CS 686-02 - Lab 4 Spring, 2018

Lab 4 - Diabetes Dataset

The goal of this assignment is to execute two classifiers (scikit-learn's implementation of Random Forest and XGBoost) against a diabetes dataset, which contains missing values. You will tune the classifiers in a principled way to gain maximum performance.

Background

We discussed the Random Forest classifier during lecture. Specifically, it applies bagging and random selection of features to achieve lower variance and therefore higher performance than a normal decision tree. XGBoost is another ensemble approach which generally gives further improvements over a single decision tree.

You are provided a diabetes dataset with missing values. This dataset contains many missing values. Your task is to impute the missing values in the data, plot the relationships between the features and the target and classify the data using the Random Forest and XGBoost algorithms.

Requirements

Perform the following steps:

1. Get the diabetes dataset from https://github.com/dbrizan/cs686-2018-01/blob/master/diabetes.csv
2. For each of the 8 features, plot the relationship between the features and the response.
3. Split the dataset into train and test by using train_test_split (from sklearn.model_selection) with `seed = 7` and `test_size = 0.2` to ensure we use a consistent train and test sets.
4. Impute the missing values using sklearn's Imputer class (from sklearn.preprocessing).
5. Classify the data using RandomForestClassifier (from sklearn.ensemble) and XGBoost. If necessary, download the XGBoost classifier with:

   ```
   conda install -c conda-forge xgboost
   ```

   … on command line for Anaconda. You should see approximately 77% accuracy for Random Forest and 79% accuracy for XGBoost.
6. Optimise your classification to achieve the maximum performance by changing parameters for the Imputer and the implementations for RandomForestClassifier and XGBoost.

Submission

Submit your source code or link to your github repository on Canvas.

Grading

Your grade for this assignment will be as follows:

- 100% = Implementation works correctly; all feature plots are present; classifiers are optimised.
- 75% = Implementation works but contains minor errors (eg. missing 1-2 required items).
- 50% = Implementation works but contains major errors (eg. missing many required items).
- 0% = Implementation not attempted or not submitted on time.