Lab Assignment 7 – Decision Trees for Dating

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

Steps

1. **Dataset:** Download the dataset from the link https://www.openml.org/d/40536. The dataset contains lot of numerical and categorical features.
2. **Preprocessing**:
    - Convert the categorical features into numerical using one hot encoding
    - Some features are having 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
    - 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).
    - 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 70-30 division.
4. **Decision Tree Modelling:** Build 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.
5. **Playing with Trees:** Change the following parameters of the decision tree and analyze their performance for training and testing using the evaluation measures
    - 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 perceptron, logistic regression etc.
7. **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.
8. **Decision Tree Regressor:** Pick a Regression dataset of your choice and perform training testing similar to above. Play with model parameters and analyse the results using regression measures.

Suggested Packages: Numpy, Pandas, Sklearn