**0College of Computer Science and Engineering**

**Department of Computer Science and Artificial Intelligence**

**CCAI-321: Artificial Neural Networks**

**Lab#8 Neural Networks using sickit-learn Python**

Student ID: 1845919

Marks Obtained = / 15 PLO = S1 - AI

**Q1. Repeat the example above, with max\_iter = 10, and report the score in the table below. Run the same script again for five times, does the score changes? Explain why.**  **[2 marks]**

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**Q2. Set the random state for the following functions: make\_classification, train\_test\_split, MLPClassifier. Check the documentation for each of these function to know how to set the random state. Next, repeat the example above, with max\_iter = 10, and report the score in the table below. Run the same script again**

**for five times, does the score changes? Explain why.**  **[2 marks]**

**A screenshot of a computer program

Description automatically generated**

**Q3. Repeat the example above, with max\_iter = 50, and report the score in the table below. Run the same script again for max\_iter = 100, 200, and 300. does the score changes? Explain why.**  **[2 marks]**

**A screenshot of a computer program

Description automatically generated**

**Q4. You are given the dataset “dataset\_spine” in csv format. Read the dataset using read\_csv and answer**

**the questions below.**  **[2 marks]**

1. **How many columns (features / attributes) does the dataset has? List the column names**
2. **How many rows (samples / records) does the dataset has?**
3. **What is the data type for each feature? Numerical, categorical, …**
4. **“class\_att” is the target value. Why is the datatype of the target value?**
5. **Is this a regression or a classification problem?**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**Q5. Train an MLPClassifier, follow the instructions below: [1+2+2+2 marks]**

1. **Split the dataset into 70% training and 30% testing**
2. **Initialize an MLPClassifier with 1 hidden layer and 7 neurons**

1. **Initialize the learning rate to 0.1 and the max\_iter to 1000**
2. **Fit the model given the training set**
3. **Predict the output of the testing set**

1. **Compute and report the score**
2. **What is the effect of the number of neurons on the result? To check its effect, repeat the steps from 2 to 6, but change the number of neurons in the hidden layer. How does this affect the score?**
3. **What is the effect of the number of hidden layers on the result? To check its effect, repeat the steps from 2 to 6, but change the number of hidden layers. How does this affect the score?**

1. **Apply gridsearch (GridSearchCV) to find the best hyperparamters for this problem. Use these Hyperparameters: number of hidden layers, number of neurons in each layer, learning rate, max iterations.**

A screenshot of a computer program

Description automatically generated

**A screenshot of a computer program

Description automatically generated**

**7-Effect of the number of neurons on the result:**

* When the number of neurons in the hidden layer is 3, the score is 0.533.
* When the number of neurons in the hidden layer is 5, the score is 0.633.
* When the number of neurons in the hidden layer is 10, the score is 0.7.
* Observing these results, we can see that increasing the number of neurons generally leads to an improvement in the score. However, it's worth noting that the score doesn't consistently increase with the number of neurons. In this case, increasing the number of neurons from 5 to 10 did not result in a significant improvement in the score.

**8- Effect of the number of hidden layers on the result:**

* When there is 1 hidden layer, the score is 0.6.
* When there are 2 hidden layers, the score is 0.533.
* When there are 3 hidden layers, the score is 0.5.
* Based on these results, we can observe that increasing the number of hidden layers does not necessarily lead to an improvement in the score. In fact, adding more hidden layers in this case seemed to decrease the score.

**9-Applying GridSearchCV to find the best hyperparameters:**

**The best hyperparameters found by GridSearchCV are:**

Number of hidden layers: (7, 7, 7)

Learning rate: 0.1

Max iterations: 1000

The best score achieved by the model with these hyperparameters is 0.814.

GridSearchCV found that having three hidden layers with seven neurons each, a learning rate of 0.1, and a maximum of 1000 iterations resulted in the best performance.