DS340W: Lab Assignment 1

Decision tree and its variants are an important family of machine learning techniques. It has been enjoying widespread implementation in industry products. In this lab assignment, you will complete a small data science project in which you will use the `scikit-learn` decision tree and random forest classifiers to evaluate the quality of white wine.

Wine is enjoyed by a wider range of consumers all over the world. Quality evaluation can be used to improve wine making and stratify wines such as premium brands. In this lab assignment, we will evaluate the quality of white wine by using 11 features (e.g., pH, alcohol). The data with 4,898 samples were collected from the northwest region of Portugal.[1]

To complete this lab, follow the steps below:

1. **Data preparation**
    - **a.** Download the data file "winequality-white.csv" from the course schedule wikipage. The head line includes all the feature names, where "quality" represents the class label. Note that the delimiter of the data is semicolon ";".
    - **b.** After loading data into your Python program, split the entire dataset randomly into 80% training and 20% testing.
2. **Parameter tuning using 5-fold cross validation**
    - **a.** Train **decision tree** classifiers using the `DecisionTreeClassifier` included in `sklearn`. You should train the classifier using at least 5 different values of `max_depth`. For each one, evaluate the decision tree using 5-fold cross-validation.
    - b. Train **random forest** classifiers using the `RandomForestClassifier` included in `sklearn`. You should train the classifier using at least 5 different values of `n_estimators`. For each one, evaluate the decision tree using 5-fold cross-validation.
3. **Test the classifiers**
    - **a.** Select the best value of the parameter (i.e., `max_depth` or `n_estimators`). Train decision tree and random forest classifiers on the entire training set.
    - b. Apply the trained classifiers to the test set. Compare the mean accuracy of the two classifiers.
    - c. Plot the confusion matrices for both classifiers summarizing your classifier performance on the test data. Hint: use `confusion_matrix` function (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

[1] http://www3.dsi.uminho.pt/pcortez/wine

4. **Visualize the decision tree**
    a. Extract and visualize the first three layers of (i) your decision tree and (ii) one tree in your random forest.

Lab assignment delivery:

Note: This is an individual lab assignment. Each student needs to work on and submit his/her report independently.

Submission: You should submit **a single .zip file** to the appropriate Canvas dropbox which includes the following content:

- **A lab report (.pdf or .docx)** summarizing your work. See below for detailed instructions.
- **A folder named "code" containing all codes (.py or .ipynb)** needed to reproduce your results. Run your code before submission to make sure it is bug-free. Use comment lines to give some annotations of the codes.

Your lab report should be in **an essay style** (NOT Q&A) and be formatted professionally. It should include the following sections:

1. **Introduction (10 pts)**
    - **a.** Describe the problem you aim to solve.
    - b. Describe the data you have. How many data samples are there in total? Among them, how many are positive samples? What are the features in each data sample?
2. **Method (30 pts)**
    - a. Understand the `sklearn` implementation of decision tree by reading the online document: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
      Then, in your own words, answer the following questions:
        - i. List all the possible criteria for measuring the goodness of a split. Which criterion is your code actually using?
        - ii. List at least five possible stopping criteria for growing the tree. Which criterion is your code actually using? Hint: check the parameters `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_leaf_nodes`, `min_impurity_decrease`, and `min_impurity_split`
    - b. Understand the `sklearn` implementation of random forest by reading the online document: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
      Then, in your own words, answer the following questions:
        - i. Explain the two randomization techniques used by a random forest to improve the diversity of the fitted trees. For each technique, which

parameter in `RandomForestClassifier` should you use to control the level of randomness?

ii. What is the default stopping criterion used by `RandomForestClassifier`?

3. **Experiments (60 pts)**
   a. **Data preparation**: describe the numbers of training samples and testing samples you obtained after splitting the dataset.
   b. **Parameter tuning:**
      i. For decision tree, include a plot of your training and validation accuracies as a function of `max_depth`. Describe which `max_depth` value had the highest validation accuracy and your explanation of the behavior you observe in your plot.
      ii. For random forest, include a plot of your training and validation accuracies as a function of `n_estimators`. Describe which `n_estimators` value had the highest validation accuracy and your explanation of the behavior you observe in your plot.
   c. Test the classifiers
      i. Describe the mean accuracies of the two classifiers. For each classifier, include a plot of the confusion matrix.
   d. Visualize the decision tree
      i. Include visualization of the first three layers of (i) your decision tree and (ii) one tree in your random forest. For each non-leaf node, describe the feature name and the split rule; for each leaf node, describe the class your decision tree would assign.