ICS 485: Machine Learning

Term Project
[Software Bug Prediction]

Abstract: Develop advanced machine learning-based classification models to satisfactorily classify **software bugs** (binary and multiclass) for data collected at the University of Geneva in Switzerland.

Dataset: The dataset is related to the bug prediction dataset. It contains data about the following software systems:

- Eclipse JDT Core
- Eclipse PDE UI
- Equinox Framework
- Lucene
- Mylyn

For each system the dataset includes the following piece of information:

- CK and other 11 object oriented metrics for the last version of the system.
- Categorized (with bug infection) post-release defect counts.

The files are defined below:

1- Eclipse_ JDT_Core_single-version-ck-oo_bugs_only.csv
2- Eclipse_PDE_UI_single-version-ck-oo_bug_only.csv
3- Equinox_Framework_single-version-ck-oo_bug_only.csv
4- Lucene_single-version-ck-oo_bug_only.csv
5- Mylyn_single-version-ck-oo_bug_only.csv

Training/Validation/Testing Set:

Divide the dataset into Training/Validation/Testing Set by randomly distributing 70% for training, 15% for validation, and 15% for the test set.

Task 1:

Provide **3 different binary classifiers** to predict the bug given the software metrics considered. In case of more than one bug, you should treat the sample as infected with a bug (class 1). Investigate the following issues:

a. Classifier optimization and hyper-parameter tuning.
b. The metrics to measure the classifier performance.
c. Make sure to run your models on the testing data.

Task 2:

Provide **3 different multiclassifiers** to predict the bugs (**0, 1, and 2**) given the software metrics considered. In case of more than one bug, you should treat the sample as infected with more than 2 bugs be considered as class 2. Investigate the following issues:

a. Classifier optimization and hyper-parameter tuning.
b. The metrics to measure the classifier performance.
c. Make sure to run your models the testing data.

Task 3:

Train a feedforward neural network to predict the bugs for the data provided. Determine the optimal training parameters for your neural network that are sufficiently general to predict the bugs on any withheld data that you will not have for testing purposes. You will, therefore, need to devise and execute a plan that uses the given data for training and testing in a manner that most closely mimics the real test (on withheld data) including:

a. Number of hidden layers
b. Hidden layer nodes
c. Training function
d. Learning function
e. Iterations (epochs)

Required:

1. Python notebooks for all the classifiers used.
2. Written report that should include:
   1) Objective
   2) Data analysis and preparation. Please provide some analysis of the data that demonstrates that you have compared the variable space for the different classes of bugs (e.g.: simple statistics, 3D cross plots, density functions, etc.).
      [Extra: consider evaluating dimension reduction and/or cluster analysis techniques.]
   3) Provide descriptive statistics on the predictors (i.e., features) as well as classes.
   4) Provide correlation analysis for features and target.
   5) Training and testing procedures for developing optimal classifiers, along with test results.
   6) Testing performance of the trained neural networks on the entire training set.
      [Extra: Some form of ensemble learning]
      [Extra: Based on the analysis and ML experiments and looking into the details of the features, provide some insights]
      Note: You can consult your instructor for clarifications and ideas.

General Instructions:

1. Create a folder with the team# and Project, e.g., Team-1_Project.
2. Once you have implemented the solutions for the needed problems, include only the python files (or Jupyter notebook files) and the assignment report in **PDF format.**
3. The report should contain a cover page having useful information and a page which tabulates involvement of every team member (as percentage) for the project. An example:

Involvement
Member-1 name: 30%
Member-2 name: 30%
Member-3 name: 40%