Assignment 3
Tree-based Methods, PCA, KNN and SVM

STAT3612: Statistical Machine Learning (Spring 2025)

DUE: **April 20, 2025, Sunday, 11:59 PM**

Goal

The three parts of this assignment is an application of solving a practical machine learning task with Decision Tree, Random Forest, Support Vector Machine and K-Nearest Neighbors techniques. You will also learn how to visualize the models by completing this assignment.
Note that you should complete this assignment using Python 3.6+.

Submission

Please submit the following **two files** in Moodle for grading:

- A PDF/HTML report of your answers to all questions. You are recommended to convert the Jupyter notebook file into PDF/HTML format and submit it as the report.
- The completed Jupyter notebook file `assign3.ipynb`.

Background

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help. The dataset in this assignment is a widely cited Kaggle dataset[1] for this issue.

Suggestion

It is highly recommended to go over the corresponding tutorial materials thoroughly before working on this assignment.

Part 1: Tree-based Methods

Q1 In this Q, we will fit a decision tree on the heart disease dataset from Kaggle. The data preprocessing code is included in `assign3.ipynb`, and use dataframes `X` and `y` in your coding.

[TOTAL: 25 points]

(a) Split the processed dataframes `X` and `y` into train and test set by using `test_size=0.4`, `random_state=2024` in your `sklearn.model_selection.train_test_split`.

[5 points]

[1] https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction/

(b) To standarlize the features, fit a `sklearn.preprocessing.MinMaxScaler((-1,1))` on train set, then apply the fitted scaler on test set.

[5 points]

(c) Based on the training set, estimate a `sklearn.tree.DecisionTreeClassifier` using `HeartDisease` as the response variable. Use `max_depth=3` and the default method `entropy`.

[10 points]

(d) Visualize the decision tree by `sklearn.tree.export_graphviz` or `sklearn.tree.plot_tree`, and print the accuracy score for the training set and the testing set.

[5 points]

Q2 In this Q, we use the same training and test set in Q1 to construct an Ensemble model, by comparing a single decision tree with the corresponding random forest.

[TOTAL: 20 points]

(a) Train a single decision tree classifier using `max_depth=2, 3, ..., 12`, `random_state=2024`. Plot the resulting training and testing accuracy scores on a graph. At what `max_depth` does test score achieve its maximum? What is the resulting maximum score? Should we continue increasing the depth and why?

[10 points]

(b) This time we train a `sklearn.ensemble.RandomForestClassifier` with the following parameters: `n_estimators=80`, `random_state=2024`, `max_features=4`, using `max_depth=2, 3, ..., 12`. Now, plot the single tree test scores together with the training and testing accuracy scores of the random forest on the same graph. At what `max_depth` does the forest test score achieve its maximum? What is this resulting maximum score? Should we continue increasing the depth and why?

[10 points]

Part 2: PCA and K-Nearest Neighbors

Q3 In this Q, we use the same training and test set in Q1 to construct the K-Nearest Neighbors algorithm.

[TOTAL: 30 points]

(a) Train a `sklearn.neighbors.KNeighborsClassifier` and print the accuracy on the test set.

[10 points]

(b) To reduce the dimension of the data, fit `sklearn.decomposition.PCA(n_components=2)` on the train set, then apply the fitted PCA on the test set and visualize the train set on a scatter plot with different classes in different colors.

[10 points]

(c) Train a `sklearn.neighbors.KNeighborsClassifier` on the PCA results and print the accuracy on the test set, compare with the Q3a, and write down your findings.

[10 points]

Part 3: Support Vector Machine

Q4 In this Q, we will construct a support vector machine with feature `Age` and `RestingBP` in the provided heart disease dataset.

[TOTAL: 25 points]

(a) Use `StandardScaler` in `sklearn.preprocessing` to scale feature `Age` and `RestingBP`.

[5 points]

(b) Use `test_size=0.3`, `random_state=2024` in your `sklearn.model_selection.train_test_split`. Train an `sklearn.svm.SVC` with the following parameters: `C=1.5`, `kernel='rbf'`.

[10 points]

(c) Use the `plot_svc` function we provide. Draw the resulting `plot_svc` graph. How many support vectors have been found?

[5 points]

(d) Draw the confusion matrix for this SVC using test data only and print the accuracy.

[5 points]