Statistical Machine Learning **Assignment 4 (Max Marks: 200 + Bonus)**

Instructions: You are allowed to discuss but the final answer should be your own. Any instance of cheating will be considered as academic dishonesty and penalty will be applied.

1. Restrict to using only Python for coding assignments.
2. You are free to use math libraries like *Numpy, Pandas*; and use *Matplotlib, Seaborn* library for plotting.
3. Add all the analysis on the question in the written format, anything not in the report is not marked.
4. Use of inbuilt function for any evaluation metric is not allowed unless stated otherwise. Each of the metrics needs to be implemented from scratch.
5. Implement code that is Modular in nature and generalized to be executed for any input.
6. Code should be submitted in Python file format only(.py)
7. Bonus should only be attempted if all other parts are done ie, bonus marks will only be given if all the initial parts are complete.

Q1. Neural Network (70 marks)

Classify (MNIST) using 3 hidden layer NN. The base architecture should be: [#input, H1: 256 nodes, H2: 128 nodes, H3: 64 nodes, Output: 10 nodes].

Unless explicitly specified in the following parts, use: (a) ReLu activation for hidden layers, (b) learning rate=0.1, (c) epochs=100, (d) # nodes as mentioned above, and (e) Softmax activation for output layer.

1) Using test data, compute the accuracy of the above-mentioned base model.
2) Vary number of nodes in the hidden layers two times, once by increasing and once by decreasing the number of nodes. Compare the performance across the base, increased and decreased networks.
3) Add one more layer H4 of 128 nodes (after H3) in the base model. Analyze accuracies of base network vs increased network.
4) **BONUS (20 + 10)**:

a) Perform the above experiments with Sigmoid, Linear and Tanh activation functions. Analyze and plot the graph on accuracy vs activation functions.

b) Come up with interesting experiments and analysis on the neural network which has not been asked in 1 to 5 tasks.

Note: The code should be modular. The bare minimum modularity required is as follows and you can create any other functions you require:

a) *Neural Net should be a class and while initializing, it should require as input the architectural details like nodes in each layer, activation to be used etc.*

b) *The class should have helper functions as follows:*

   i) ***Fit** function with inputs as the data, learning rate, batch size and epochs.*

   ii) ***Predict** function which returns the probabilities for the different classes for the given data.*

   iii) ***Score** which takes as input the output of the predict function and the actual labels to give the accuracy.*

Unsupervised Learning

Q2. Autoencoder (30 marks)

1. For this part, you are allowed to use any library (Pytorch/Keras) for Autoencoder implementation.
2. Use MNIST default train-test split for the task given below. Normalize the data in the range [0,1].
3. For this part, you are allowed to use any library as per your convenience.

Train an autoencoder with random initialization of weights to find the reduced feature representation with respect to reconstruction error. The base architecture of the autoencoder is

{input:786, hidden: [256,128,64,128,256], output:786}

Use MSE as Loss function in the training process.

A. For all layers of encoder and decoder use the following activation
    a. Sigmoid
    b. ReLU

   Report the reconstruction visualisation, train error and test error.

B. After getting the reduced dimension, train a neural network of the given architecture

   {input: 64, hidden : [32], output:10}.

   and report classification accuracy on test data with ReLU activation and also plot the confusion matrix.

C. **(Bonus: 5 marks)** Perform PCA on MNIST dataset with n_components = 64. Train NN as in part (B) and report classification accuracy and confusion matrix. Compare and contrast the results with part B and report your observations.

Q3. Feature selection, Feature extraction and Clustering (50 marks)

In this question, you will apply unsupervised learning techniques to identify the segment of population that have greater than 50k earning per year. You are given two sets of data, Dataset 1. Population.csv contains information about the general population, while Dataset 2. more_than_50k.csv contains same information about the people who are making more than 50k per year. You are required to do feature selection, feature extraction and clustering. The goal of the problem is to cluster data in general population and more_than_50k population and analyze which clusters are over-represented in general population vs more_than_50k population and vice versa.

1. For this part, you are allowed to use any library as per your convenience.
2. Submit a well organized IPython/Jupyter Notebook as per sections given in the ***Question3.txt.*** After each section report observation and analysis for the given section. The submitted notebook will act as the report for this question.
3. Do submit the IPython/Jupyter Notebook exported in .py format too.

Dataset Contains:

1. population.csv : General Population Data
2. more_than_50k.csv : Dataset for Population having more than 50k Annual Income
3. Data Description.csv : Contains description for the features in the dataset.

The assessment will be done on the basis of the following components:

1. Working codes **(150)**
2. Analysis and clarity of results (drawing comparisons across different parts), clarity of the report, and understanding the theoretical concepts/viva **(50)**