Decision Trees and Random Forests

Train_test_split

- Split the data to train set test set and validation set with the ratio 0.8 , 0.2 ,0.2 respectivly

Decision Trees

- Train the Decision Trees on train set and predict on train and test set
- Calculate the accuracy_score for train and test set
- plot accuracy_score vs max_depth with the following values: max_depth = [1,2,5,10,100,1000,None]
- plot accuracy_score vs min_samples_split with the following values: min_samples_split = [0.05,0.1,0.3,0.5,0.7,0.9,0.99]
- plot accuracy_score vs **min_samples_leaf** with the following values: min_samples_leaf = [0.05,0.1,0.3,0.5]
- plot accuracy_score vs **max_features** with the following values: max_features = [0.05,0.1,0.3,0.5]
- **In each graph, the train accuracy_score and test accuracy_score must be printed**

Decision Trees

- From each graph choose the parameter that give you the best accuracy_score on train and test set
- For the choosen parameter run the model and print the accuracy_score on train and test

Random Forest

- Take the parameters from previous and insert them to random forest classifier and insert n_estimators=100
- Run the model and print the accuracy_score on train and test
- plot accuracy_score vs **n_estimators** with the following values:
  n_estimators = [1,5,10,50,100,200,300,400,500,700,1000]
- **In the graph, the train accuracy_score and test accuracy_score must be printed**
- Choose the n_estimators that gives you the best accuracy_score
  run the model, and print the accuracy_score for train and test set
- **Print the accuracy_score also for the validation set**

Evaluate Model Performance

- For this exercise use the random forest from previous
- Calculate:
    - confusion_matrix
    - accuracy_score
    - precision_score
    - recall_score
    - f1_score
- Plot the ROC curve and choos the threshold that gives the best result
- Print the f1_score of the validation set with the chosen threshold and compare it with the default f1_score