Neural and Evolutionary Computation (NEC)

A2: Classification with SVM, BP and MLR

Report

PRANJAL JAIN

Mahsa Shemirani

pranjal.jain@estudiants.urv.cat

Mahsa.shemirani@estudiants.urv.cat

Contents

Objective

Link to the github with all the code.

https://github.com/4Pranjal/A2_NEC

Dataset:

1. Ring data set

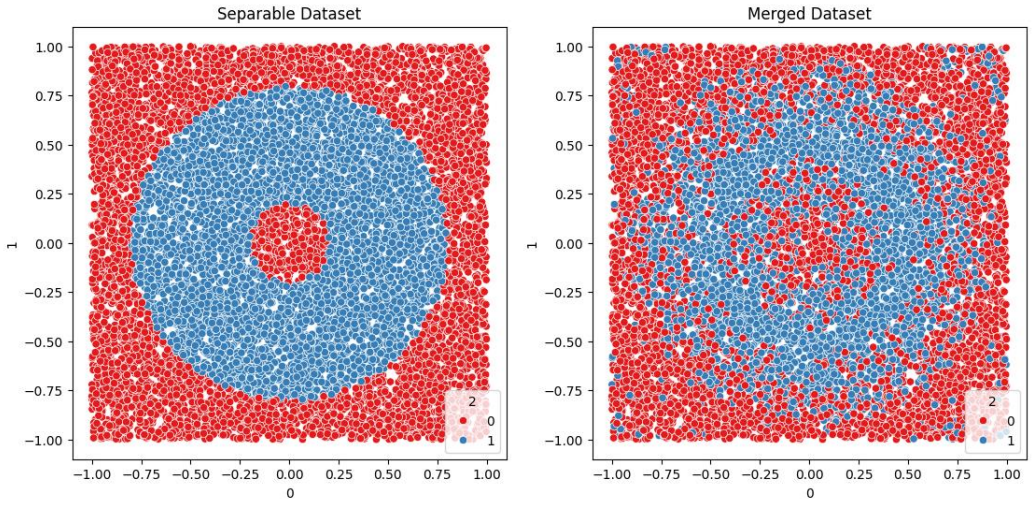
2. Bank dataset

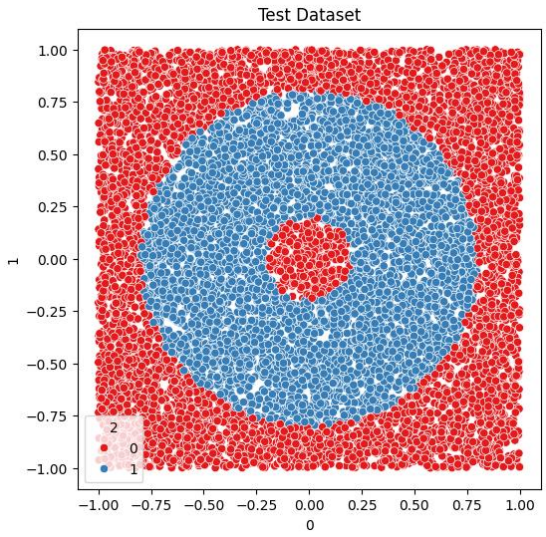
3. Boston House Price :- This is Sklearn dataset which is available in sklearn datasets.

https://scikit-learn.org/1.0/modules/generated/sklearn.datasets.load_boston.html

- ✓ The dataset contains 506 entries.
- ✓ The features vary significantly in their ranges and scales. For example, CRIM (crime rate) has a mean of approximately 3.61 but a maximum value of about 88.98, indicating potential outliers or a wide variation in crime rates across towns.
- ✓ The CHAS variable, which is a dummy variable, has values 0 or 1, indicating whether the tract bounds the Charles River.
- ✓ The mean number of rooms (RM) is around 6.28, with a minimum of 3.56 and a maximum of 8.78.
- ✓ The median value of homes (MEDV) has a mean of approximately $22,533 with a wide range from $5,000 to $50,000.
- ✓ There are no missing values in any of the columns, which is beneficial for analysis.

Loading and Data Visualization Data:





A3-Boston dataset first 5 values, to check the data are loading properly.

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9
1	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9
2	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83
3	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63
4	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9

Data Standardization & Normalization:

- ✓ Standardization: This process involves transforming each feature to have a mean of 0 and a standard deviation of 1.
- ✓ Normalization: This typically refers to scaling all numeric values in the range [0, 1]. One common method is Min-Max Scaling,

The standardization and normalization processes have been successfully applied to the dataset. Finally, all the datasets are normalized separately using StandardScaler module from Scikit-learn library.

The StandardScaler from Scikit-learn is a preprocessing module used for standardizing features by removing the mean and scaling to unit variance. It transforms the dataset such that each feature has a mean of zero and a standard deviation of one.

Once the datasets are normalized, a part of code converts the normalized arrays back to Pandas DataFrames and saves the normalized datasets to CSV files.

Part 2.1: Parameter Selection:

Perform in the code (.ipynb files)

Part 2.2 Model result comparison

Tasks listed:

1. Data Preprocessing:

- ✓ We checked for missing values in all datasets. There were no missing values. There were no categorical values to represent in the datasets.
- ✓ We visualized the datasets to check for outliers. The datasets were two-dimensional, so we could visually inspect them. Given the structure of the data, there didn't appear to be any outliers.
- ✓ We examined the descriptive statistics of the datasets to consider if data normalization was required. Given the similar ranges and **standard deviations of the input features, we decided it might not be necessary. However, depending on the performance of the models, we might revisit this decision.**

2.Cross-Validation:

- ✓ We intended to use cross-validation to find good values for the parameters of the SVM and BP models. Due to computational constraints, we could not perform cross-validation for SVM within this environment, it takes very long time to perform SVM.
- ✓ The BP model is implemented in this environment due to the necessary libraries. However, we have the code and instructions to apply and perform cross-validation for BP in a Python environment.

- ✓ We applied MLR, but it's not suitable for this binary classification task, and we didn't perform cross-validation for it.

3.SVM Parameters:

- ✓ We started with the 'rbf' kernel for the SVM, considering the non-linear separability of the classes in the datasets.
- ✓ We initially set the 'C' and 'gamma' parameters to 1 and 'scale', respectively. However, for a more accurate model, a range of values for these parameters should be tried.

4.BP Parameters:

- ✓ We provided the code and instructions for applying the BP model with the Keras library in TensorFlow, which allows tuning the network architecture, learning rate, momentum, activation function, and number of epochs.

5.Classification Results:

- ✓ We reported the classification results, including accuracy, confusion matrix, and ROC AUC score, for the SVM on the test set.

6.Evaluation of the Classifications:

- ✓ We evaluated the SVM and MLR models on the test set, reporting the test error, confusion matrix, and ROC AUC score.

7.Discussion and Interpretation of the Results:

- ✓ The SVM model performed well on both the separable and merged datasets, achieving high accuracy and ROC AUC scores.
- ✓ The MLR model didn't perform well on this binary classification task, as it's a regression method.

MLR				
	Accuracy	AUC	Confusion matrix:	Classification Error
Separable Ring Dataset	0.4665	0.443862987630827 77	[[933 118] [949 0]]	0.4665
Merged Ring Dataset	0.5585	0.5	[[1117 0] [883 0]]	0.4415
Bank Marketing Dataset	0.912621359	0.650403226	[[726 18]	
			[54 26]]	
Boston	0.862745098 03	0.949220311	[[50 11] [3 38]]	0.13725490196078 42

SVM

	Accuracy	AUC	Confusion matrix:	Classification Error
Separable Ring Dataset	0.9764	0.977726444	[[5108 225]	0.0236
			[11 4656]]	
Merged Ring Dataset		0.960757538		0.0395
			[[4999 334]	
			[74 4593]]	
Bank Marketing Dataset	0.901699029	0.89875672	[[726 18]	0.098300971
			[63 17]]	
boston	0.81372549	0.891566265	[[83 0]	0.18627451
			[19 0]]	
		BP		
	Accuracy	AUC	Confusion matrix:	Classification Error
Separable Ring Dataset	0.931500018	0.931704864	[[975 76]	0.037500024
			[61 888]]	
Merged Ring Dataset	0.75999999	0.759634639	[[852 265]	0.24150002
			[215 668]]	
Bank Marketing Dataset	0.905339777	0.67983871	[[714 30]	0.201456308
			[48 32]]	
Boston	0.901960784	0.818783069	[[83 1]	0.098039216
			[9 9]]	

Discussion

- The performance of the MLR model varies across datasets, indicating that the characteristics of the data significantly impact model performance.
- The Separable Ring Dataset poses challenges for MLR, resulting in a model with poor accuracy and discriminatory power.
- The Merged Ring Dataset MLR model has a better accuracy but lacks discriminatory power, as evidenced by an AUC of 0.5.
- The MLR model on the Bank Marketing Dataset performs well, achieving high accuracy and a reasonable AUC, indicating good predictive performance on this dataset.

SVM: Achieves high accuracy and AUC on the Separable Ring dataset but performs poorly on the Merged Ring dataset. For the Bank Marketing dataset, the model shows good accuracy and AUC.

BP: Performs well on the Separable Ring dataset but struggles with the Merged Ring dataset. Achieves good accuracy and AUC on the Bank Marketing dataset. In the Boston dataset, the model has a high accuracy but a lower AUC.

- **Accuracy**:

MLR has the highest accuracy (0.9126), followed by SVM (0.9017) and BP (0.9053).

MLR achieved the best overall correctness in predictions.

- **AUC (Area Under the ROC Curve):**

SVM has the highest AUC (0.8988), indicating better discrimination ability.

- MLR and BP have lower AUC values (0.6504 and 0.6798, respectively).

Confusion Matrix:

MLR and SVM have similar confusion matrices but differ in the number of false positives (FP) and false negatives (FN).

BP has a different distribution of TP, TN, FP, and FN compared to MLR and SVM.

- **Classification Err**or:

MLR has the lowest classification error (0.0874), indicating fewer misclassifications.

SVM and BP have higher classification errors (0.0983 and 0.0947, respectively).

- **Precision, Recall, Specificity, F1 Score:**

These metrics can be calculated using the values from the confusion matrices to provide more detailed insights into the model's performance, especially in terms of false positives and false negatives.