Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.

Prof. Dr. Mohamed Abougabal
Prof Dr. Nagwa M. El-Makky
Special Topics in Information Systems

Assignment 2
Data Classification

Objectives

1. Applying preprocessing techniques learnt before and see their effects on classification performance.
2. Exploring different classification models and tune their hyper-parameters
3. Exploring different techniques for evaluating classification models
4. Learning how to analyze observed results and explain observations in a detailed report.

Problem Statement

You are given the Wisconsin Diagnostic Breast Cancer (WDBC) Data Set; the dataset is found in wdbc.data and the description is found in wdbc.names.

The data set consists of 32 attributes and 569 instances. The **first attribute** is the **instance Id.** The **second attribute** represents the **class label**, **Diagnosis (M = malignant, B = benign).** The rest of the 30 attributes are real-valued features that are computed for each cell nucleus. The features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image.

It is required to investigate the dataset deeper, split it into train and test datasets with **class labels M = malignant** and **B = benign**. You are required to apply visualization and preprocessing techniques on the dataset, to construct classification models using different approaches such as **Decision Trees, AdaBoost** and **Random Forests** and tune the hyper-parameters of these models. You are also required to compare the performance of models with each other.

Lab session

1. Visualization and Dataset Splitting

Visualize the dataset using different ways (e.g., histograms, box plots, scatter plots, line plot, correlation matrix, etc.) Use the most discriminative plots and discuss your observations on the data in your report. Split your dataset randomly such that your training data would form 70% of the dataset and the test data would form 30% of it. Since the dataset is unbalanced, use Stratified Splitting so that class distribution in training and testing sets is approximately the same as that in the initial dataset.
Hint: you can use `StratifiedShuffleSplit` from `Sklearn` to create these splits.

2. Preprocessing

Apply any required preprocessing techniques (e.g., feature normalization, feature selection/ feature projection, etc.) on both the training and testing sets.

3. Classification

In this step, it is required to apply the following classification models and to perform hyper-parameter tuning using **cross-validation**.

a. Classification models:

You are required to build models using the following classification techniques:

- **Decision Tree:**
  Parameters to be tuned: `max_depth`
  Hint: use **`DecisionTreeClassifier`** from **`Sklearn`**
- **AdaBoost:**
  Parameters to be tuned: `n_estimators, learning_rate`
  Hint: use **`AdaBoostClassifier`** from **`Sklearn`**
- **Random Forest:**
  Parameters to be tuned: `n_estimators, max_depth`
  Hint: use **`RandomForestClassifier`** from **`Sklearn`**

b. Hyper-parameter tuning:

Use the **cross-validation** approach on the pre-processed training dataset (as discussed in class) to get the best parameter values for each classifier. Test the models trained with best obtained parameter values on the separate pre-processed testing set.

Hint: you can use **`GridSearchCV`** from **`Sklean`** to perform parameter tuning.

c. Evaluation:

Report for each model the following performance metrics: **precision, recall, and F-measure** as well as the resultant confusion matrix using the test data.

Report Requirements

Your report should contain the following:

- Comments on all visualizations
- Analysis and comparison of the performance results obtained in the evaluation part
- Analysis of the effect of hyper-parameter tuning on the performance of different classifiers

Notes

- **This lab session needs time. So, try to start working on it early.**
- **Use Google Colab.**
- You should work in groups of two. Each student should answer any question in the lab session.
  - You should deliver a well-documented code as well as a report showing all your work and conclusions.
  - Copied assignments will be penalized; so not delivering the assignment would be much better.
- You should write your code in python.

References

[1] Chapter 8 of the first reference (J. Han, M. Kamber, and J. Pie, “Data Mining: Concepts and Techniques”, 3rd Edition, Morgan Kaufmann, 2012).
[2] Raschka, S. (2015). Python Machine Learning. Packt Publishing.
[3] Dua Dheeru and Efi Karra Taniskidou. 2018. UCI Machine Learning Repository. (2018). http://archive.ics.uci.edu/ml