**BT-3172: Special Topics in Bioinformatics**

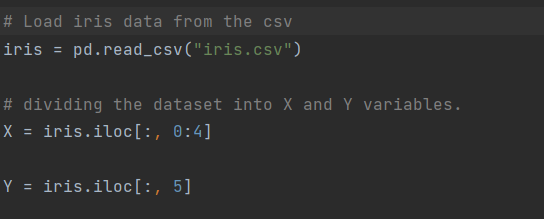
**Lab 11: Machine learning in Bioinformatics-II: Artificial Neural Networks.**

**Name: Anushka Udara**

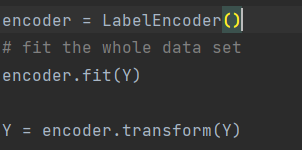
**Index number: s14234**

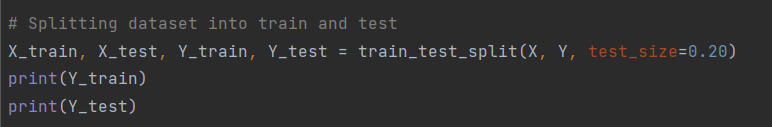
In this practical, you will learn how to use Python scikit-learn package to implement an Artificial Neural Network (ANN) using the popular iris data set.

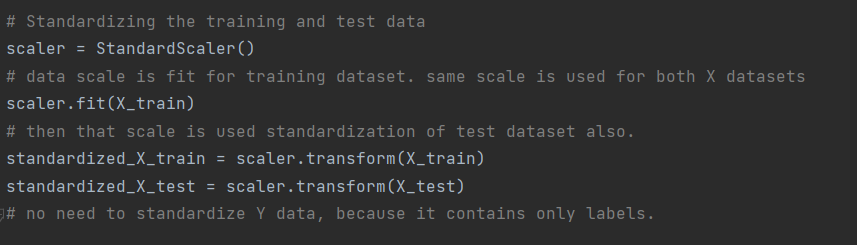
After using PyCharm to write your scripts, **copy the codes to the appropriate space below the questions**. Also, submit the Python files separately so we can test them. Use the following format to name each script: YourIndexNo\_PrimaryQuestion.py (submit two programs for the two questions)

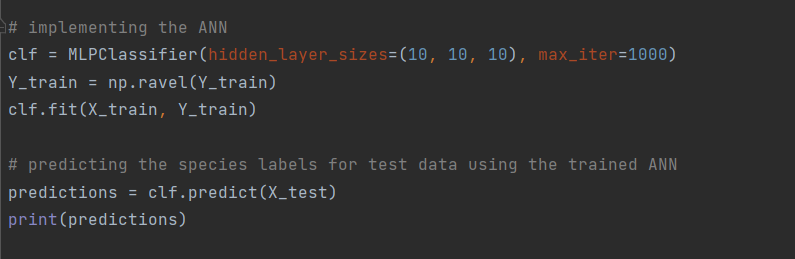
1. Implementing the ANN.
   1. Pre-processing the Iris dataset.
      1. Load the Iris dataset into a Pandas DataFrame from the given CSV file. Then, split the measurement data and class labels (species) to X and Y variables, respectively. (Hint: Pandas *iloc* function is useful for quick manipulation and indexing of Pandas DataFrames.)
      2. Output the species category values of the Y variable. Now, replace the category. values with integer numbers Then, again output the Y variable values. (Hint: use Pandas unique() function to get the values and Scikit-Learn's LabelEncoder class for the replacement)



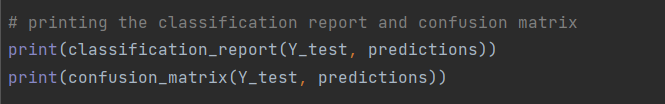
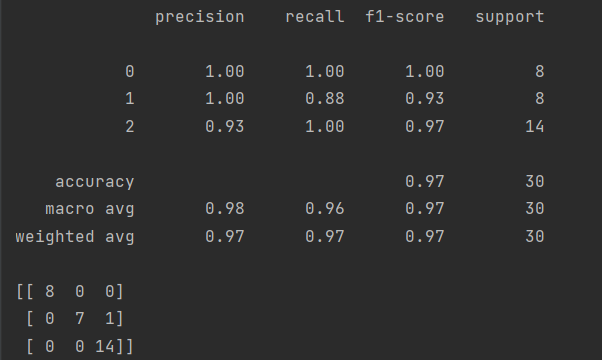


* + 1. To reduce the over-fitting, split the Iris dataset (X and Y variables) to train and test data. Use 20% data for testing.
    2. Standardize the training and testing data of the input variable



* 1. Using the Scikit-Learn package, train an ANN that contains 3 hidden layers (10 neurons each) for a maximum of 1000 iterations. Then, use the trained ANN for predicting the species labels of the test data. (Hint: when passing the training data for y (class variable), you have to convert the column data into a 1-D array using ravel()).



* 1. Evaluate the ANN using the classification report and the confusion matrix generated by the Scikit-Learn package. Save the output in the space below. What can you conclude about the performance of the network from the generated outputs?

**In the classification report,**

**Precision** is the percentage of correct positive predictions relative to total positive predictions. This ANN has a precision of 1.0 for predicting labels for both 0 and 1 labels, and 0.93 precision of predicting, 2 labels.

**Recall** is the percentage of correct positive predictions relative to total actual positives. This ANN has a recall of 1.0 for both 0 and 2 labels, and 0.88 for 1 label.

**F1 Score** is a weighted harmonic mean of precision and recall. The closer to 1, the better the model. This ANN has 1.0 for 0 label, 0.93 for 1 label and 0.97 for 2 label.

The accuracy percentage is 97%.

**The confusion matrix shows the number of true negatives, false negatives, true positives and false positives.**

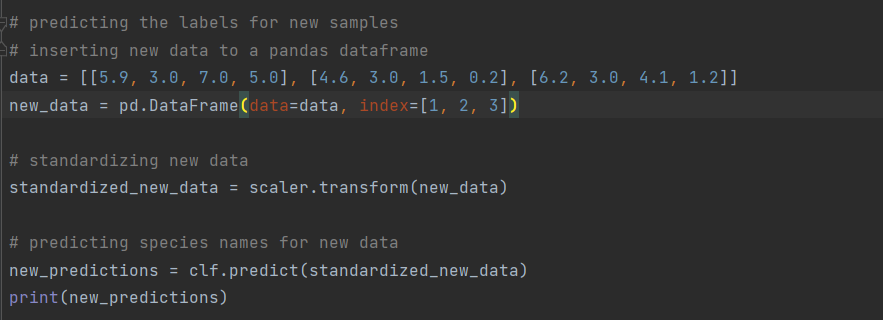
In this model, we have eight zero expected labels, predicted as zero. Seven , one labels predicted as one and one, one label predicted as 2. Fourteen two labels predicted as two.

* 1. A student has recorded the following measurements for 3 iris plants. Using the ANN you trained in (II), predict the species of the 3 plants.

Plant 1: 5.9,3.0,7.0,5.0

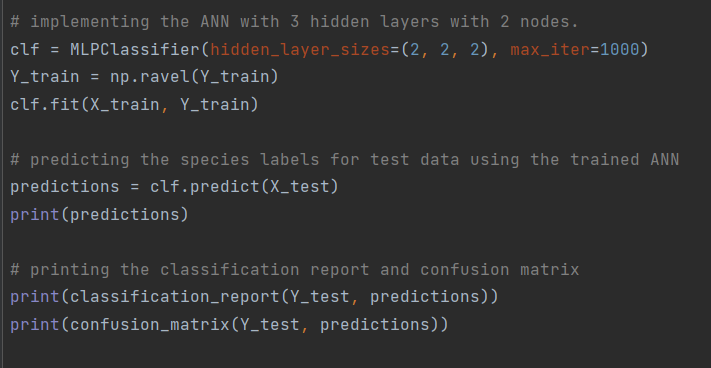
Plant 2: 4.6, 3.0, 1.5, 0.2

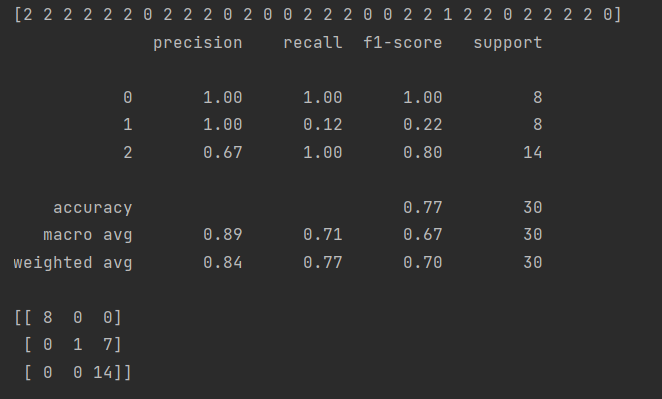
Plant 3: 6.2, 3.0,4.1,1.2





* 1. Now, change the number of neurons in each hidden layer to 2 and repeat (II) and (III). Did you notice a performance difference? Save the classification report and confusion matrix, interpret them here, and explain the reason for the performance difference.



**In the classification report,**

**Precision** is the percentage of correct positive predictions relative to total positive predictions. This ANN has a precision of 1.0 for predicting labels for both 0 and 1 labels, and 0.67 precision of predicting, 2 labels.

**Recall** is the percentage of correct positive predictions relative to total actual positives. This ANN has a recall of 1.0 for both 0 and 2 labels, and 0.12 for 1 label.

**F1 Score** is a weighted harmonic mean of precision and recall. The closer to 1, the better the model. This ANN has 1.0 for 0 label, 0.22 for 1 label and 0.80 for 2 label.

The accuracy percentage is 77%.

**The confusion matrix shows the number of true negatives, false negatives, true positives and false positives.**

In this model, we have eight zero expected labels, predicted as zero. One, one label predicted as one and seven, one labels predicted as 2. Fourteen, two labels predicted as two.

**When comparing the two models,** t**he accuracy has decreased to 97% from 77% and the prediction error has increased from predicting one label wrong to predicting seven labels wrong. This can be due the change in number of neurons in each hidden layer from 10 to 2. The best number of neurons in hidden layers for the model, can be decided based on the results accuracy provided by the model for several datasets.**