Lab Assignment 9 – Classification Model Comparison

Build different **Machine Learning Classifiers** using **Sklearn** for classifying whether an individual will get a Match or not in a Speed Dating experiment. Do a comparative study of the models.

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. **Classification Models:** Build different classification models from Sklearn such as Perceptron, Logistic Regression, Decision Tree Classification, Random Forest Classification, Adaboost Classification, Gradient Boost Classification and Neural Network (from keras). Train these models with default parameters. For Neural Networks train a simple Predict the labels in the testing set. Apply classification metrics such as accuracy, confusion matrix, precision, recall, f-measure, ROC Curve etc. Visualize the classification metrics as graphs.
5. **Comparative Study:** Compare the 7 classification models by plotting bar graphs of precision, recall, f-measure and accuracy. Plot the confusion matrix of models individually. Plot the ROC curves of different models in same line graph.
6. **Playing with Models:** Change the hyper parameters of the classification models and improve their accuracy on testing set. You can try different strategies to see whether testing accuracy goes up 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 accuracy for the testing data for each setup.

Suggested Packages: Numpy, Pandas, Sklearn