Machine Learning – Assignment I

All about supervised learning

- **Requirements**: For each assignment in this course, ensure you submit the following:
- **Document**: A comprehensive document in either PDF or Word format detailing your implementation process and any challenges you encountered.
- **Source Code & Compiled File**: Include both your source code and the compiled file (in .exe, .dmg, or .sh format). Accompany these with a README file that provides instructions on how to launch the compiled program.
- **Code Comments:** Ensure that your source code contains key comments explaining crucial parts of your implementation. If any portion of your code is derived from existing sources on the Internet, provide appropriate citations within your comments.
- **Note**: For this assignment, you are **prohibited** from using external packages, with the exception of those used for visualization purposes.

- **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.

- **(40pt) Classification Task:**
  Implement the following algorithms manually, without relying on libraries like scikit-learn. After implementation, compare the performance of these classifiers. You should also select an evaluation criterion of your choice and justify its selection. Ensure to plot the training/validation loss for each algorithm, and all results should be evaluated on the test set.

  1. **Linear Classifier (5pt)**:
     - Implement a basic linear classifier.
  2. **K-NN Classifier (10pt)**:
     - Implement the K-NN classifier using three different distance metrics.
     - Specify and explain the distance metrics used.
  3. **Naïve Decision Tree Classifier (10pt)**:
     - Implement a basic decision tree classifier.
     - Note: This is a naïve implementation without any pruning.
  4. **Decision Tree with Pruning (15pt)**:
     - Implement a decision tree classifier that incorporates a pruning algorithm.
     - Clearly explain the type of pruning algorithm and criterion used to prune nodes in the decision tree.

- **(40pt) Feature engineering:**

  - (10pt) Implement an algorithm that can determine the "feature importance" for both linear classifiers and decision trees. Explain the rationale behind your chosen algorithm.

  - (10pt) Utilize SHAP (https://shap.readthedocs.io/en/latest/) with your implemented algorithm to assess feature importance. Compare your findings with those obtained using SHAP.

  - (20pt) It is known that sometimes the original feature set may not be effective. Designing new features based on the original set is crucial for model performance. Based on your observations and experience, propose an algorithm that can derive new features to enhance model accuracy.

- **(30pt) 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:

  1. (10pt) se k=3,5,10, and make some discussions of your observation.

  2. (10pt) 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.

  3. (10pt) 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.