Fall 2023: CSCI 4/5587 Programming Assignment #1

DUE: Monday, Oct 23rd, 2023 (**Softcopy** @10 PM via Moodle)

Instructions

- **All work must be your own other than the instructor-provided data/code and hints to be used. You are NOT to work in teams on this assignment.**
- **Bonus marks: 5 points** for a well-presented report and Python code (in jupyter notebook).

Problem Description:

Given the IRIS dataset (consists of 150 samples, four input features, and three different output classes), train and compute the performances of the following classifiers using 10-fold cross-validations (10 FCV):

(1) [5 × 6 = 30 points] The classifiers are: (a) ETC (Extra Tree Classifier), (b) Bagging, (c) DTC (Decision Tree Classifier), (d) LR (Logistic Regression), (e) SVC (support vector classifier), and (f) kNN (k-Nearest Neighbor).

[Hints: you should be able to import those classifiers by calling the following lines respectively:

```
    from sklearn.tree import ExtraTreeClassifier,
    from sklearn.ensemble import BaggingClassifier,
    from sklearn.tree import DecisionTreeClassifier,
    from sklearn.linear_model import LogisticRegression,
    from sklearn.svm import SVC,
    from sklearn.neighbors import KNeighborsClassifier.
```

]

(2) [7 × 6 = 42 points] Compute and show the following performance metrics for each of the classifiers: (a) accuracy, (b) balanced accuracy, (c) Matthews Correlation Coefficient, (d) Sensitivity, (e) Specificity, (f) F1-score, and (g) confusion matrix.

(3) [14 × 2 = 28 points] Build two different ensemble classifiers by Stacking [1-4] – each of the classifiers will have a base layer and a meta-layer. Each base-layer will consist of three base-classifiers, and each meta-layer will consist of one classifier - taken from the classifiers listed in Question #1. Compute and show these two classifiers' performance in terms of the metrics listed in Question #2.

Stacking refers to a method to blend estimators where the base estimators are individually fitted on some training data while a final or meta estimator is trained using the stacked predictions of these base estimators. In your Stacking-based classifier constructions, the base classifiers will provide three class-classification probabilities [hints: use model_instance.predict_proba(X_test)], for each sample to the meta classifier. Thus, the meta classifier will be trained using the original 4 input features plus 3 probabilities from each of the three base classifiers, i.e., the meta classifier will have in total (4 + 3 × 3) or 13 input features.

Submission via Canvas:

(1) A report in ~.pdf or ~.docx, containing each of the classifiers' performance metrics listed in Question #1 and Question #3 using Table(s).
(2) Your python code in jupyter notebook format/file.
(3) Additional datasets (if any) that you may have created and used to build the classifiers based on Stacking – so that the grader can run and check your code smoothly.

Compress all three items in a folder as ~.zip and submit via Canvas.

References:

[1] D. H. Wolpert, "Stacked Generalization," *Neural Networks, Elsevier.*, vol. 5, pp. 241-259, 1992.

[2] A. Mishra, P. Pokhrel, and M. T. Hoque, "StackDPPred: A Stacking based Prediction of DNA-binding Protein from Sequence," *Oxford Bioinformatics*, vol. 35, pp. 433–441, 2019.

[3] S. G. Gattani, A. Mishra, and M. T. Hoque, "StackCBPred: A Stacking based Prediction of Protein-Carbohydrate Binding Sites from Sequence," *Carbohydrate Research, Elsevier.*, 2019.

[4] S. Iqbal and M. T. Hoque, "PBRpredict-Suite: A Suite of Models to Predict Peptide Recognition Domain Residues from Protein Sequence," *Oxford Bioinformatics* 2018

--- x ---