Machine Learning Assignment 2
Classification using Logistic Regression and Decision Trees

Submission deadline: March 1, 2020

In this assignment, you are asked to implement Logistic Regression Classifier and Decision Tree classifier. You also need to use scikit-learn to test out these classifiers.

Task 1: Dataset Generation [5 + 10 = 15]

Download the *winequality-red.csv* from this link. It contains various chemical properties of red wine samples, along with the quality of wine. We want to train classifiers to predict the quality.
We shall create two modified datasets from this data.

A. Convert all the values in quality attribute to 0 (bad) if the value is less than or equal to ‘6’ and to 1 (good) otherwise. Normalize all the other attributes between 0 and 1 by **min-max scaling**. **Use this dataset (dataset A) for Logistic Regression**.
B. Convert all the values in quality attribute to 0 (bad) if the value is less than ‘5’, to 1 (good) if the value is ‘5’ or ‘6’ and to 2 (great) otherwise. Normalize all the other attributes by **Z-score normalization**, and segregate them into 4 equal spaced bins each giving the values between [0 to 3], and replace the values for that attribute with the number corresponding to the interval they belong.
For example, suppose after normalization an attribute has values between [-0.5,1.5], i.e., minimum value of the attribute is -0.5 and maximum value is 1.5, then form 4 bins:
bin 0: [-0.5,0.0],
bin 1: [0.0,0.5],
bin 2: [0.5,1.0],
bin 3: [1.0,1.5].
Now, if a data instance has a value of 0.73 for that attribute, replace 0.73 with 2.
Use this dataset (dataset B) for Decision Tree.

Task 2: Logistic Regression [25 + 5 + 5 = 35]

Use **dataset A** for this part.

1. Implement a standard Logistic Regression Classifier as discussed in class. Do NOT use scikit-learn for this part.
2. Test out the implementation of Logistic Regression from scikit-learn package, using **saga solver** and using no regularization penalty.
3. Cross validate both the classifiers with 3-folds and print the mean accuracy, precision and recall for the class 1 (good) for both the classifiers. You may or may not use the scikit-learn implementations for computing these metrics and cross validation.

Task 3: Decision Tree [35 + 5 + 10 = 50]

Use **dataset B** for this part.

1. Implement the standard **ID3 Decision tree algorithm** as discussed in class, using information gain to choose which attribute to split at each point. Stop splitting a node if it has less than 10 data points. Do NOT use scikit-learn for this part.
2. Test out the implementation of Decision Tree Classifier from scikit-learn package, using information gain. Here also stop splitting a node if it has less than 10 data points.
3. Cross validate the classifiers with 3-folds and print the mean macro accuracy, macro precision and macro recall for both the classifiers. You may or may not use the scikit-learn implementations for computing these metrics and cross validation.

Submission Instructions

Submit separate codes for each task and put them in a folder called “src”. Comment your codes sufficiently, use meaningful variable names. Keep the datasets in a separate folder called “data”. Also submit a README file which will contain the instructions on how to execute your codes, and the four sets of scores from Tasks 2 and 3.

All source codes, result files and the README file must be uploaded via the course Moodle page, as a **single compressed file (.tar.gz or .zip)**. The compressed file should be named as:
{ROLL_NUMBER}_ML_A2.zip or {ROLL_NUMBER}_ML_A2.tar.gz
Example: If your roll number is 16CS60R00, then your submission file should be named as 16CS60R00_ML_A2.tar.gz or 16CS60R00_ML_A2.zip

Note that the evaluators can deduct marks if the deliverables are not found in the way that has been asked for the assignment, or if your codes are not understandable.

You can use one of C / C++ / Java / Python for writing the codes; no other programming language is allowed. Note that for Tasks 2.2 and 3.2 you have to use Python and the scikit-learn package. You need to submit **raw codes** in one of the specified languages, which can be executed on a Linux system from the console. Other platform-specific or software-specific formats such as ipython notebooks are not allowed.

You **cannot** use any library/module meant for Machine Learning for implementing your models (except where mentioned). You can use libraries for other purposes, such as generation and formatting of data. Also you **should not use any code available on the Web**. **Submissions found to be plagiarised or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.**

Submission deadline: March 1, 2020, 23:59 IST [hard deadline]

For any questions about the assignment, contact the following TAs:
1. Soham Poddar (sohampoddar26 @ gmail . com)
2. Paheli Bhattacharya (paheli.cse.iitkgp @ gmail . com)