Machine Learning – Assignment II

All about supervised learning

- **Requirements**: The requirements of each assignment of this course at least include a) a full document in PDF/Word format with implementation details and difficulties you met, 2) source code and the compiled file (in exe/dmg/sh) and its readme to indicate how to launch it, and 3) key comments in your source code. If your code was referred from an existing source on the Internet, please cite it accordingly. Note that the packages CAN NOT be used in this assignment except for the visualization functions.

- **Problem set (110pt):**

Dataset: We are going to explore real-world applications and scenarios for this assignment. The dataset and its metadata can be found on Kaggle below

https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification?select=train.csv

Note that all the training/testing processes should perform on train.csv only, and you need to manually & randomly partition the train.csv into train/validation/test datasets.

1. **(50) Classification Task:** Please implement the following algorithm manually without using scikit-learn or something like that. Make the performance comparison (also choose one of the criteria and show me why that is your choice) among these classifiers.
   i. **Naïve Bayer classifier.**
   ii. **Random Forest Classifier.** Note that you can implement the simplified version of the random forest; the full function is unnecessary in this assignment. Basically, you are required to implement the bagging in #data and feature dimension with the predefined ratio. The other internal hyper-parameters of the #tree, #max-depth of each tree, etc., could be predefined by yourself, just mentioned in your report.
   iii. **Random Forest Classifier (scikit-learning):** Use the existed package to implement the random forest and make the performance comparison.
   iv. **XGBoost, Catboost, LightGBM (can use other public code sources).**

2. **(60pt) Cross-Validation**: Based on **Problem 1**, use *k*-fold cross-validation to verify the stability of each classifier. Note that cross-validation could adopt any existing package. Answer the questions below:

   i. Use k=3,5,10, and make some discussions of your observation.

   ii. Now you have a test dataset you have partitioned from train.csv. Please design an algorithm that can merge/aggregate the predicted results from *k* classifiers in k-fold cross-validation. Compare the performance and complexity of the cross-validation with Problem 1.

   iii. How do we know the performance of one model is really better than another one? Please compare the result in 5-fold cross-validation and the result of Problem 1 to justify which is “REALLY” better. Also show me why.