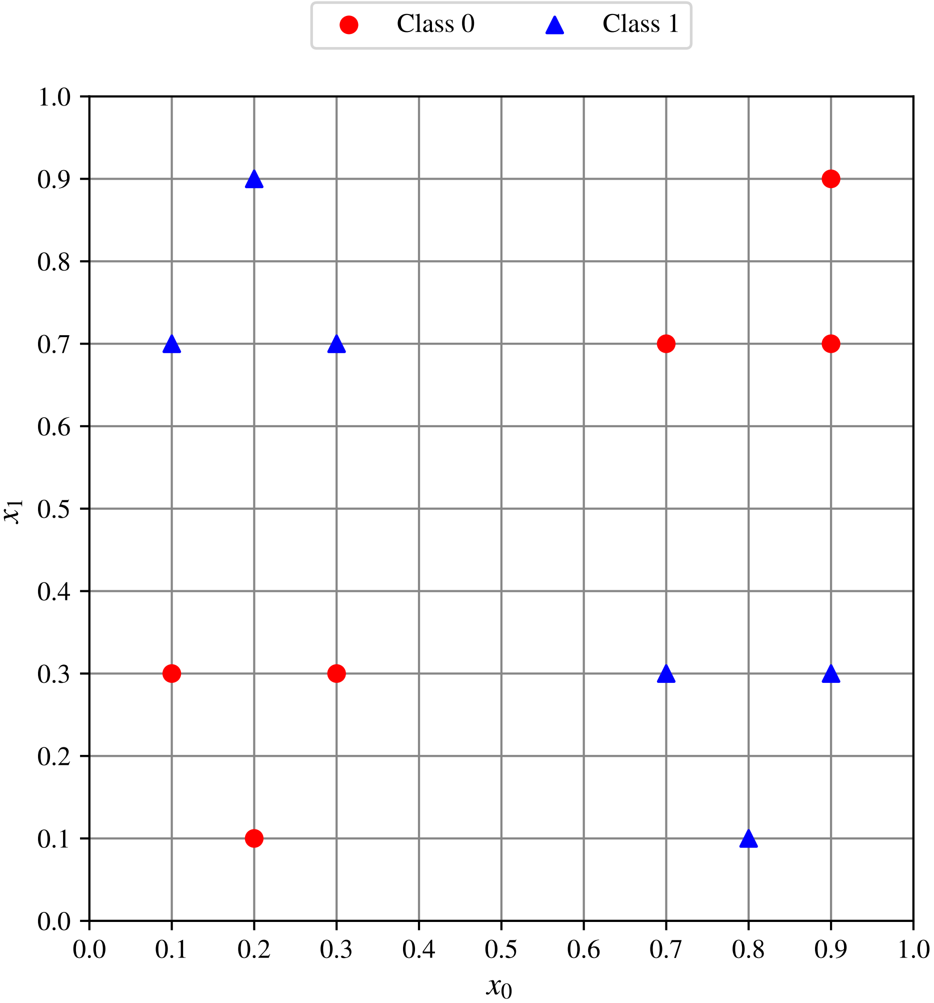
6 Decision Trees and Random Forest

To obtain extra points for the written exam, prepare the solution for **Programming Task 6.3** using the Jupyter notebook provided and upload your work to StudOn. Students may discuss with each other while preparing the solution, but each student must submit their work individually. Refer the Supplements page in StudOn for details regarding the submission deadline.

6.1 Decision Tree for classification



The scatter plot of a binary classification dataset is given in the figure above. The goal of this task is to apply a classification tree model to this dataset. The node purity is measured by the Gini score:

$$G = \sum_{k=1}^{N_k} p_k \, (1 - p_k), \tag{6.1}$$

where p_k is the fraction of samples in the node that belong to class k. The objective during training is to split the dataset into regions that are concentrated with samples of a particular class. Such regions will have low Gini scores. The decrease in Gini score is quantified by the goodness-of-split:

$$\Delta G = G_P - \left(\frac{N_L}{N_P} \cdot G_L + \frac{N_R}{N_P} \cdot G_R \right), \tag{6.2}$$

where G_P, G_L, G_R are the Gini scores of the parent node, left node and right node and N_P, N_L, N_R are the number of samples in the parent node, left node and right node.

i. Calculate the goodness-of-split at the root node for each of the following split functions:

   (a) $x_0 \leq 0.25$

   (b) $x_1 \leq 0.5$

   (c) $x_0 \leq 0.85$

ii. Grow the tree until each leaf node has a Gini score of 0. Sketch the tree indicating the selection function and the Gini score at each node.

iii. Using the tree, predict the output for the following input data samples:

   (a) $\mathbf{x} = [0.7 \quad 0.9]^\top$

   (b) $\mathbf{x} = [0.5 \quad 0.5]^\top$

6.2 Programming Task: Regression Trees

The datasets in files `train-reg-tree.csv` and `test-reg-tree.csv` contain samples from a synthetic dataset for training a Regression Tree. The dataset consists of 3 columns: the first two columns, denoted as x1 and x2, represent the input features for each data sample, and the last column represents target value denoted by y. There are 200 samples in the `train-reg-tree.csv` and 100 samples in the `test-reg-tree.csv`

Given a node M containing N_M samples, the node impurity can be expressed by the node sample variance:

$$V_M = \frac{1}{N_M} \sum_{i=1}^{N_M} (Y_i - \overline{Y}_M)^2, \tag{6.3}$$

where Y_i is the target of the sample i and $\overline{Y}_M$ is the mean target of all samples in the node. Similar to equation 6.2, the goodness-of-split is given by:

$$\Delta V = V_P - \left(\frac{N_L}{N_P} \cdot V_L + \frac{N_R}{N_P} \cdot V_R \right), \tag{6.4}$$

where V_P, V_L, V_R are the variances of the parent node, left node and right node respectively.

i. Complete the missing code in the implementation of the Regression tree.

ii. Train the above regression tree using the train dataset and obtain the mean square error (MSE) for the model predictions on the test dataset.

iii. Compare your results using the model trained using the DecisionTreeRegressor class from the scikit-learn library.

6.3 Programming Task: Song popularity prediction using Random Forest

The goal of this task is to train a random forest model that predicts the song popularity using the datasets already provided in task 4.3.

i. Implement a function that draws a bootstrap sample of size N from the train dataset, where N can be specified by the user.

ii. Complete the implementation of the random forest algorithm. For this task you may use the DecisionTreeClassifier from the scikit-learn library. The other parts of the random forest algorithm must be implemented using only Scipy/Numpy.

iii. Train the model for the dataset from `train-songs.csv` using the parameters given below.

Parameter	Value
Number of trees	100
Maximum features per tree	2
Bootstrap sample size	20000
Minimum node size	1
Maximum tree depth	10

iv. Calculate the accuracy of the model using the test dataset and compare your results with the RandomForestClassifier from the scikit-learn library using the following parameters.