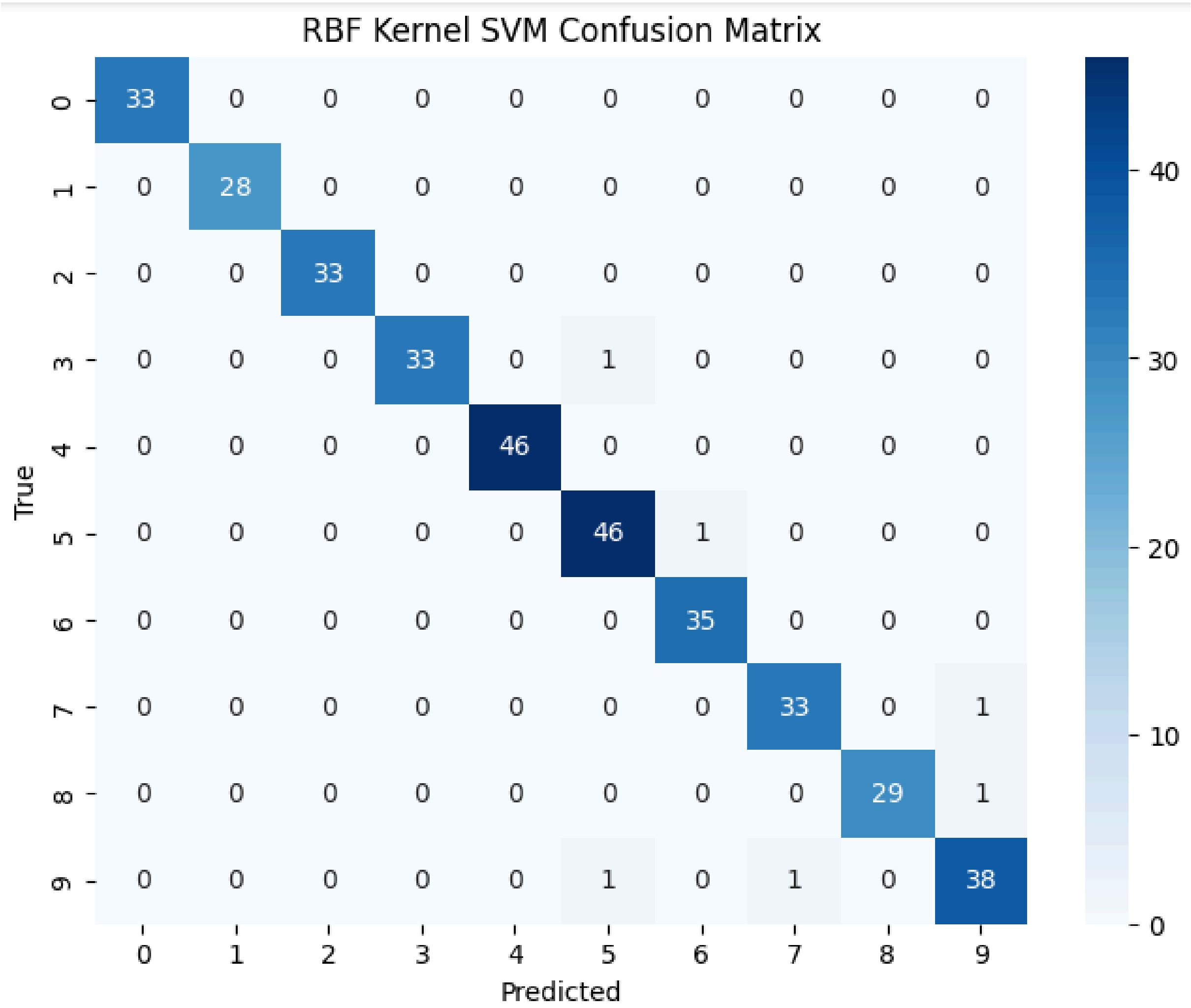
- After training, make predictions on the test set with each model. Calculate and report the accuracy for each model, and generate confusion matrices to interpret classification results comprehensively.

Linear Kernel SVM:

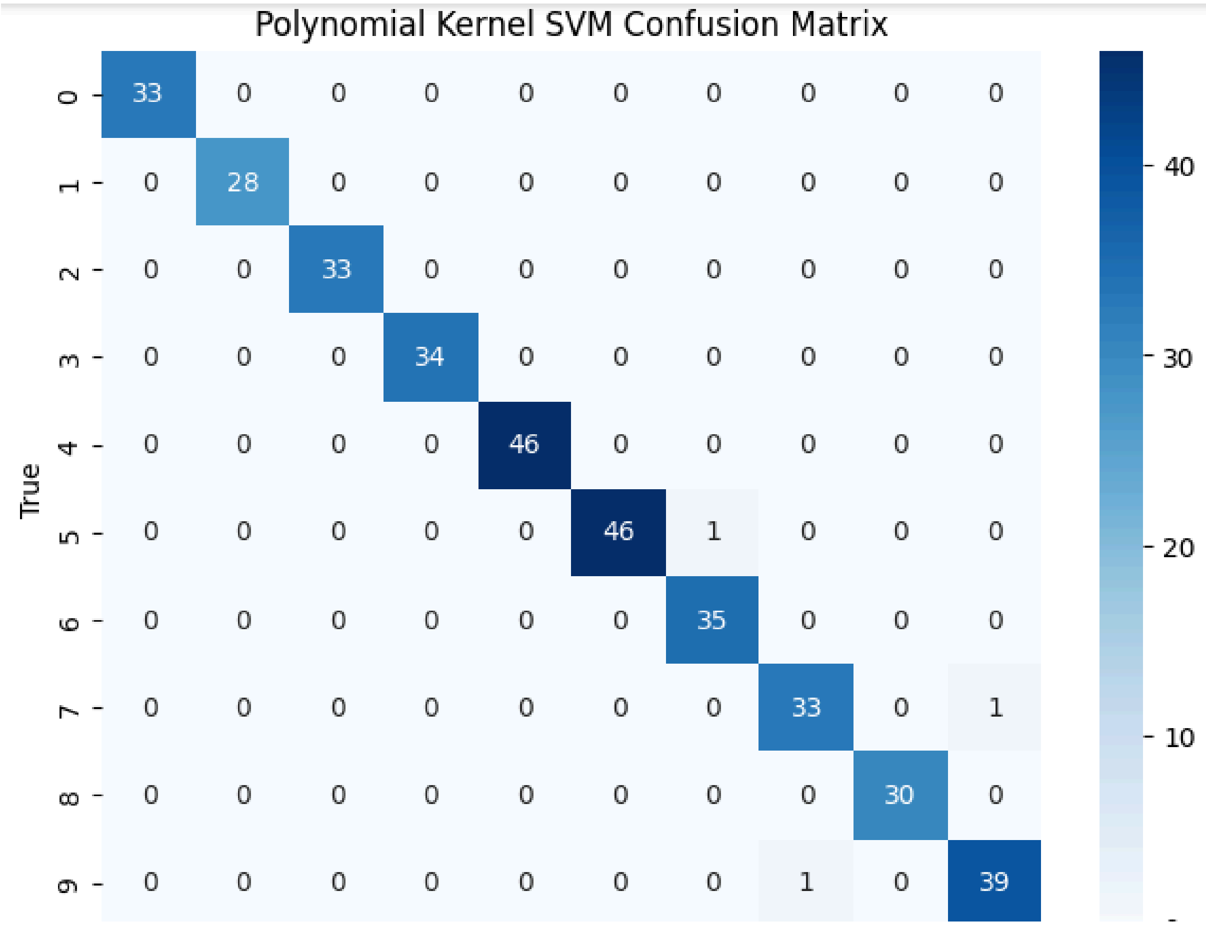
Accuracy: 0.9833



RBF Kernel SVM:
Accuracy: 0.9833



Polynomial Kernel SVM:
Accuracy: 0.9917

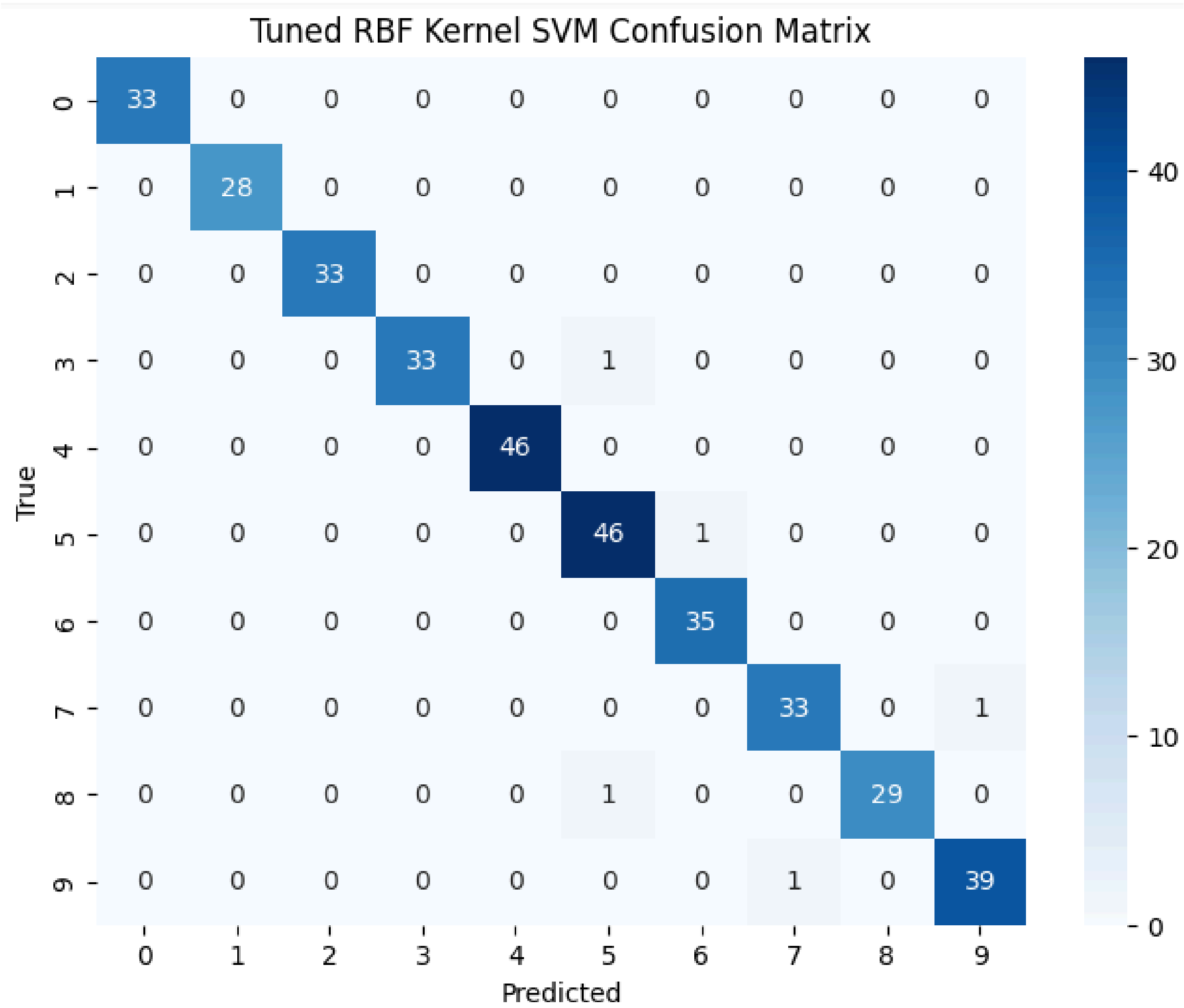


5. Hyperparameter Tuning (Optional):

- Employ grid search or random search methods to identify the optimal hyperparameters (C and gamma) for the RBF kernel. Evaluate the performance of the tuned model and compare it against the baseline models to assess improvements in classification accuracy.

Metrics and parameters for the tuned RBF Kernel SVM model

Metric	Value
Best Parameters	**C: 10,gamma: 0.1,kernel: rbf**
Best Estimator	**SVC(C=10, gamma=0.1)**
Accuracy	**0.9861**



Part 2: SVM Regression for Boston Housing Prices

Learning Objectives

1. Understand SVM regression principles: Gain insight into how Support Vector Machines (SVM) can be adapted for regression tasks, focusing on the concepts of margin and kernel functions.

2. Prepare and Preprocess Housing Data for SVR: Learn the importance of data preprocessing, including feature standardization, to enhance model performance.

3. Train SVR Models with Linear and RBF Kernels: Develop and fit SVM regression models using different kernel types to understand their effects on prediction accuracy.

4. Evaluate Regression Performance Using Relevant Metrics: Assess model performance through metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2).

5. Optimize Hyperparameters for Improved Prediction: Explore techniques like grid search to fine-tune model hyperparameters, enhancing the predictive capabilities of SVR models.

Lab Steps

1. Load the Boston Housing Dataset:

- The dataset can be accessed using the fetch_openml function from sklearn.datasets. It is crucial to split the dataset into training (80%) and testing (20%) sets to evaluate model performance accurately.

2. Data Preprocessing:

- Standardization of features is performed using StandardScaler. This step is vital as it scales the input data, ensuring that each feature contributes equally to the model's learning process.

3. SVR Model Training:

- Two SVR models are created: one utilizing a linear kernel (kernel='linear') and the other employing a Radial Basis Function (RBF) kernel (kernel='rbf'). Each model is trained on the scaled training data.

4. Model Evaluation:

Predictions are made on the test set using both models. Performance is quantified using the following metrics:

- **Mean Squared Error (MSE)**: Measures the average squared difference between predicted and actual values, emphasizing larger errors.
- **Mean Absolute Error (MAE)**: Computes the average absolute difference between predicted and actual values, providing a straightforward measure of prediction accuracy.
- **R-squared (R^2)**: Indicates the proportion of variance in the target variable explained by the model, offering insight into model fit.

SVR Linear Kernel model

Metric	Value
Mean Squared Error (MSE)	**28.92**
Mean Absolute Error (MAE)	**3.10**
R-squared (R^2)	**0.61**

SVR RBF Kernel model

Metric	Value
Mean Squared Error (MSE)	**25.67**
Mean Absolute Error (MAE)	**2.73**
R-squared (R^2)	**0.65**

SVR RBF kernel model tuned with Best Parameters

Metric	Value
Best Parameters	**C: 100, gamma: 0.1**
Mean Squared Error (MSE)	**12.55**
Mean Absolute Error (MAE)	**2.17**
R-squared (R^2)	**0.83**

5. Hyperparameter Tuning (Optional):

- Grid search is utilized to identify optimal hyperparameters, specifically the regularization parameter C and the kernel coefficient gamma for the RBF kernel. This process involves systematically evaluating model performance across different combinations of hyperparameters, ensuring the best model configuration is selected.

In this exploration of SVM for both classification and regression tasks, we have gained valuable insights into the strengths and weaknesses of Support Vector Machines, as well as the significance of kernel functions and hyperparameters.

Classification Task: MNIST Handwritten Digits	**Regression Task: Boston Housing Prices**
1.Strengths: SVM proved effective for classifying handwritten digits, particularly due to its ability to handle high-dimensional data with a clear margin of separation. The use of different kernels (linear, RBF, polynomial) allowed for flexibility in adapting to the complexity of the data.	**1.Strengths:** SVM regression (SVR) effectively captured relationships in the housing price data, with the ability to handle non-linear relationships through kernel functions. The flexibility of kernel choice allowed for tailored modeling to the data structure.
2.Weaknesses: SVM can be computationally expensive, especially with large datasets, and may struggle	**2.Weaknesses:** Similar to classification, SVR can be sensitive to outliers, which may

with imbalanced classes unless appropriately handled. The choice of kernel and hyperparameters significantly impacts performance, necessitating careful tuning.	disproportionately affect performance metrics. Additionally, the need for feature standardization underscores the importance of data preprocessing.
3.Impact of Kernels:The linear kernel performed well with simpler decision boundaries, while the RBF kernel excelled in capturing more complex relationships in the data. The polynomial kernel provided intermediate performance, showcasing the trade-offs between model complexity and overfitting.	**3.Impact of Kernels:**The linear kernel was suitable for simpler relationships, while the RBF kernel captured more complex, non-linear patterns, illustrating the importance of kernel selection based on data characteristics.
4.Hyperparameter Sensitivity: The hyperparameter tuning, especially for the RBF kernel (C and gamma), demonstrated that optimal values can significantly enhance accuracy. The exploration of different hyperparameter combinations is critical for achieving the best model performance.	**4.Hyperparameter Sensitivity:**The optimization of hyperparameters (C and gamma) was crucial for improving model accuracy and generalization. Grid search provided a systematic approach to identifying the best parameters, demonstrating the importance of this step in the modeling process.

Potential Areas for Further Exploration:

1. **Advanced Kernels**: Investigating other kernel functions such as sigmoid or custom kernels could provide additional insights into model performance for specific datasets.
2. **Feature Engineering**: Exploring feature selection and engineering techniques could further improve model accuracy by focusing on the most informative variables.
3. **Scalability and Speed**: Researching methods to enhance SVM's scalability, such as using stochastic optimization techniques or approximations, could make it more viable for large datasets.
4. **Ensemble Methods**: Combining SVM with ensemble techniques like bagging or boosting may yield improved robustness and performance in both classification and regression tasks.

Conclusion:

SVM is a powerful tool for both classification and regression tasks, with distinct advantages depending on the application. Understanding kernel selection and hyperparameter tuning is vital for leveraging SVM's full potential in real-world predictive modeling scenarios.