Mateo W. Racca

UNIVERSITAT ROVIRA i VIRGILI
Escola Tècnica
Superior d'Enginyeria

NEURAL AND EVOLUTIONARY COMPUTING

(MESIIA): Assignment #2: Classification with SVM, BP and MLR

Part 1

Dataset 1 (Ring Dataset) and Dataset 2 (Bank Dataset)

Data Normalization

Type of Normalization: For both datasets, Z-score normalization (Standard Scaling) was applied to the input variables.

Ring Dataset: given that the dataset is intended for a classification task with a model sensitive to the scale of input features (e.g., SVM), standardizing the features ensures they contribute equally to the distance calculations. This step is crucial because differences in scale among features can significantly impact the performance of such models.

Bank Dataset: as this dataset contains a mix of categorical and numerical features. Numerical features were standardized for similar reasons as the Ring dataset. For the categorical features, one-hot encoding was applied to convert them into a numerical format, facilitating their use in machine learning algorithms.

Handling Output Variables: no specific normalization was applied to the output variables as they are used for classification tasks (binary or multiclass) and are typically encoded as discrete labels.

Dataset 3 (Diabetes Dataset)

Data Source: the Diabetes Dataset was sourced from https://www.kaggle.com/datasets/mathchi/diabetes-data-set#

The dataset was examined for any missing values to ensure data integrity. Handling missing data is crucial as they can lead to inaccurate models and biased analyses. Regarding outliers, the dataset was reviewed as outliers can distort the results of an analysis and affect the performance of many machine learning models.

Data Normalization: Z-score normalization was applied to the input variables. This step was taken to standardize the scale of the numerical features, enhancing the performance of machine learning algorithms that are sensitive to the scale and distribution of the input variables.

In summary, the preprocessing steps for these datasets were carefully selected based on their specific characteristics and the requirements of the expected analysis or machine learning tasks. Normalization, in particular, plays a critical role in preparing the data for algorithms that rely on distance metrics. Additionally, handling categorical values, missing values, and outliers ensures that the datasets are well-suited for robust and accurate predictive modeling.

Part 2

Datasets: The results of the preprocessing of the three datasets (Ring, Bank, and Diabetes) was splitted for training and testing.

Importing Libraries: At the beginning of the script, I've included essential Python libraries such as Pandas for data manipulation, NumPy for numerical operations, and various modules from scikit-learn (sklearn) for machine learning tasks. Additionally, I've imported libraries like Matplotlib for creating plots and Seaborn for enhanced visualization.

Function Definitions: I've defined three critical functions within the script:

- split_features_target(df): This function extracts features and the target variable from a Pandas DataFrame. It also handles potential missing values in the target and ensures it's in a numeric format when needed.
- visualize_data(X, y, title): This function employs Principal Component Analysis (PCA) for dimensionality reduction and generates scatterplots to visualize data points, allowing for easy insights into dataset distribution.
- classification_error(y_true, y_pred): Here, I've created a function to calculate the classification error rate, which is essential for assessing model performance.

Model Initialization: I've carefully initialized three machine learning models:

- svm_model: A Support Vector Machine (SVM) model configured with an RBF kernel, probability estimates, and specific hyperparameters like C and gamma.
- mlp_model: A Multi-Layer Perceptron (MLP) Classifier with a specified neural network architecture and training parameters.
- log_reg_model: A logistic regression model with polynomial features and regularization settings.

Model Dictionary: To easily manage these models throughout the script, I've organized them into a dictionary named models, associating each with a human-readable name.

Dataset Loop: In the heart of the script, I've structured a loop to handle multiple datasets. Here's what it does:

- Data Preprocessing: I've accounted for missing values by removing rows with incomplete data in both training and test datasets. This ensures clean, usable data for model training and evaluation. Standardization of data is also performed using StandardScaler for consistency in feature scaling.
- Model Training: For each dataset, I've trained each of the three machine learning models on the preprocessed training data using the fit() method.
- Model Evaluation: I've calculated crucial metrics such as accuracy, classification error, and confusion matrices to gauge how well the models perform on the test data.
- Data Visualization: To gain insights, I've harnessed PCA to reduce feature dimensions, and I've created scatterplots to visually represent the data in both training and test sets.
- For specific models like MLP and Logistic Regression, I've gone a step further by computing and plotting ROC curves and AUC scores, providing a more comprehensive view of model performance in binary classification scenarios.

Results:

Ring dataset

SVM Accuracy on Ring: 0.9791

SVM Classification Error on Ring: 2.09%

SVM Confusion Matrix on Ring: [[5308 25]

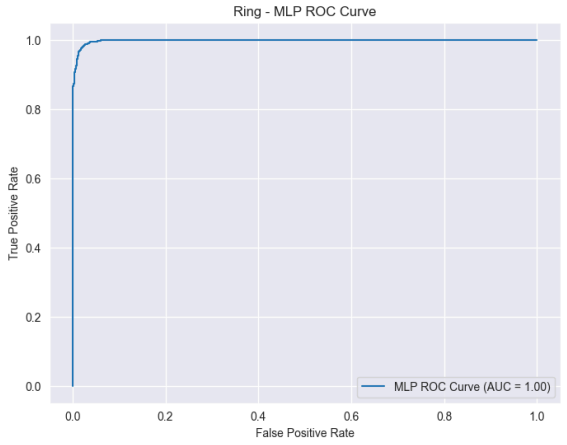
[184 4483]]

MLP Accuracy on Ring: 0.9601

MLP Classification Error on Ring: 3.99%

MLP Confusion Matrix on Ring: [[5292 41]

[358 4309]]

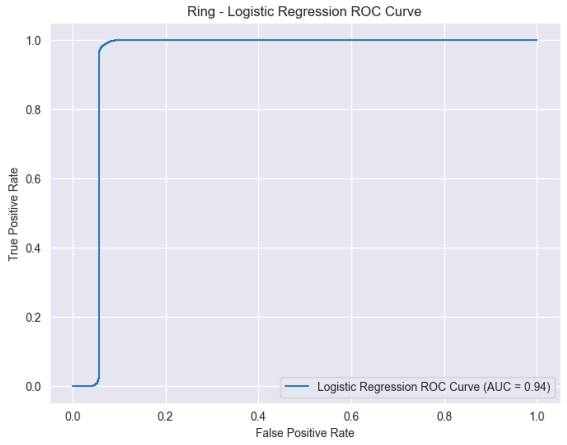


Logistic Regression Accuracy on Ring: 0.9086

Logistic Regression Classification Error on Ring: 9.14%

Logistic Regression Confusion Matrix on Ring: [[5033 300]

[614 4053]]



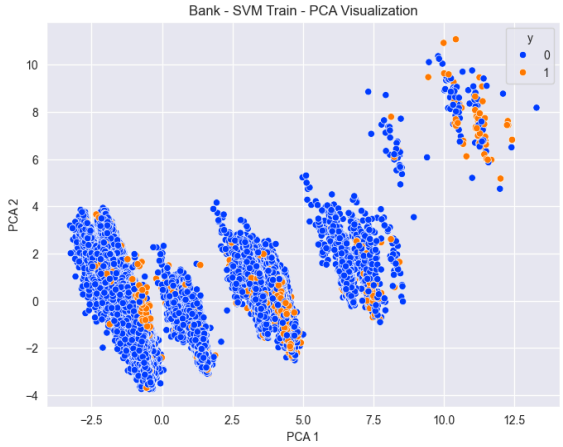
Bank dataset

SVM Accuracy on Bank: 0.6939791211459092

SVM Classification Error on Bank: 30.60208788540908%

SVM Confusion Matrix on Bank: [[5584 114]

[2407 133]]

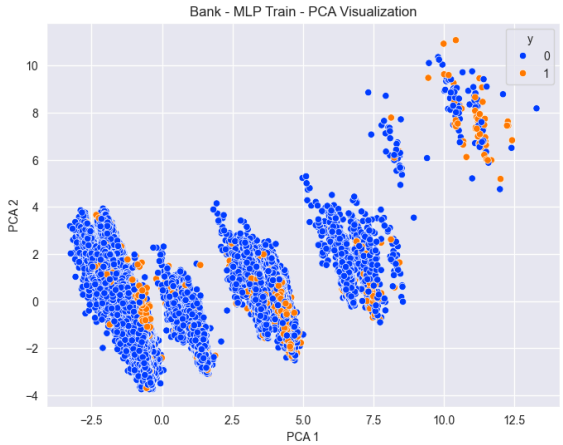


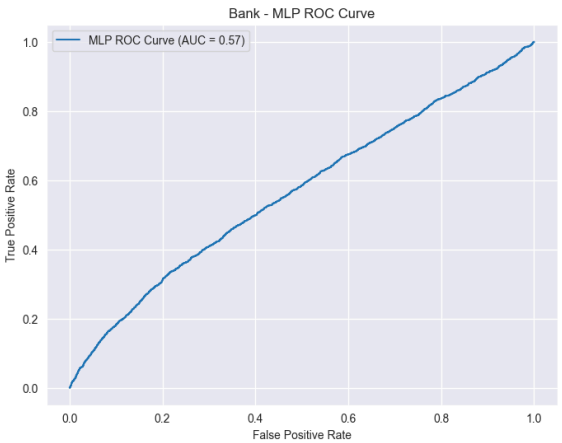
MLP Accuracy on Bank: 0.682689973294489

MLP Classification Error on Bank: 31.731002670551106%

MLP Confusion Matrix on Bank: [[5199 499]

[2115 425]]



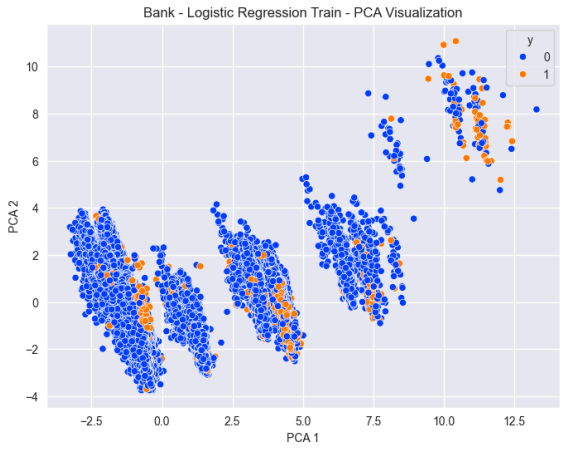


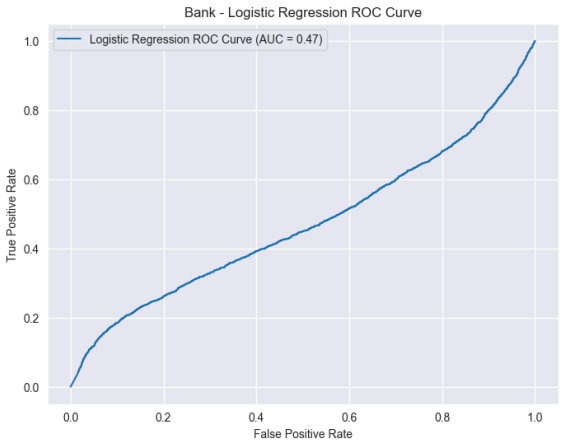
Logistic Regression Accuracy on Bank: 0.6624180626365622

Logistic Regression Classification Error on Bank: 33.75819373634377%

Logistic Regression Confusion Matrix on Bank: [[4895 803]

[1978 562]]





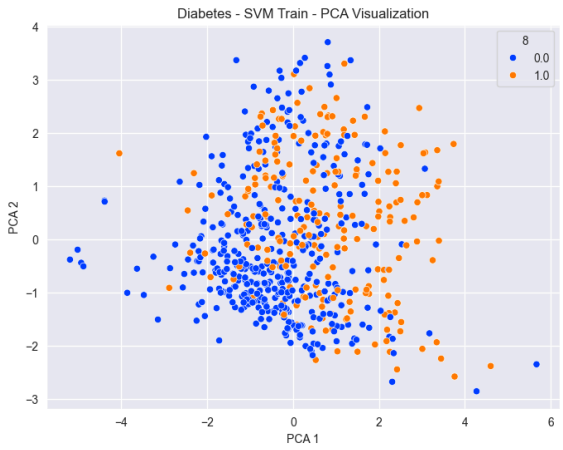
Diabetes dataset

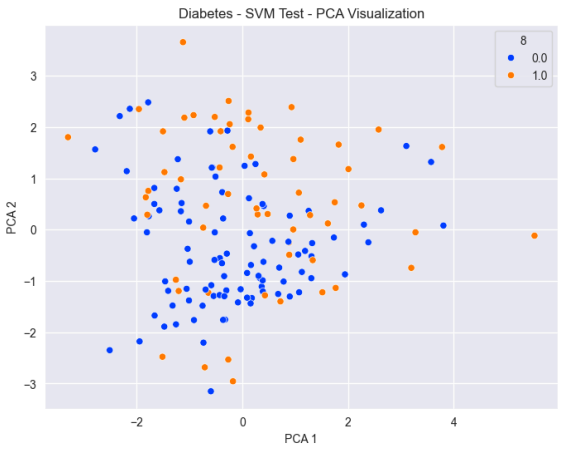
SVM Accuracy on Diabetes: 0.7792207792207793

SVM Classification Error on Diabetes: 22.07792207792208%

SVM Confusion Matrix on Diabetes: [[84 12]

[22 36]]



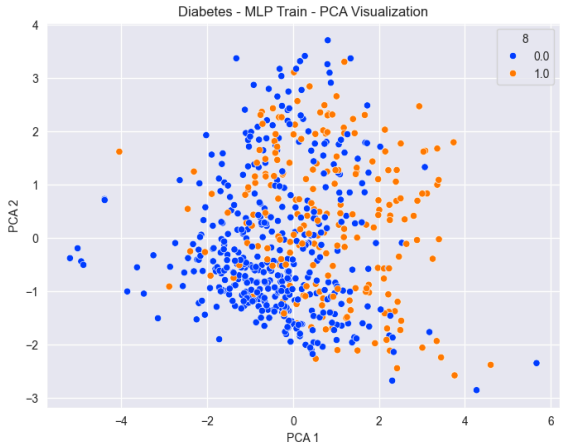


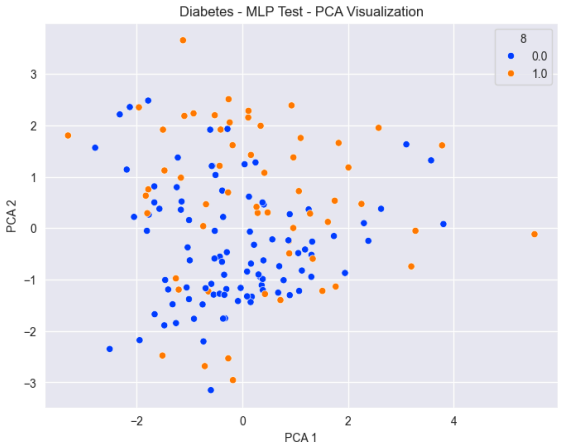
MLP Accuracy on Diabetes: 0.7467532467532467

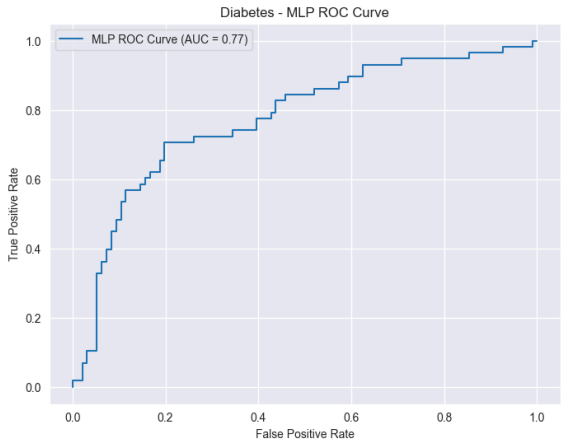
MLP Classification Error on Diabetes: 25.324675324675326%

MLP Confusion Matrix on Diabetes: [[77 19]

[20 38]]





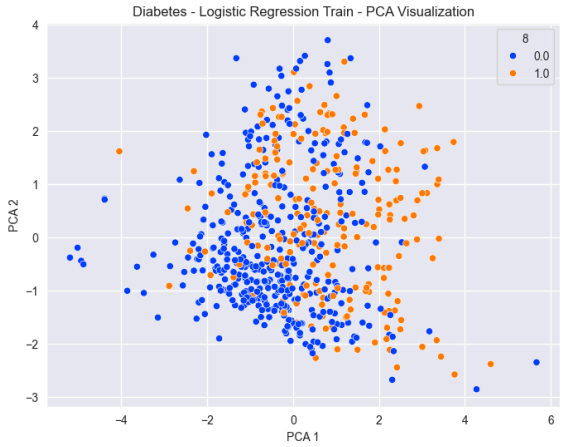


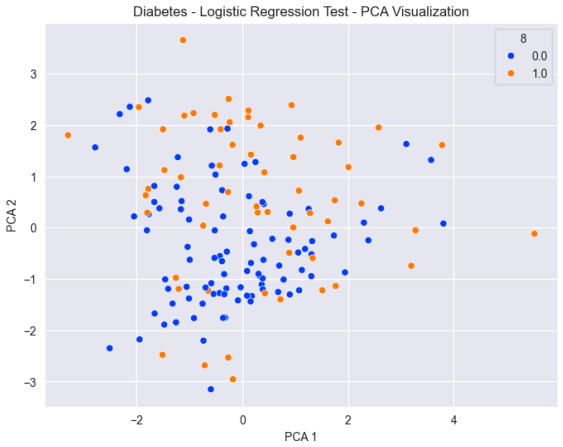
Logistic Regression Accuracy on Diabetes: 0.6883116883116883

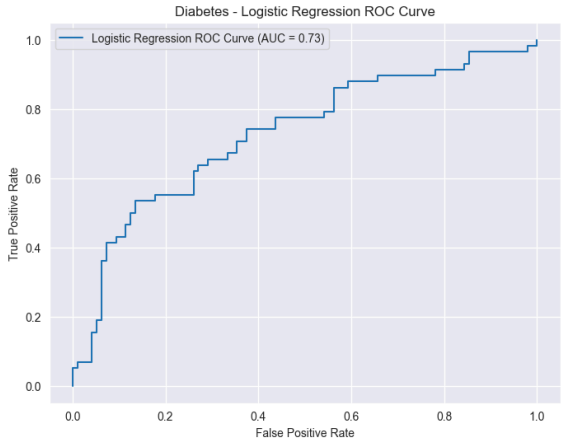
Logistic Regression Classification Error on Diabetes: 31.16883116883117%

Logistic Regression Confusion Matrix on Diabetes: [[74 22]

[26 32]]







Conclusion

The provided results reveal distinct performance characteristics of the SVM, MLP, and Logistic Regression models across the Ring, Bank, and Diabetes datasets.

Ring Dataset: The SVM model outperforms others with a high accuracy of 97.91% and a low classification error. MLP also performs well, but with slightly lower accuracy and higher error. Logistic Regression lags behind in both accuracy and error rate, indicating its lesser suitability for this dataset.

Bank Dataset: All models show reduced performance compared to the Ring dataset. SVM leads with about 69.4% accuracy, followed closely by MLP and then Logistic Regression. The higher classification errors across all models suggest more complexity or noise in this dataset.

Diabetes Dataset: Again, SVM performs the best among the three, with approximately 77.9% accuracy. MLP's accuracy is moderately lower, and Logistic Regression shows the least accuracy. The errors are higher for all models compared to the Ring dataset but lower than the Bank dataset.

These results highlight the importance of model selection based on dataset characteristics. SVM consistently shows the best performance, suggesting its robustness, especially in handling complex patterns. The varying performance of MLP and Logistic Regression across datasets indicates their sensitivity to data features and distribution.