Exercise Sheet 8

1. Exercise: Decision Tree (Classification)

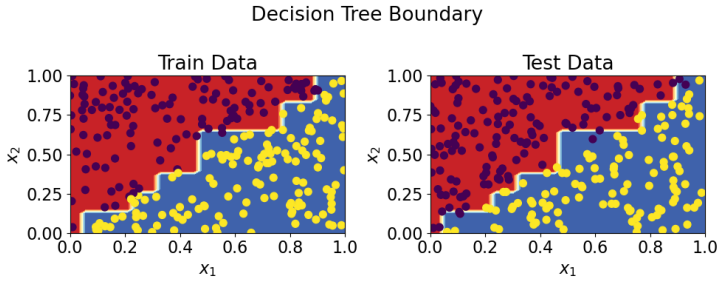
(a) Load the dataset `data_exercise_1.pkl` provided in the Moodle-Course.

(b) Split the dataset into random train and test subsets, with the train subset accounting for 50% of the total data.

(c) Implement a decision tree using the `sklearn` library. Fit the decision tree using the train dataset.

(d) Utilize the trained decision tree to predict the labels of the test dataset. Evaluate the prediction performance by applying the F-Score metric.

(e) Generate the following figure with two subplots that shows the train dataset (using ground truth labels) on the left subplot and the test dataset (using ground truth labels) on the right subplot. Additionally plot the learned decision boundary of the decision tree using the built-in function `DecisionBoundaryDisplay(*, xx0, xx1, response, xlabel=None, ylabel=None)` from `sklearn.inspection`.



2. Exercise: Ensemble Method - Random Forest Algorithm (Classification)

Ensemble methods using decision trees are a form of machine learning model which combine the predictions from multiple decision trees to make a final prediction. The idea behind these methods is that by combining the results of several simple models (like decision trees), you can build a model that is more powerful and robust than any of the individual on their own. To implement such an ensemble method based on decision trees, follow these steps:

(a) Load the dataset `data_exercise_2.pkl` provided in the Moodle course.

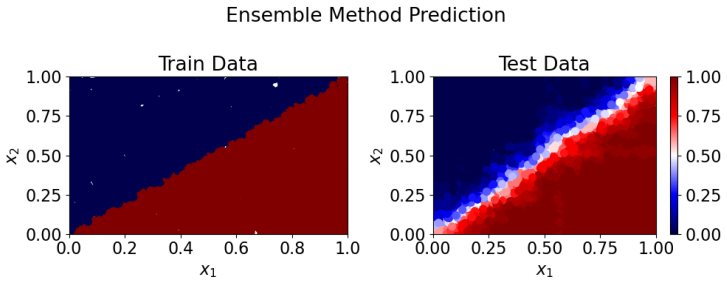
(b) Generate and train $N = 30$ different decision trees by resampling the data with replacement (also known as bootstrapping) using the `sklearn` library. Therefor, repeatedly split the dataset into random train and test subsets, with the train subset accounting for only 1% of the total data. Hence, each tree is trained on a different set of observations.

(c) For the evaluation of the ensemble method, split the dataset into random train and test subsets, with the train subset accounting for only 50% of the total data.

(d) During the prediction phase, each decision tree independently generates a class prediction from the test dataset. The concluding prediction is then established based on the principle of majority voting. Apply this for the generated test data set.

(e) Evaluate the prediction performance of the ensemble method using Cross-Entropy loss.

(f) Generate the following figure with two subplots that shows the train dataset (using ground truth labels) on the left subplot and the test dataset (using the predicted label probability of the ensemble method) on the right subplot.



3. Exercise: Gini Impurity

(a) Load the dataset `gini_imp.pkl` provided in the Moodle course.

(b) Write a function called `gini_impurity()`, which calculates and returns the Gini Impurity of a dataset.

(c) Write a function called `split_data()` that splits a data set based on the x_1 feature. Given a position value x_{div}, the data is split into two subsets according to the values of that feature. Data samples $x_{1,i} <= x_{\text{div}}$ are assigned to set 1; data samples $x_{1,i} > x_{\text{div}}$ are assigned to set 2.

(d) Write a python function `split_gini_impurity()` to compute the overall Gini Impurity upon splitting a dataset. This function should first calculate the Gini Impurity for each partition separately, and subsequently generate a weighted sum considering the individual (split) dataset sizes.

(e) Compute the Gini Impurity for the split positions $x_{\text{div}} = [-5, -2, 0, 1, 10]$ and generate a plot, where the split position x_{div} is marked with a red dashed line. Replace the question marks in the subplot titles with the computed Gini Impurity for each split.