Data Science with Python
Day 04 - Activity

Principal component analysis, Decision Trees, Random Forest

1. Use the 'Breast Cancer Wisconsin (Diagnostic)' dataset which an inbuilt dataset in scikit-learn, to answer below questions

   a. Load the breast cancer dataset and explore its features. Visualize the distribution of the target variable (malignant or benign) using a bar chart.
   b. Use principal component analysis (PCA) to reduce the dimensionality of the breast cancer dataset and visualize the transformed data in two dimensions.
   c. Determine the number of principal components needed to retain 95% of the explained variance in the breast cancer dataset using PCA.

2. Use the given heart diseases dataset(heart.csv) to answer below questions.

   Attribute Information:

   - age
   - sex
   - cp - chest pain type (4 values)
   - trestbps - resting blood pressure
   - chol - serum cholestoral in mg/dl
   - fbs - fasting blood sugar > 120 mg/dl
   - restecg - resting electrocardiographic results (values 0,1,2)
   - thalach - maximum heart rate achieved
   - exang - exercise induced angina
   - oldpeak - ST depression induced by exercise relative to rest
   - slope - the slope of the peak exercise ST segment
   - ca - number of major vessels (0-3) colored by flourosopy
   - thal - 0 = normal; 1 = fixed defect; 2 = reversable defect
   - target - the presence of heart disease in the patient. It is integer valued 0 = no disease and 1 = disease.

A)

   a) Load the dataset into a pandas dataframe. Explore the dataset by checking the head, shape, and data types of the columns.
   b) Split the dataset into the feature matrix (X) and the target vector (y).
   c) Split the data into a training set and a testing set using the train_test_split function from scikit-learn.
   d) Build a decision tree classifier model using the DecisionTreeClassifier class from scikit-learn and train the decision tree classifier model on the training set.

e) Use the trained model to make predictions on the testing set and evaluate the performance of the model by calculating the accuracy, precision, recall, and F1 score.

f) Visualize the decision tree using the plot_tree function from scikit-learn.

B)

a) Load the heart disease dataset into a Pandas dataframe and split it into training and testing sets using Scikit Learn

b) Preprocess the data by scaling the features in the training and testing sets using the StandardScaler from Scikit Learn.

c) Build a Random Forest classifier from Scikit Learn and fit it to the training data. Evaluate the performance of the model on the testing set using metrics such as accuracy, precision, recall, and F1-score

d) Use the feature_importances_ attribute of the Random Forest classifier to identify the most important features in the dataset

e) Use a grid search with cross-validation to experiment with different hyperparameters of the Random Forest classifier, such as the number of estimators, maximum depth, and minimum samples split, and report the best hyperparameters and their corresponding performance metrics