Table of Contents

[Project Overview 1](#_Toc158899940)

[Problem Statement 1 1](#_Toc158899941)

[Question 1](#_Toc158899942)

[Code 1](#_Toc158899943)

[Output 1](#_Toc158899944)

[Problem Statement 2 3](#_Toc158899945)

[Question 3](#_Toc158899946)

[Code 3](#_Toc158899947)

[Output 3](#_Toc158899948)

[Problem Statement 3 3](#_Toc158899949)

[Question 3](#_Toc158899950)

[Code 3](#_Toc158899951)

[Output 3](#_Toc158899952)

[Observations 4](#_Toc158899953)

[Problem Statement 4 5](#_Toc158899954)

[Question 5](#_Toc158899955)

[Code 5](#_Toc158899956)

[Output 5](#_Toc158899957)

[Observations 6](#_Toc158899958)

[Problem Statement 5 7](#_Toc158899959)

[Question 7](#_Toc158899960)

[Code 8](#_Toc158899961)

[Output 8](#_Toc158899962)

[Observations 10](#_Toc158899963)

[Project 2 Conclusion 10](#_Toc158899964)

[Appendix 11](#_Toc158899965)

[Problem Statement 1 11](#_Toc158899966)

[Code 11](#_Toc158899967)

[Problem Statement 2 12](#_Toc158899968)

[Code 12](#_Toc158899969)

[Problem Statement 3 13](#_Toc158899970)

[Code 13](#_Toc158899971)

[Problem Statement 4 14](#_Toc158899972)

[Code 14](#_Toc158899973)

[Problem Statement 5 15](#_Toc158899974)

[Code 15](#_Toc158899975)

**List of Tables**

[Table 1: Split of Data into Training, Validation and Testing with values. 1](#_Toc160069853)

[Table 2: Cross-Validation Scores for each iteration and the average. 2](#_Toc160069854)

# Project Overview

The project 3 aims to expand learning through practice upon the concepts of Neural Network multilayer perceptron learning algorithms with focus upon MLPClassifier and the MLPRegressor.

# Problem Statement 1

## Question

The expectation is to write a python code that splits the Wisconsin Breast Cancer data set available in the *Sklearn* dataset library into Training/Validation and Testing subsets through a split ratio of 80% and 20% respectively. The key objective is to mention and describe how the split is conducted along with providing the random value used for the split for the purpose of recreation.

## Code Snippet

The following is the code snippet for the question given:



The entire code block for the Problem Statement 1 can be found in the appendix within the Code 1 of this document.

## Output



The above output clearly identifies the total number of samples in the Wisconsin Breast Cancer Dataset which is 569 and the number of samples divided for Training and Validation is 455 i.e. 80% (79.96%). Additionally the Training and Validation is further split in half with 227 samples shared with Training and 228 samples shared with Validation. The number of samples divided for Testing is 114 i.e. 20% (20.03%). The random state value provided for the split is given as 19.

|  |  |  |
| --- | --- | --- |
| **Category** | **Percentage split** | **Number of rows (samples)** |
| Training and Validation | 80% | 455 (227+228) |
| Training | 50% of 80% | 227 |
| Validation | 50% of 80% | 228 |
| Testing | 25% | 114 |

Table 1: Split of Data into Training, Validation and Testing with values.

# Problem Statement 2

## Question

The requirement here is to either create an additional split for the validation sub-dataset or to use cross validation technique to tune the overall program’s MLP Hyperparameters.

## Code Snippet

The following is the code snippet for the question given:



The entire code block for the Problem Statement 2 can be found in the appendix within the Code 1 of this document.

## Output



The output at first displays the cross-validation scores with a cross validation count of number of folds is given as 10. This means that the validation subset is divided into 10 equal folds and each fold is compared with the remaining 9 other to evaluate the model’s performance based on 10 iterations (in this case). The average value of these 10-fold iterations is also displayed and the value obtained here is 0.9164. This shows that the model is already leaning on a good learning curve.

|  |  |
| --- | --- |
| **Iteration Count** | **Cross Validation Scores** |
| 1 | 0.97826087 |
| 2 | 0.91304348 |
| 3 | 0.93478261 |
| 4 | 0.82608696 |
| 5 | 0.93478261 |
| 6 | 0.91111111 |
| 7 | 0.91111111 |
| 8 | 0.95555555 |
| 9 | 0.86666667 |
| 10 | 0.93333333 |
| **Average** | **0.9164734299516908** |

Table 2: Cross-Validation Scores for each iteration and the average.

# Problem Statement 3

## Question

The expectation with this question is to provide a procedure to document the design process and to identify and mention the tradeoffs considered in using a MLPClassifier.

## Design Process Procedure

The following are the general steps involved (along with short descriptions in relation to the Project 3 fulfillment) in the establishment of design as a precursor to the MLPClassifier:

1. Problem Definition:

In this case, the problems are clearly defined in what is to be achieved along with the necessary inputs (Ex. Choice of Dataset), procedural steps and the necessary outputs (Ex. MLP and Cross Validation scores).

1. Data Collection and Preprocessing:

This step involved prepping the data (cleaning, normalization etc.) before operations can be done using it. However, in this case, since we are already using a prepped dataset such as the Wisconsin Breast Cancer Dataset from the Sklearn dataset library, it is directly used for operations.

1. Model Architecture:

The model architecture focusses on the finer details of how the data is to be analyzed based on a certain set of hyperparameters such as number of layers with the number of neurons in each layer along with the number of epochs that is to be conducted et al.

1. Training and Hyperparameter tuning:

This stage of the process trains the model from the split training sub-dataset and tuning the hyperparameters such as the number of neurons, learning rate etc. The usage of cross validation technique is also considered for hyperparameter to achieve the best score for the model.

1. Model Evaluation:

This step focusses on the model’s performance upon the unseen data i.e. testing sub-dataset. This evaluation is done by using metrics such as scores and the expected efficiency is obtained based on the problem statement’s requirement.

1. Documentation

This stage is a vital step of the process since it involves documenting the entire process procedure, the decisions made and the tuning of the hyperparameters to achieve desired results and to satisfy the problem statement. This step also provides an insight upon the entire activity conducted from the start till the end.

## Considered MLPClassifier tradeoffs

1. Increasing the complexity of a model does improve the model’s efficiency and its ability to solve complex problems, however, tuning the complexity of a model greater than a limit causes overfitting in the model.
2. Increasing the number of layers or the number of neurons improves the performance of the model but also increases the training time for any dataset.
3. MLP Classifiers are black box systems due to their complexity and thus leaves little room for interpretation and decision making in complex scenarios. However, the replacement for MLP Classifiers can either be linear classifiers or decision trees which can interpret much easily. But the performance of an MLP Classifier is unmatched.
4. Data Availability affects the model complexity as well i.e. with abundant labelled examples from the dataset, complex models like MLP Classifiers can be used else there is a risk of overfitting when the dataset isn’t large and unlabeled.

# Problem Statement 4

## Question

The requirement here is to use an MLP Classifier to train, validate and test an MLP Model. Any number of features from the dataset can be used. The random state value is to be mentioned for the purpose of model recreation.

## Code Snippet



The entire code block for the Problem Statement 4 can be found in the appendix within the Code 1 of this document.

## Output



It can be observed that the output prints the score of all the all the three subsets separately.

# Problem Statement 5

## Question

The requirement is to provide a list of hyperparameters that were used as default values as input along with

## Code

The entire code block for the Problem Statement 5 can be found in the appendix of this document.

## Output

The program runs over the dataset of Breast Cancer from the *SKLearn* repository. The following are the plots obtained for the three good and three poor performing features. Both the features that perform a good job and a poor job are separated by classes and are plotted alongside each other.

A graph of blue squares

Description automatically generated

Figure 4: Three features that do a good job as per their Logistic Regressor Coefficients

Figure 4 has the score of each feature’s logistic regressor coefficient upon the bar chart plotted The model has selected “Worst Area”, “Worst Radius Feature” and “Worst Texture” to be the features that do a good job in predicting disease progression.

A white rectangular object with black text

Description automatically generated with medium confidence

Figure 5:Three features that do a poor job as per their Logistic Regressor Coefficients

Figure 5 has the score of each feature’s logistic regressor coefficient upon the bar chart plotted The model has selected “Mean Radius”, “Worst Compactness Feature” and “Worst Perimeter” to be the features that do a poor job in predicting disease progression.



The above is also the output from the program upon the terminal. Here, the score of all features is first printed and there after the Model Efficiency Score is printed.

This Model Efficiency score is the score based on the Validation split of the dataset. The five best features are then selected based on the score from the All-features list. The Final score is the based on the Testing split of the dataset. The top three example values for Benign and Malignant is also provided upon using K Nearest Neighbor Classifier (with n\_neighbors = 5).

## Observations

The flow of the program with the plotting of the good and poor features to the validation score and then the testing score, exemplifies that the data learning curve by the model has improved based on the ratio split of the Training, Validation and Testing.

The benign and malignant set of features after the K Nearest Neighbor classification is also displayed along with score to the top three values in each category.

# Project 2 Conclusion

Project 2 has been the most challenging task that has been endured so far. The convolution in understanding and further continued learnings from each problem statement has not only been frustrating but also fruitful towards understanding each model and the concept of splitting data for Training, Validation and Testing in Machine Learning. The problems revolve only around this splitting concept and helps one solidify their reasoning upon the knowledge of these Machine Learning concepts.

# Appendix

## Code 1



## Problem Statement 2

### Code



## Problem Statement 3

### Code



## Problem Statement 4

### Code





## Problem Statement 5

### Code

