Tutorial on Machine Learning.
Part 2. Descriptor Selection Bias.

Igor Tetko, Igor Baskin and Alexandre Varnek

1. Introduction

The *n*-fold cross-validation technique is widely used to estimate the performance of QSAR models. In this procedure, the entire dataset is divided into *n* non-overlapping pairs of training and test sets. Each training covers *(n-1)/n*th of the dataset while the related test set covers the remaining 1/*n*th. Following developments of models with the training set, the predictions for the test set are performed. Thus, predictions are made for all molecules of the initial dataset, since each of them belongs to one of the test sets. This tutorial demonstrates how crucial unbiased descriptor selection is for correct assessment of the prediction performance of the models. In principle, two scenarios are possible:

- selection of descriptors using all molecules from the parent data set followed by *n*-fold cross-validation (*internal CV*);
- selection of descriptors is performed on each fold using *(n-1)/n*th of the dataset (*external CV*).

The models obtained on each fold of the *internal CV* are based on the same set of descriptors, whereas corresponding models for the *external CV* may involve different descriptors. Notice that frequently reported *Leave-One-Out* cross-validation typically represents the *internal CV*.

Here, we will show that only *external CV*, in which the information of test compounds is not used for the development of the models, can be used for reasonable assessment of accuracy of predictions (see also [1-3]). For this purpose, a set of random numbers will be used as molecular descriptors to develop a model for boiling points of alkanes. Thus, a procedure leading to “robust” QSAR models, e.g. models with high Pearson correlation coefficient R^2, based on those descriptors is definitely not correct one.

The tutorial consists in 3 parts (sections 4.1 - 4.3):

- descriptors selection is performed before the model development (internal CV, Fig 1A);
- descriptors selection is used to optimize the model (internal CV, Fig 1B);
- descriptors are selected in parallel with the development of the model (external CV, Fig 1C).

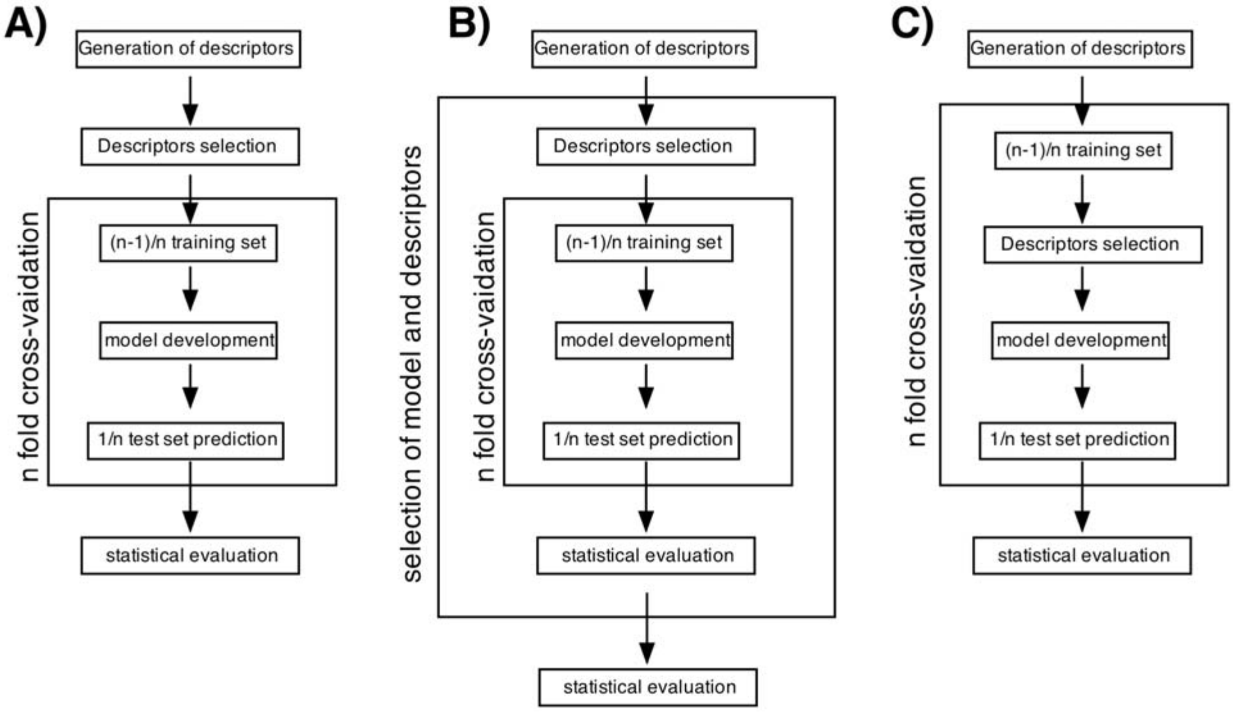


Figure 1. Internal (A, B) and External (C) cross-validation procedures.

2. Datasets and Descriptors

A database containing the values of the boiling points of 74 alkanes [4] is used. Two sets of descriptors are used: 100 and 1000 descriptors representing random numbers. The *k* Nearest Neighbors (kNN, or **IBk** in Weka's terminology) and Multi-Linear Regression (MLR) methods will be used for the modeling.

3. Files

1. *alkan-bp-louse100.arff* – dataset with 100 descriptors taking random values
2. *alkan-bp-louse1000.arff* – dataset with 1000 descriptors taking random values
3. *preselected_descr.arff* – dataset with preselected descriptors
4. *knn_descr.arff* – dataset with descriptors selected by the kNN procedure

4. Modeling

4.1 Internal Cross-Validation using Preliminary Selected Descriptors

In *Weka*, in order to select descriptors (attributes in Weka's terminology), one should specify a **Search Method** and an **Attribute Evaluator**. The **Search Method** stands for a search algorithm (such as *ExhaustiveSearch*, *GeneticSearch*, *BestFirst*, etc), whereas the **Attribute**

Evaluator specify a way how to compute the value being optimized in the course of descriptors selection.

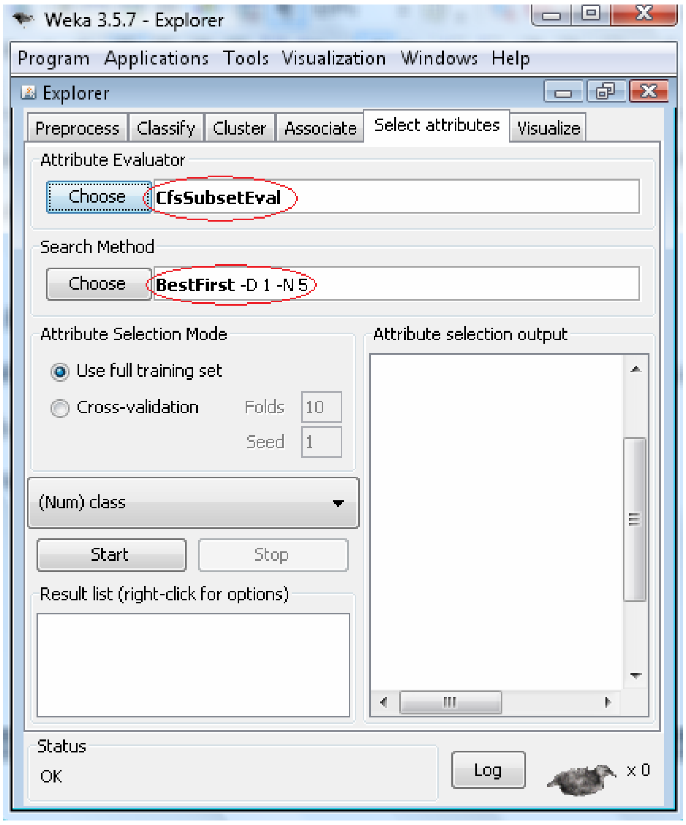
The default **Search Method** in *Weka* is **BestFirst**. It searches the space of descriptor subsets by greedy hill-climbing augmented with a backtracking facility. The **BestFirst** method may start with the empty set of descriptors and searches forward (default behavior), or starts with the full set of attributes and searches backward, or starts at any point and searches in both directions (by considering all possible single descriptor additions and deletions at a given point).

The default **Attribute Evaluator** in *Weka* is **CfsSubsetEval**. This method evaluates the worth of a subset of descriptors by considering the individual predictive ability of each one along with the degree of redundancy between the descriptors. Subsets of descriptors that are highly correlated with the property/activity values and having low intercorrelation are preferred (see [5]).

For this problem, **BestFirst** (search method) and **CfsSubsetEval** (attribute evaluator) combination is as efficient as best variable selection techniques - genetic algorithm or simulated annealing - but it is much quicker. This is why these default settings were selected for the tutorial.

- Start Weka
- Select the item **Explorer** from the **Applications** menu
- Click on ***Open file…*** and load the file *alkan-bp-louse1000.arff.*
- Click on the ***Select attributes*** item

The following window appears:



Weka 3.5.7 - Explorer

Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Num) class

Status

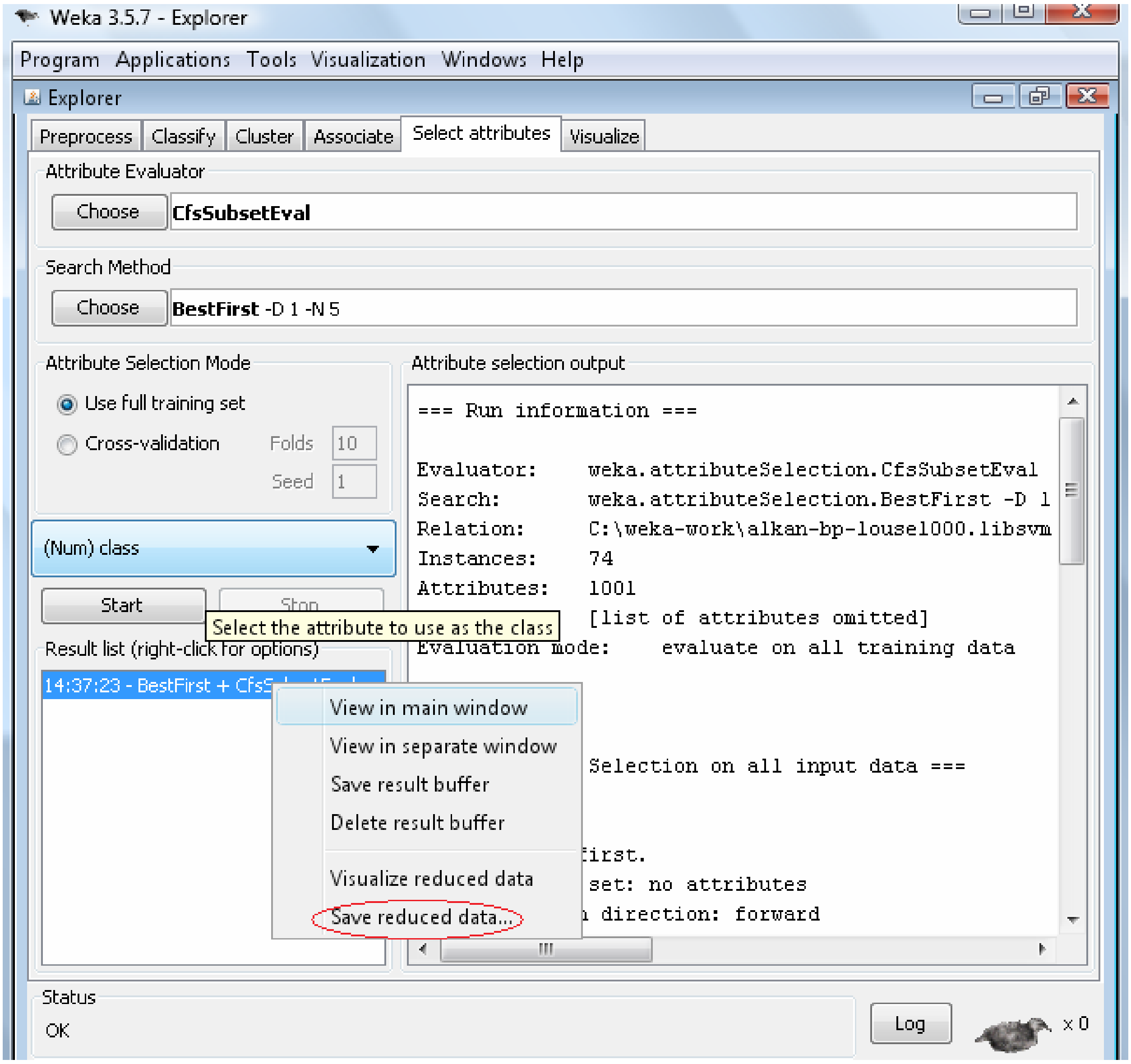
OK

- Click on ***Start***

Computation finishes after 2-3 seconds (application of the genetic algorithm to the same dataset would require hours).

- Click on the new line in the *Result list* with the right mouse button
- From the pop-up menu, select the item ***Save reduced data…***

The program window looks as a following:



CfsSubsetEval

BestFirst -D 1 -N 5

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Num) class

14:37:23 - BestFirst + CfsS

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval
Search:       weka.attributeSelection.BestFirst -D 1
Relation:     C:\weka-work\alkan-bp-louse1000.libsvm
Instances:    74
Attributes:   1001
              [list of attributes omitted]
Evaluation mode:    evaluate on all training data



            Selection on all input data ===


           irst.
            set: no attributes
           n direction: forward
```

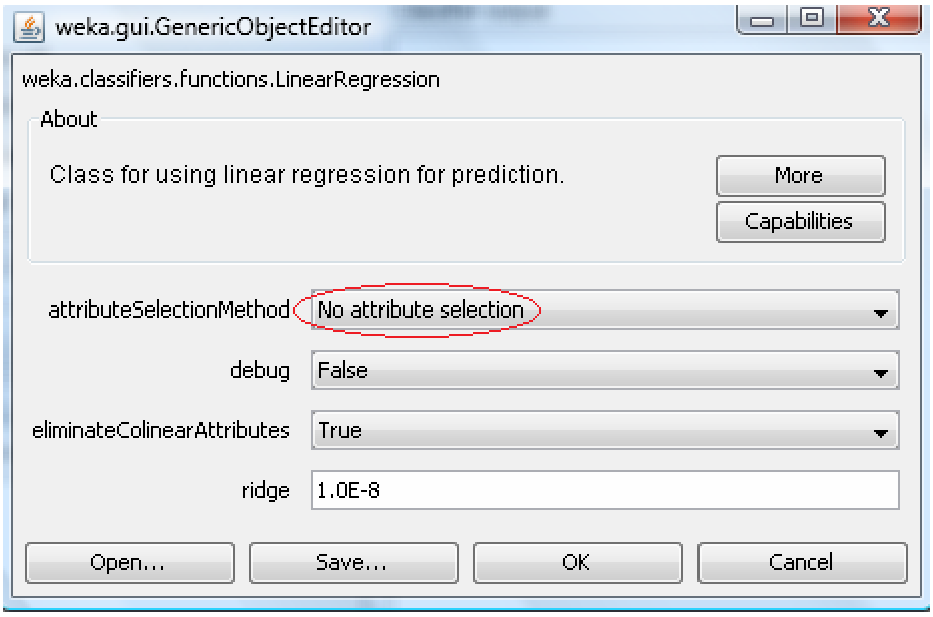
Status
OK

- Save the dataset with 30 selected descriptors to file *preselected_descr.arff*

Selected descriptors are supposed to be used in the Multiple Linear Regression model.

- Switch to the **Preprocess** submode of the **Explorer** mode

- Click on ***Open file…*** and open the file *preselected_descr.arff*
- Switch to the ***Classify*** submode
- Click on ***Choose***
- From the hierarchical list of machine learning methods choose ***weka/classifiers/functions/LinearRegression***
- To change options for the MLR method, click on **LinearRegression.** A new window with parameters of MLR appears.
- Change **attributeSelectionMethod** to ***No attribute selection***
  All settings are shown on the snapshot below



weka.classifiers.functions.LinearRegression

About

Class for using linear regression for prediction.

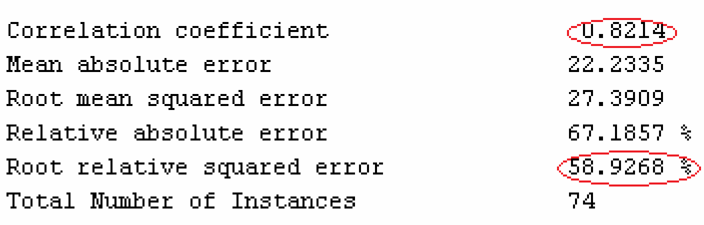
attributeSelectionMethod: No attribute selection

debug: False

eliminateColinearAttributes: True

ridge: 1.0E-8

- Press the ***OK*** button in this window
- Press the Start button in order to test MLR on the subset with selected descriptors. The obtained results are:



```
Correlation coefficient                  0.8214
Mean absolute error                     22.2335
Root mean squared error                 27.3909
Relative absolute error                 67.1857 %
Root relative squared error             58.9268 %
Total Number of Instances               74
```

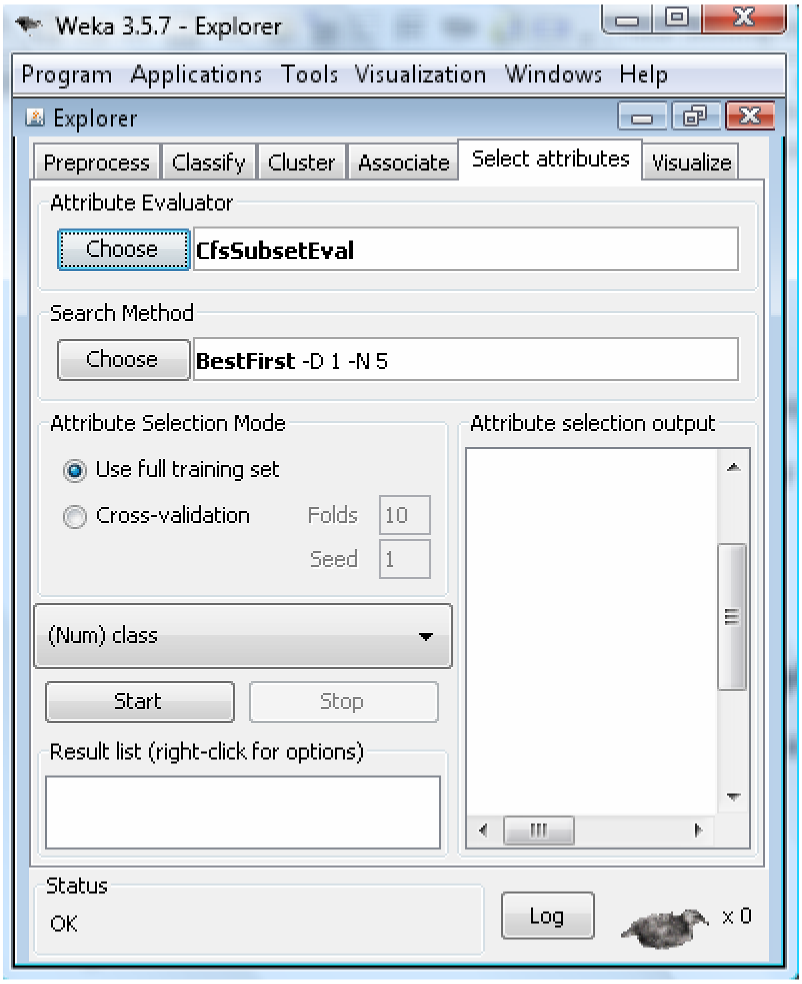
Thus, this study shows that the correlation coefficient between predicted in cross-validation values of property and their experimental value is **0.8214**. This indicates a statistical significance of the model based on random numbers. One can show that the similar situation is observed for some other popular machine learning methods (PLS, kNN, etc.). Hence, the whole procedure of building the model using a set of preliminary selected descriptors is erroneous.

4.2 Internal cross-validation using descriptors selected in course of model building

In this section, the *k* Nearest Neighbor approach will be used both to optimize the *k* parameter and to select an optimal set of descriptors by minimizing the cross-validation error of the method.

- Start or restart *Weka*
- Select the item **Explorer** from the **Applications** menu
- Click on ***Open file…*** and load the file *alkan-bp-louse100.arff.*
- Click on the label ***Select attributes***

The following window appears.



Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

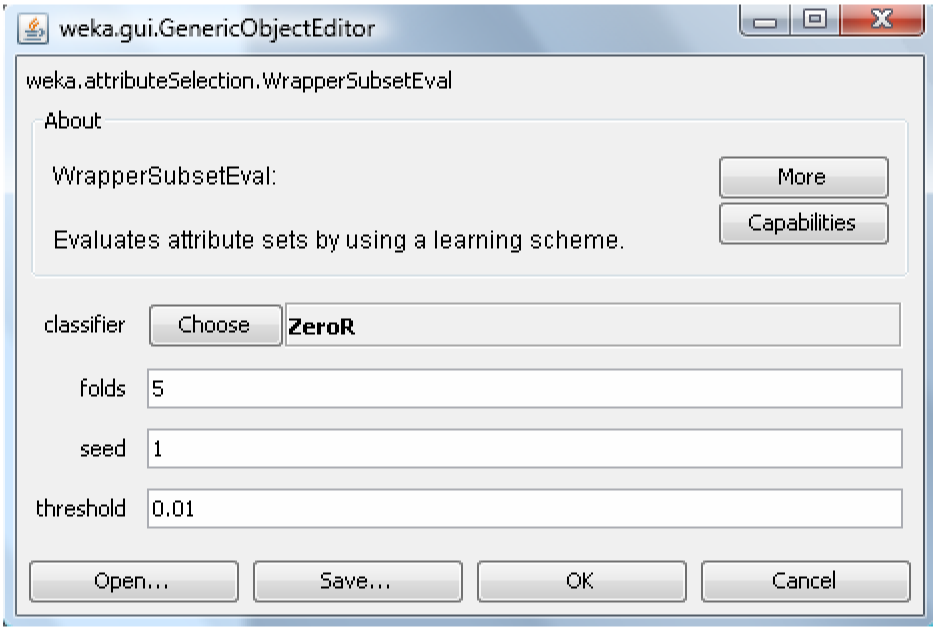
(Num) class

Status
OK

Under these settings, the minimization of the cross-validation error of kNN models is used to guide the descriptors selection.

- Click on the ***Choose*** button under the label **Attribute Evaluator**
- Choose menu item ***WrapperSubsetEval***
- Click on ***WrapperSubsetEval***

The following window pops up.



weka.attributeSelection.WrapperSubsetEval

WrapperSubsetEval:

Evaluates attribute sets by using a learning scheme.

classifier ZeroR

folds 5

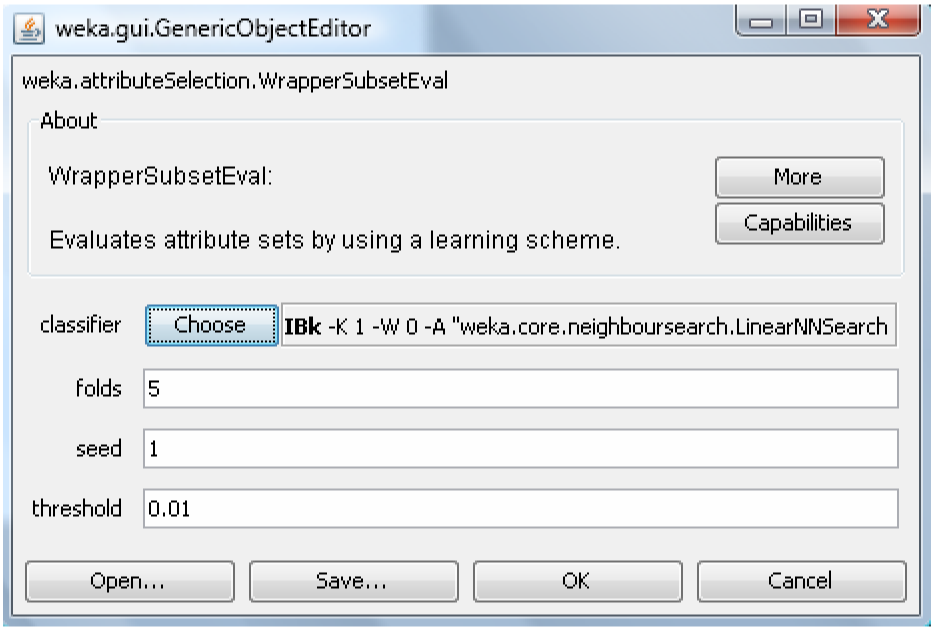
seed 1

threshold 0.01

The default method ZeroR should be changed to kNN.

- Click on the ***Choose*** button near the **classifier** label
- Select from the hierarchical list of machine learning methods ***weka/classifiers/lazy/IBk***.

The settings are shown on the snapshot below.



weka.attributeSelection.WrapperSubsetEval

WrapperSubsetEval:

Evaluates attribute sets by using a learning scheme.

classifier **IBk** -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch

folds 5

seed 1

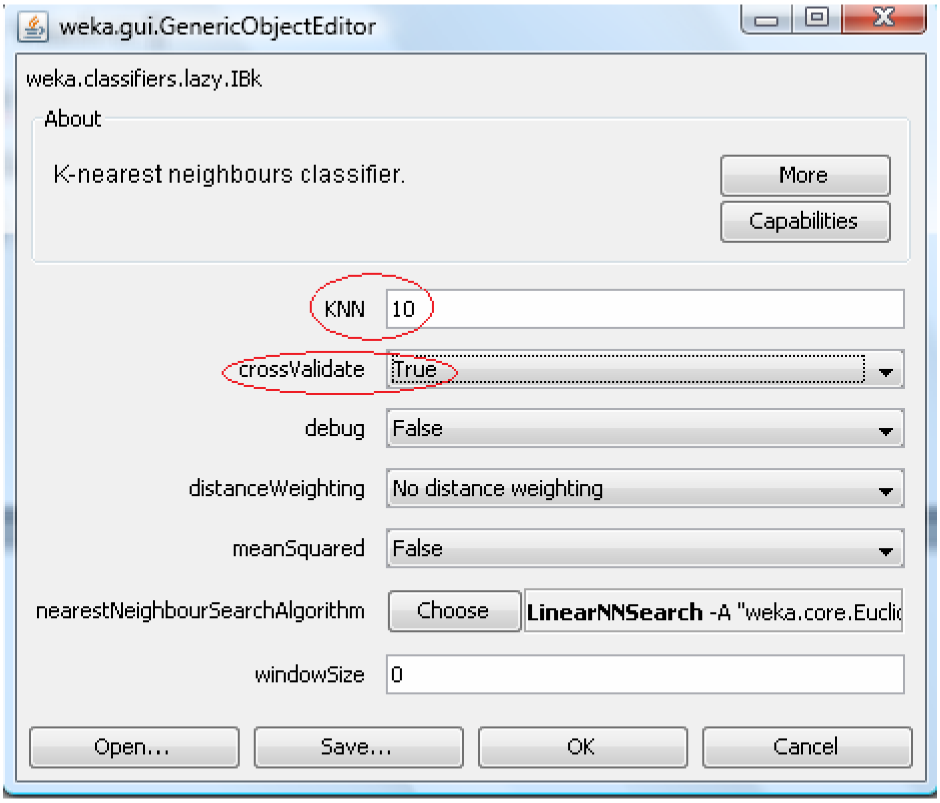
threshold 0.01

- Click on **IBk**

A window with default parameters of the kNN method (fixed value of k = 1) appears on the screen. In order to search an optimal value of k in the range from 1 to 10:

- Change **kNN** to 10
- Select ***True*** for **crossValidate**

The corresponding window is:



weka.classifiers.lazy.IBk

K-nearest neighbours classifier.

KNN 10

crossValidate True

debug False

distanceWeighting No distance weighting

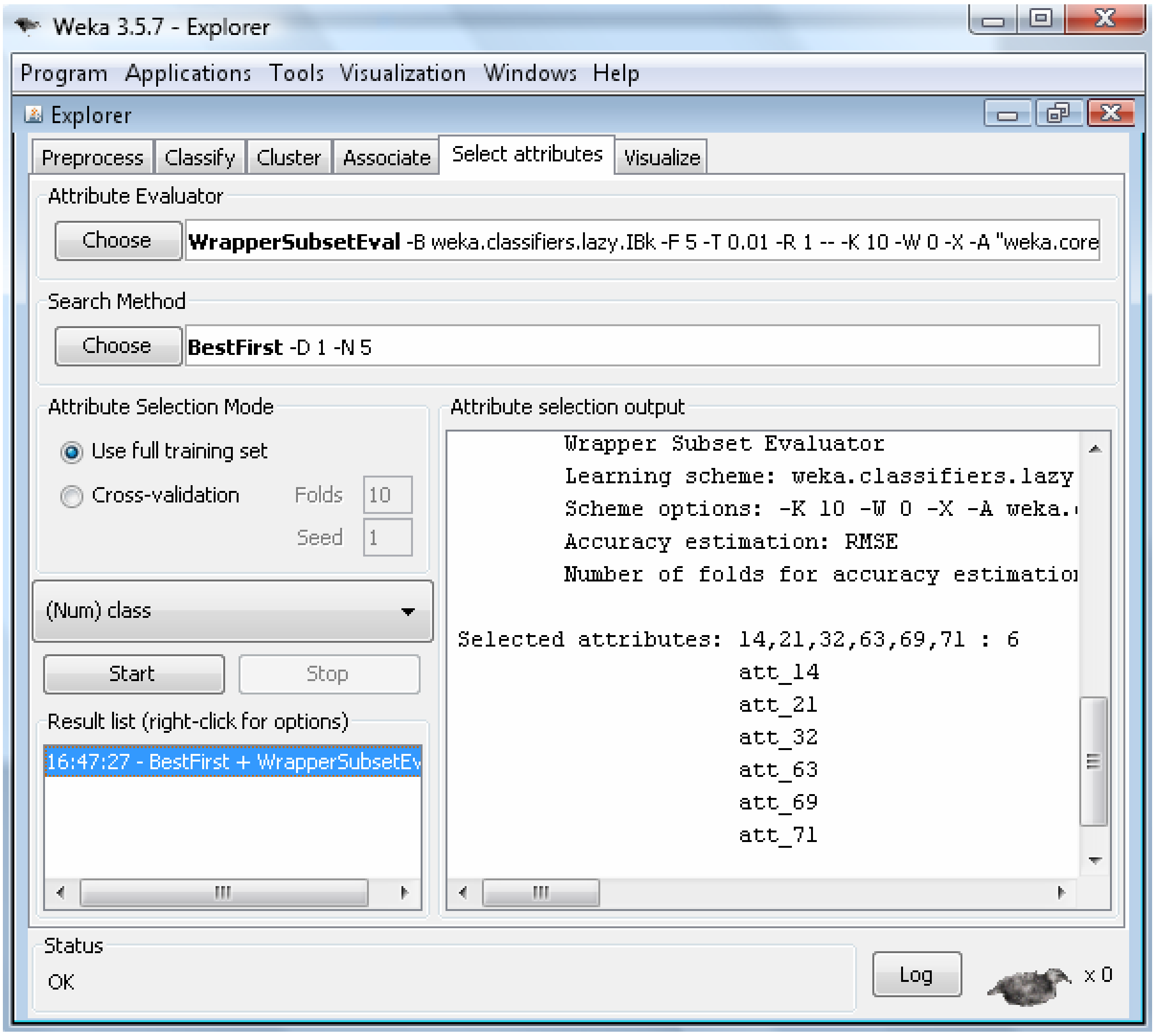
meanSquared False

nearestNeighbourSearchAlgorithm LinearNNSearch -A "weka.core.Euclic

windowSize 0

- Click on the ***OK*** button in the window with *k*-Nearest Neighbors classifier
- Click on the ***OK*** button in the window with WrapperSubsetEval
- In the remaining window click on the ***Start*** button

  Computations may take 1-2 minutes depending on the speed of computer. The resulting window is:



Attribute Evaluator

WrapperSubsetEval -B weka.classifiers.lazy.IBk -F 5 -T 0.01 -R 1 -- -K 10 -W 0 -X -A "weka.core

Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Num) class

Result list (right-click for options)

16:47:27 - BestFirst + WrapperSubsetEv

Attribute selection output

```
          Wrapper Subset Evaluator
          Learning scheme: weka.classifiers.lazy
          Scheme options: -K 10 -W 0 -X -A weka.
          Accuracy estimation: RMSE
          Number of folds for accuracy estimatio

Selected attributes: 14,21,32,63,69,71 : 6
                     att_14
                     att_21
                     att_32
                     att_63
                     att_69
                     att_71
```

Status

OK

One can see that 6 attributes (descriptors) have been selected.

In order to save the selected descriptors,

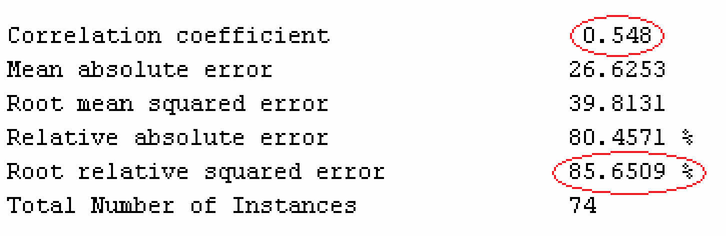
- Click on corresponding line in the *Result list* (left side, on the bottom) with the right mouse button
- From the pop-up menu, select the item ***Save reduced data…***
- Save the dataset with 6 selected descriptors to file *knn_descr.arff*

Selected descriptors are supposed to be used in the k Nearest Neighbors model.

- Switch to the **Preprocess** submode of the **Explorer** mode
- Click on the ***Open file…*** button and open the file *knn_descr.arff*
- Switch to the ***Classify*** submode
- Click on the ***Choose*** button near the **classifier** label
- Select from the hierarchical list of machine learning methods ***weka/classifiers/lazy/IBk***.
- Click on **IBk**

- Change **kNN** to 10
- Select ***True*** for **crossValidate**
- Click on the ***OK*** button in the window with K-nearest neighbors classifier
- Click on the ***Start*** button

  The following results are displayed:



```
Correlation coefficient                  0.548
Mean absolute error                     26.6253
Root mean squared error                 39.8131
Relative absolute error                 80.4571 %
Root relative squared error             85.6509 %
Total Number of Instances               74
```

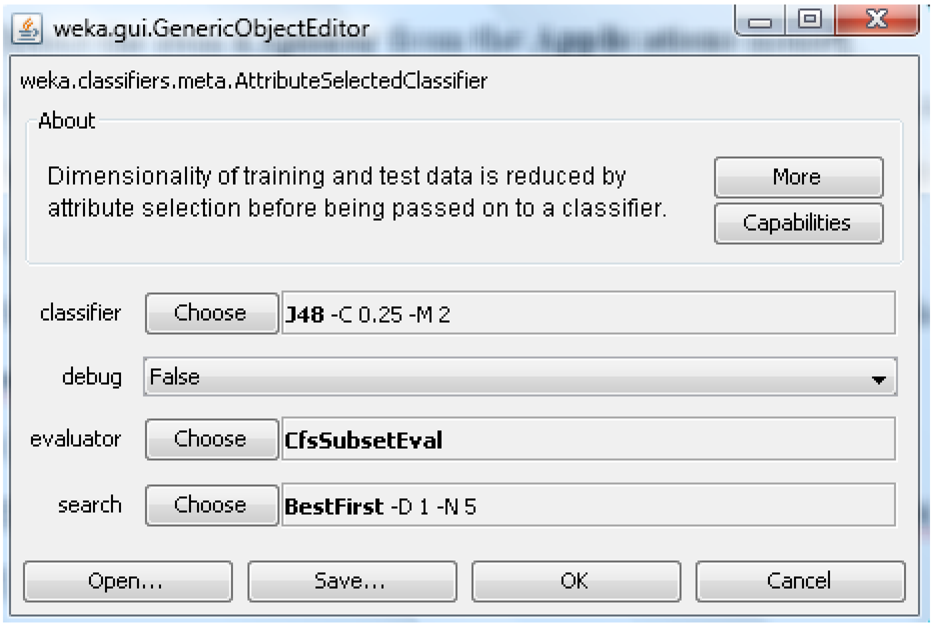
As in previous case, the correlation coefficient significantly deviates from zero. Hence, the descriptor selection bias takes place in the case of internal cross-validation procedures.

4.3. External cross-validation

Here, a separate set of selected descriptors is formed in each fold of the cross-validation procedure.

- Start the Weka program
- Select the item **Explorer** from the **Applications** menu
- Click on the button ***Open file…*** and load the file *alkan-bp-louse100.arff.*
- Switch to the **Classify** submode by clicking on label ***Classify***
- Click on the ***Choose*** button near the **classifier** label
- Select from the hierarchical list of machine learning methods ***weka/classifiers/meta/AttributeSelectedClassifier***.
- Click on the word ***AttributeSelectedClassifier***

This method provides a correct external cross-validation and performs variable section for each cross-validation fold. The following window appears:



weka.gui.GenericObjectEditor

weka.classifiers.meta.AttributeSelectedClassifier

About

Dimensionality of training and test data is reduced by attribute selection before being passed on to a classifier.

classifier: J48 -C 0.25 -M 2

debug: False

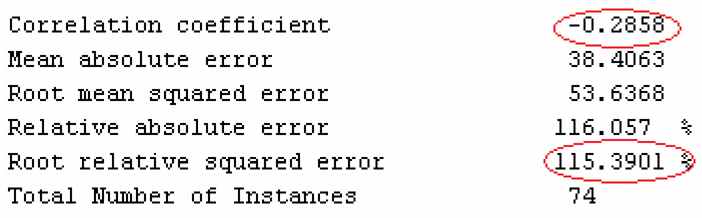
evaluator: CfsSubsetEval

search: BestFirst -D 1 -N 5

Here, the kNN method will be used instead of **J48** (default setting)

- Click on the ***Choose*** button near the **classifier** label
- Select from the hierarchical list of machine learning methods ***weka/classifiers/lazy/IBk***.
- Click on **IBk**
- Change **kNN** to 10
- Select ***True*** for **crossValidate**
- Click on the ***OK*** button in the window with k-Nearest Neighbors classifier
- Click on the ***OK*** button in the window with parameters of ***AttributeSelectedClassifier***
- In the remaining window click on ***Start***

The obtained results are:



```
Correlation coefficient                 -0.2858
Mean absolute error                     38.4063
Root mean squared error                 53.6368
Relative absolute error                116.057  %
Root relative squared error            115.3901 %
Total Number of Instances               74
```

These results show that the models are not predictive at all. This looks reasonable since those models involve random numbers as descriptors.

5. Conclusions

External cross validation should be used to assess predictive performance of QSAR models. Using test set compounds in descriptors selection procedure leads to erroneous or overfitted models.

6. References

1. Hawkins, D. M., The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004,** *44 (1)*, 1-12.

2. Tetko, I. V.; Sushko, I.; Pandey, A. K.; Tropsha, A.; Zhu, H.; Papa, E.; Öberg, T.; Todeschini, R.; Fourches, D.; Varnek, A., Critical assessment of QSAR Models to predict environmental toxicity against Tetrahymena pyriformis: Focusing on applicability domain and overfitting by variable selection. *J. Chem. Inf. Model.* **2008,** *(submitted).*

3. Tetko, I. V.; Solov'ev, V. P.; Antonov, A. V.; Yao, X.; Doucet, J. P.; Fan, B.; Hoonakker, F.; Fourches, D.; Jost, P.; Lachiche, N.; Varnek, A., Benchmarking of linear and nonlinear approaches for quantitative structure-property relationship studies of metal complexation with ionophores. *J. Chem. Inf. Model.* **2006,** *46 (2)*, 808-819.

4. Needham, D. E.; Wei, I. C.; Seybold, P. G., Molecular modeling of the physical properties of alkanes. *J. Am. Chem. Soc.* **1988,** *110 (13)*, 4186-4194.

5. Hall, M. A. Correlation-based Feature Selection for Machine Learning. Ph.D diss, Waikato University, Hamilton, NZ, 1998.