School of Computer Science Engineering and Technology

Course- BTech
Course Code- 301
Year- 2022
Date- 14-02-2022

Type- Core
Course Name-AIML
Semester- Even
Batch- 4th Sem (SPL)

Lab Assignment No. 5

Objective: To Implement decision tree classifier using Sklearn

Problem Statement: Build a Decision Tree Classifier using Sklearn for classifying whether an individual will get a Match or not in a Speed Dating experiment.

About Dataset: This data was gathered from participants in experimental speed dating events from 2002-to 2004. During the events, the attendees would have a four-minute "first date" with every other participant of the opposite sex. At the end of their four minutes, participants were asked if they would like to see their date again. They were also asked to rate their date on six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests. The dataset also includes questionnaire data gathered from participants at different points in the process. These fields include: demographics, dating habits, self-perception across key attributes, beliefs on what others find valuable in a mate, and lifestyle information.

Steps

1. **Dataset:** Download the dataset from the link https://www.openml.org/d/40536 . The dataset contains a lot of numerical and categorical features. (5)
2. **Pre-processing:** (25)
    a. Convert the categorical features into numerical using one-hot encoding
    b. Some features have range values like [num1, num2]. Process them by a) creating two columns for each number in the set [] or b) take average value of num1 and num2
    c. Features such as 'race' and 'race_o' contain multiple nominal values. This cannot be processed directly using the one hot encoding. Hence find the unique values in that column and create "multi hot encoding" (i.e., more than one value of 1's in the representation).
    d. Perform range normalization on numerical features, not in the range of 0 to 1.
3. **Data Splitting:** Split the dataset into training and testing using 75-25 divisions. (10)
4. **Decision Tree Classifier:** Build a Decision tree using Sklearn with default parameters. Predict the labels in the testing set. Apply classification metrics such as confusion matrix, precision, recall, f-measure etc. Visualize the classification metrics as graphs. (25)
5. **Playing with Trees:** Change the following parameters of the decision tree and analyze their performance for training and testing using the evaluation measures (25)
    - criterion{"gini", "entropy"}
    - splitter{"best", "random"}
    - max_depth
    - min_samples_split
    - min_samples_leaf
    - max_features
    - random_state
    - max_leaf_nodes

6. **Comparison:** Compare the performance of the Decision tree model with other classification models such as logistic regression (10)

Suggested Platform: Python: Azure Notebook/Google Colab Notebook, packages such as numpy, nltk, regular expression package re.

Additional (Not a part of the evaluation**)**

*** Accuracy improvement: You can try different strategies to see whether testing error comes down or not. Strategies can be different 1. Encoding of features, 2. removal of some features, 3. normalization methods, 4. Shuffling of training samples. Check the model error for the testing data for each setup.