UNIVERSITY
OF MANNHEIM
School of Business Informatics
and Mathematics

Data Mining

Exercise 5: Classification

5.1. Learning a classifier for the Iris Data Set

In the last exercise, you have learned lazy classification models for the Iris dataset. Now try a Decision Tree based approach with 10-fold cross-validation.

1. Discretise the Iris data set into three bins. Then use the DecisionTreeClassifier with a 10-fold stratified cross validation and compute the accuracy. Afterwards plot the decision tree.
2. Remove the discretization and adjust the max_depth parameter of DecisionTreeClassifier to increase the accuracy. Does the accuracy change? Compare the complexity of the two models. Which model should be preferred according to Occam's razor?

5.2. Parameter optimization

In Exercise 4.1 we have used the German credit data set from the UCI data set library (http://archive.ics.uci.edu/ ml/index.html), which describes the customers of a bank with respect to whether they should get a bank credit or not. The data set is provided as *credit-g.arff* file in ILIAS.

1. (recap) Go back to the results of exercise 4.1.4, in which you have compared Naïve Bayes, Logistic Regression, k-NN (k=5) classifiers. In that exercise you
    a. Used the 10-fold cross-validation approach.
    b. Balanced the training set multiplying the "bad customer" examples.
    c. Evaluated the results, setting up your cost matrix to ((0,100)(1,0)) – thus, you assumed you will lose 1 unit if you refuse a credit to a good customer but you lose 100 units if you give a bad customer a credit.

    Rerun your process to get the performance results. Now additionally use a *Decision Tree Classifier. How does it perform? What are the default parameters of this classifier?*

2. Now, try to find a more appropriate configuration for the Decision Tree classifier. Use the *GridSearchCV* from scikit-learn. Try the following parameters of the Decision Tree:
    - criterion: ['gini', 'entropy']
    - 'max_depth': [1, 2, 3, 4, 5, None] (What does None mean?)
    - 'min_samples_split': [2,3,4,5]

    You should come up with 48 (2 x 6 x 4) combinations.

    What is the best configuration for the data set and the classification approach?

3. What is the cost of misclassification for this configuration?
4. How does the optimal decision tree differ from the one you have learned in 5.2.1?