EE769 Introduction to Machine Learning (Jan 2023 edition)
Electrical Engineering, Indian Institute of Technology Bombay
Programming Assignment – 2 : Classification and Feature Engineering

Instructions:
a) Name files A2_<RollNo>.<extension>. Submit three files (or links to these): an .ipynb, a .py, and a video demo (approx 10 minutes)
b) Use good coding practices such as avoiding hard-coding, using self-explanatory variable names, using functions (if applicable). This will also be graded.
c) You may use libraries such as scikit-learn, and need not code anything from scratch.
d) Cite your sources if you use code from the internet (line-by-line or block-by-block). Also clarify what you have modified.

Objective 1: Learn various steps and due diligence needed to train successful classification models.
Background: Some experiments were conducted on mice to see if a treatment of Down's syndrome works or not. Mice were divided into control and diseased (genotype), treated or untreated and whether it shows a particular behavior or not (treatment_behavior). Readings for 77 proteins were recorded for the mice, but some of the readings were discarded if they seemed unreliable (out of range). Your job is to develop a pre-processing pipeline and a classifier, and also find out which subset of proteins is important in predicting which class. Specifically:

1. Let your code read the data directly from https://www.ee.iitb.ac.in/~asethi/Dump/MouseTrain.csv [0]
2. Perform exploratory data analysis to find out: [3]
    a. Which variables are usable, and which are not?
    b. Are there significant correlations among variables?
    c. Are the classes balanced?
3. Develop a strategy to deal with missing variables. You can choose to impute the variable. The recommended way is to use multivariate feature imputation (https://scikit-learn.org/stable/modules/impute.html) [3]
4. Select metrics that you will use, such as accuracy, F1 score, balanced accuracy, AUC etc. Remember, you have two separate classification tasks – one is binary, the other has four classes. You may have to do some reading about multi-class classification metrics. [0]
5. Using five-fold cross-validation (you can use GridSearchCV from scikit-learn) to find the reasonable (I cannot say "best" because you have two separate classifications to perform) hyper-parameter settings for the following model types:
    a. Linear SVM with regularization as hyperparameter [2]
    b. RBF kernel SVM with kernel width and regularization as hyperparameters [2]
    c. Neural network with single ReLU hidden layer and Softmax output (hyperparameters: number of neurons, weight decay) [2]
    d. Random forest (max tree depth, max number of variables per node) [2]
6. Check feature importance for each model to see if the same proteins are important for each model. Read up on how to find feature importance. [3]
7. See if removing some features systematically will improve your models (e.g. using recursive feature elimination https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html). [3]
8. Finally, test a few promising models on the test data: https://www.ee.iitb.ac.in/~asethi/Dump/MouseTest.csv [2]

Objective 2: Practice using pre-trained neural networks to extract domain-specific features for new tasks.

9. Read the pytorch tutorial to use a pre-trained "ConvNet as fixed feature extractor" from https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html and you can ignore "finetuning the ConvNet". Test this code out to see if it runs properly in your environment after eliminating code blocks that you do not need. [2]
10. Write a function that outputs ResNet18 features for a given input image. Extract features for training images (in image_datasets['train']). You should get an Nx512 dimensional array. [2]
11. Compare L2 regularized logistic regression, RBF kernel SVM (do grid search on kernel width and regularization), and random forest (do grid search on max depth and number of trees). Test the final model on test data and show the results -- accuracy and F1 score. [3]
12. Summarize your findings and write your references. [2]