Lab Course Machine Learning
Exercise Sheet 9

Prof. Dr. Dr. Lars Schmidt-Thieme, Shereen Elsayed
Information Systems and Machine Learning Lab
University of Hildesheim

January 14th, 2022
Submission on January 21th, 2022 at 12 noon, (on learnweb, course code 3116)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a jupyter notebook detailing your solution.
2. Please set the seed(s) to 3116.
3. Please explain your approach i.e. how you solved a given problem and present your results in form of graphs and tables.
4. Please submit your jupyter notebook to learnweb before the deadline. Please refrain from emailing the solutions except in case of emergencies.
5. Unless explicitly noted, you are not allowed to use scikit, sklearn or any other library for solving any part.
6. Please refrain from plagiarism.

Exercise 1: Implement Decision Tree (10 Points)

In this task you will implement a decision tree. More specifically, we would be following the example in the lecture slides, and build a decision tree for classification. In particular you have to implement *Learn-Decision-Tree* with an appropriate *Quality-criterion* and *Predict-Decision-Tree*.

Datasets

1. **Classification Datasets:** You can use one of the two datasets (or optionally, both datasets).
   (a) Car Evaluation dataset D_1: Target attribute **safety**:{low, med, high}. `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`
   (b) Iris dataset D_2: Target attribute **class**:{Iris Setosa, Iris Versicolour, Iris Virginica}. `https://archive.ics.uci.edu/ml/datasets/Iris`

Part A: (5 Points): Basic working with MCR In Part A, you have to split data into three parts train, validation and test (70%, 15% and 15% respectively). Using the train data you will build a decision tree. Use **Misclassification Rate (MCR)** as a *Quality-criterion*. Please use the validation split to configure the following hyperparameter:

1. Defining an appropriate stopping criteria i.e. max depth, gain is too small or reduction in cost is small

Please also plot the following:

1. At each decision step (or split) present the probability of each class using histogram (properly labeled figure)
2. Print your tree using a breath first tree traversal.
3. On the validation-set measure the cross entropy loss (i.e. logloss, note that this time problem is not binary classification).

Part B: (5 Points): Experimenting with other *Quality-criterion*: In Part B, you will implement **Information Gain** as the quality criterion.

1. Use the train and validation splits from Part A.
2. modify the *Quality-criterion* to **Information Gain**.
3. At each decision step, plot the **Information Gain**.
4. Compare the validation set results for both *Quality-criterion*, output one value for test-set.

Exercise 2: Gradient Boosted Decision Trees (10 Points)

In this exercise, you are tasked to build a Gradient Boosted Decision Tree Classifier for a binary classification task. You need to go through the following slides and follow the tutorial at the end.

- Predictive Analytics: Ensemble of Gradient-Boosted Decision Trees (link: `https://www.ismll.uni-hildesheim.de/lehre/ba-18w/script/4_predictive-analytics-xgboost.pdf`)

Concretely, the tasks are as follows:

1. Generate a binary classification toy dataset from the scikit-learn utility "make-moons". Please generate 100 samples, for 10 different levels of noise which should give you a toy-dataset of 1000 samples. Here sample refers to a single point in 2-D, and it's corresponding label (0 vs. 1) denoting membership in either of the two moons. Visualize the 10 different pairs of so-called moons.
2. Generate train/validation/test splits with the ratios like before.
3. Please keep max depth of trees to 2 i.e root node then leaf nodes (also called stumps), and tune number of trees in the ensemble on the validation set.
4. Report test-accuracy.