Programming Assignment 5: Random Forests

Advances in Data Mining

Due Date: Nov 5, 23:59

Introduction

In this assignment, you are asked to implement Random Forest models for both classification and regression tasks. You will learn how to use a Random Forest classifier and regressor on different datasets. You are required to train, experiment and evaluate your models using 10-fold cross-validation and to tune hyperparameters to individuate the best possible settings for both tasks.

Task Descriptions

The assignment is divided into two main tasks: classification and regression.

1. Random Forest Classification

You are asked to implement a Random Forest classifier using the wine quality dataset from the UCI ML repository. Your tasks are:

- Implement `random_forest_experiment` to train a classifier and perform 10-fold cross-validation. Getting as input the feature matrix, the target vector and two tunable parameters, being the number of trees in the model and the number of samples per leaf, you are expected to implement a Random Forest classifier (using `scikitlearn`), to evaluate your model through 10-fold cross-validation and to return the scores based on cross-validation accuracy.
- Implement `get_best_hyperparameters` to tune `n_estimators` and `min_samples_leaf`, selecting and returning as a dictionary the combination, out of the values stated in `n_estimator_list` and `min_samples_leaf_list` in the `Main` function, with the highest cross-validation accuracy.
- Complete `manually_entered_best_params_and_accuracy` with the best parameters and accuracy score obtained through `get_best_hyperparameters` .

2. Random Forest Regression

For the regression task, you are asked to implement a Random Forest regressor to predict the number of bike rentals based on features from the bike-sharing dataset.

- Implement `random_forest_regression_experiment` to train the regressor and perform 10-fold cross-validation. Getting as input the feature matrix, the target vector and the two tunable parameters, you are expected to implement a Random Forest regressor (using `scikitlearn`), to evaluate your model through 10-fold cross-validation and to return the average negative mean squared error (MSE).
- Implement `get_best_hyperparameters` to tune `n_estimators` and `min_samples_leaf`, selecting and returning as a dictionary the best combination of parameters, out of the values stated in `n_estimator_list` and `min_samples_leaf_list` in the `Main` function, based on cross-validation MSE.
- Complete `manually_entered_best_params_and_mse` with the best parameters and MSE obtained through `get_best_hyperparameters`.

Datasets

The provided `.py` files contain instructions about how to directly import the necessary datasets from the UCI repository. In case of any problem in downloading the data, we also provide them in two separate `.zip` files attached to this assignment.

Wine Quality (Classification)

Use the `ucimlrepo` library to fetch the wine quality dataset (ID: 186) or import the data related to both white and red whine by the provided zipped directory. The feature matrix contains wine properties, while the target variable indicates the wine quality.

Bike Sharing (Regression)

The bike-sharing dataset can be downloaded from the UCI ML repository (ID: 275) or can be imported from the provided zipped directory that contains data aggregated both per day and per hour. Preprocess the dataset by extracting information from categorical feature such as `dteday`. The target variable for our regression task is `cnt`, which represents the total number of rentals.

Submission

You are provided with files `Task_Classification.py` and `Task_Regression.py`, which include the skeleton codes for both tasks. Implement the following functions:

- `random_forest_experiment` (classification)
- `random_forest_regression_experiment` (regression)
- `get_best_hyperparameters` (for both tasks)

- Complete the manual functions for best hyperparameters and performance indicators (for both tasks).

Submit both your Python file named respectively `Task_Classification_<student_id>.py` and `Task_Regression_<student_id>.py`, where `<student_id>` is your student number without the leading 's'. Test your code before submission. The last file you submit will be evaluated.

Good luck with the last one of your programming assignment!