**Lab Assignment #3 – Use MLPs for prediction/classification problems**

Due Date: By the dropbox deadline.

Purpose: The purpose of this Lab assignment is to:

* Apply MultiLayer Perceptron (MLP) for solving various prediction/classification problems
* Design and implement Python applications that incorporate MLP solutions

References: Read reference books, Stanford tutorial on logistic regression (http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/), and the lecture slides. This material provides the necessary information that you need to complete the exercises.

* Your submission **MUST** include the **dataset files (if any)** that you may have used. Instead of using a dataset file, **if you need call some function to fetch a dataset**, then that should be fine too.
* Your submission **must** be a **zip** file **if compressed**. Any other compression **is not accepted**.
* You MUST run the program of an exercise in the relevant .ipynb file and **MUST retain** **the output that gets generated** (Note that the textual output of a program in an .ipynb file stays in that .ipynb file if you do not delete the output).
* You **MUST** create a **demo video** of less than 3 minutes of your solution. **Do not** **show yourself** in the demo video. Upload your video in your personal youtube account or google-drive account and share its link with the instructor through the **Comments** **box** of submission pageas mentioned next (Do not share the video publicly).
* During submission at the dropbox, you should see a **Comments** **box** present near the bottom of the submission page. **Write the link of your video** in this **Comments** **box**.
* Next, upload your solution and submit.

**Exercise 1: Multi-layer Neural Network**

Write a scikit-learn based application to predict the secondary school student performance using the MLP model **MLPClassifier**. The dataset is present in file **student.cleaned.data.csv**. The features to be taken into account are **traveltime, studytime, failures, famrel, freetime, gout, health**. The target should be **G3**. In **G3** column, assume the values less than 10 to be 0, and the values equal to or more than 10 to be 1. Evaluate the accuracy of the model. Compare the results with those of a logistic regression model from lab2.

(5 marks)

**Exercise 2: Multi-layer Neural Network**

Write a scikit-learn based application to classify MNIST digits using the MLP model **MLPClassifier**. The dataset is from http://yann.lecun.com/exdb/mnist/. You **must** use a tensorflow function **to just fetch** the data. The description about this tensorflow function is in this page: <https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data>

Rest of the functionality must be accomplished using scikit-learn library. Train the model using the **top 60 rows** out of 60000 rowsofthe training data (present in x\_train; see below how to obtain the training data in x\_train). Test the model using **top 10 rows** out of 10000 rows oftest data (present in x\_test; see below how to obtain the test data in x\_test). Evaluate the accuracy of the model. **Compare the result with those of the SVM model used in lab2.**

**Note:** If you go down the aforementioned webpage you will see an example usage of the method keras.datasets.mnist.load\_data. It returns four objects of type *ndarray*. Those four objects are x\_train, x\_test, y\_train, y\_test. **The shape of x\_train is (60000, 28, 28) implying 60000 rows of images, each image consists of 28 rows of pixels, 28 columns of pixels. You need to reshape x\_train to (60000, 784) so that it becomes a matrix of 60000 rows and 784 columns, thereby enabling it to be used by methods of sklearn models. Similarly, the shape of x\_test is (10000, 28, 28). You need to reshape x\_test to (10000, 784) so that it can be used by sklearn methods.** Note that the integer **784** **=** **28 \* 28**. (For an analogy, you may recall that the sklearn method **sklearn.datasets.load\_digits** returns an object of type *Bunch*. The *Bunch* object in turn has an attribute named *data* that stores a *ndarray* or a *dataframe* of shape (1797, 64). This shape implies that the *ndarray* object or the *dataframe* object contains 1797 rows and 64 columns. Note that the integer **64** **=** **8 \* 8**.

(5 marks)

**Evaluation:**

|  |  |
| --- | --- |
| **Design and Functionality:**  Correct design and implementation of requirements  Code explanation if asked | 90% |
| **Documentation of code using comments:**  At least a single-line comment for each functionality | 10% |
| **Total** | 100% |

You must name your Jupyter notebook file(s) according to the following rule:

**YourFullname\_COMP377Labnumber\_Exercisenumber**.ipynb

Example: **JohnSmith\_COMP377Lab1\_Ex1**.ipynb

**Submission rules:**

Submit your solution as a **zip file** that is named according to the following rule:

**YourFullname\_COMP377Labnumber.zip**

Example: **JohnSmith\_COMP377Lab1.zip**

Use 7-zip to compress files (https://www.7-zip.org/download.html).