Machine Learning

Lab 08: Decision trees

Task 1.

In this task you will analyse the performance of a decision tree classifier for the Iris dataset.

A. Load the Iris dataset from the dataset repository of SciKit learn.
B. Create a loop to evaluate the performance of different Decision Tree Classifiers of increasing maximum depth ranging from a minimum of max_depth=1 to a maximum of max_depth=10.
C. For each of these Decision Tree Classifiers create a loop to run the following evaluation 100 times: each time create a k-fold cross validation creating n_splits=5 random splits, train the classifier on the training data and evaluate the classifier on both the test as well as the training data. Collect the accuracy score of each evaluation in a list.
D. For each maximum tree depth evaluated in subtask C, plot the decision tree for one of the training instances. What can you observe?
E. Plot the mean prediction accuracy both on the training as well as on the test data as collected in subtask C against the maximum tree depth. Which tree-depth performs best?

Task 2. (Matrix inverse)

The $\mathrm{n}\times\mathrm{n}$ identity matrix is the quadratic matrix that has ones on the diagonal and zeros everywhere else, i.e.

$$I_n = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

If we multiply this matrix with a vector it does not change, i.e.

$$I_n a = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = a$$

A quadratic $n \times n$ matrix can have an inverse, i.e. a matrix that multiplied with it evaluates to the identity matrix, i.e.

$$A^{-1}A = I$$

This inverse matrix A^{-1} may or may not exist. If it exists, we call the matrix regular, if it does not exist we call the matrix singular. An outer product AA^T, for a matrix with more rows than columns is always singular.

A. Create a non-zero 3×2 matrix A. Calculate its inner product $A^T A$ and its inverse $(A^T A)^{-1}$. Print all three matrices. Now try to calculate the inverse of the outer product AA^T and observe what happens.

The inverse of a matrix is useful for solving systems of linear equations. Given the $n \times n$ matrix A and the n-vector b the solution to the linear equation system

$$Ax = b$$

can be found by multiplying both sides of the equation from left with A^{-1}. Because we already saw that

$$A^{-1}Ax = Ix = x$$

we can obtain the solution as

$$x = A^{-1}b$$

B. Create a non-zero 2-vector x and calculate $b = A^T Ax$ using the matrix product from subtask A. Now use the inverse from subtask A and calculate $(A^T A)^{-1}b$. Print all matrices and observe the result.