UNIVERSITY OF MANNHEIM

Data and Web Science Group
Prof. Dr. Christian Bizer
B6 – B1.15
68159 Mannheim

# Data Mining – FSS 2020

## Exercise 5: Classification

### 5.1. Parameter optimization

In Exercise 4.2 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. You need to use the RapidMiner ARFF reader operator to import the data set. Please also have a look at the data set documentation that is included in the file.

1. (recap) Go back to the results of exercise 4.2.4, in which you have compared Rule Induction, k-NN (k=5) and Decision Tree classifiers. In that exercise you
    a. Used the 10-fold X-Validation approach.
    b. Balanced the training multiplying the "bad customer" examples by using *Filter Examples* and *Append* operators.
    c. Used the *Performance (Costs)* operator to evaluate the results, setting up your cost matrix to ((0,100)(1,0)) – that is, you assumed you will lose 1 Unit if you refuse a credit to a good customer, but that you lose 100 Units if you give a bad customer a credit.

    Rerun your process to get the performance results. What were the default parameters of the *Decision Tree* operator?

    **Solution**: Rerun the process from 4.2.4.

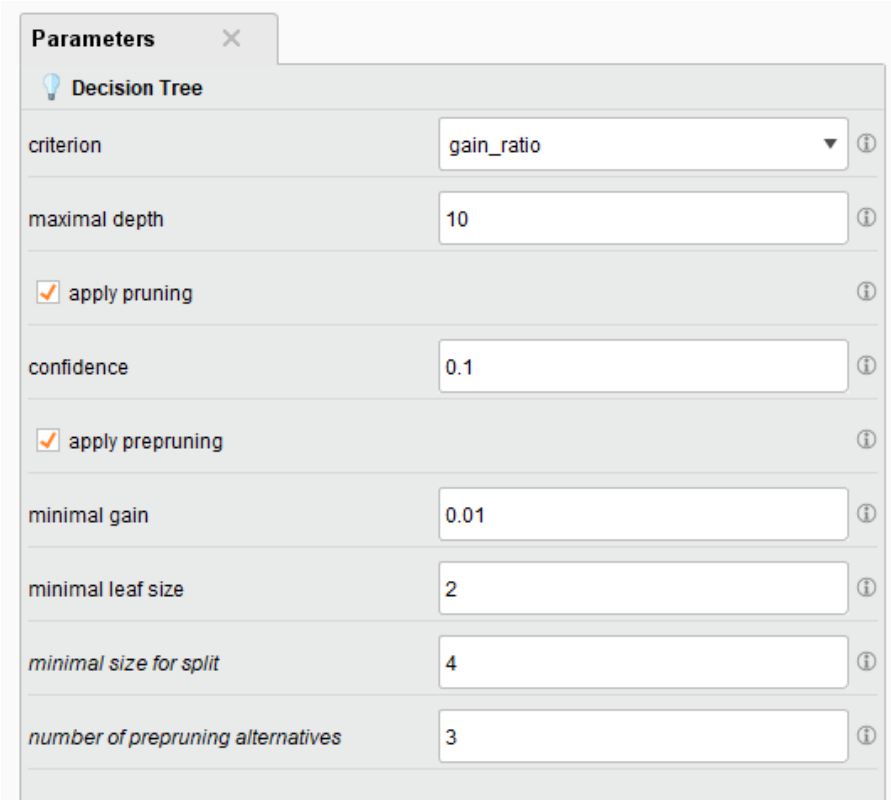
    Misclassification costs you should get are:

    *11.223 for rule induction,*

    *17.617 for k-NN (k=5),*

    ***11.437 for decision trees**.*

    Parameters of the Decision Tree operator:




2. Now try to find a more optimal configuration for the Decision Tree operator. Use Nested X-Validation and the *Optimize Parameter* Operator with an inner X-Validation. Then use the "Edit Parameter Settings" option of this operator to let RapidMiner test different combinations of parameters. Try the following parameters of the *Decision Tree* operator:
   - CRITERION (information_gain, gain_ratio, gini_index, accuracy)
   - APPLY_PRUNING (true, false)
   - MAXIMAL_DEPTH (try the range from -1 to 50 with 10 steps). What does -1 mean?

   You should come up with 88 (4 x 2 x 11) combinations.

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

   **Solution:** Build up the process as described and select the three parameters Decision Tree.criterion, Decision Tree.apply_pruning and Decision Tree.maximal_depth in the optimization operator. Set the boundaries for the maximal_depth.

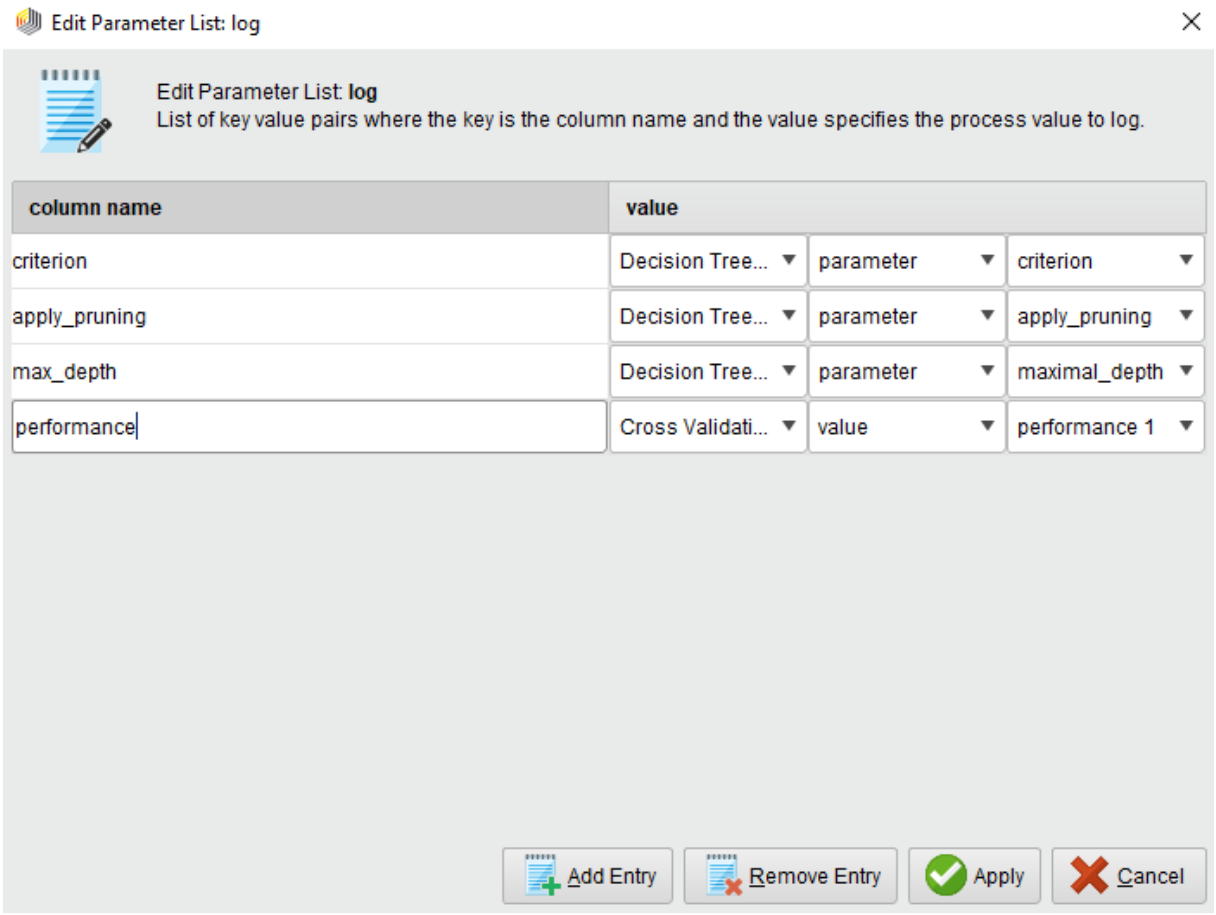
   **Conclusion:** The optimal setup is:

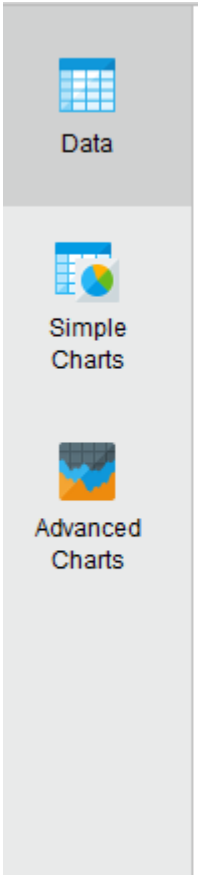
   - X-Validated Performance (Misclassification cost) = 6.782 +/- 1.342
   - Decision Tree.criterion = gain_ratio
   - Decision Tree.apply_pruning = true
   - Decision Tree.maximal_depth = 14

3. What is the misclassification cost for this configuration? Can the same cost be obtained with other setups? Use the *Log* operator to find out.

**Solution:** Place the Log operator after the X-Validation operator within the Optimize Parameters Operator. In the results perspective you can see a tab called “log” which presents – already before the whole process is done – the intermediate results.

This is a way to configure the Log operator:






Log (88 rows, 4 columns)

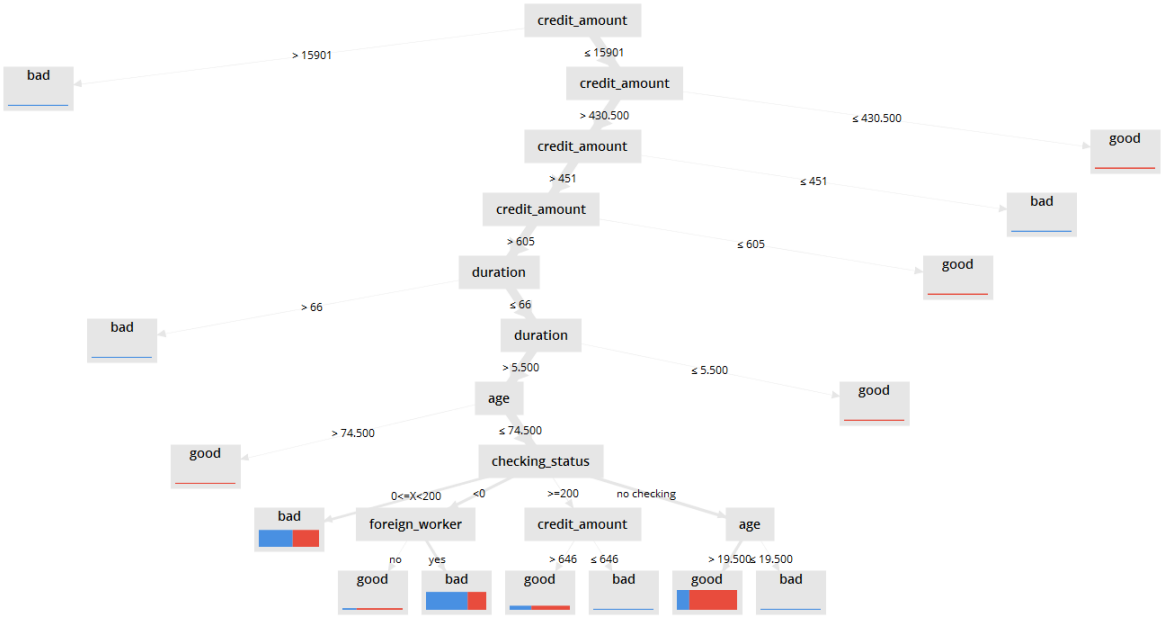
| criterion | apply_pruning | max_depth | perf... ↑ |
|---|---|---|---|
| gain_ratio | true | 14 | 6.782 |
| gain_ratio | false | 14 | 6.782 |
| gain_ratio | true | 19 | 8.357 |
| gain_ratio | false | 19 | 8.359 |
| information_gain | false | 4 | 8.653 |
| information_gain | true | 4 | 8.654 |
| gini_index | false | 4 | 9.705 |
| gini_index | true | 4 | 9.706 |
| gain_ratio | true | 25 | 10.127 |
| gain_ratio | false | 25 | 10.128 |
| accuracy | false | 4 | 10.411 |
| accuracy | true | 4 | 10.412 |

**Conclusion:** Actually, there are two configuratios that give you the best results (least misclassification costs). When you sort the log (data view) by performance (cost) you can see that gain_ratio as a criterion together with both modalities for pruning and with the max_depth equal to 14 gives you the same result.

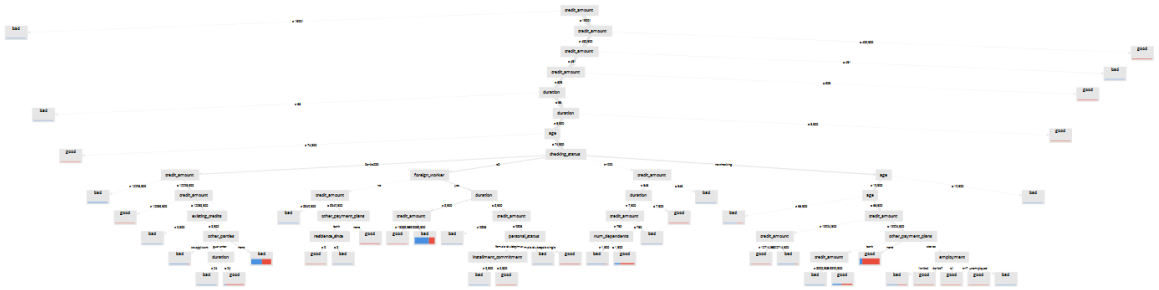
4. How does the optimal decision tree differ from the one you have learned in 4.2.4?

**Solution:** Set up two x-validation processes with DecisionTree operators inside, with default and optimal parameters, respectively.

The resulting decision tree for the default configuration:




The resulting decision tree for the optimal configuration:



In the optimal decision tree more attributes are examined, which makes more sense than looking just at the credit amount.

## 5.2. Open Competition: Finding rich Americans

The Adult data set from the UCI data set library (http://archive.ics.uci.edu/ml/datasets/Adult) describes 48842 persons from the 1994 US Census. The data set is provided as *adult.arff* file on the website of this course.

Your task is to find a good classifier for determining whether a person earns over 50.000 $ a year. Beside of being accurate, your classifier should also have balanced precision and recall.

To evaluate your classifiers use split validation (split ratio=0.8, linear sampling).

In order to find the best classifier, you may experiment with:

1. different algorithms
2. different parameter settings
3. the balance of the two classes in the data set

4. the set of attributes that are used or not used
5. other preprocessing techniques

People are described by the following 14 attributes:

| | |
|---|---|
| **age** | continuous |
| **workclass** | Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. |
| **fnlwgt** | continuous |
| **education** | Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. |
| **education-num** | continuous |
| **marital-status** | Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. |
| **occupation** | Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. |
| **relationship** | Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. |
| **race** | White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. |
| **sex** | Female, Male. |
| **capital-gain** | continuous |
| **capital-loss** | continuous |
| **hours-per-week** | continuous |
| **native-country** | United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands. |

In order to increase your understanding of the data set, you might want to visualize different attributes or attribute combinations.

**Solution**: When looking at the dataset (Meta Data View) the following points can be found out:

1. Big dataset with over 45k records (in comparison to the other dataset which were processed in the exercises)
2. The dataset is unbalanced 3:1 (“<=50k”, “>50k”)
3. Numerical and nominal attributes are included
4. Ranges of numerical attributes differ e.g. education-num [1; 16] and capital-gain [0; 100k)
5. Some attribute values are missing e.g. workclass and marital-status

**Conclusion**: In the first place we just try all discussed classification methods without any pre-processing and have a look at the accuracy and precision/recall.

A → <= 50K, B→ > 50K

| Method | Accuracy | Recall/ Precision | Description |
|---|---|---|---|
| Decision Tree (Default) | 83,12% | r(A) = 98,91% | - complex tree |

| | | | |
|---|---|---|---|
| | | r(B) = 34,73%<br>p(A) = 82,28%<br>p(B) = 91,26% | - bad recall for B |
| Decision Tree (Size for Split 10, Min Leaf Size: 5) | 83.20% | r(A) = 99.33%<br>r(B) = 33.78%<br>p(A) = 82.13%<br>p(B) = 94.31% | - simpler tree |
| Decision Tree (Size for Split 20, Min Leaf Size: 10) | 82.96% | r(A) = 98.77%<br>r(B) = 34.57%<br>p(A) = 82.22%<br>p(B) = 90.13% | |
| Decision Tree (Size for Split 50, Min Leaf Size: 25) | 82.96% | r(A) = 98.76%<br>r(B) = 34.57%<br>p(A) = 82.22%<br>p(B) = 90.13% | |
| k-NN (k=3) | 76.41% | r(A) = 88.43%<br>r(B) = 39.60%<br>p(A) = 81.77%<br>p(B) = 52.77% | - 2:55 processing time<br>- produces bad results for r(B) and p(B) |
| k-NN (k=5) | 77.64% | r(A) = 91.80%<br>r(B) = 34.28%<br>p(A) = 81.06%<br>p(B) = 57.70% | - Processing time similar to k=3 (2:55)<br>- Produce bad results for r(B) and p(B) |
| k-NN (k=10) | 79.72% | r(A) = 97.25%<br>r(B) = 23.98%<br>p(A) = 80.26%<br>p(B) = 73.29% | - Processing time similar to k=3 (2:55)<br>- Produce bad results for r(B) |
| Naïve Bayes | 83.65 | r(A) = 93.55%<br>r(B) = 53.33%<br>p(A) = 85.99%<br>p(B) = 72.97% | - so far best recall for B |

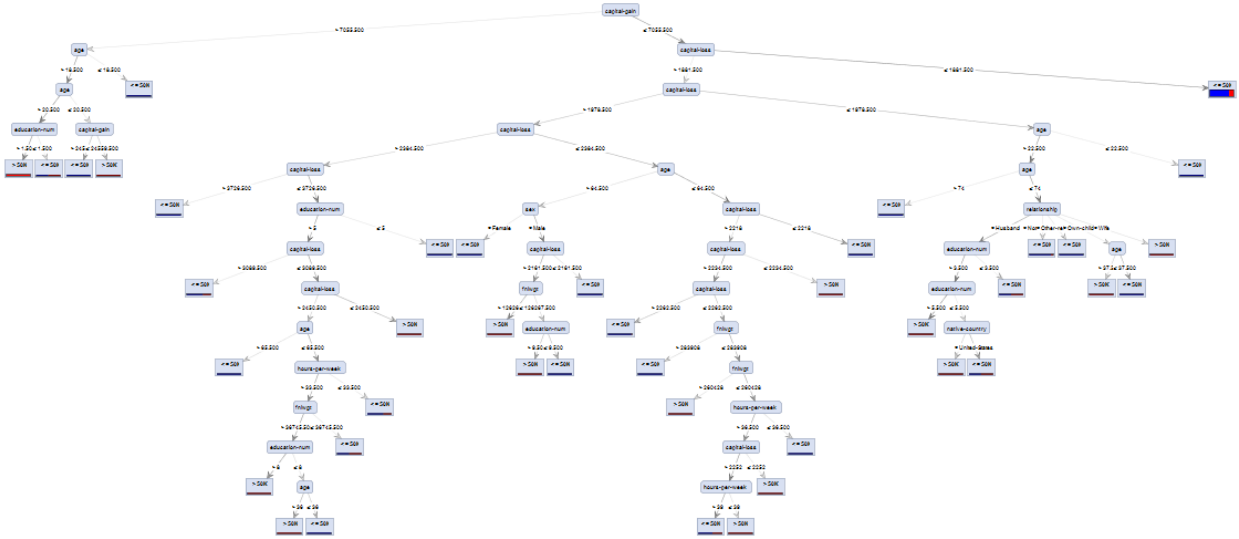


*Figure 1: decision tree with default settings*

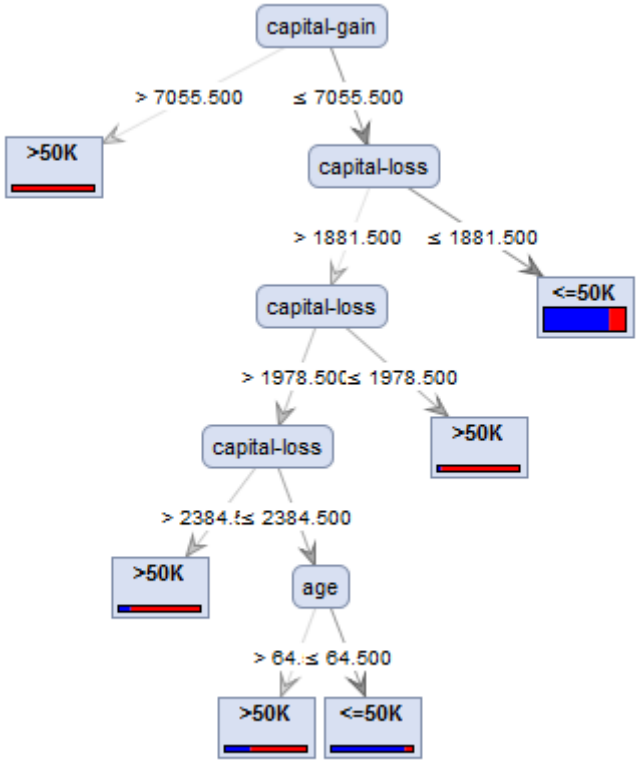



*Figure 2: decision tree with size for split=50, min leaf size=25*

In a second step we balance the data (almost 1:1) set and try our most promising methods again.

| Method | Accuracy | Recall/ Precision | Description |
|---|---|---|---|
| Decision Tree (Size for Split 10, Min Leaf Size: 5) | 74.88% | r(A) = 70.33%<br>r(B) = 89.36%<br>p(A) = 95.46%<br>p(B) = 48.65% | - Decrease in accuracy<br>- Strong increase in r(B)<br>- Strong decrease in p(B) |
| k-NN (k=5) | 60.10% | r(A) = 56.75%<br>r(B) = 70.76%<br>p(A) = 86.05%<br>p(B) = 33.98% | - Processing time increase<br>- Results get worse<br>- Recall on B increase to over 70%<br>- but Precision on B decreased |
| Naïve Bayes | 83.48% | r(A) = 90.98%<br>r(B) = 56.60%<br>p(A) = 87.75%<br>p(B) = 67.53% | - Fast processing time<br>- Improved results on r(B) with slight decrease of accuracy |

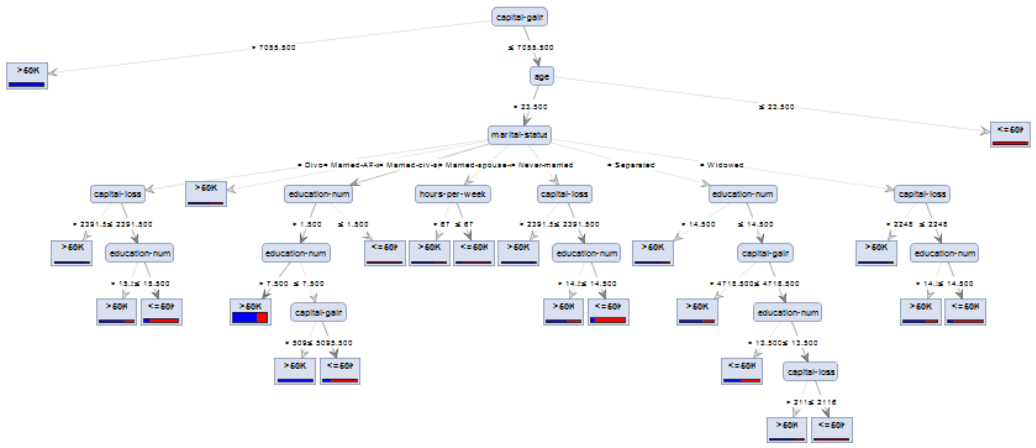



*Figure 3: decision tree trained with balanced data*

Alternatively sampling could be used to balance the data using the *Sample* operator. As we have at least 11k records of each class we can use this as balance value.

| Method | Accuracy | Recall/ Precision | Description |
|---|---|---|---|
| Decision Tree (Size for Split 4, Min Leaf Size: 2, Default config) | 72.21 % | r(A) = 65.92%<br>r(B) = 91.47%<br>p(A) = 95.95%<br>p(B) = 46.70% | |
| k-NN (k=5) | 62.43% | r(A) = 62.60%<br>r(B) = 61.90%<br>p(A) = 83.42%<br>p(B) = 35.08% | |
| Naïve Bayes | 83.94% | r(A) = 91.15%<br>r(B) = 61.86%<br>p(A) = 87.98%<br>p(B) = 69.52% | - Improved results on r(B) with slight decrease of accuracy |

Taking a look at the attributes (e.g. having run Naïve Bayes) some attributes might support the learning of a good model more than others. Attributes which are not overlapping or are too similar according to our two classes are: relationship, occupation, education, marital status, cap_gain, cap_loss. It is possible to have more or less in the selection.

Running the Naïve Bayes again with selected attributes plus discretization (cause of the high ranges of numerical values) the following results can be achieved:

Accuracy: 79.64%

r(A) = 81.23%

r(B) = 74.75%

p(A) = 90.79%

p(B) = 56.58%

Especially precision and recall are distributed equally over the two classes

## 5.3. Have RapidMiner help you!

RapidMiner offers operators to automate the process of

- feature selection (automatically try different attribute combinations in order to find the attribute combination that works best for learning) as well as

- to automatically try out different parameter settings of the learning algorithms in order to find the parameter setting that results in the best performance of the learned model.

- Rapidminer Tutorial on Accidental Contamination through Feature Selection and Parameter Optimization
    - https://rapidminer.com/blog/learn-right-way-validate-models-part-4-accidental-contamination/