CISC/CMPE 471: Assignment 2

Classification of Cancer Type using Gene Expression Data

What to submit:
One archive file containing your code, appropriately commented plus a report document containing the required output and answers to questions

In this assignment, you are asked to examine a small gene expression dataset and try to classify leukemia patients into one of two classes. This dataset comes from a proof-of-concept study published in 1999 by Golub et al. It showed how new cases of cancer could be classified by gene expression monitoring (via DNA microarray) and thereby provided a general approach for identifying new cancer classes and assigning tumors to known classes. These data were used to classify patients with acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL).
In the uploaded files, there are two csv files containing the initial (training, 38 samples) and independent (test, 34 samples) datasets used in the original paper. These datasets contain measurements corresponding to ALL and AML samples from Bone Marrow and Peripheral Blood. Note that in these files, rows correspond to different genes and columns correspond to patients. The third file holds the labels for all 72 patients.

Tasks:

Task 1- Data Preparation: (20%)

1. Show the distribution of data in two classes (e.g., using a *barplot*) in the combined dataset
2. Encode the labels
3. Remove all the 'Call' columns from both data files
4. Associate the train and test data to the labels
5. Compute and display summary statistics for the data

- Normalize the data, if necessary

Task 2- Dimensionality Reduction: (20%)

Principle Component Analysis (PCA) is an unsupervised, non-parametric statistical technique primarily used for dimensionality reduction in data analytics. High dimensionality means that the dataset has a large number of features, like the case with the current dataset.

1. Research and write a short paragraph on a high-level description of PCA method and how it is used for reducing the length of the input feature vectors.
2. Use PCA from *sklearn* to select features that account for 90% of data variance in trainset
3. Visualize the trainset in 3D space when the first 3 PCA components are selected

Task 3- Data Analysis: (60%)

Notes:

- **a)** For the following subtasks, **use the original dataset (without dimensionality reduction)**
- b) You may use *GridSearchCV* from *sklearn* to try and determine the best hyperparameters. In all cases, you should show your exploration that resulted in the selected model parameters.
- c) Performance of all models should be reported on the test set by calculating accuracy, sensitivity, specificity, and confusion matrix.

1. Establish a simple baseline model, by assigning the label of the majority class to all data and calculating the accuracy on test set. Your models should not perform worse than this.
2. Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.
3. Use *logistic regression* from *sklearn* to predict the cancer type and report its performance by calculating accuracy, sensitivity, specificity, and confusion matrix.
4. Use a *decision tree* from *sklearn* to predict the cancer type and report its performance as in subtask 2.
5. It has been mentioned in class, that a Random Forest classifier is an ensemble of several decision trees. Use a *random forest* from *sklearn* to predict the cancer type and report its performance as in subtask 2.
6. Build an ANN (using tensorflow-keras) to predict the cancer type and report its performance as in subtask 2. Choose an appropriate architecture. Explain how you have decided to choose the parameters such as learning rate, batch size, number of epochs, size of the hidden layer, number of hidden layers, etc.
   As the epochs go by, we expect that its error on the training set naturally goes down. But we are not actually sure that overfitting is happening or not. One thing we can do, is to further split the training set into train and validation and monitor the validation loss as we do for the train loss. If after a while, the validation error stops decreasing, this indicates that the model has started to overfit the training data. With *Early Stopping*, you just stop training as soon as the validation error reaches the minimum. Try to add Early stopping using *keras callbacks* to your model.
7. Each time you train a network, there is a set of new initialization of parameters, (e.g. weights in neural networks) and so the performance differs. How can you make sure that your results are reproducible?
8. Which of the above results in the best performance? Why do you think that is?
9. Repeat parts 2-8 using the dimensionally reduced dataset (Task 2) and compare the performance of all models with the original dataset. What are your conclusions?