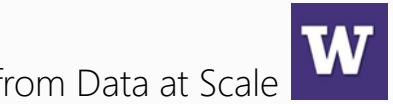
Data Science

Deriving Knowledge from Data at Scale

Nov 23rd, 2015



Updated Teams

student	Team	
Velazquez-Muriel, Javier	T1	home site quote
Moreno Diaz Covarrubias, Marciano Alberto	T1	home site quote
Kramer, Aleksey	T10	wal mart
Bodas, Jagger	T10	wal mart
Featherly-Bean, Winston	T10	wal mart
Quindara, Czarina	T11	key point facial image
Keeton, Jeff	T11	key point facial image
Severaid, Trevor	T12	homesite
Bose, Atanu	T13	recipe
Desai, Devesh	T13	recipe
Fannin, Calvin	T14	wal mart
Godwin, Headieh	T15	digit recognizer
Smith, Daniel	T15	digit recognizer
Liu, Kai	T16	right whale
Guerin, Adrian	T17	recipe
Patterson, Michael	T17	recipe
Byers, Tyler	T18	how much it rain
Kumar, Alok	T2	wal mart
Purayil, Binu	T2	wal mart
Prout, Matthew	T3	wal mart
Han, Sujin	T3	wal mart
Choudhury, Kailash	T4	wal mart
Bane, Ryan	T4	wal mart
Williams, Tamara	T4	wal mart
Macnab, Angus	T5	digit recognizer
Rosenbloom, Anton	T6	identify right whale
Dybdahl, Eric	T6	identify right whale
Srinivasan, Sundar	T6	identify right whale
Churchill, Dean	T7	how much it rain
Thompson, Mike	T8	code
Kaltsukis, Ryan	T9	wal mart
Chandra, Saunak	T9	wal mart

Project Part 1 (Nov16)
Project Part 2 (Nov30)
Project Part 3 (Dec 7)

Lecture 8 Agenda

- Quick note on PCA
- Recap SVM with an example 30
- Attribute Selection from Lecture 7 30
- Data Transformation 75
- Course Evaluation 30

A Practical Guide to Support Vector Classification

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin

Department of Computer Science

National Taiwan University, Taipei 106, Taiwan

http://www.csie.ntu.edu.tw/~cjlin

Initial version: 2003 Last updated: April 15, 2010

Abstract

The support vector machine (SVM) is a popular classification technique. However, beginners who are not familiar with SVM often get unsatisfactory results since they miss some easy but significant steps. In this guide, we propose a simple procedure which usually gives reasonable results.

1 Introduction

SVMs (Support Vector Machines) are a useful technique for data classification. Although SVM is considered easier to use than Neural Networks, users not familiar with it often get unsatisfactory results at first. Here we outline a "cookbook" approach which usually gives reasonable results.

Note that this guide is not for SVM researchers nor do we guarantee you will achieve the highest accuracy. Also, we do not intend to solve challenging or difficult problems. Our purpose is to give SVM novices a recipe for rapidly obtaining acceptable results.

Although users do not need to understand the underlying theory behind SVM, we briefly introduce the basics necessary for explaining our procedure. A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one "target value" (i.e. the class labels) and several "attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

Given a training set of instance-label pairs $(\mathbf{x}_i, y_i), i = 1, \ldots, l$ where $\mathbf{x}_i \in R^n$ and $\mathbf{y} \in \{1, -1\}^l$, the support vector machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) require the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{1}$$

UCI Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 295 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. Our old web site is still available, for those who prefer the old format. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If you wish to donate a data set, please consult our donation policy. For any other questions, feel free to contact the Repository librarians. We have also set up a mirror site for the Repository.

Supported By: In Collaboration With: Rexa.info

Latest News:

2013-04-04: Welcome to the new Repository admins Kevin Bache and Moshe Lichman!
2010-03-01: Note from donor regarding Netflix data
2009-10-16: Two new data sets have been added.
2009-09-14: Several data sets have been added.
2008-07-23: Repository mirror has been set up.
2008-03-24: New data sets have been added!
2007-06-25: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope

Featured Data Set: Balance Scale

Task: Classification
Data Type: Multivariate
 # Attributes: 4
 # Instances: 625

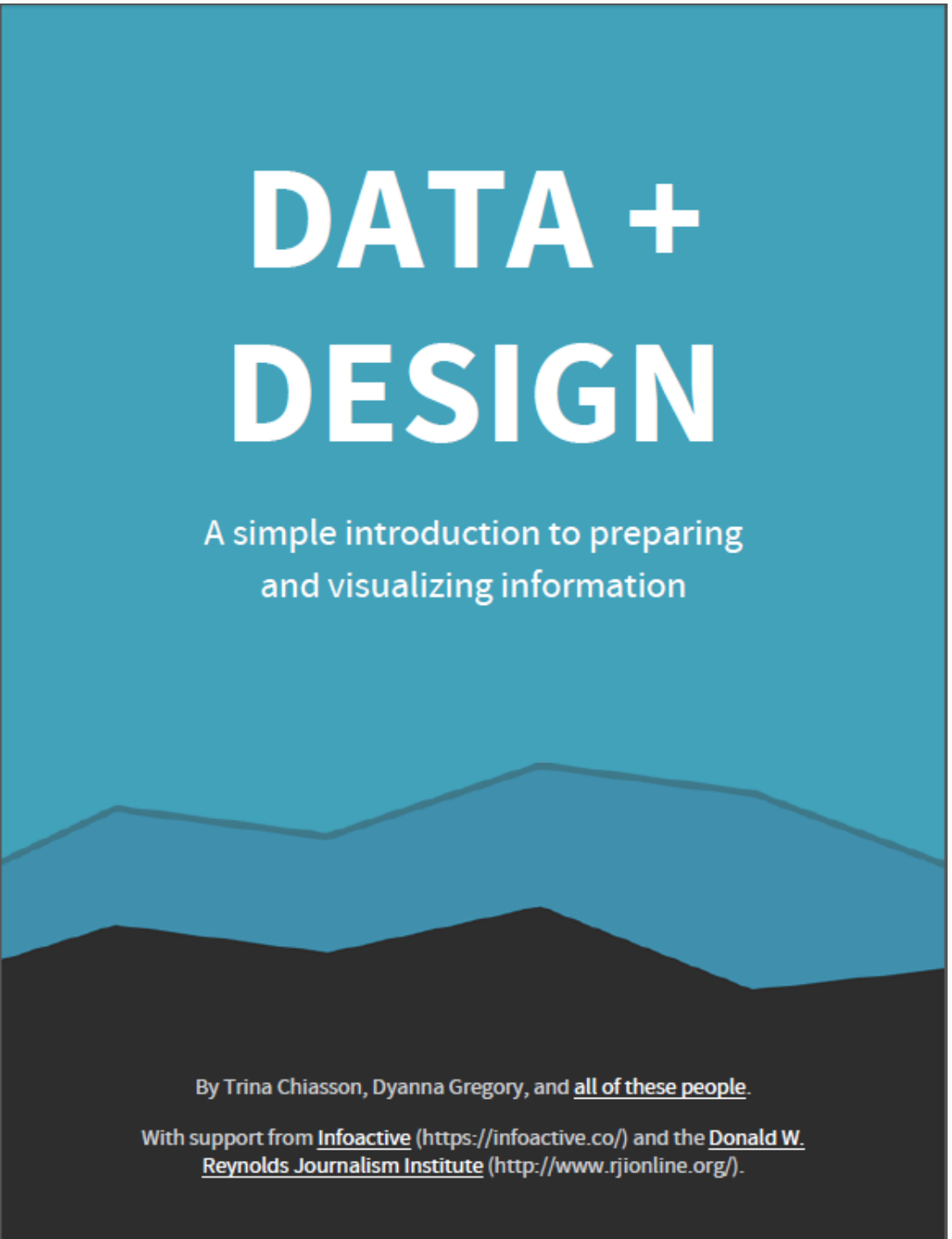
http://archive.ics.uci.edu/ml/index.html

Newest Data Sets:

2014-07-25: REALDISP Activity Recognition Dataset
2014-07-22: Perfume Data
2014-06-18: Gesture Phase Segmentation
2014-06-12: Parkinson Speech Dataset with Multiple Types of Sound Recordings
2014-06-01: Tennis Major Tournament Match Statistics
2014-05-29: BlogFeedback

Most Popular Data Sets (hits since 2007):

594361: Iris
415131: Adult
354842: Wine
289367: Breast Cancer Wisconsin (Diagnostic)
287988: Car Evaluation
232020: Abalone

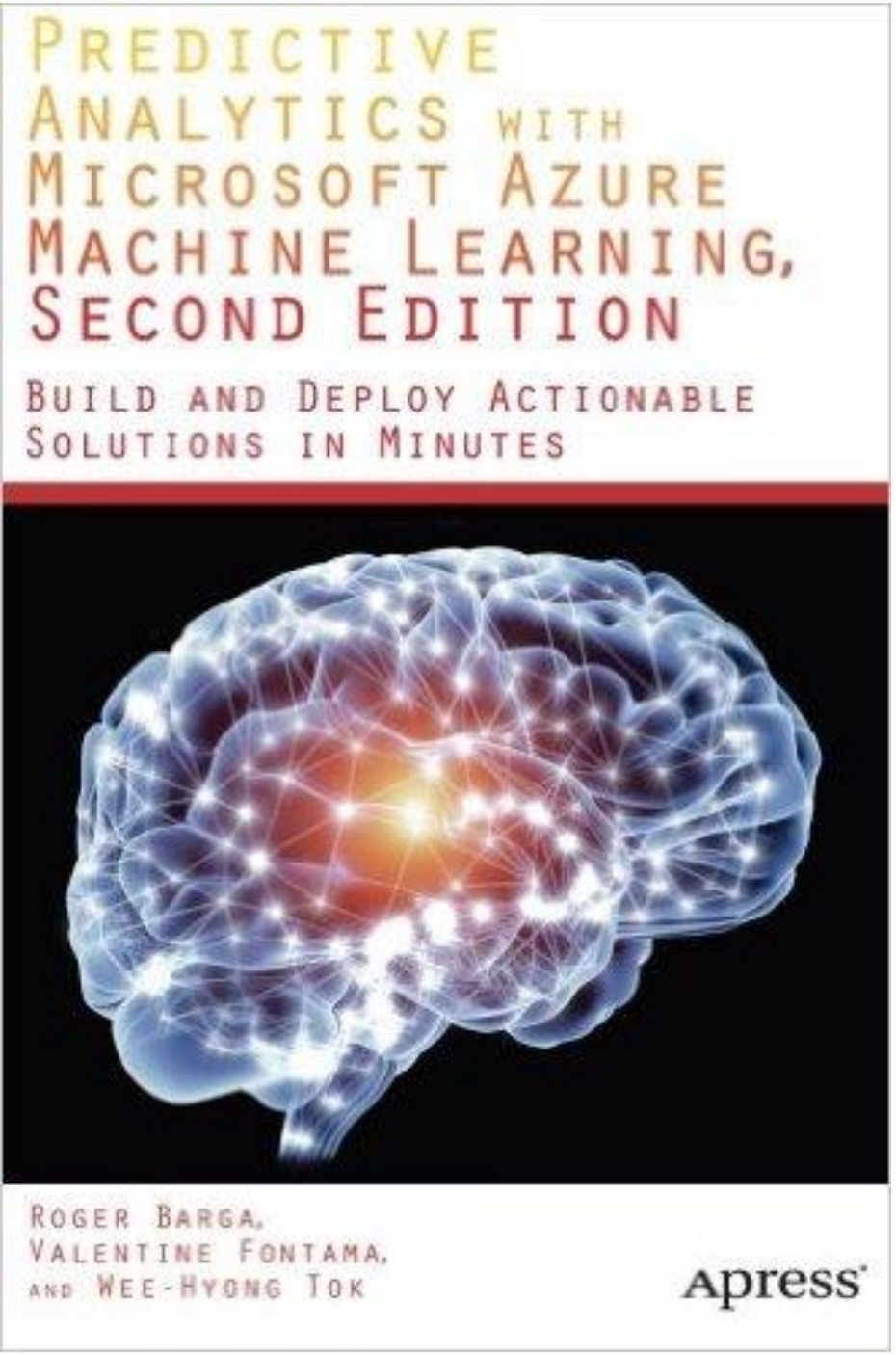


DATA + DESIGN

A simple introduction to preparing and visualizing information.

Over 50 people worked hard over the course of many months to create this book, which we're delighted to offer as a free resource. If you do use or build upon this work, make sure to give credit to the contributors who made it happen by including a link to infoactive.co/data-design (https://infoactive.co/data-design).

The source code can be found and forked on Github (https://github.com/infoactive/data-design/).



- Just out: Second edition of the Azure ML book.

SVM example

Support Vector Machines *
The Interface to `libsvm` in package `e1071`

by David Meyer
FH Technikum Wien, Austria
David.Meyer@R-Project.org

August 5, 2015

"Hype or Hallelujah?" is the provocative title used by Bennett & Campbell (2000) in an overview of Support Vector Machines (SVM). SVMs are currently a hot topic in the machine learning community, creating a similar enthusiasm at the moment as Artificial Neural Networks used to do before. Far from being a panacea, SVMs yet represent a powerful technique for general (nonlinear) classification, regression and outlier detection with an intuitive model representation.

The package `e1071` offers an interface to the award-winning[1] C++-implementation by Chih-Chung Chang and Chih-Jen Lin, `libsvm` (current version: 2.6), featuring:

- C- and ν-classification
- one-class-classification (novelty detection)
- ϵ- and ν-regression

and includes:

- linear, polynomial, radial basis function, and sigmoidal kernels
- formula interface
- k-fold cross validation

For further implementation details on `libsvm`, see Chang & Lin (2001).

Basic concept

SVMs were developed by Cortes & Vapnik (1995) for binary classification. Their approach may be roughly sketched as follows:

Class separation: basically, we are looking for the optimal separating hyperplane between the two classes by maximizing the *margin* between the classes' closest points (see Figure 1)—the points lying on the boundaries are called *support vectors*, and the middle of the margin is our optimal separating hyperplane;

*A smaller version of this article appeared in R-News, Vol.1/3, 9.2001

[1]The library won the IJCNN 2001 Challenge by solving two of three problems: the Generalization Ability Challenge (GAC) and the Text Decoding Challenge (TDC). For more information, see: http://www.csie.ntu.edu.tw/~cjlin/papers/ijcnn.ps.gz.

A Practical Guide to Support Vector Classification

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin

Department of Computer Science
National Taiwan University, Taipei 106, Taiwan
http://www.csie.ntu.edu.tw/~cjlin

Initial version: 2003 Last updated: April 15, 2010

Abstract

The support vector machine (SVM) is a popular classification technique. However, beginners who are not familiar with SVM often get unsatisfactory results since they miss some easy but significant steps. In this guide, we propose a simple procedure which usually gives reasonable results.

1 Introduction

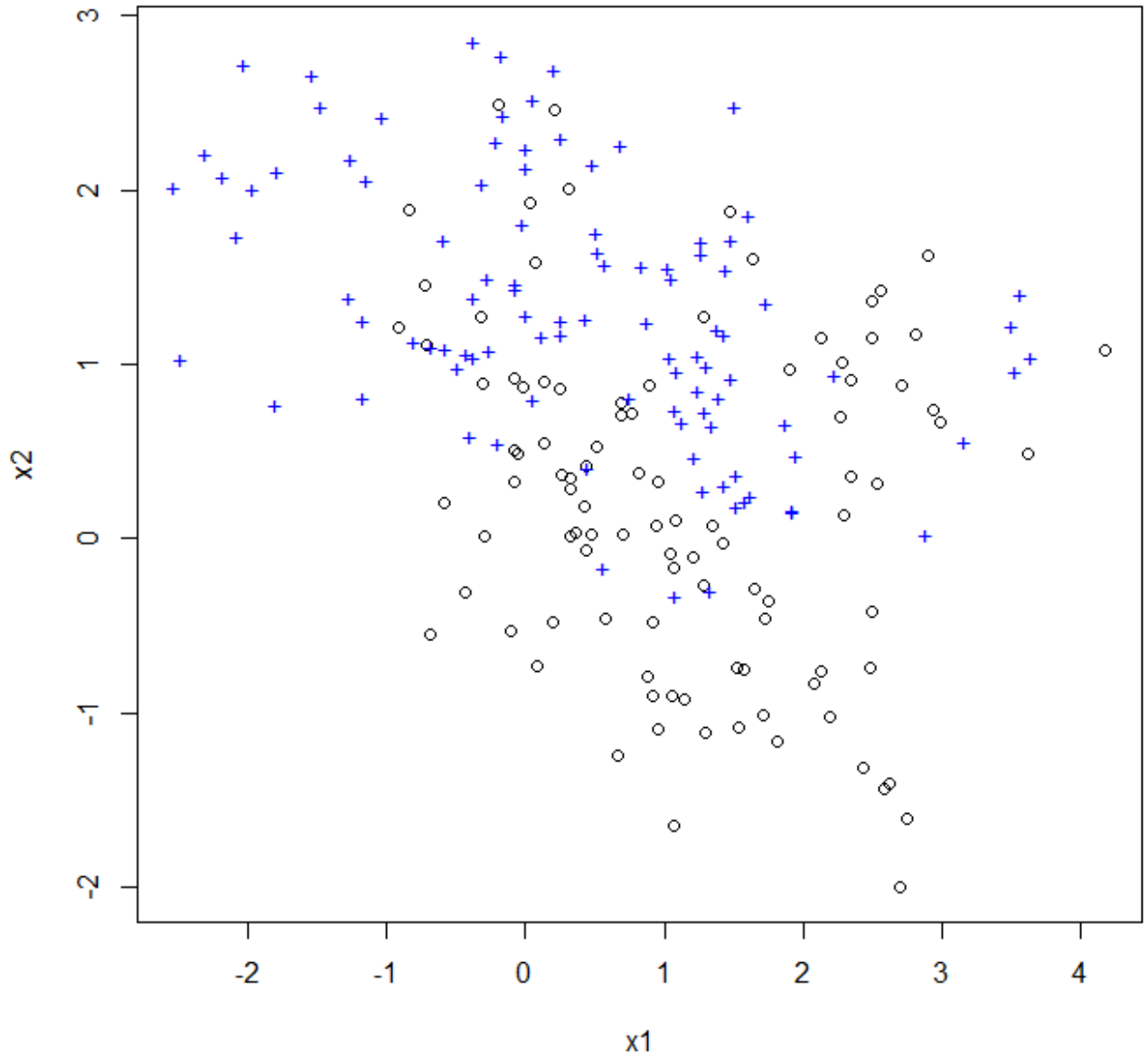
SVMs (Support Vector Machines) are a useful technique for data classification. Although SVM is considered easier to use than Neural Networks, users not familiar with it often get unsatisfactory results at first. Here we outline a "cookbook" approach which usually gives reasonable results.

Note that this guide is not for SVM researchers nor do we guarantee you will achieve the highest accuracy. Also, we do not intend to solve challenging or difficult problems. Our purpose is to give SVM novices a recipe for rapidly obtaining acceptable results.

Although users do not need to understand the underlying theory behind SVM, we briefly introduce the basics necessary for explaining our procedure. A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one "target value" (i.e. the class labels) and several "attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

Given a training set of instance-label pairs $(\mathbf{x}_i, y_i), i = 1, \ldots, l$ where $\mathbf{x}_i \in R^n$ and $\mathbf{y} \in \{1, -1\}^l$, the support vector machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) require the solution of the following optimization problem:

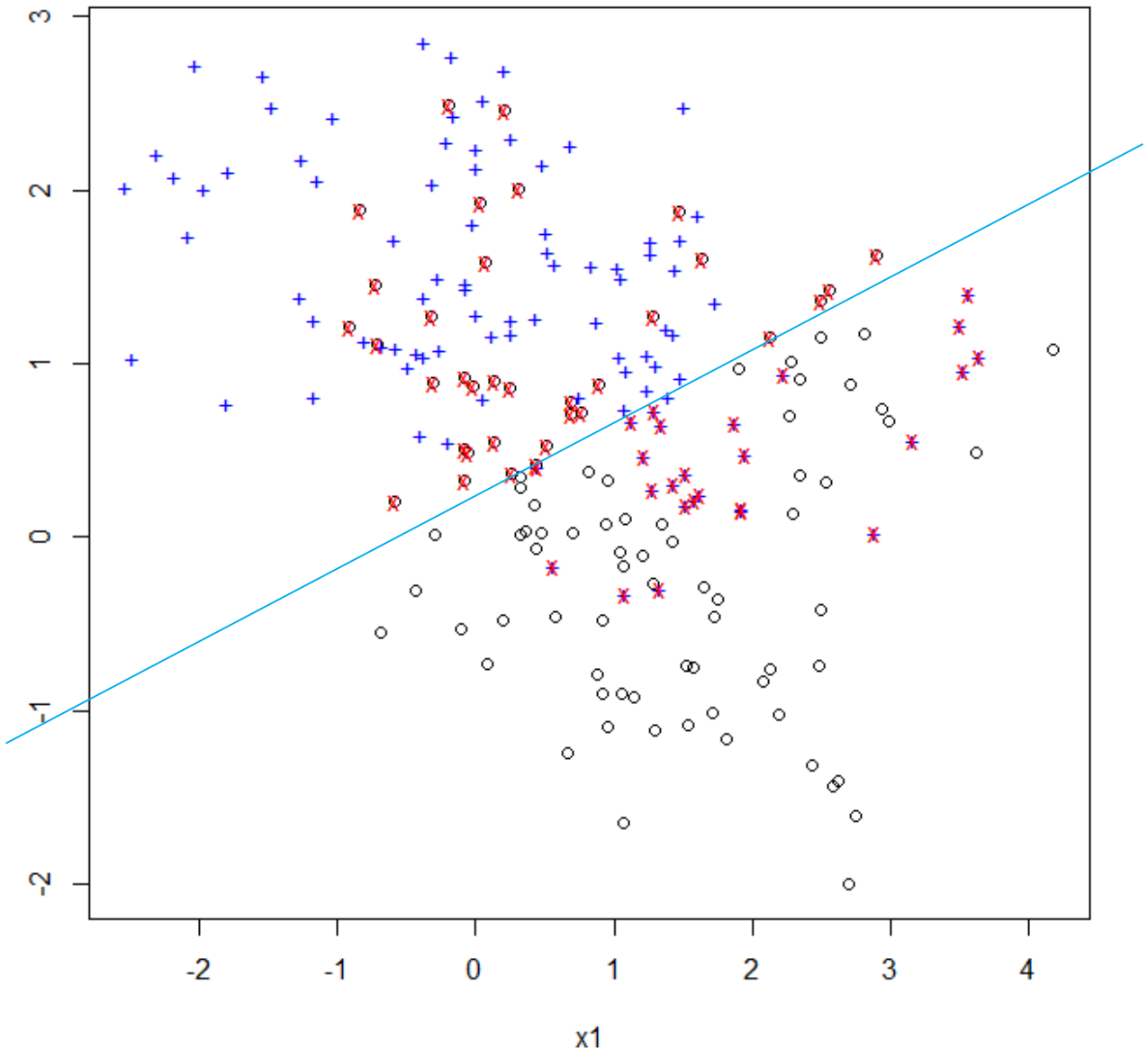
$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{1}$$



Attributes: x1, x2
Response: y = binary (2 classes)

SVM is a powerful tool to identify a boundary in separating these 2 classes.

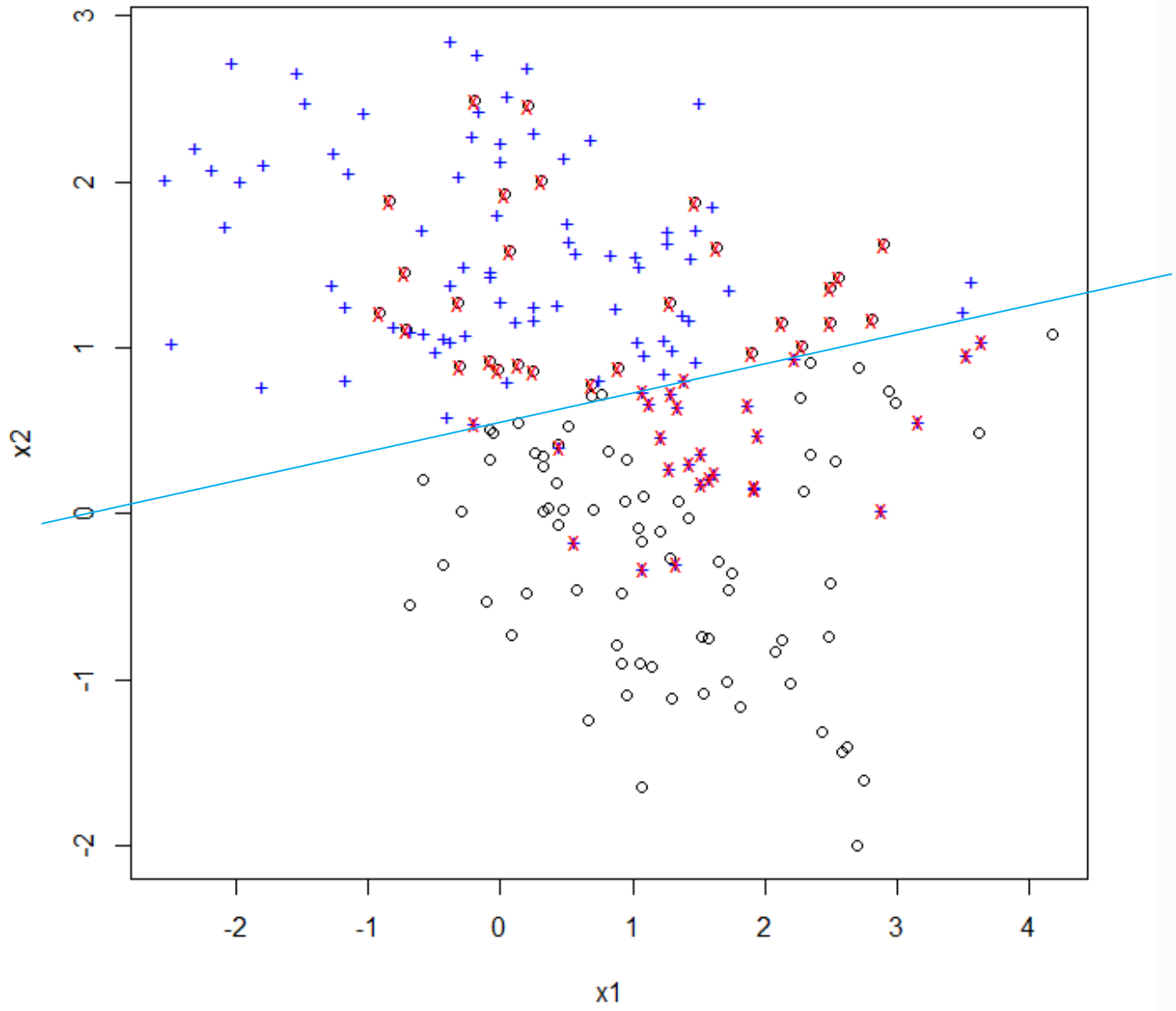
Ref: Hastie, Tibshirani and Friedman
Ref: library(e1071) in R
Ref: David.Meyer@R-Project.org



Linear boundary
C=1e-4

Confusion:

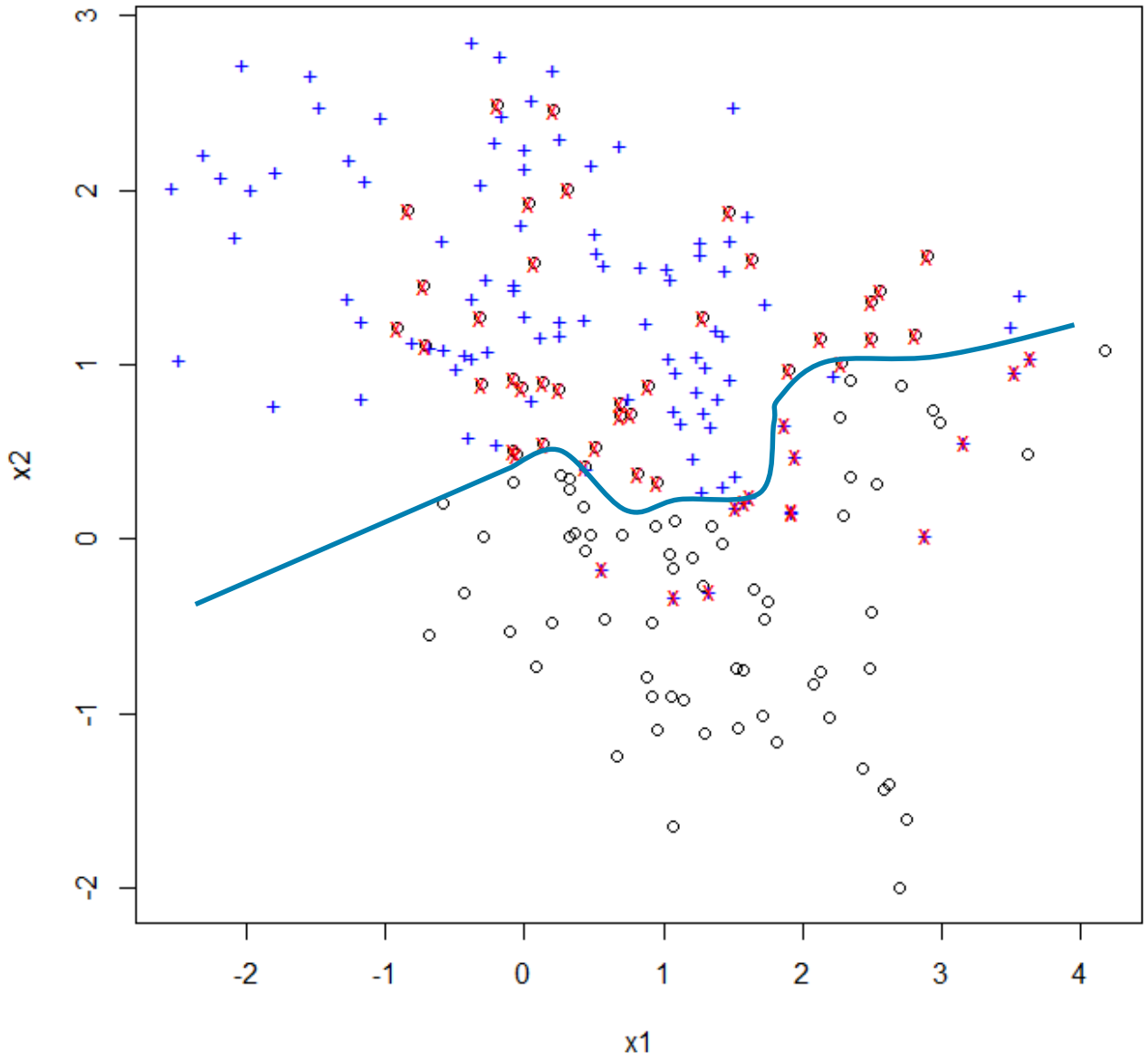
	predict 0	1
Original 0	66	34
Original 1	25	75



Linear boundary
C=1

Confusion matrix:

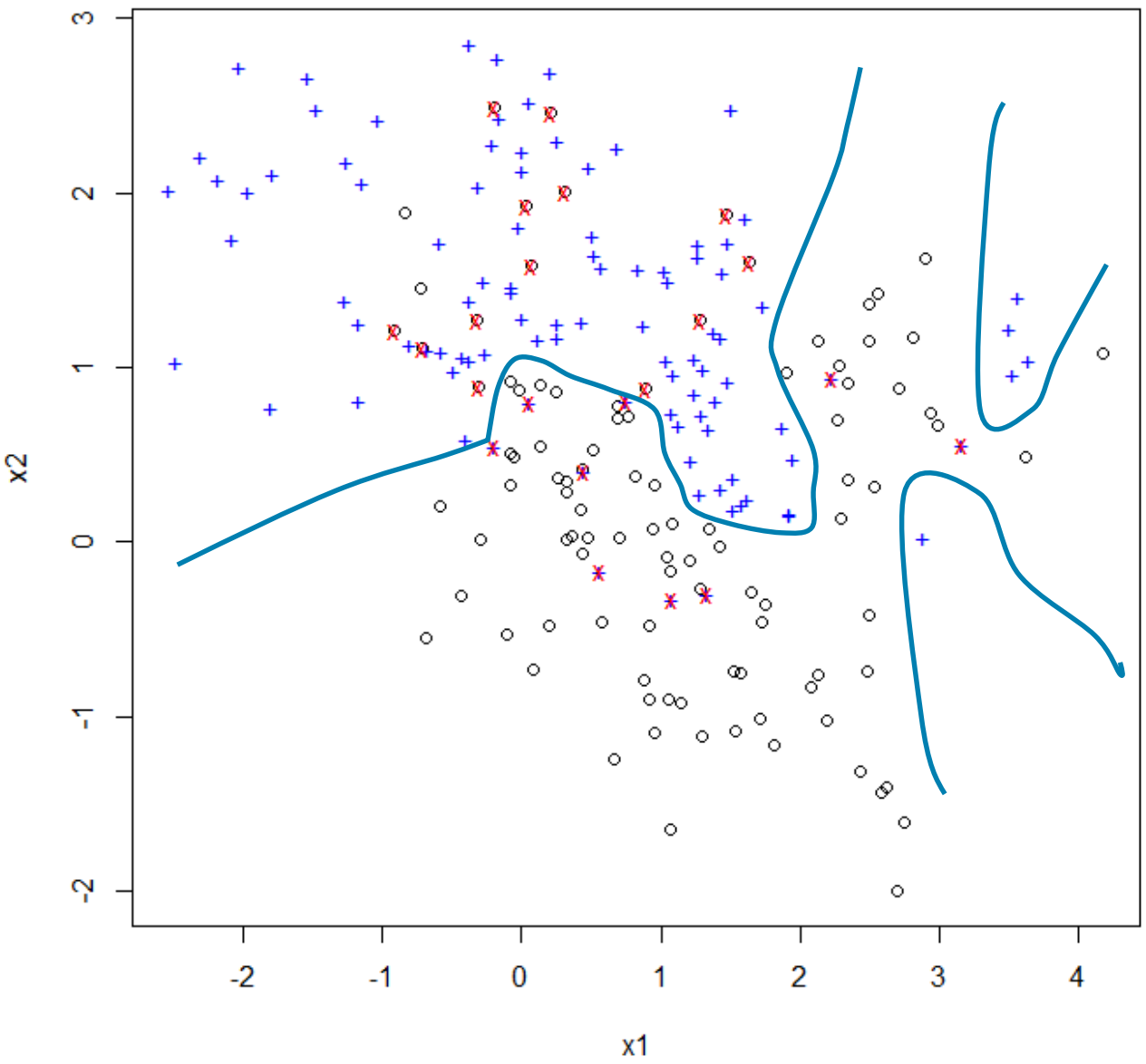
	predict 0	predict 1
Original 0	72	28
Original 1	26	74



Polynomial boundary
C=1, degree=3, gamma=1

Confusion matrix:

	predict	
	0	1
Original 0	63	37
Original 1	14	84

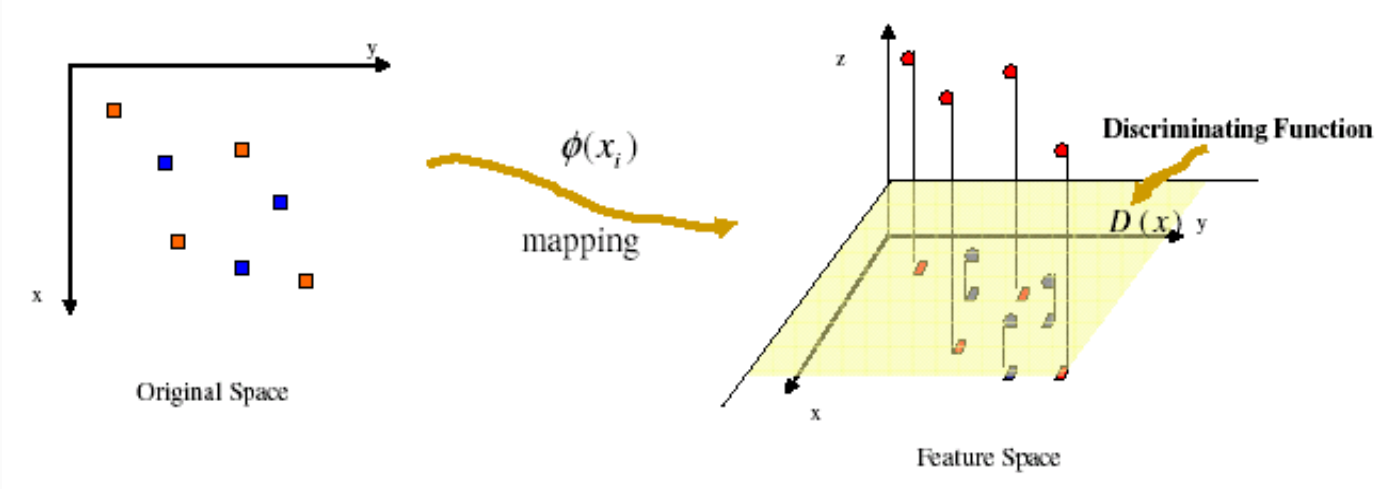


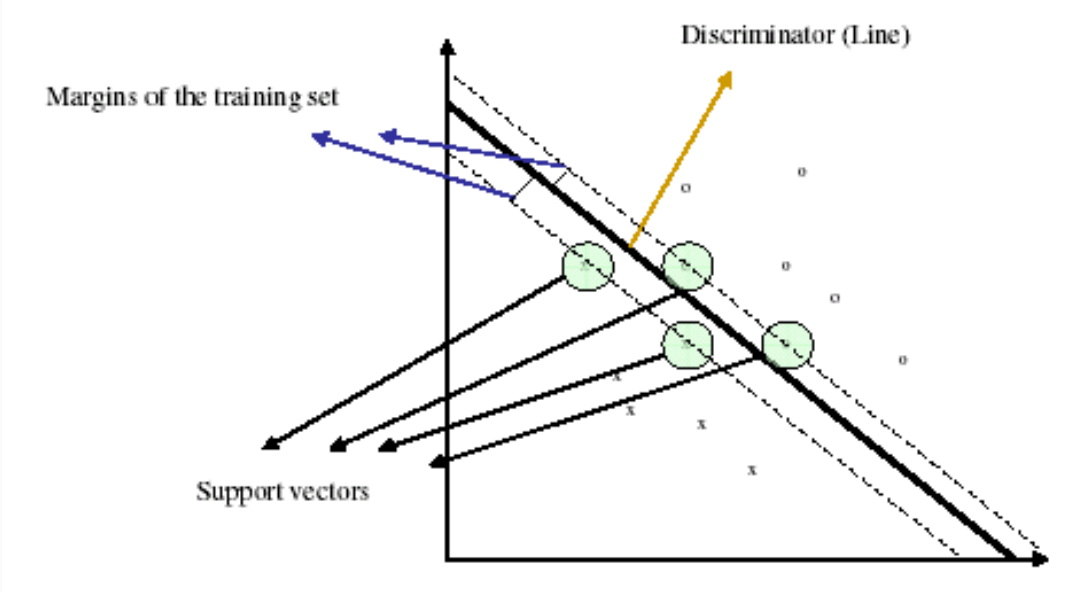
Radial basis function boundary
Gamma=10

Confusion matrix:

	predict 0	predict 1
Original 0	87	13
Original 1	09	91

Support Vector Machines





Goal: to find discriminator
That maximize the margins

Practical tips/workflow for using an SVM

- Transform data to the format of an SVM package
  - {red, green, blue} → {0,0,1} {0,1,0} {1,0,0}
  - Efficient if the number of values is not large
  - If the range is large use single number encoding

Practical tips/workflow for using an SVM

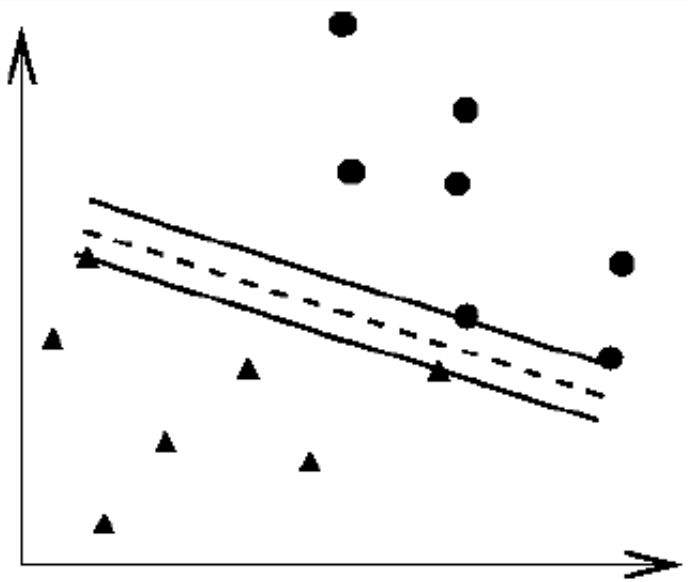
- Transform data to the format of an SVM package
- Conduct simple scaling on the data
  - -1…1 or 0…1 to avoid overfitting

Practical tips/workflow for using an SVM

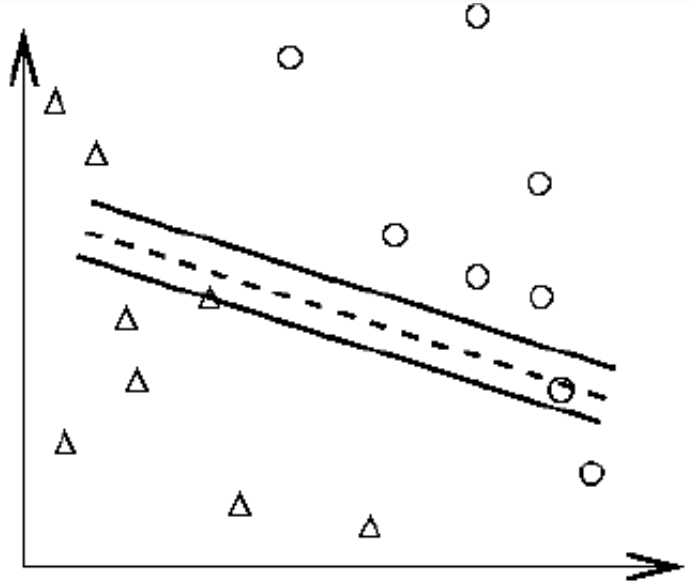
- Transform data to the format of an SVM package
- Conduct simple scaling on the data
- Try the linear kernel
- Consider the RBF kernel
  - The RBF kernel has only two hyperparameters, C & Gamma

Practical tips/workflow for using an SVM

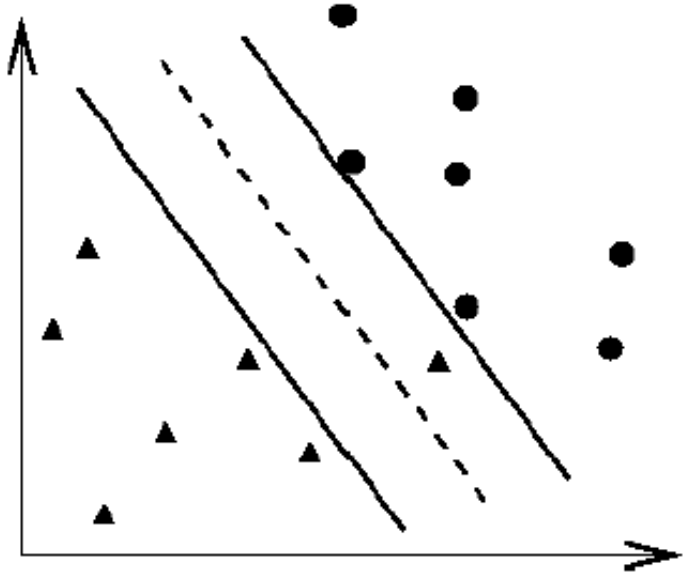
- Transform data to the format of an SVM package
- Conduct simple scaling on the data
- Try the linear kernel
- Consider the RBF kernel
- Grid search to find the best parameter C & Gamma
  - C = 2^{-5}, 2^{-3} , 2^{-1}, … , 2^{15}
  - Gamma = 2^{-15}, 2^{-13}, … , 2^{3}
  - Test on validation data



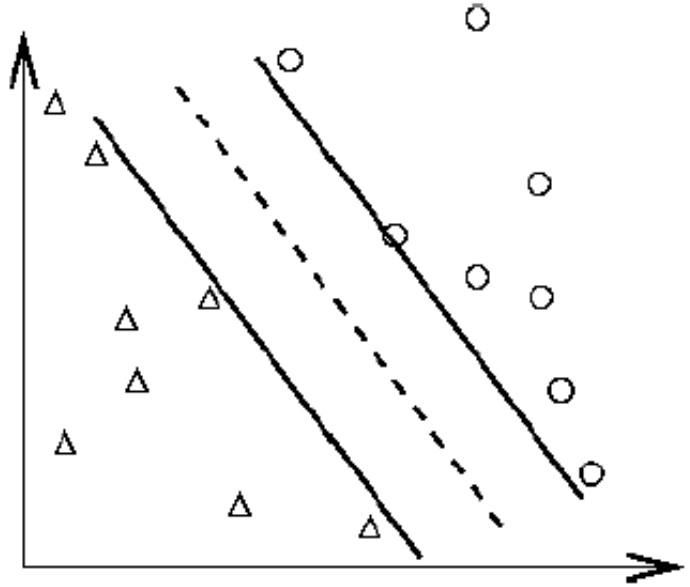
(a) Training data and an overfitting classifier



(b) Applying an overfitting classifier on testing data



(c) Training data and a better classifier



(d) Applying a better classifier on testing data

Optimizing Parameters

SMO and it's complexity parameter ("-C")

- load your dataset in the Explorer
- choose weka.classifiers.meta.CVParameterSelection as classifier
- select weka.classifiers.functions.SMO as base classifier within CVParameterSelection and modify its setup if necessary, e.g., RBF kernel
- open the ArrayEditor for *CVParameters* and enter the following string (and click on *Add*):

  C 2 8 4

This will test the complexity parameters 2, 4, 6 and 8 (= 4 steps)

- close dialogs and start the classifier
- you will get output similar to this one, with the best parameters found in bold:

```
Cross-validated Parameter selection.
Classifier: weka.classifiers.functions.SMO
Cross-validation Parameter: '-C' ranged from 2.0 to 8.0 with 4.0 steps
Classifier Options: **-C 8** -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.RBFKernel -C
```

Optimizing Parameters

LibSVM and the gamma parameter of the RBF kernel ("-G")

- load your dataset in the Explorer
  - choose weka.classifiers.meta.CVParameterSelection as classifier
  - select weka.classifiers.functions.LibSVM as base classifier within CVParameterSelection and modify its setup if necessary, e.g., RBF kernel
  - open the ArrayEditor for *CVParameters* and enter the following string (and click on *Add*):

    G 0.01 0.1 10

This will iterate over the gamma parameter, using values from 0.01 to 0.1 (= 10 steps)

- close dialogs and start the classifier
- you will get output similar to this one, with the best parameters found in bold:

```
Cross-validated Parameter selection.
Classifier: weka.classifiers.functions.LibSVM
Cross-validation Parameter: '-G' ranged from 0.01 to 0.1 with 10.0 steps
Classifier Options: **-G 0.09** -S 0 -K 2 -D 3 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1
```

Optimizing Parameters

GridSearch

weka.classifiers.meta.GridSearch is a meta-classifier for exploring 2 parameters, hence the *grid* in the name. Instead of just using a classifier, one can specify a base classifier **and** a filter, which both of them can be optimized (one parameter each).

For each of the two axes, X and Y, one can specify the following parameters:

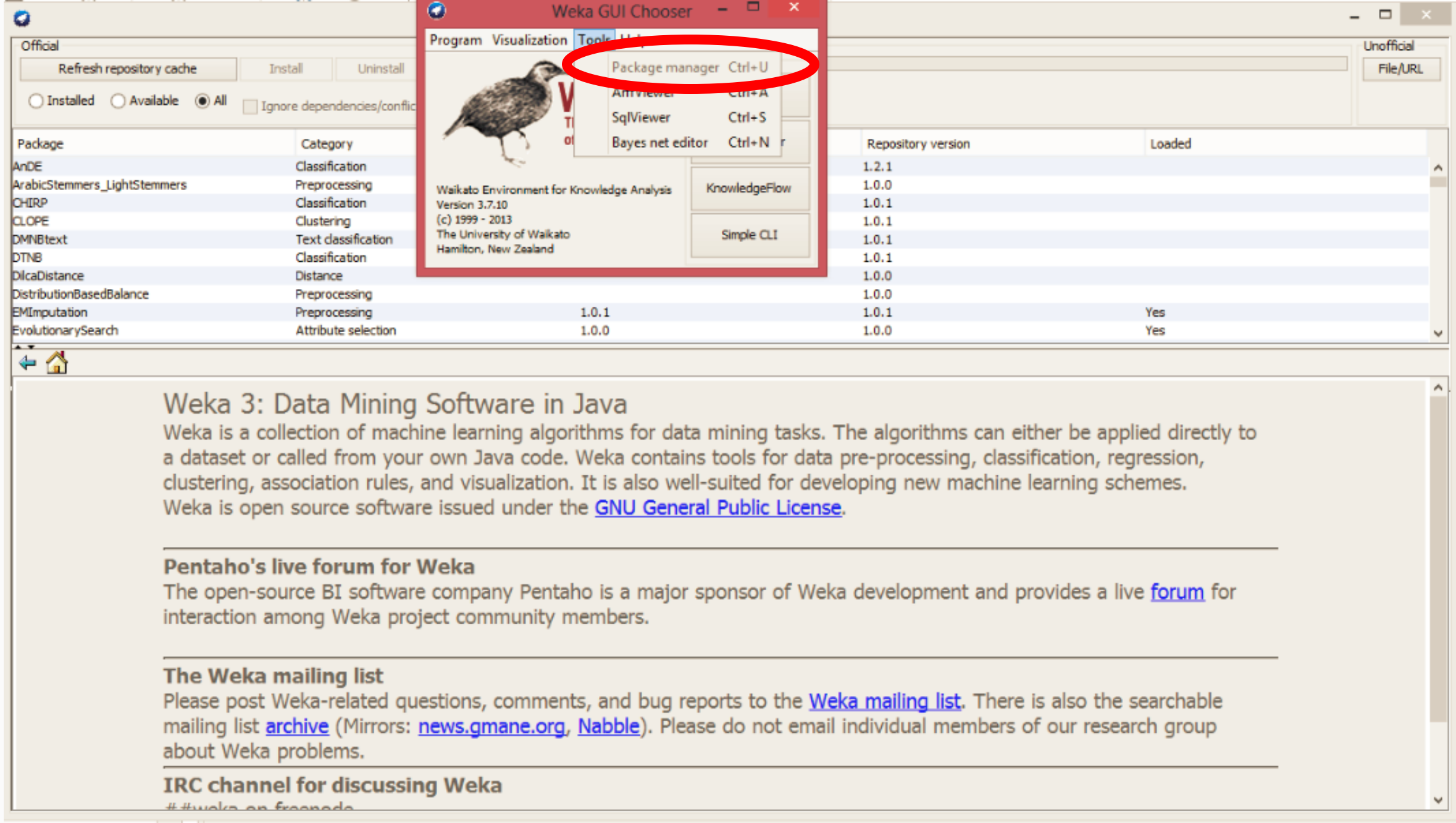
- min, the minimum value to start from.
- max, the maximum value.
- step, the step size used to get from min to max.

GridSearch can also optimized based on the following measures:

- Correlation coefficient (= CC)
- Root mean squared error (= RMSE)
- Root relative squared error (= RRSE)
- Mean absolute error (= MAE)
- Root absolute error (= RAE)
- Combined: (1-abs(CC)) + RRSE + RAE
- Accuracy (= ACC)

At the coal face, the unglamorous side of data science…


Weka GUI Chooser
Program Visualization Tools Help
WEKA
The University of Waikato
Waikato Environment for Knowledge Analysis
Version 3.7.10
(c) 1999 - 2013
The University of Waikato
Hamilton, New Zealand
Applications
Explorer
Experimenter
KnowledgeFlow
Simple CLI


Weka GUI Chooser
Program Visualization Tools
Package manager Ctrl+U
SqlViewer Ctrl+S
Bayes net editor Ctrl+N
Waikato Environment for Knowledge Analysis
Version 3.7.10
(c) 1999 - 2013
The University of Waikato
Hamilton, New Zealand
KnowledgeFlow
Simple CLI
Official
Refresh repository cache
Install
Uninstall
Installed
Available
All
Ignore dependencies/conflic
Unofficial
File/URL
Package
Category
Repository version
Loaded
AnDE Classification 1.2.1
ArabicStemmers_LightStemmers Preprocessing 1.0.0
CHIRP Classification 1.0.1
CLOPE Clustering 1.0.1
DMNBtext Text classification 1.0.1
DTNB Classification 1.0.1
DilcaDistance Distance 1.0.0
DistributionBasedBalance Preprocessing 1.0.0
EMImputation Preprocessing 1.0.1 1.0.1 Yes
EvolutionarySearch Attribute selection 1.0.0 1.0.0 Yes
Weka 3: Data Mining Software in Java
Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.
Weka is open source software issued under the GNU General Public License.
Pentaho's live forum for Weka
The open-source BI software company Pentaho is a major sponsor of Weka development and provides a live forum for interaction among Weka project community members.
The Weka mailing list
Please post Weka-related questions, comments, and bug reports to the Weka mailing list. There is also the searchable mailing list archive (Mirrors: news.gmane.org, Nabble). Please do not email individual members of our research group about Weka problems.
IRC channel for discussing Weka

Packages to install…

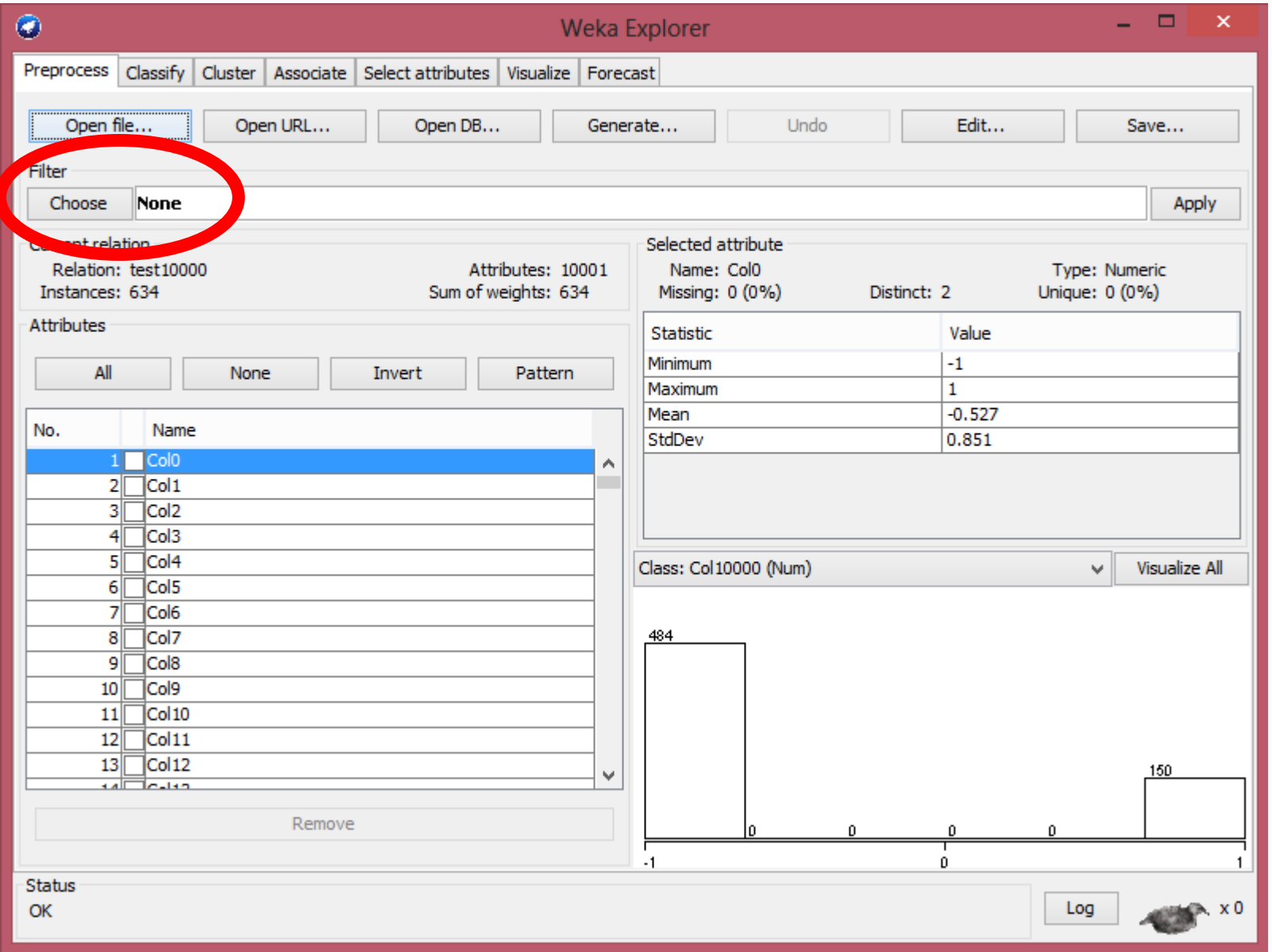
Package	Category
Evolutionary Search	Attribute Selection
attributeSelectionSearchMethods	Attribute Selection
chiSquaredAttributeEval	Attribute Selection
classifierBasedAttributeSelection	Attribute Selection
filteredForwardSelection	Attribute Selection
linearForwardSelection	Attribute Selection
J48graft	Classification
LibLinear	Classification
LivSVM	Classification
multilayerPerceptrons	Classification
WekaExcel	Converter

Packages to install…

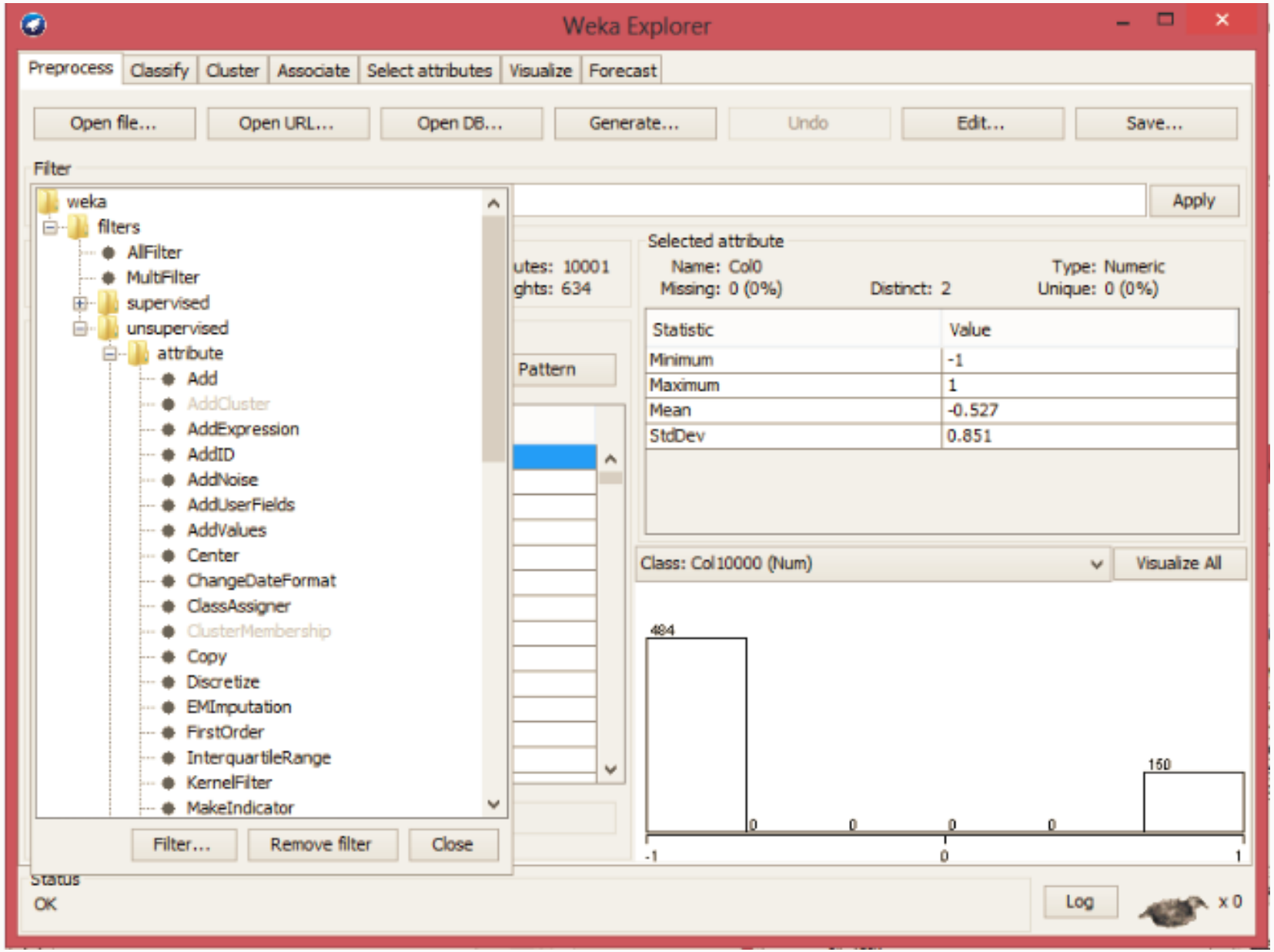
Package	Category
ensembleLibrary	Ensemble Learning
ensembleOfNestedDichotomies	Ensemble Learning
Grading	Ensemble Learning
multiBoostAB	Ensemble Learning
Stacking	Ensemble Learning
EMImputation	Preprocessing
Normalize	Preprocessing
timeseriesForecasting	Time Series

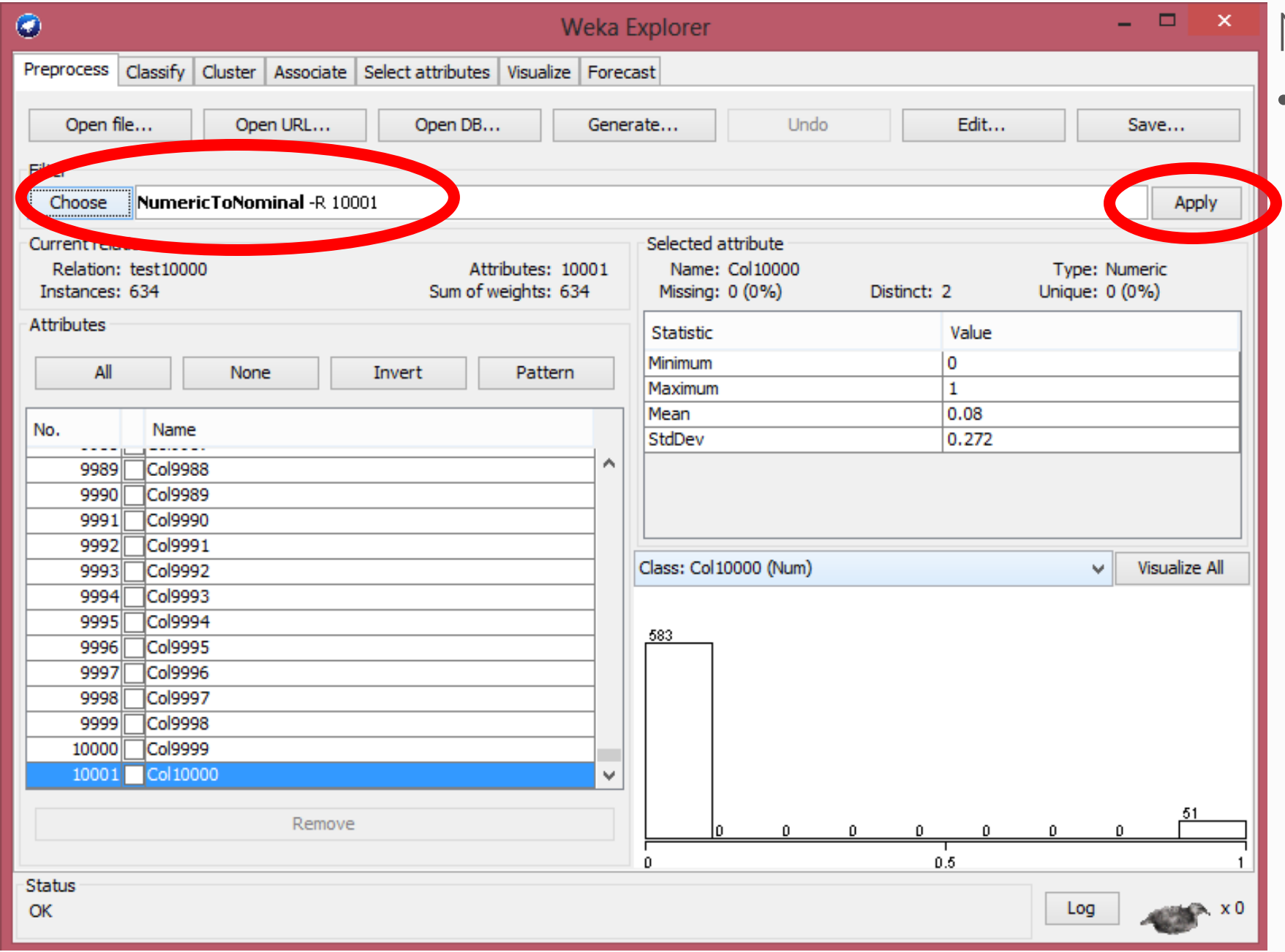
Filtering

Load the data file Thrombin_test_10000CSV into WEKA



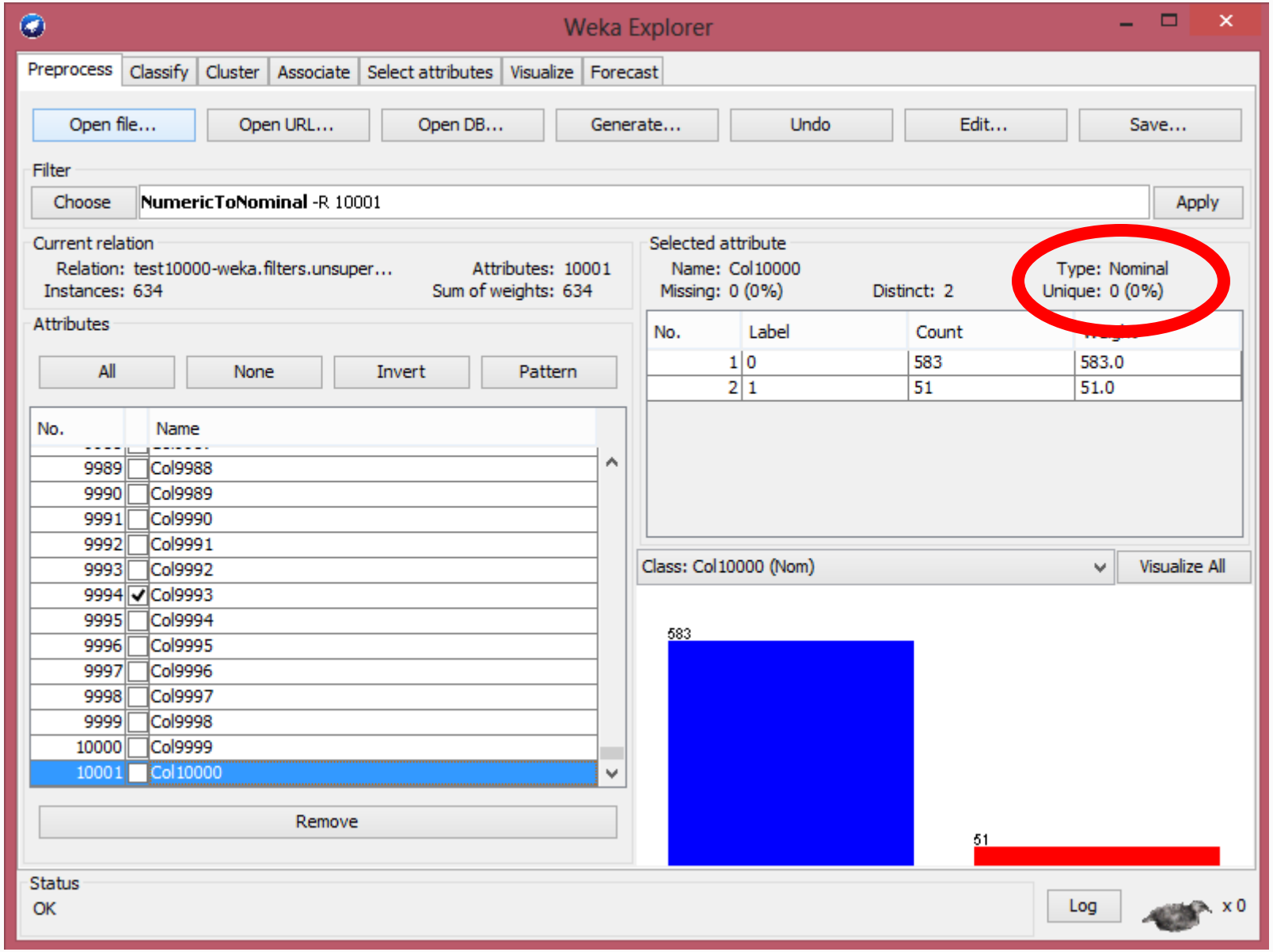
filters → unsupervised → attribute

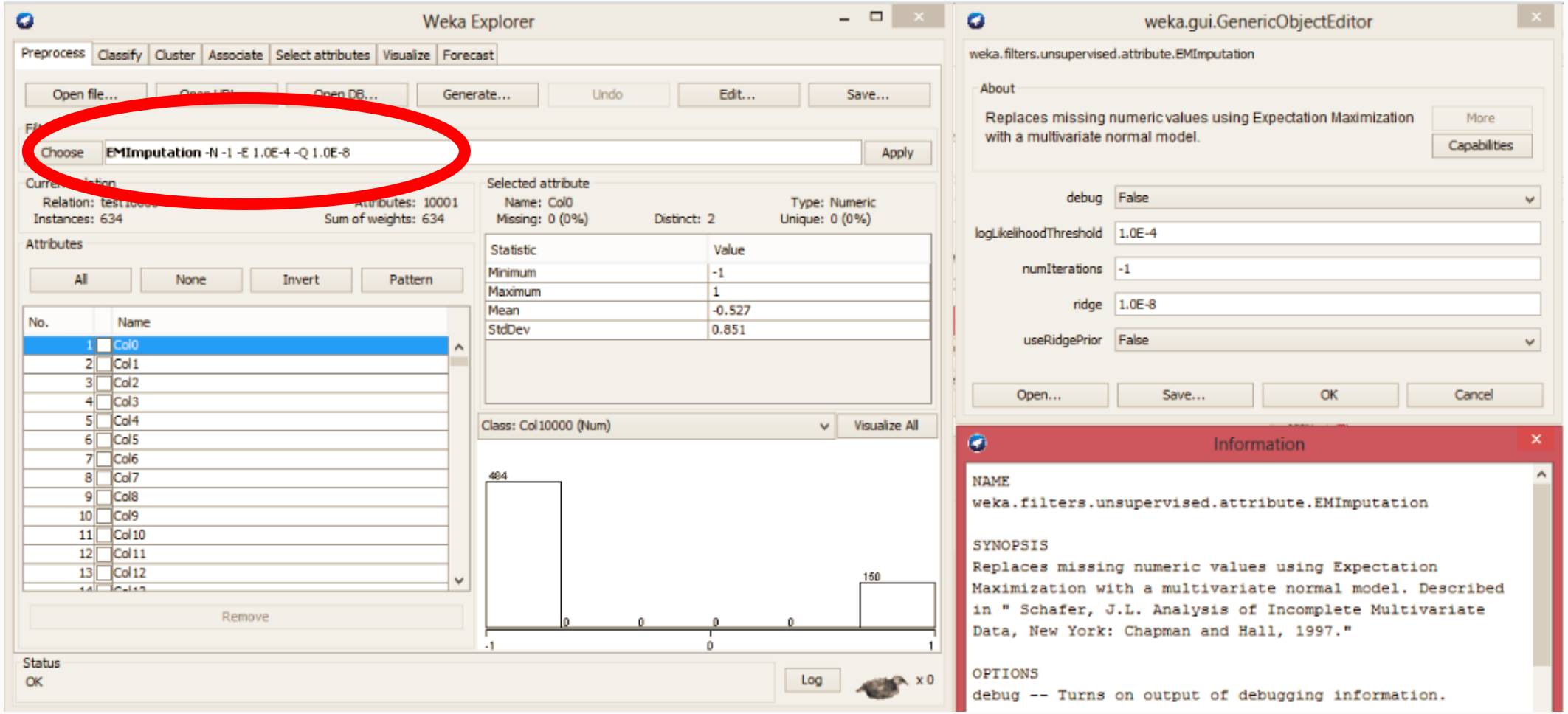




NumericToNominal

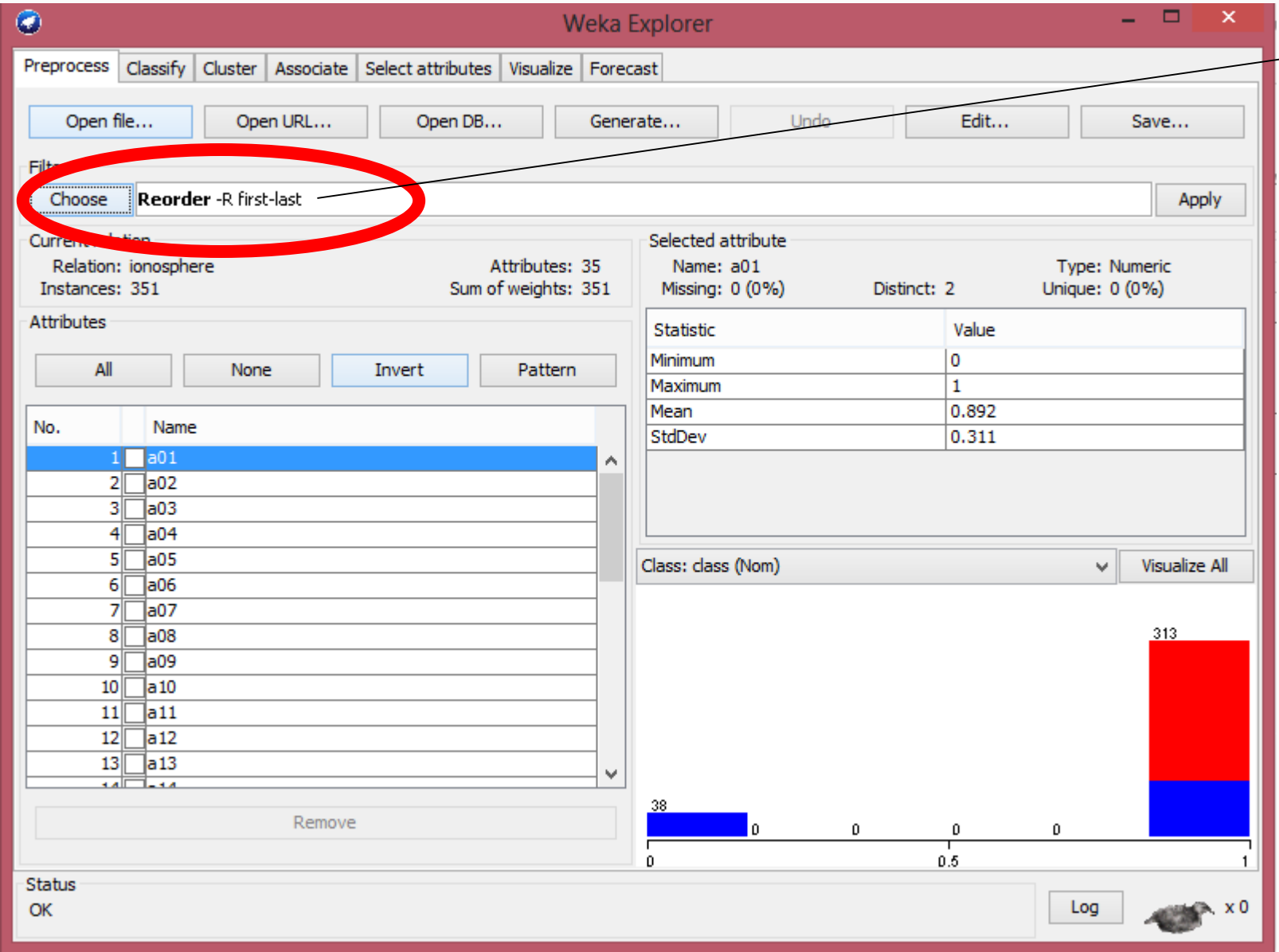
- Column 10001


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Forecast
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
NumericToNominal -R 10001
Apply
Current relation
Relation: test10000-weka.filters.unsuper...
Attributes: 10001
Instances: 634
Sum of weights: 634
Selected attribute
Name: Col10000
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)
No. Label Count
1 0 583 583.0
2 1 51 51.0
Attributes
All
None
Invert
Pattern
No. Name
9989 Col9988
9990 Col9989
9991 Col9990
9992 Col9991
9993 Col9992
9994 Col9993
9995 Col9994
9996 Col9995
9997 Col9996
9998 Col9997
9999 Col9998
10000 Col9999
10001 Col10000
Remove
Class: Col10000 (Nom)
Visualize All
583
51
Status
OK
Log
x 0



The imputed values are the same every time. There is no randomness involved. It uses the expectation maximization procedure described in the book "Schafer, J.L. Analysis of Incomplete Multivariate Data" to find the parameters of a multivariate normal distribution fit to the observed data, and then *fills in the missing values for an instance with the expected values* based on the multivariate normal distribution and observed values for the instance.

WEKA is rather stubborn about last column is the class…

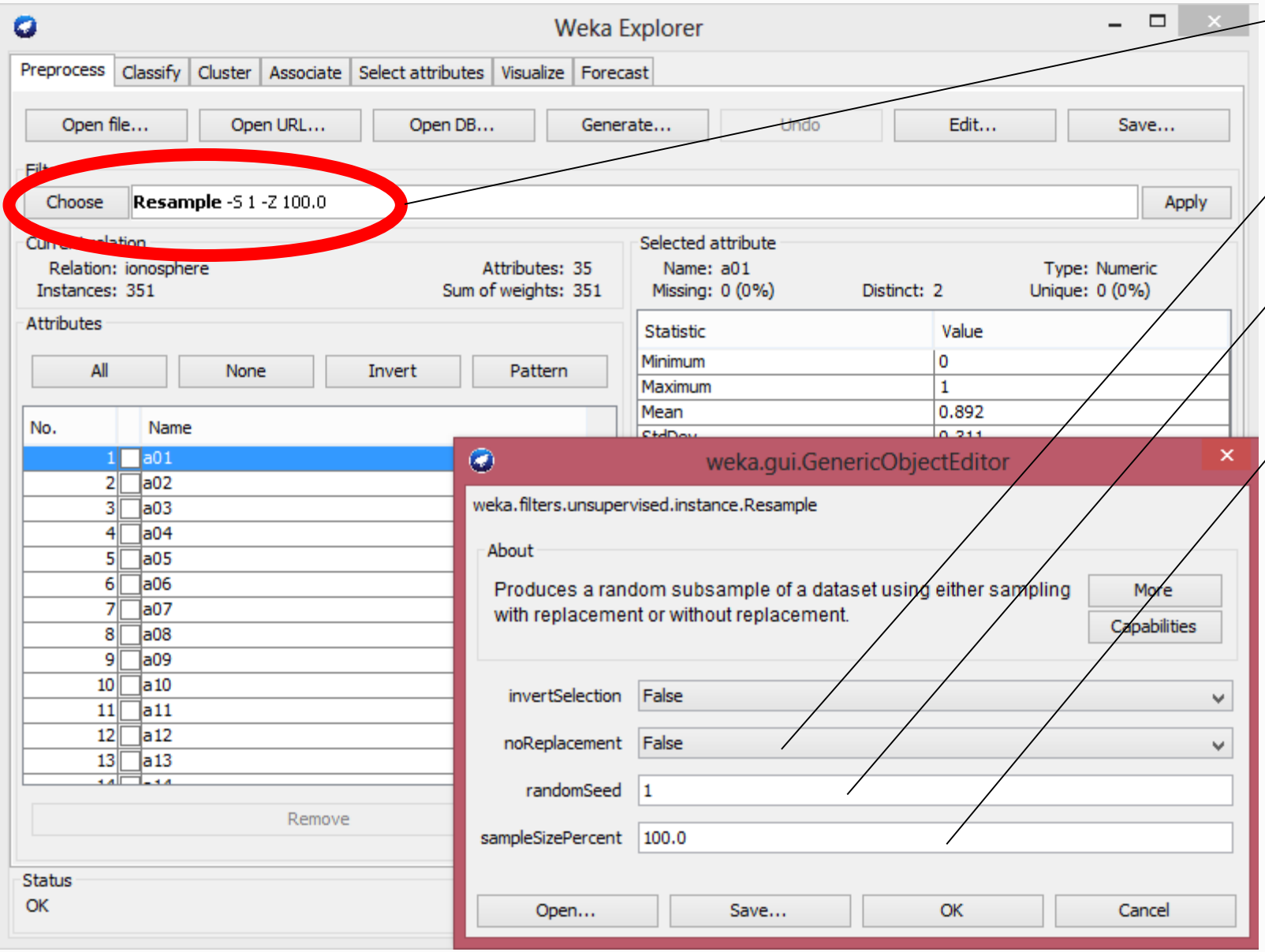


Specify the new ordering…

2-last, 1

1-5, 7-last, 6

filters → unsupervised → instance (rows)



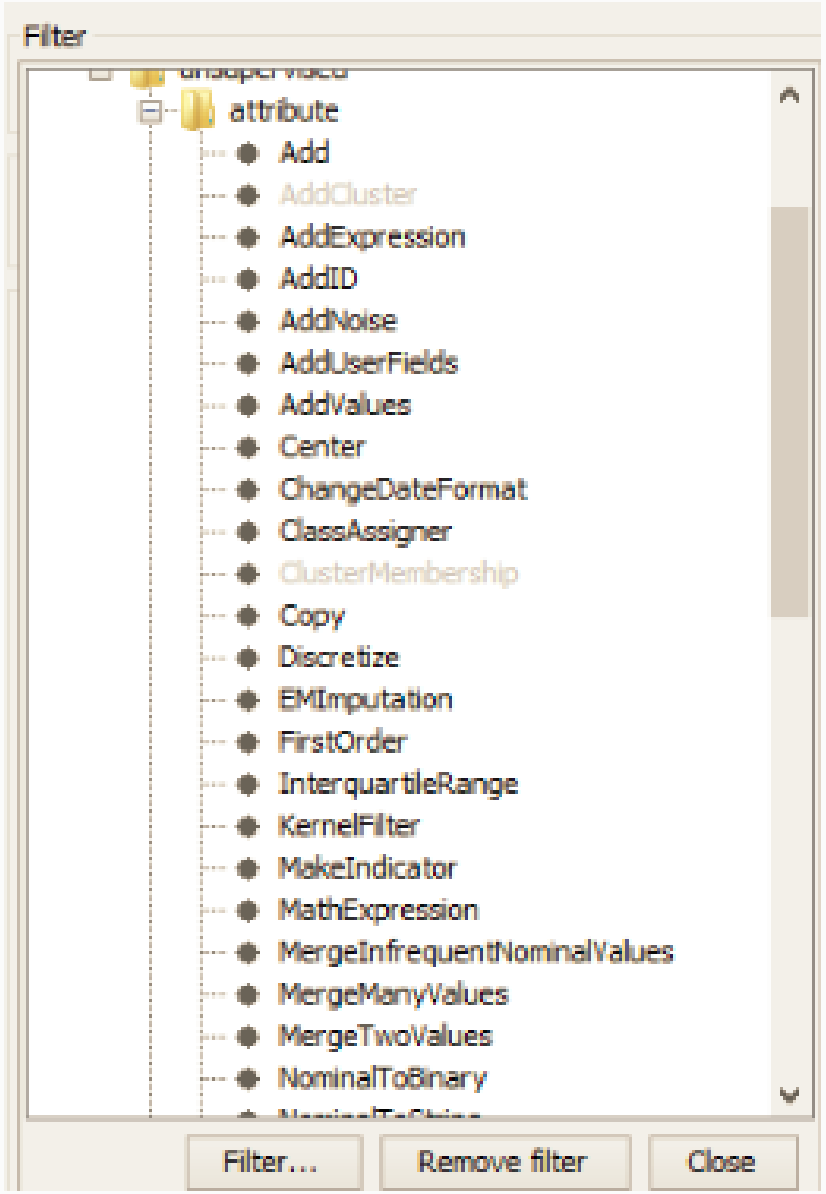
Resample, a random subsample with or without replacement;

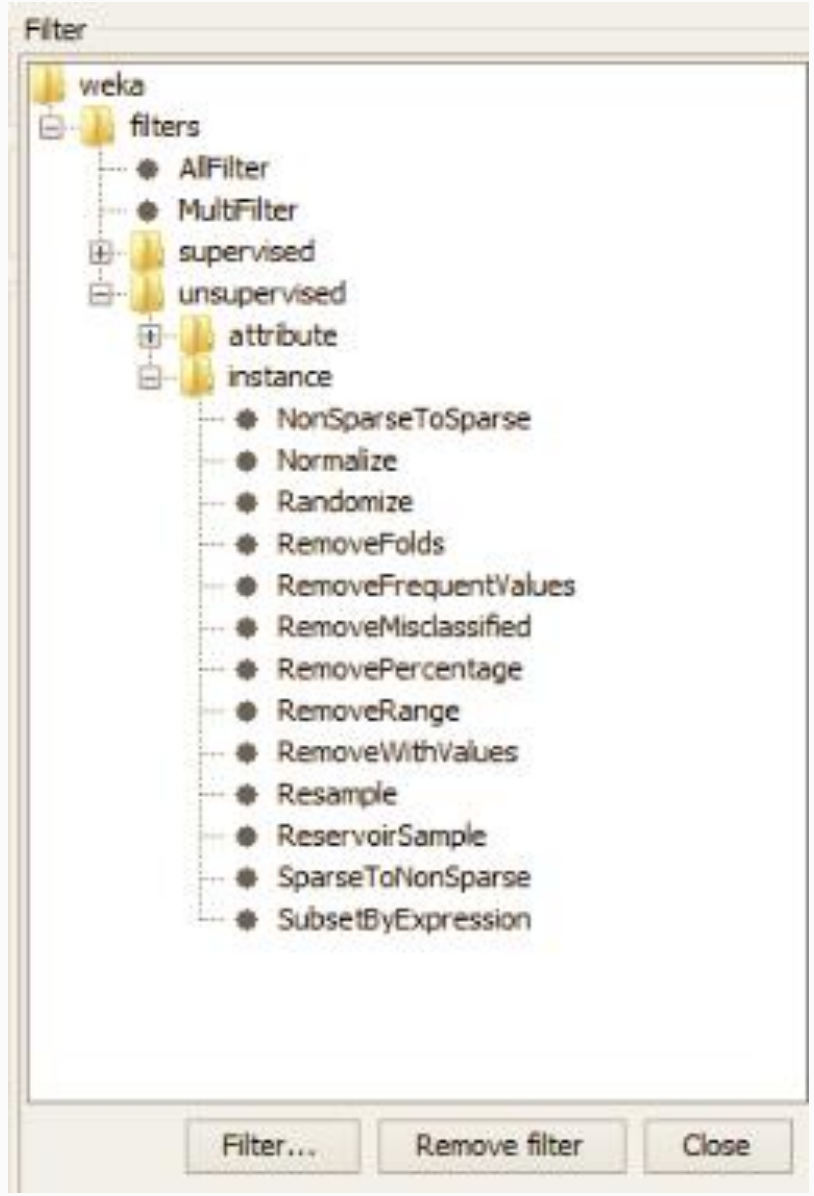
To replace or not...

Same random seed, will result in same (repeatable) sample.

Sample size, as percentage of original data set size.

Spend an afternoon exploring instance and attribute filters…





Attribute Selection

Problem: Where to focus attention?

- A universal problem of machine learning is where to focus the attention of the algorithm.
- What aspects of the problem at hand are important/necessary to solve it?
- Discriminate between the relevant and irrelevant parts of experience (feature space).

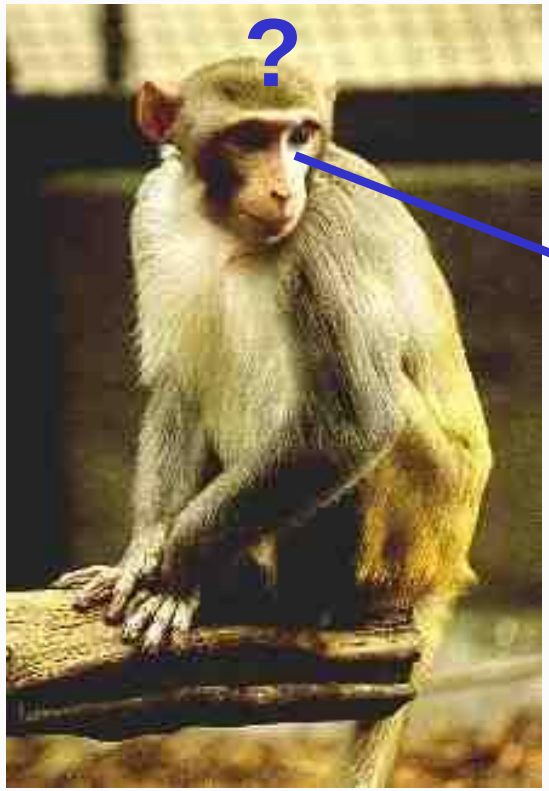
What is Feature selection ?

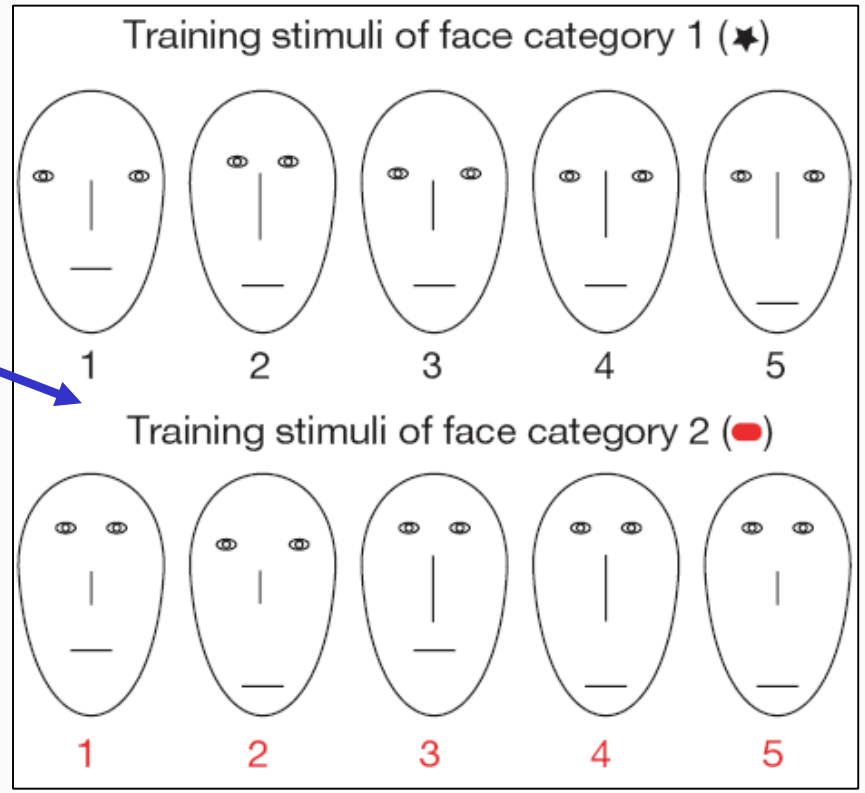
- Feature selection:
  Problem of selecting some subset of a learning algorithm's input variables upon which it should focus attention, while ignoring the rest

- Humans/animals do that constantly!

Motivational example from Biology

Monkeys performing classification task



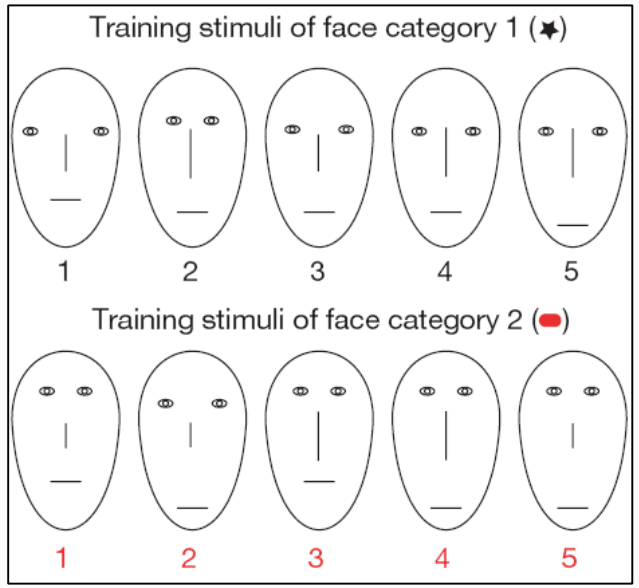


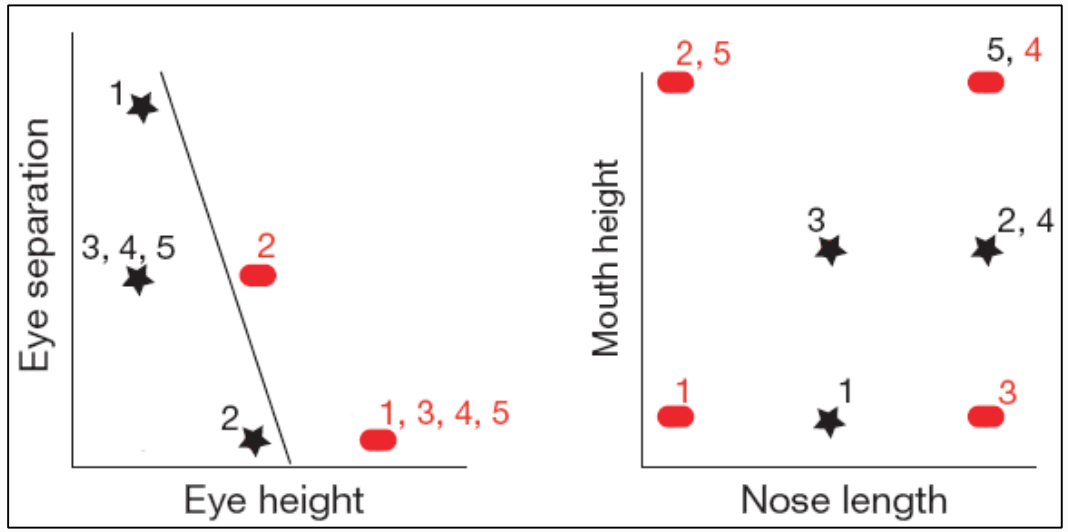
N. Sigala & N. Logothetis, 2002: *Visual categorization shapes feature selectivity in the primate temporal cortex.*

[1] Nathasha Sigala, Nikos Logothetis: Visual categorization shapes feature selectivity in the primate visual cortex. *Nature* Vol. 415(2002)

Motivational example from Biology

Monkeys performing classification task





Diagnostic features:
- Eye separation
- Eye height

Non-Diagnostic features:
- Mouth height
- Nose length

Motivational example from Biology

Monkeys performing classification task

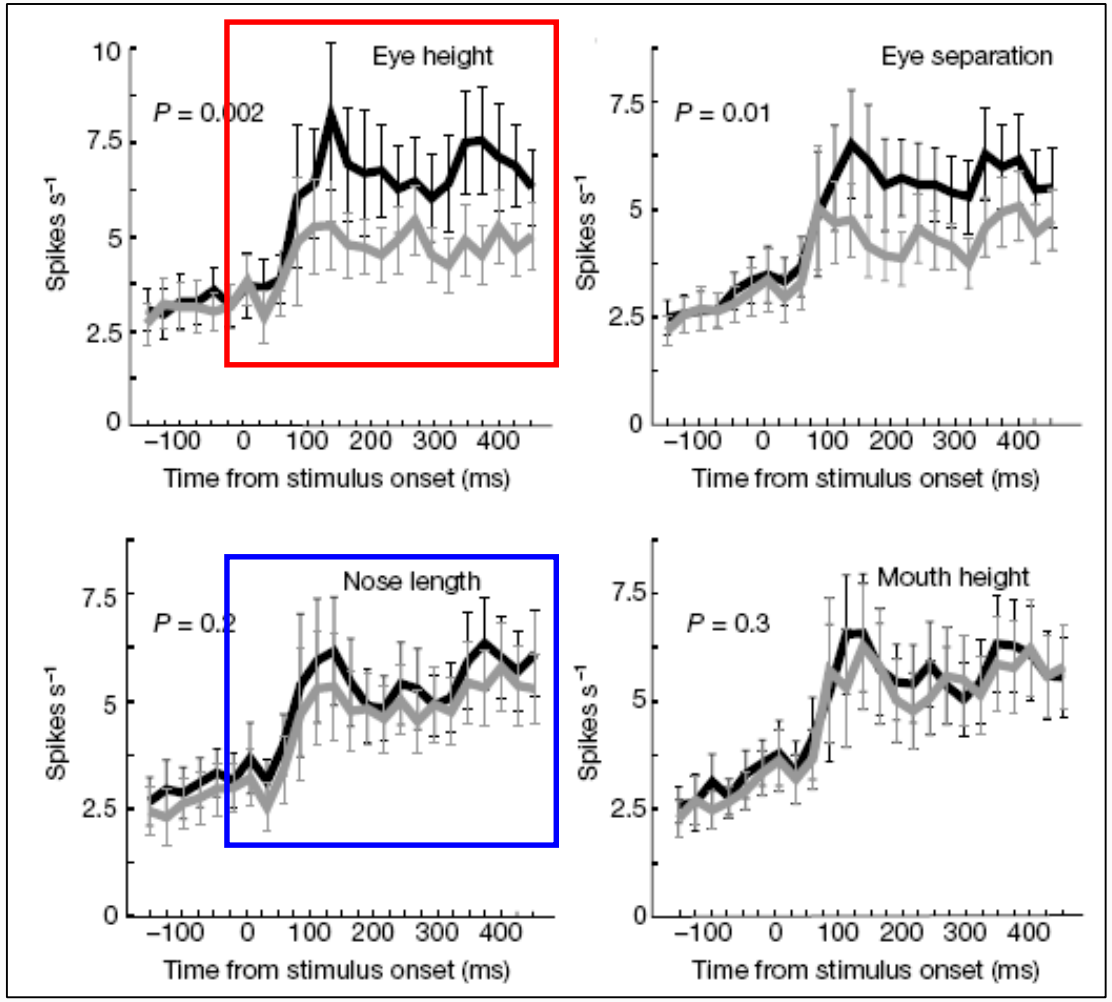
Results:

- Activity of a population of 150 neurons in the anterior inferior temporal cortex was measured
- 44 neurons responded significantly differently to at least one feature
- After Training: 72% (32/44) were selective to one or both of the diagnostic features (and not for the non-diagnostic features)

Motivational example from Biology

Monkeys performing classification task

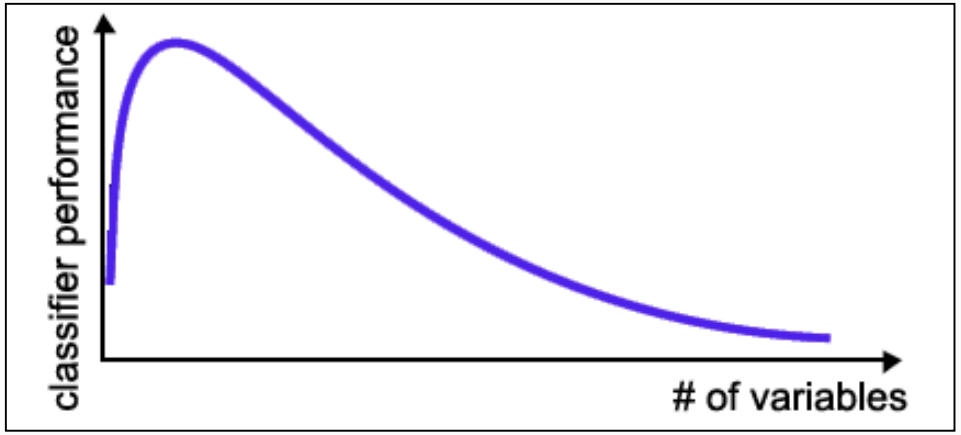
Results: (population of neurons)



Curse of Dimensionality

- The required number of samples (to achieve the same accuracy) grows exponentially with the number of variables!
- In practice: number of training examples is fixed!

  => the classifier's performance usually will degrade for a large number of features!



In many cases the information that is lost by discarding variables is made up for by a more accurate mapping/sampling in the lower-dimensional space !

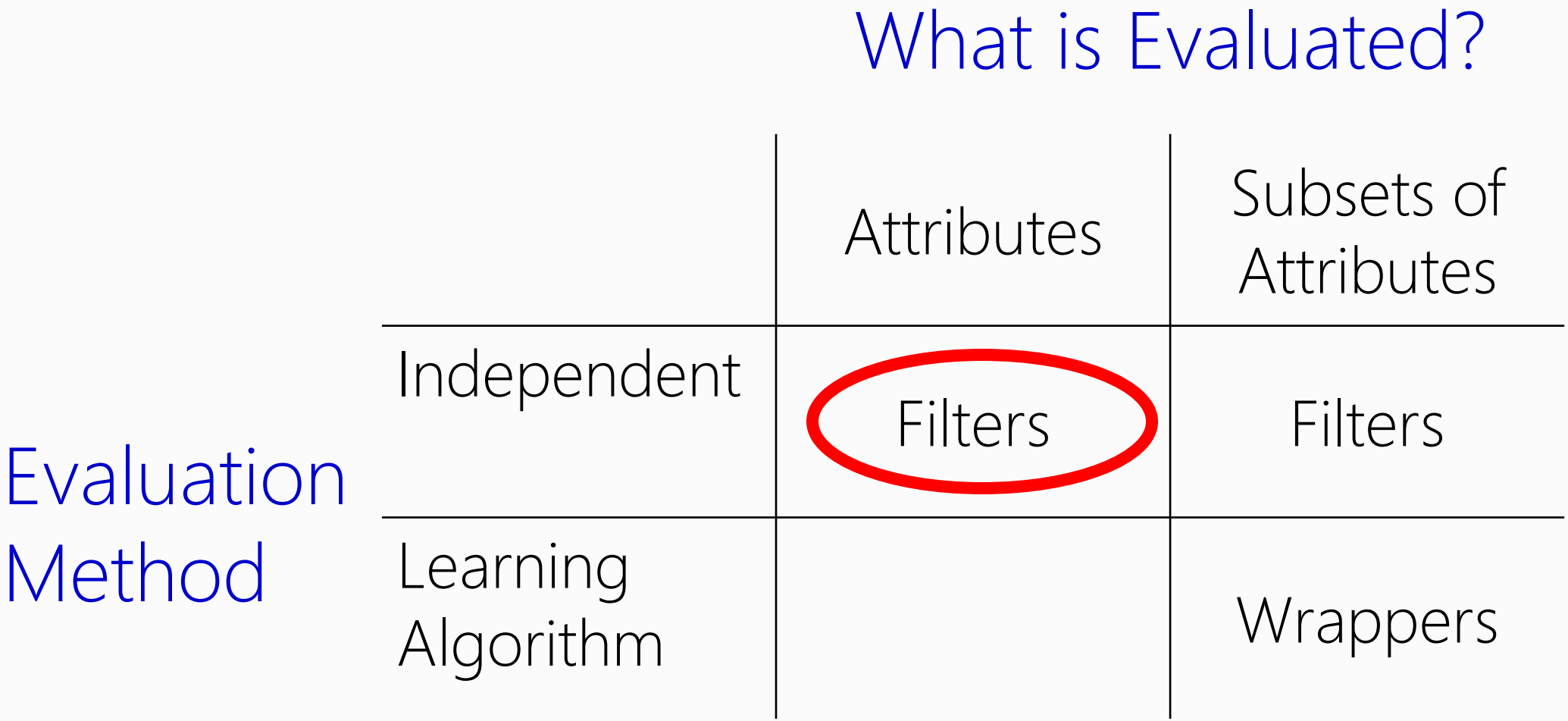
Attribute Selection Methods

Evaluation Method \ What is Evaluated?	Attributes	Subsets of Attributes
Independent	Filters	Filters
Learning Algorithm		Wrappers

What you need to remember…

- It's **not all about accuracy**, feature selection can result in a model that is easier and faster to train, less features to maintain, questions to ask, all very pragmatic.
- **Filtering is fast**, **linear**, and **intuitive** as it identifies a statistical dependency;
- **Filtering** is **model oblivious**, most relevant may not be optimal for modeling;
- Features found with wrappers **may not be optimal** for other learners;
- **Wrappers** are **model-aware**, relevance is clear, but **slow** and **nonintuitive**;
- **PCA and SVD are lossy**, algorithms such as SVM may perform better on entire data set – if they are able to handle that many attributes/data;
- PCA and SVD **work on the entire data set**, which may not fit into memory;
- If the number of features is huge and many irrelevant, it makes sense to **start with fast feature filtering first**, eliminate rubbish, and then proceed with a more sophisticated feature search;

Attribute Selection Methods



Filters

Results in either

Ranked list of attributes

- Typical when each attribute is **evaluated individually**
- You must **select** how many to keep

A selected subset of attributes

- Forward selection
- Best first
- Random search such as genetic algorithm

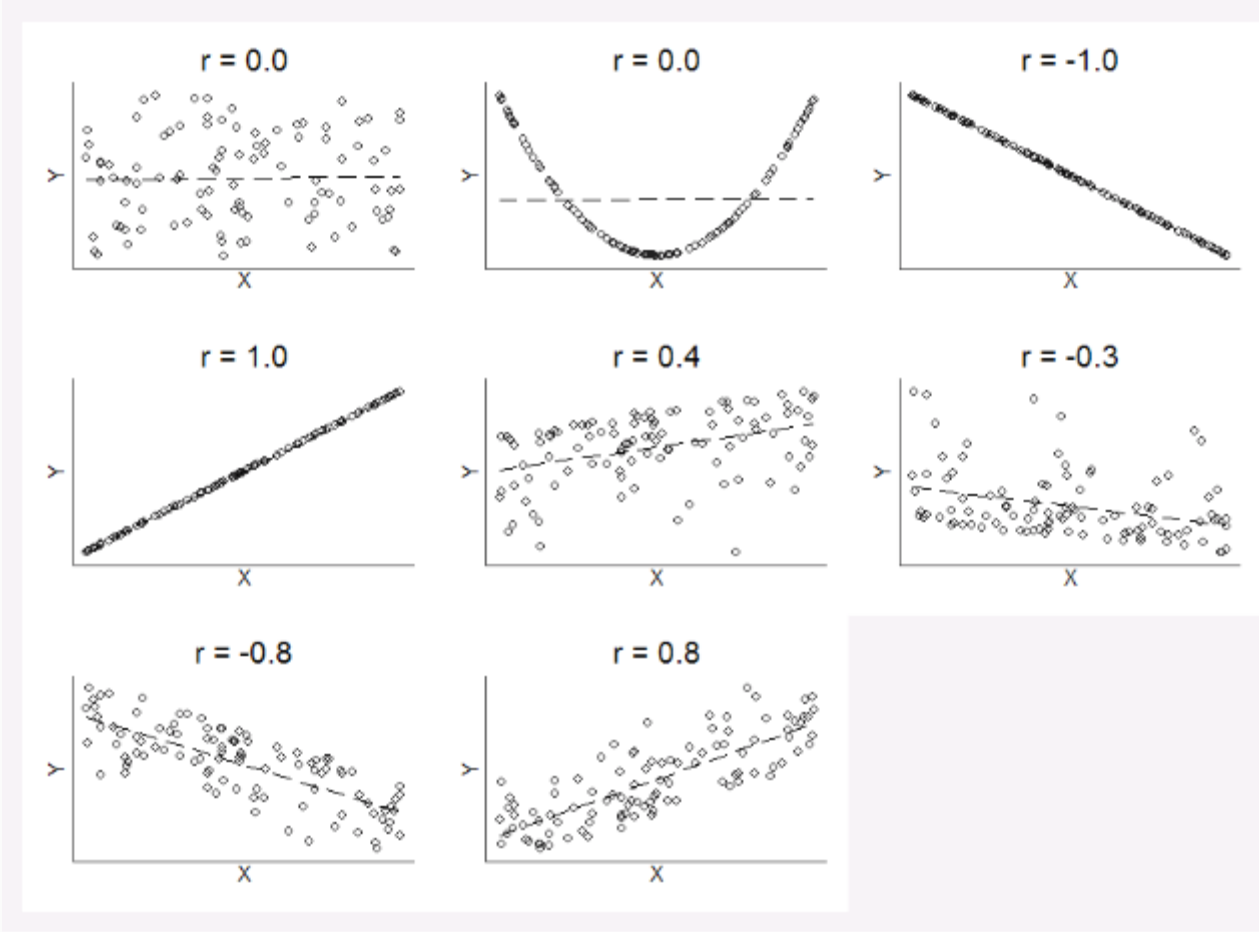
Filter Evaluation Examples

- Information Gain
- Gain ration
- Pearson Correlation

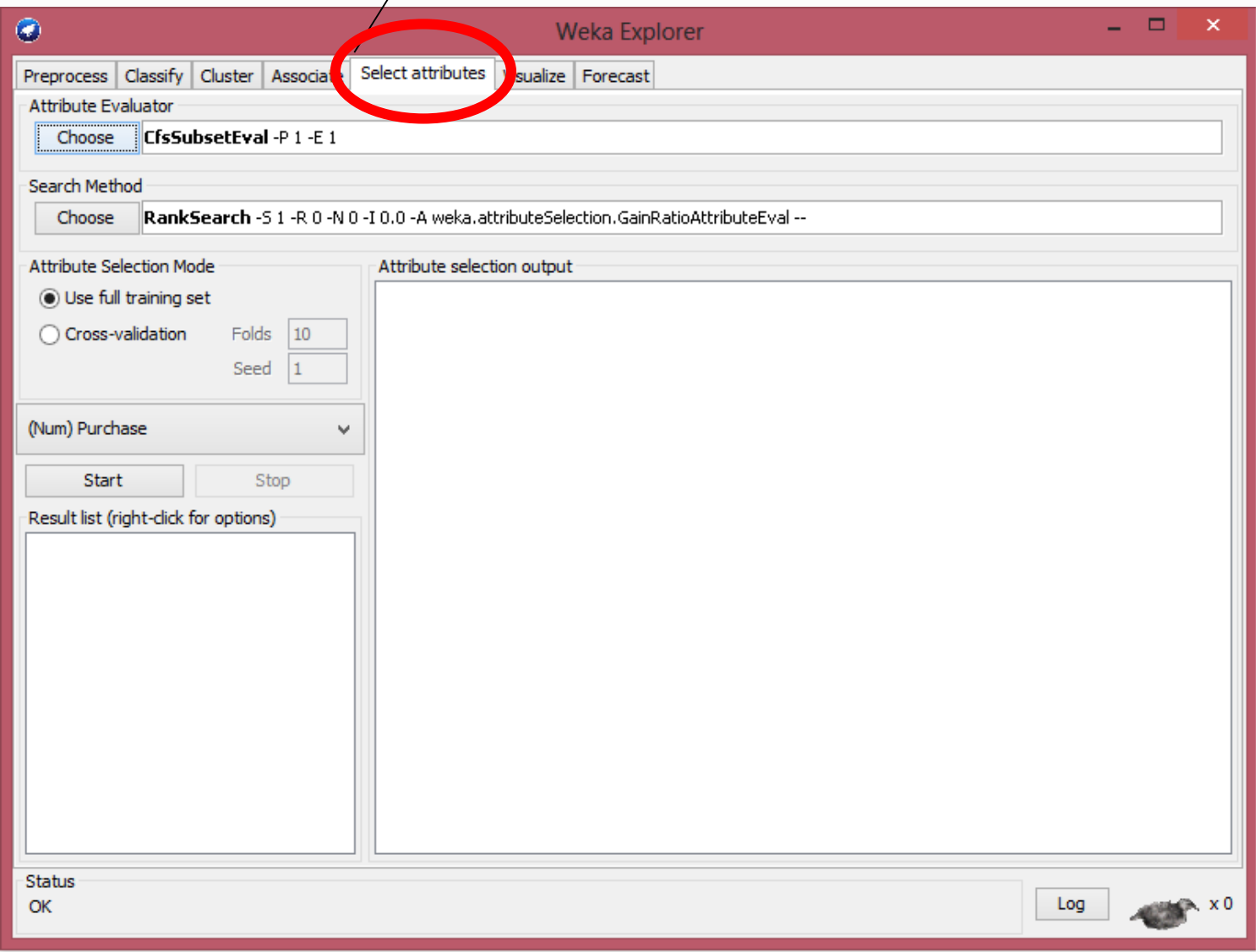
- Correlation
  - High correlation with class attribute
  - Low correlation with other attributes

Pearson's Product Moment Correlation

A correlation coefficient shows the degree of linear dependence of x and y. In other words, the coefficient shows how close two variables lie along a line. If the coefficient is equal to 1 or -1, all the points lie along a line. If the correlation coefficient is equal to zero, there is no linear relation between x and y. However, this does not necessarily mean that there is no relation between the two variables. There could e.g. be a non-linear relation.

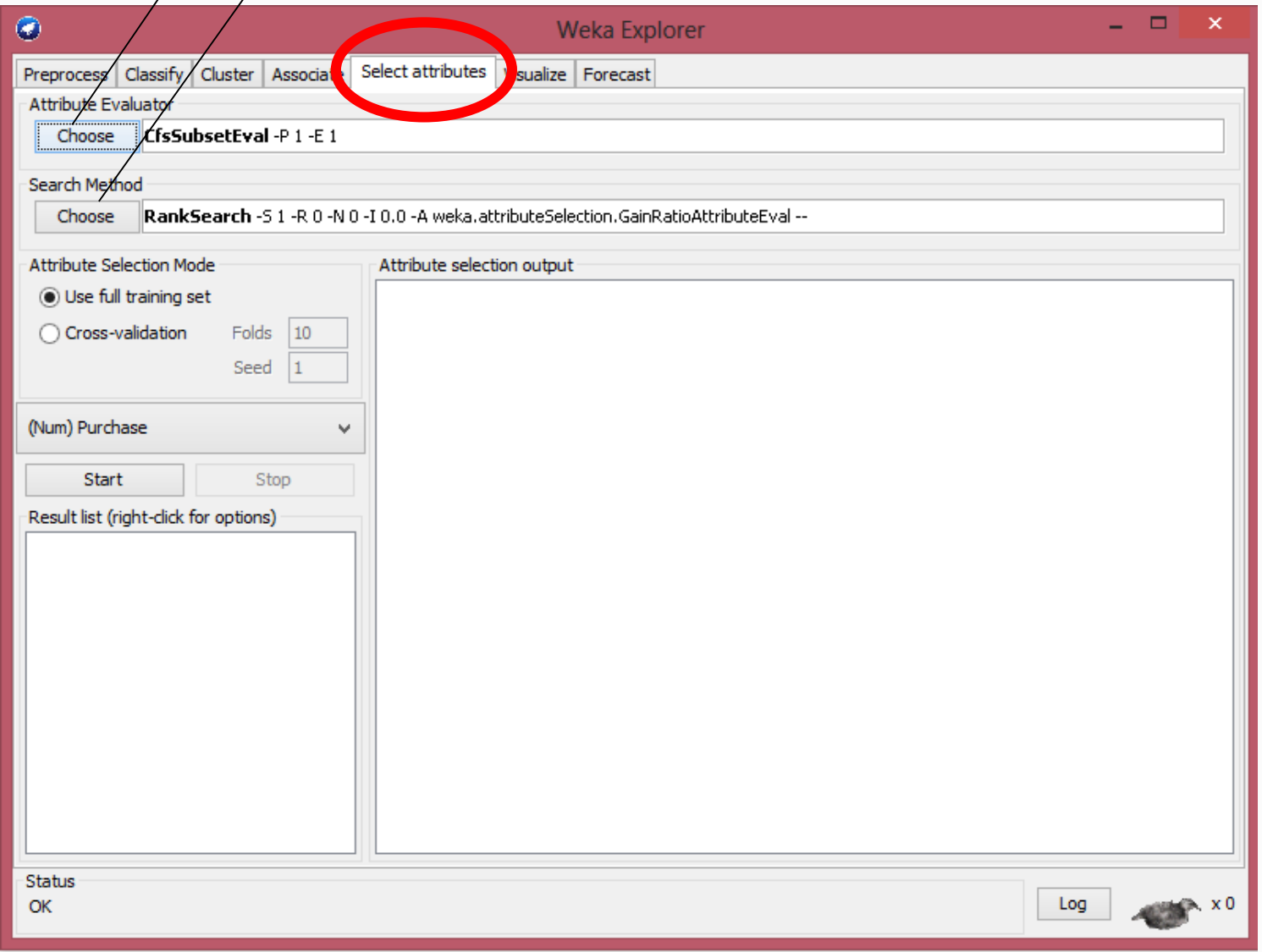


Tab for selecting attributes in a data set…

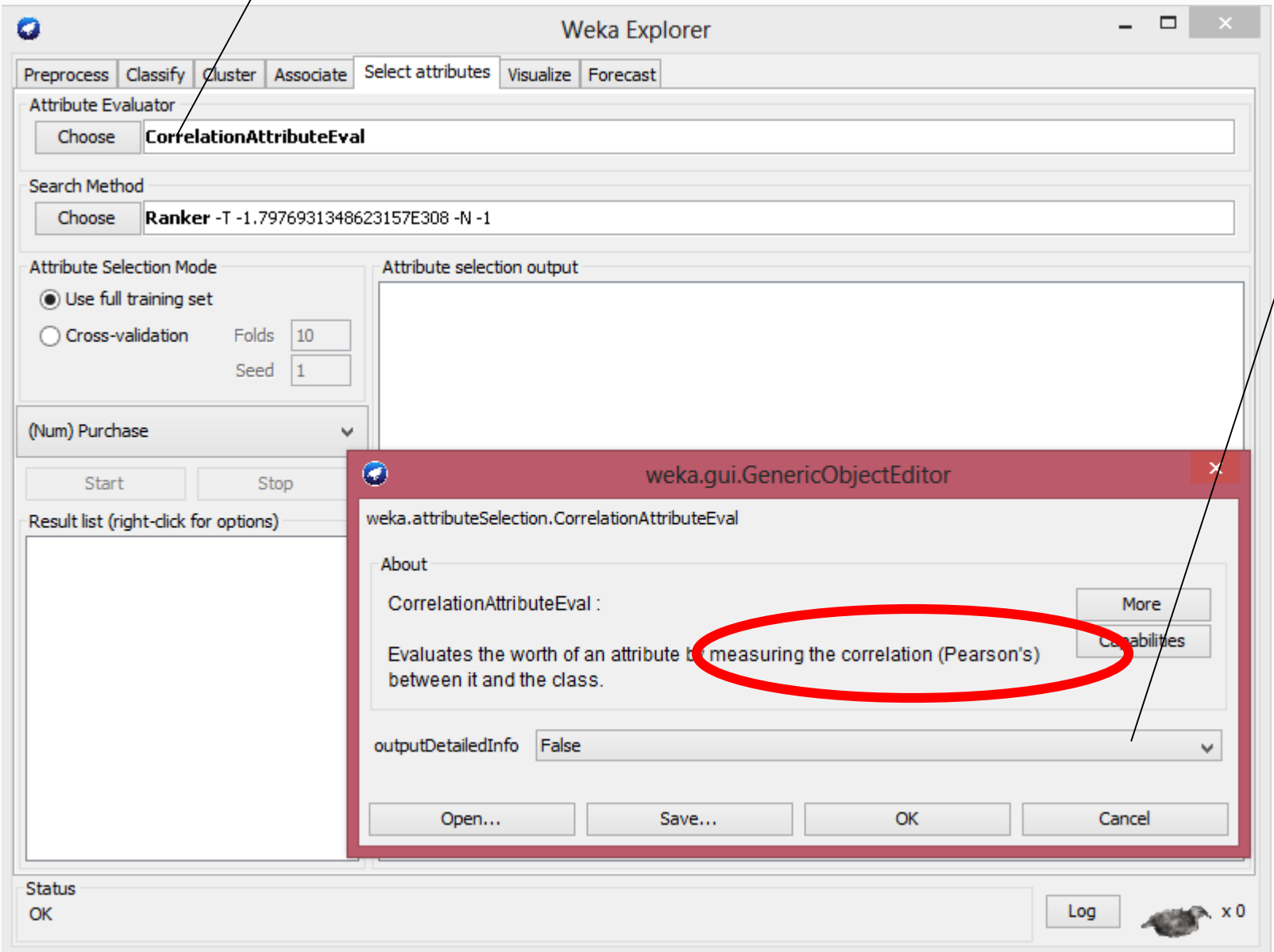


Interface for classes that evaluate attributes…

Interface for **ranking** or **searching** for a subset of attributes…

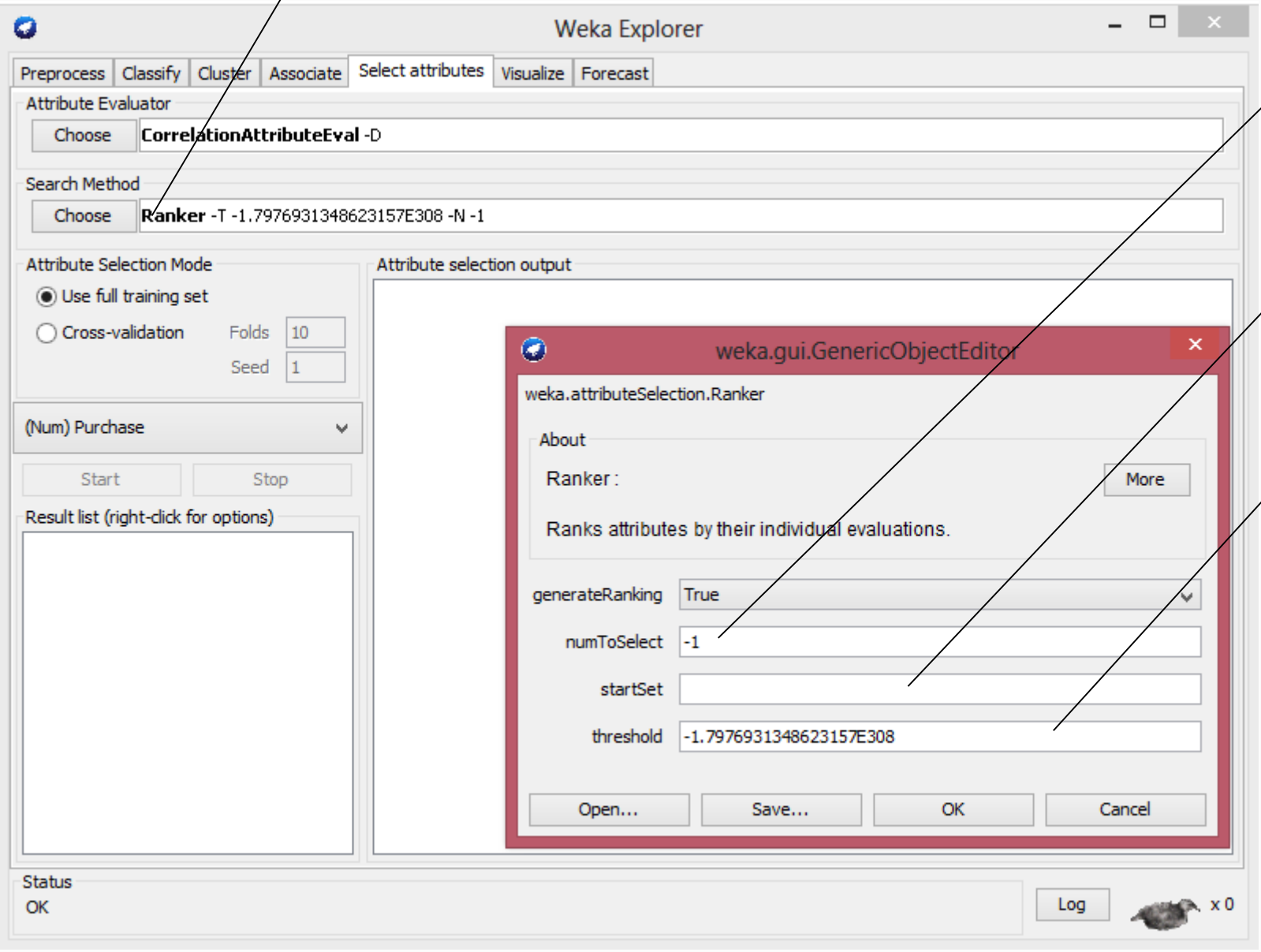


Select **CorrelationAttributeEval** for Pearson Correlation…



False, doesn't return R score
True, returns R scores;

Ranks attributes by their individual evaluations, used in conjuction with GainRatio, Entropy, Pearson, etc...



Number of attributes to return, -1 returns all ranked attributes;

Attributes to ignore (skip) in the evaluation forma: [1, 3-5, 10];

Cutoff at which attributes can be discarded, -1 no cutoff;

Pearson Correlation Exercise…

Predicting Self-Reported Health Status

The Data Set, NHANES_data.csv (National Health and Nutrition Examination Survey)

How would you say your health in general is?
Excellent predictor of mortality, health care utilization & disability
How I processed it…

- 4000 variables;
- Attributes with > 30% missing values removed (dropped column);
- 105 variables remaining;
- Chi-square test, variable and target, remove variables with P value < .20;
- Impute all missing values using model based approach (cluster impute);
- 85 variables remaining;

Pearson Correlation Exercise…

- Load NHANES_data.csv
- Convert the last column from numeric to nominal
- Find the top 15 features using Pearson Correlation

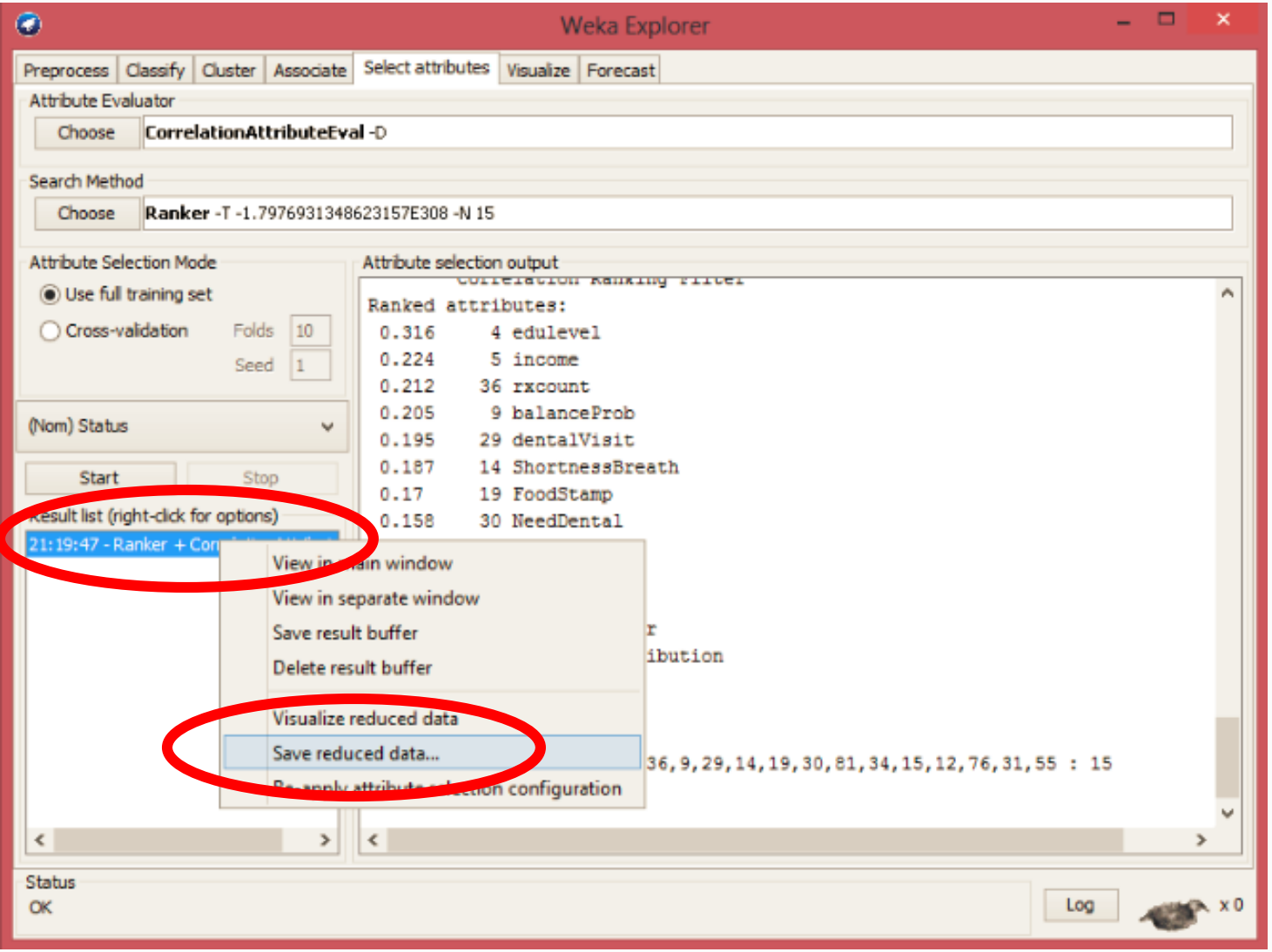
Pearson Correlation Exercise…

- Load NHANES_data.csv
- Convert the last column from numeric to nominal
- Find the top 15 features using Pearson Correlation

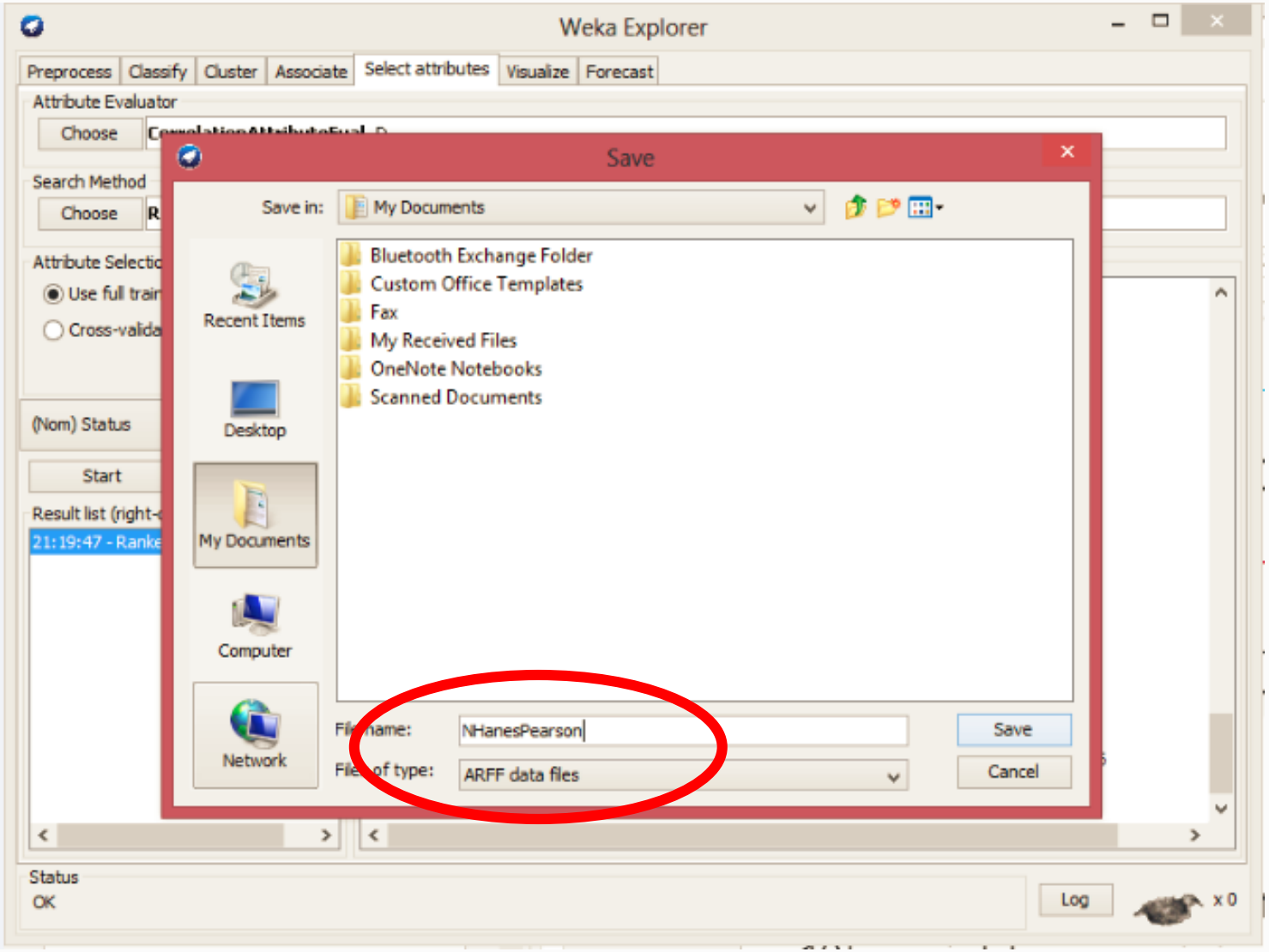
- Try at least one other Attribute Evaluation technique
  - ChiSquared
  - GainRatio
  - InformationGain
  - …

Other Attribute Evaluation Metrics

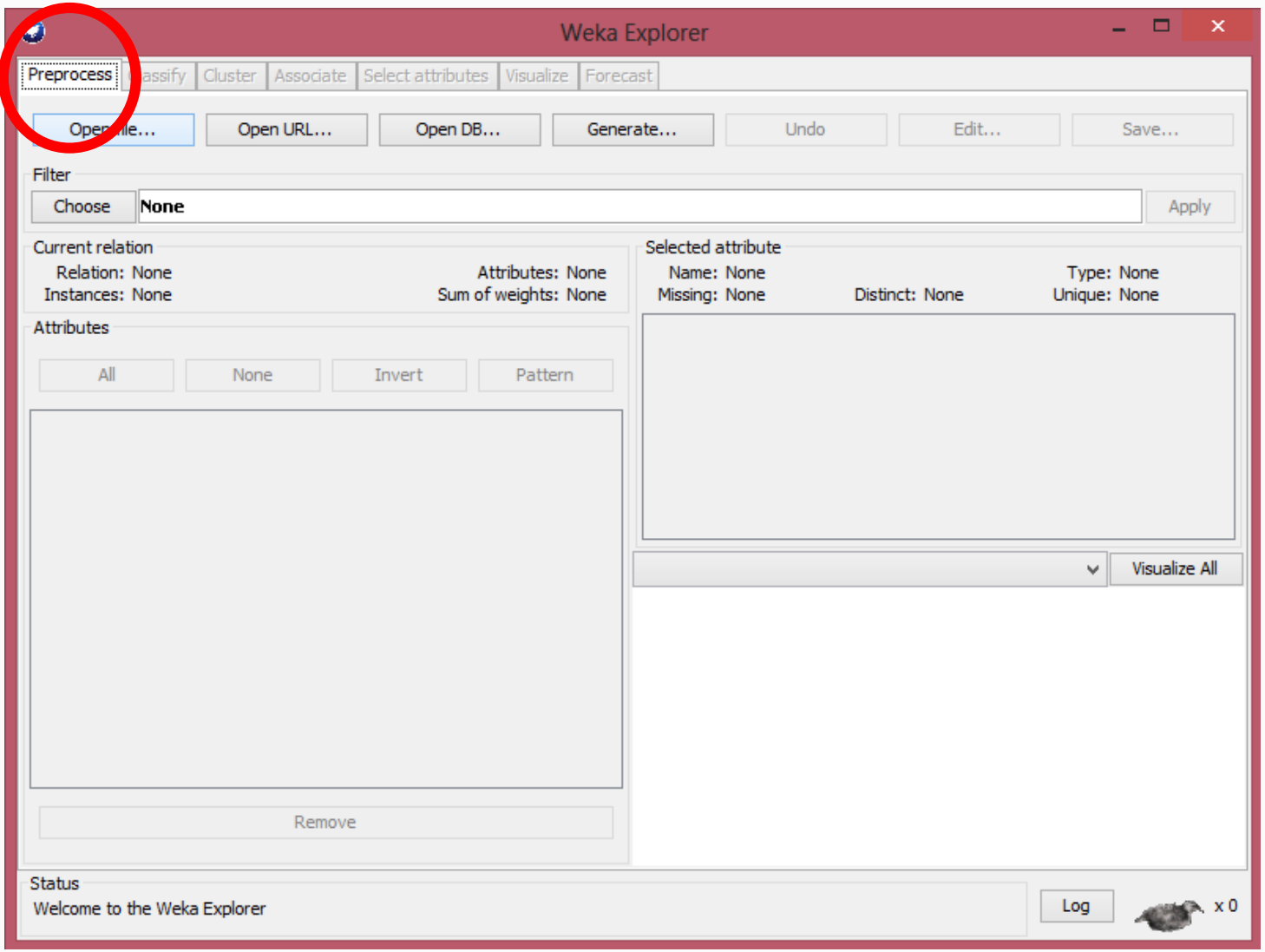
- OneRAttributeEval: Evaluates the worth of an attribute by using the OneR classifier.
- GainRatioAttributeEval: Evaluates the worth of an attribute by measuring the gain ratio with respect to the class. GainR(Class, Attribute) = (H(Class) - H(Class | Attribute)) / H(Attribute).
- InfoGainAttributeEval: Evaluates the worth of an attribute by measuring the information gain with respect to the class. InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute).
- ChiSquaredAttributeEval: Evaluates features individually by measuring their chi-squared statistic with respect to the classes.
- ReliefFAttributeEval: Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class



- **Right Click** on the new line in the Result list;
- From the pop-up menu, select the item **Save reduced data**…



- **Right Click** on the new line in the Result list;
- From the pop-up menu, select the item **Save reduced data**…
- Save the dataset with 15 selected attributes to file *NHanesPearson.arff*



- **Right Click** on the new line in the Result list;
- From the pop-up menu, select the item **Save reduced data**...
- Save the dataset with 15 selected attributes to file *NHanesPearson.arff*
- Switch to the **Preprocess** mode in Explorer
- Click on Open file... and open the file *NHanesPearson.arff*
- Switch to the **Classify** submode
- Click on Choose, select classifier and use this feature set and data to build a predictive model;

But, before we get too carried away…

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize | Forecast

Attribute Evaluator

Choose **CorrelationAttributeEval**

Search Method

Choose **Ranker** -T -1.7976931348623157E308 -N 15

Attribute Selection Mode

- Use full training set
- Cross-validation Folds 10 Seed 1

(Nom) Status

Start | Stop

Result list (right-click for options)

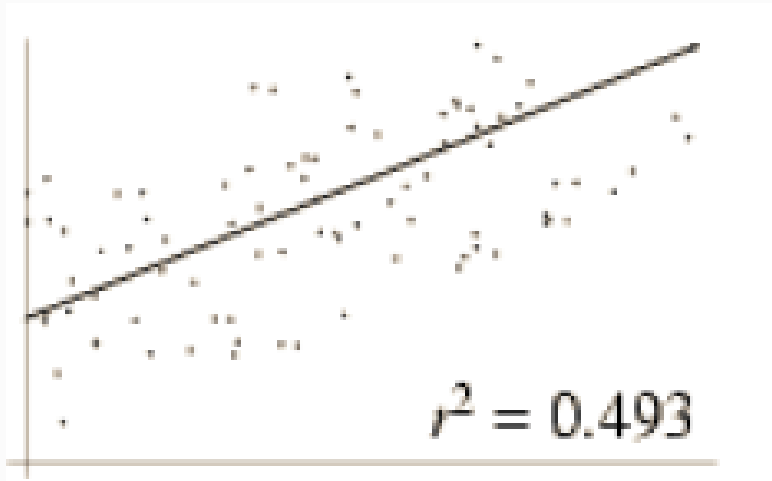
21:53:59 - Ranker + CorrelationAttribute

Attribute selection output

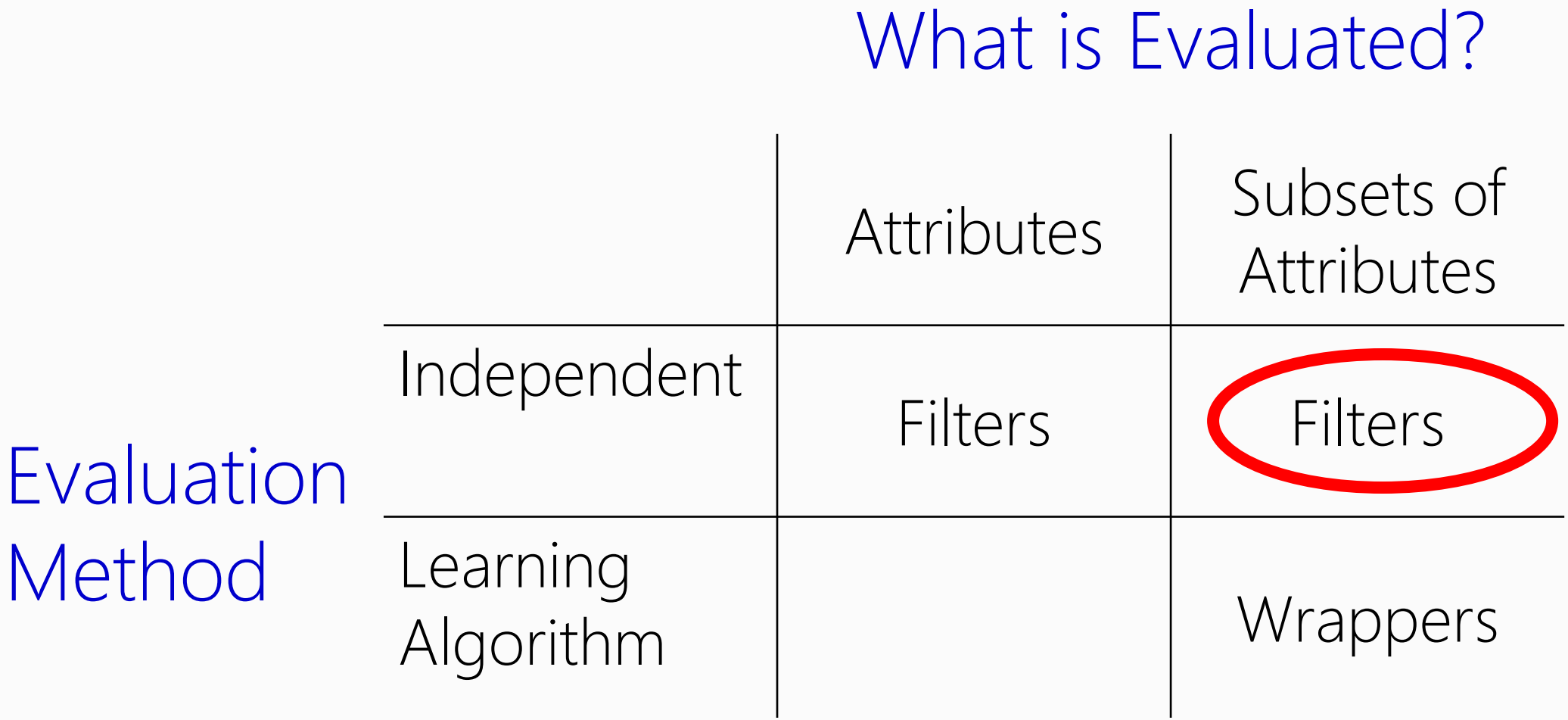
```
Ranked attributes:
 0.316     4 edulevel
 0.224     5 income
 0.212    36 rxcount
 0.205     9 balanceProb
 0.195    29 dentalVisit
 0.187    14 ShortnessBreath
 0.17     19 FoodStamp
 0.158    30 NeedDental
 0.158    81 VitaminD
 0.152    34 activity
 0.152    15 Restaurant
 0.149    12 ChestPainEver
 0.148    76 RedCellDistribution
 0.143    31 vigorousAct
 0.141    55 g_tocopherol

Selected attributes: 4,5,36,9,29,14,19,30,81,34,15,12,76,31,55 : 15
```

Status: OK Log x 0

- Anything below 0.3 isn't highly correlated with the target…

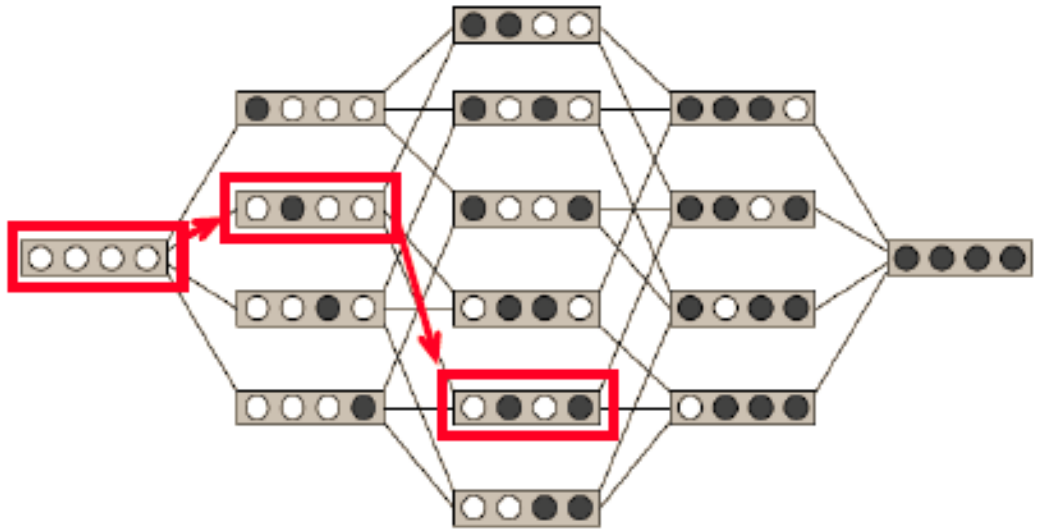


Attribute Selection Methods



Forward Search

- Define an intial subset
  - begin with empty set

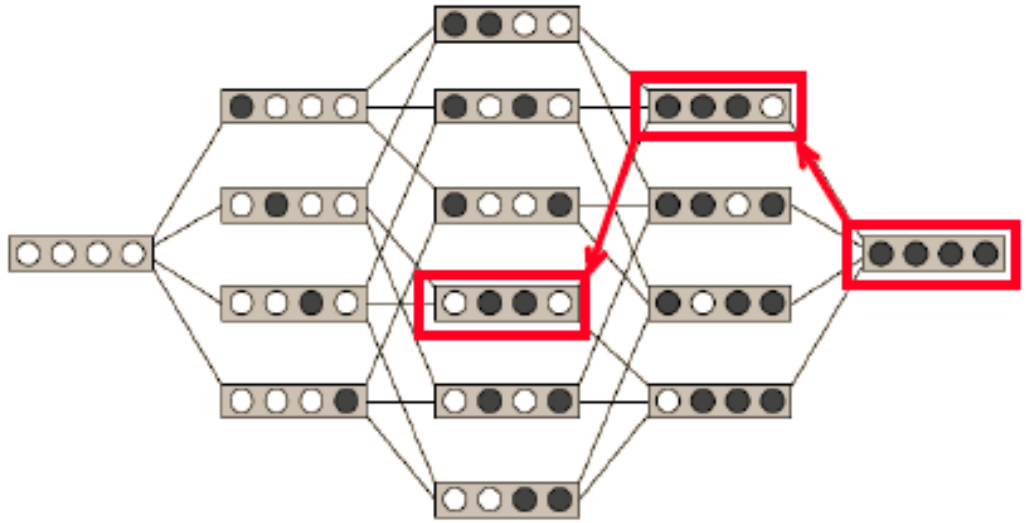


- Choose a strategy to update subset
  - Filter: add feature that increases the most the relevance/redundancy compromize
  - Wrapper: add feature that increases the most the performances of the model

- Decide when to stop
  - Filter: needs a supplementary criterion
  - Wrapper: stop when adding a feature increases the generalization error

Backward Search

- Define an intial subset
  - begin with the full set

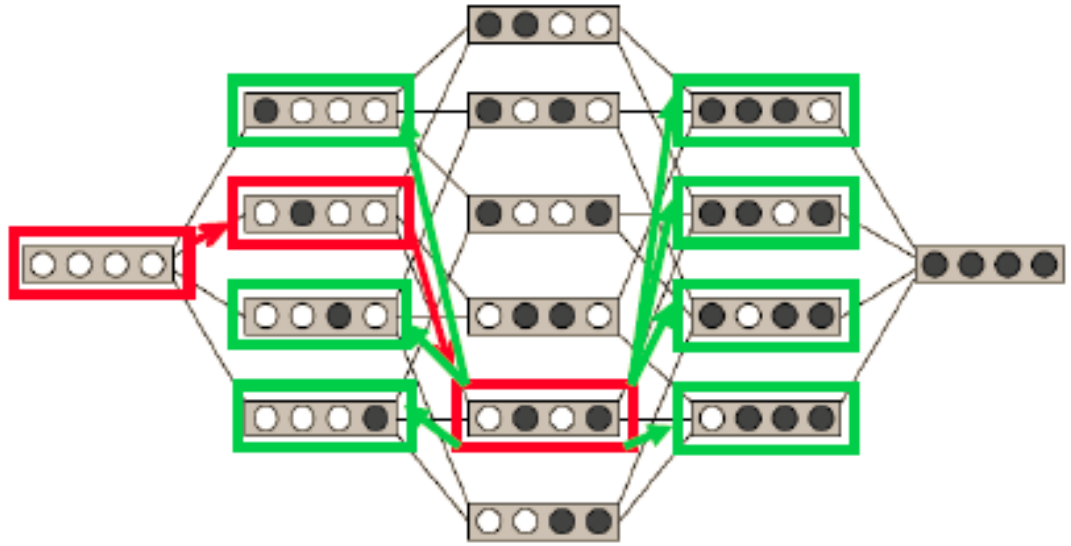


- Choose a strategy to update subset
  - Filter: remove feature that increases t
    most the relevance/redundancy compromize
  - Wrapper: remove feature that increases the most the performances of the model

- Decide when to stop
  - Filter: needs a supplementary criterion
  - Wrapper: stop when removing a feature increases the generalization error

Variants

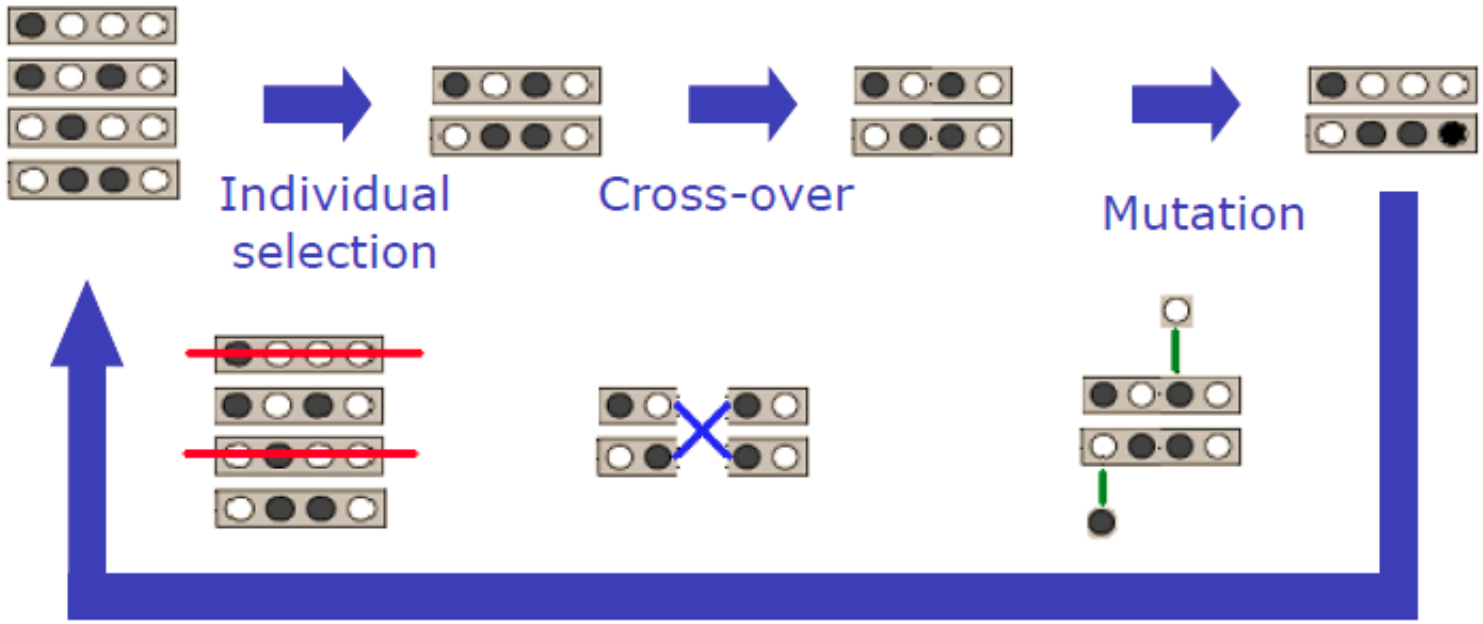
- Forward-backward
  - At each step, consider all additions and removals of one variable, and select the best result
  - Wrapper: this makes sense
  - Filter: in theory, the mutual information cannot increase with less variables



Genetic Algorithms

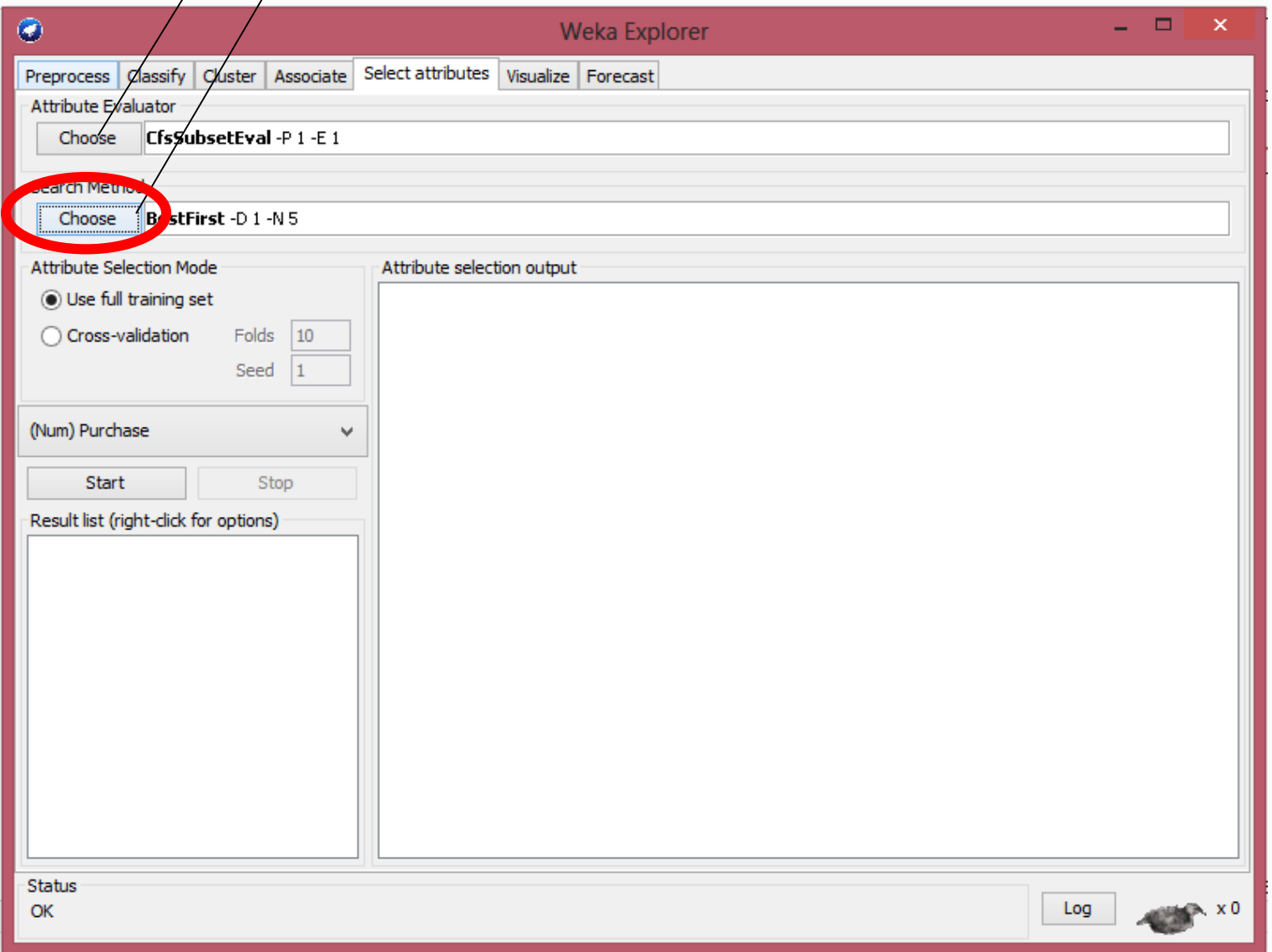
"clever" random exploration of the space of subsets

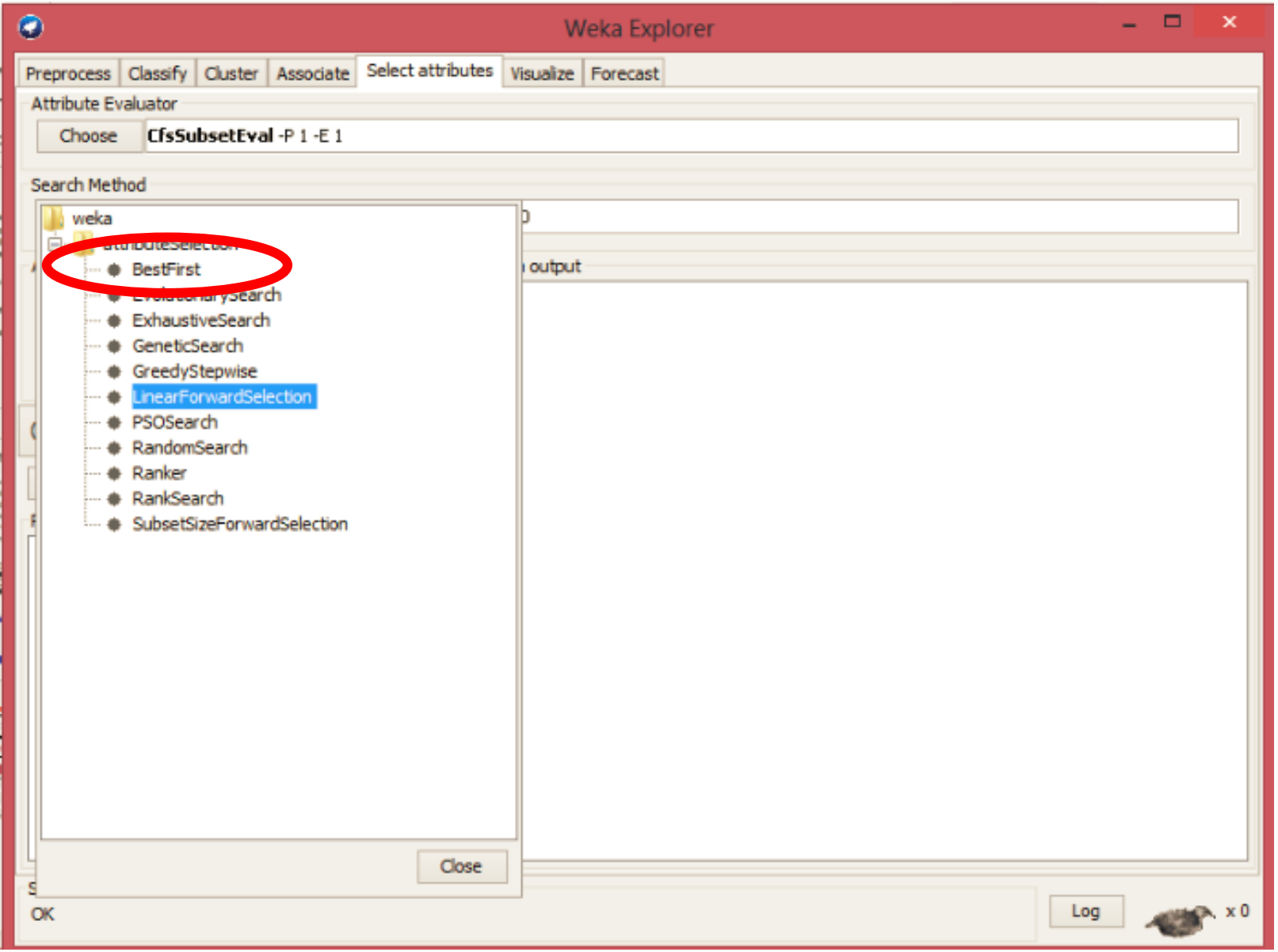
1. Draw initial population (candidate subsets)
2. Select individuals
3. Apply cross-overs and mutations on individuals
4. Repeat from 2 until a new population is generated
5. Select best individuals and repeat from 1



Interface for classes that evaluate attributes…

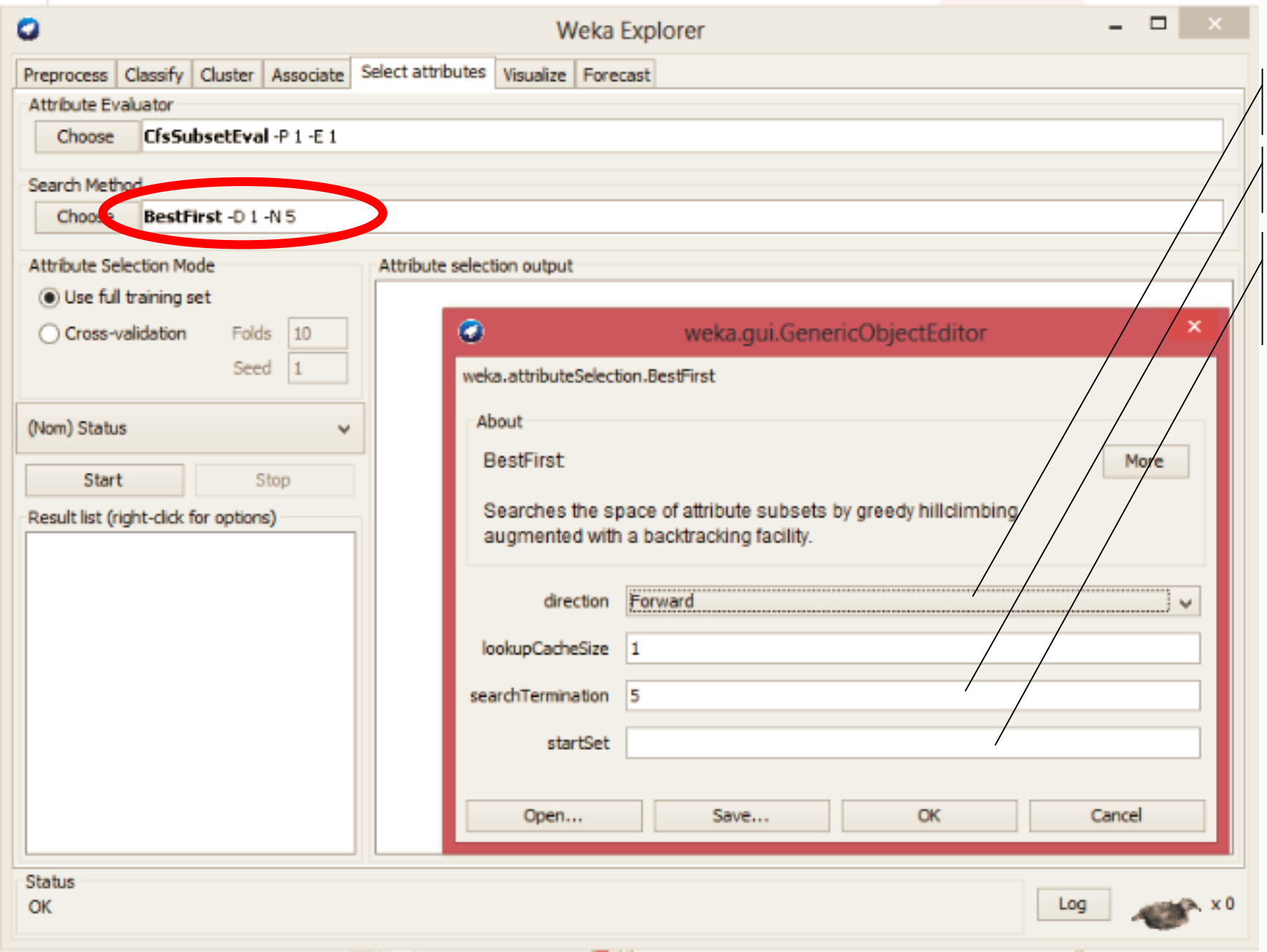
Interface for **ranking** or **searching** for a subset of attributes…





This is an incredibly rich collection of search strategies, from the tried and true forward/backward, to genetic, greedy, random search, exhaustive,…

The best offered in a free tool for building predictive models…

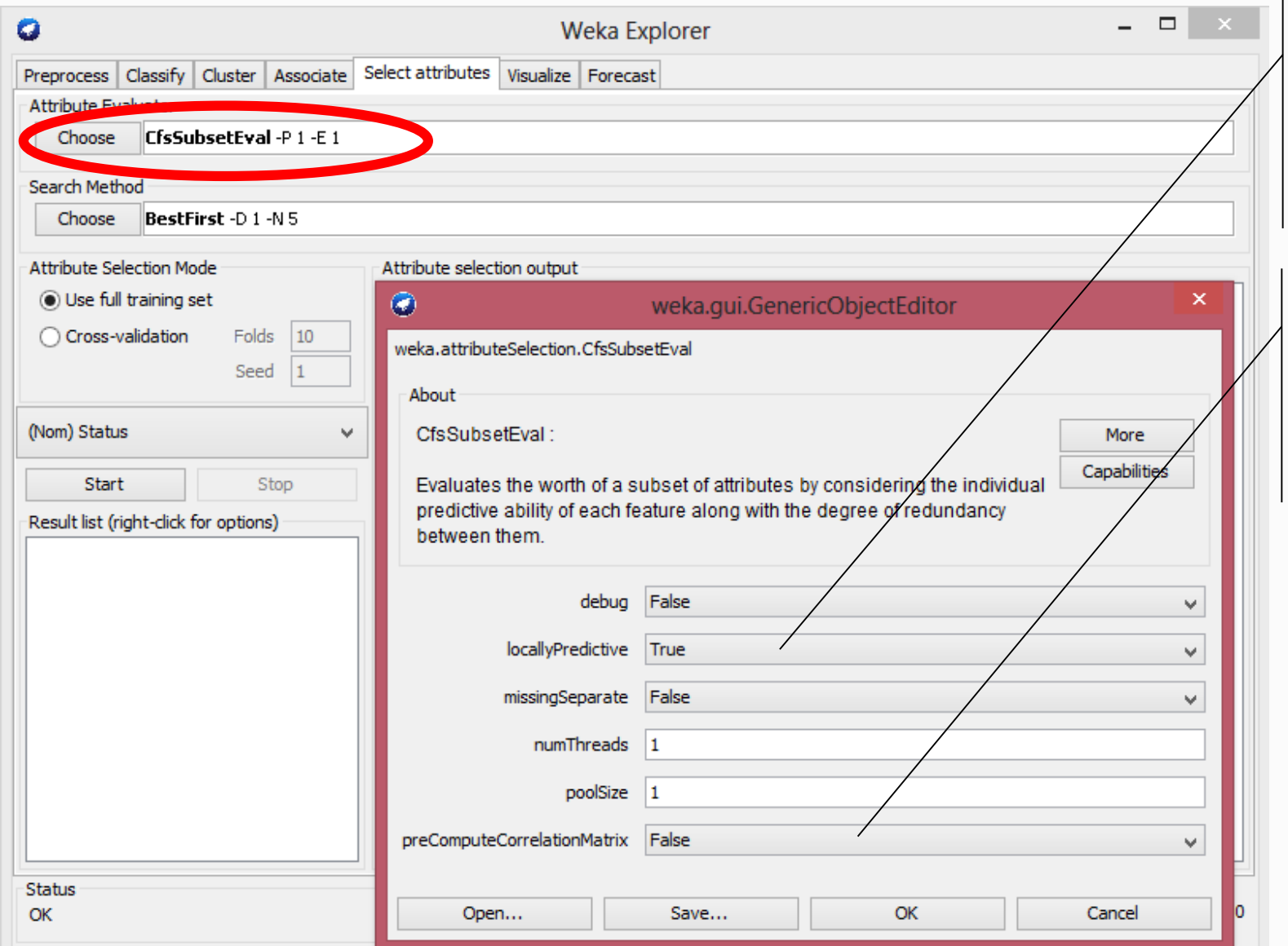


Forward, Backward, Bi-Directional

Cutoff for backtracking…

Attributes to "seed" the search, listed individually or by range.

CfsSubsetEval, evaluates redundancy between attributes, discarding attributes that are highly redundant with attributes already in selection, evaluates correlation with target. Selects features that are correlated with class, low intercorrelation with other features.



True: Adds features that are correlated with class and NOT intercorrelated with other features already in selection.
False: Eliminates redundant features.

Precompute the correlation matrix in advance, useful for fast backtracking, or compute lazily. When given a large number of attributes, compute lazily…

Attribute Search Exercise…

- Load NHANES_data.csv
- Convert the last column from numeric to nominal
- Set the search method as Best First, Forward
- Set the attribute evaluator as CfsSubsetEval
- Run across all attributes in data set…

Attribute Selection Methods

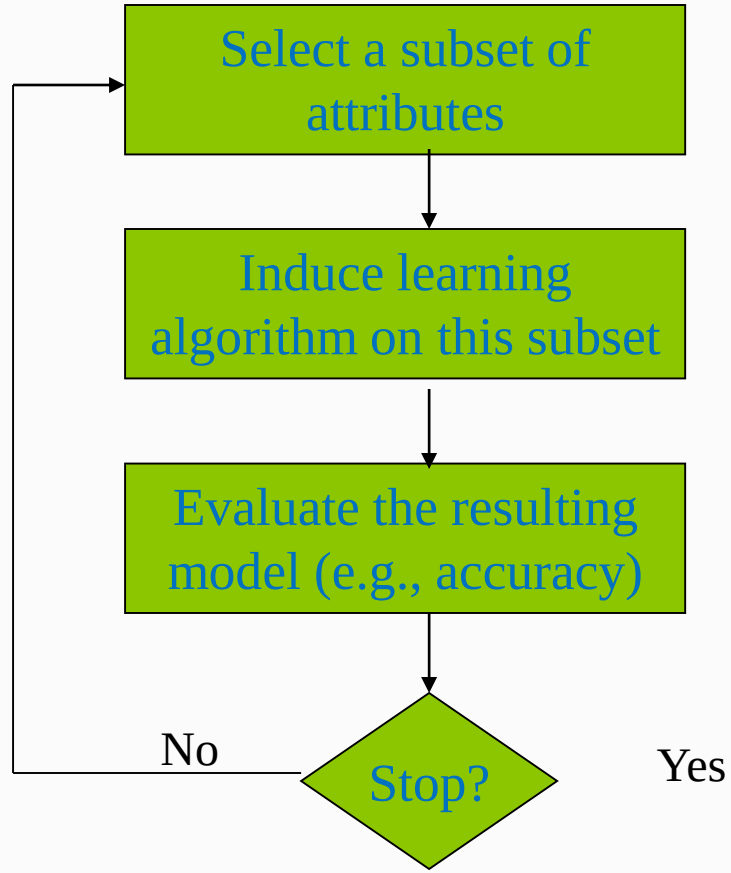
What is Evaluated?

Evaluation Method

	Attributes	Subsets of Attributes
Independent	Filters	Filters
Learning Algorithm		Wrappers

Wrappers

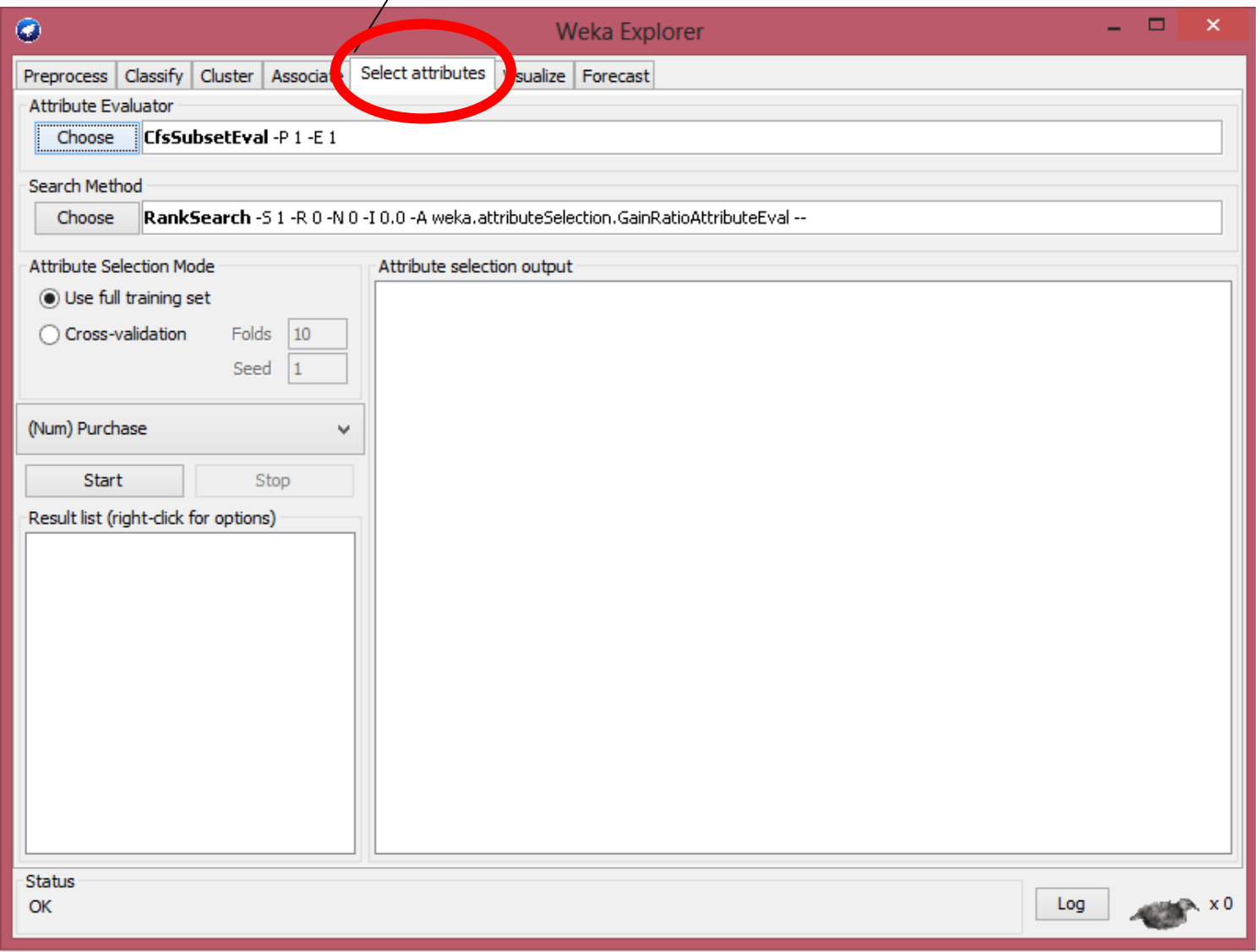
- "Wrap around" the learning algorithm
- Must therefore always evaluate subsets
- Return the best subset of attributes
- Apply for each learning algorithm
- Use same search methods as before



Actually, you've already been using wrappers…

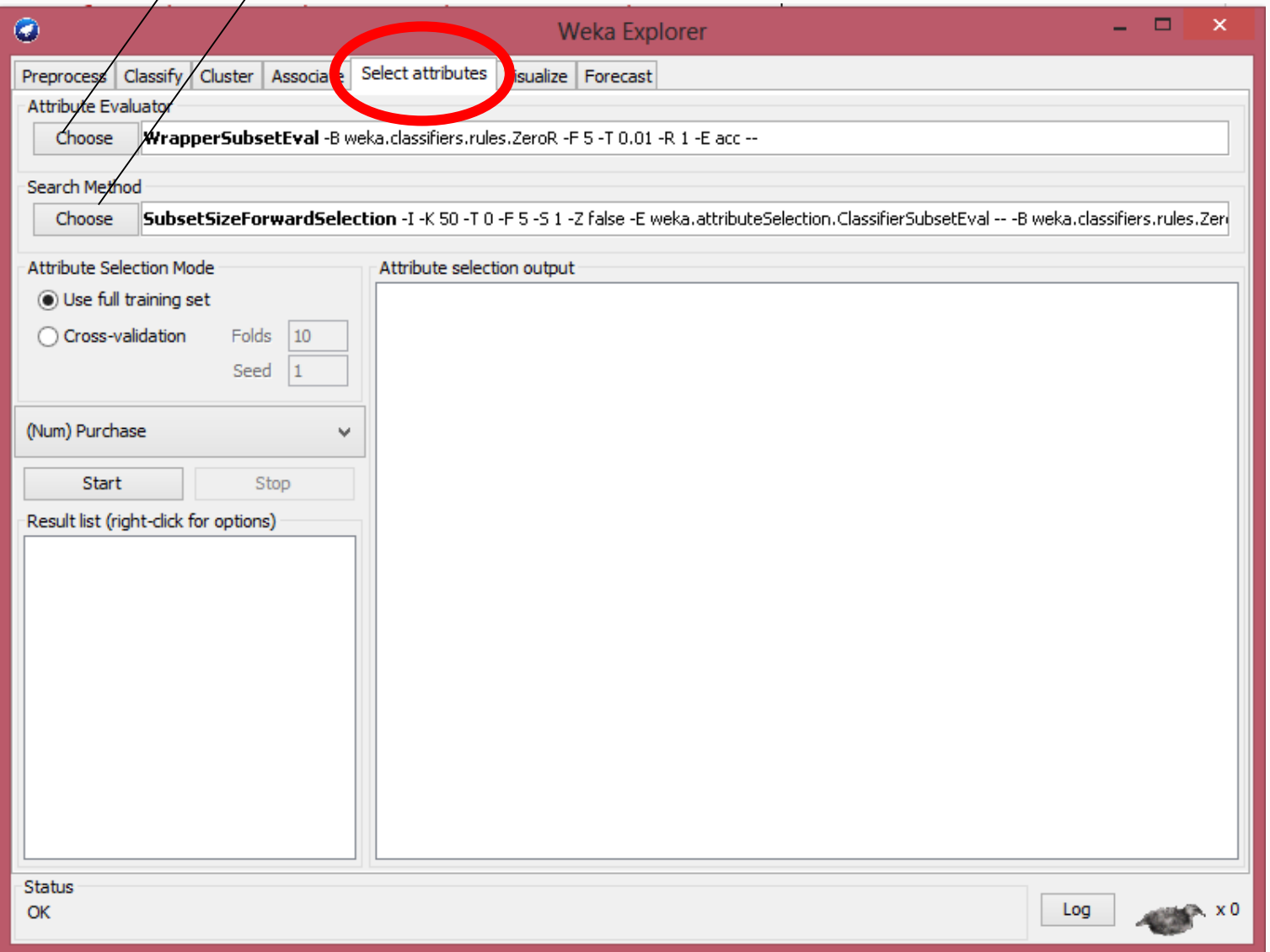
- If no polynomial time algorithm exists to solve a problem it is called *NP-complete*
- Finding the optimal decision tree is an example of a NP-complete problem
- However, ID3 and C4.5 are polynomial time algorithms
  - Heuristic algorithms to construct solutions to a difficult problem;
  - Greedy hill climbing, find the attribute which reduces entropy the most, add to attribute set, repeat…
  - "Efficient" from a computational complexity standpoint but still has a scalability problem;

Tab for selecting attributes in a data set…



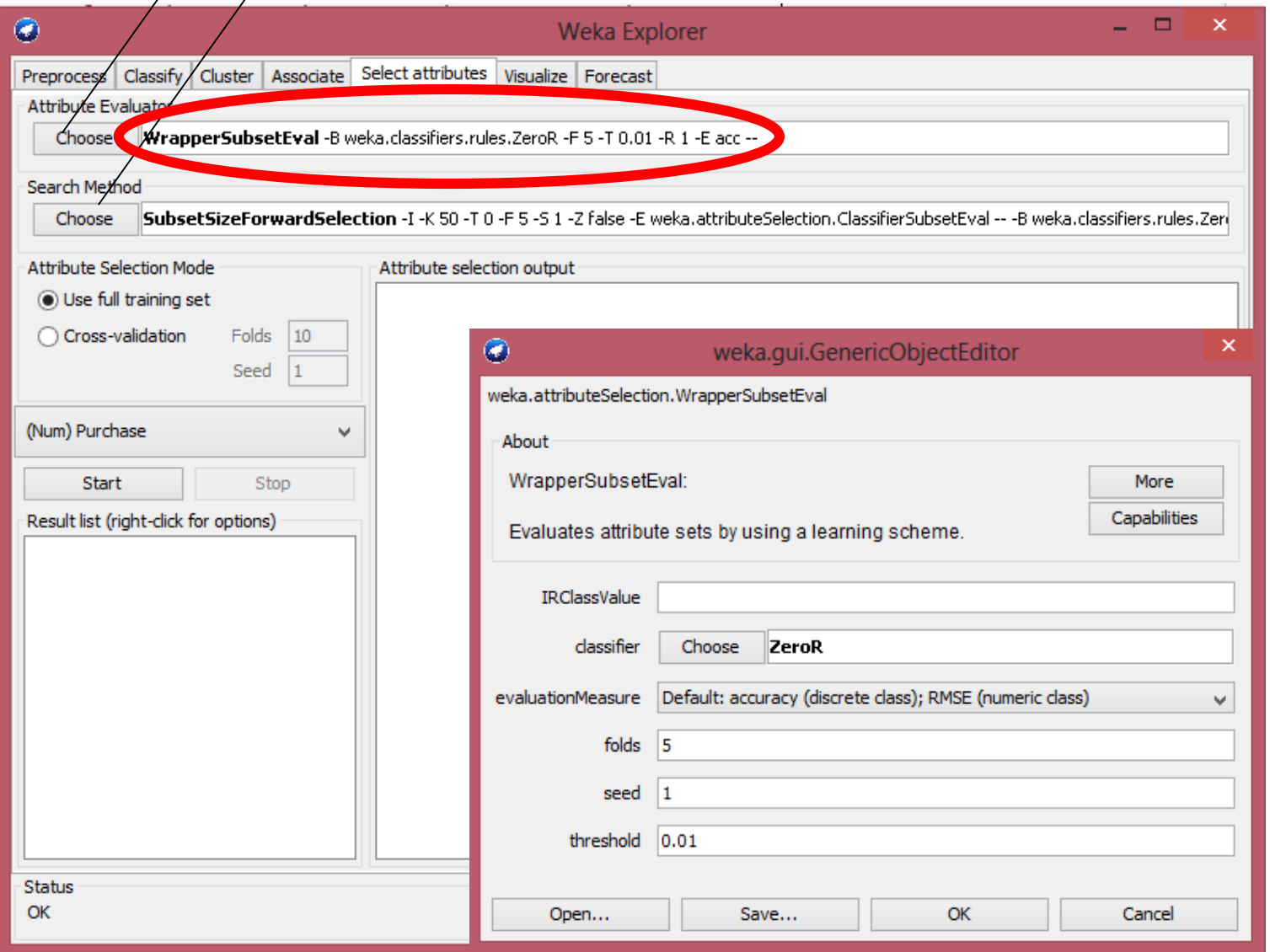
Interface for classes that evaluate attributes…

Interface for **ranking** or **searching** for a subset of attributes…



Interface for classes that evaluate attributes…

Interface for **ranking** or **searching** for a subset of attributes…

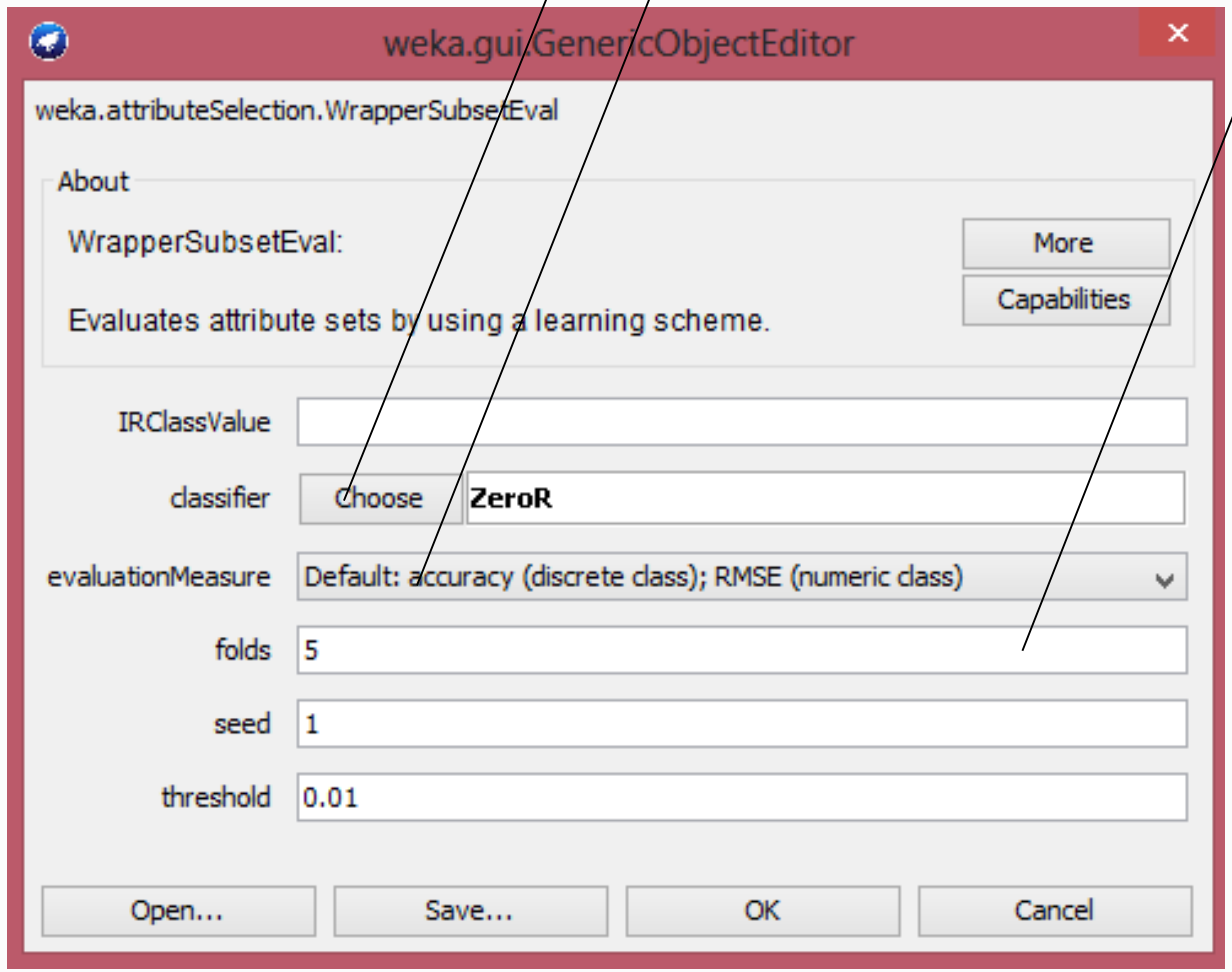


WrapperSubsetEval – evaluates attributes subset using learning scheme.

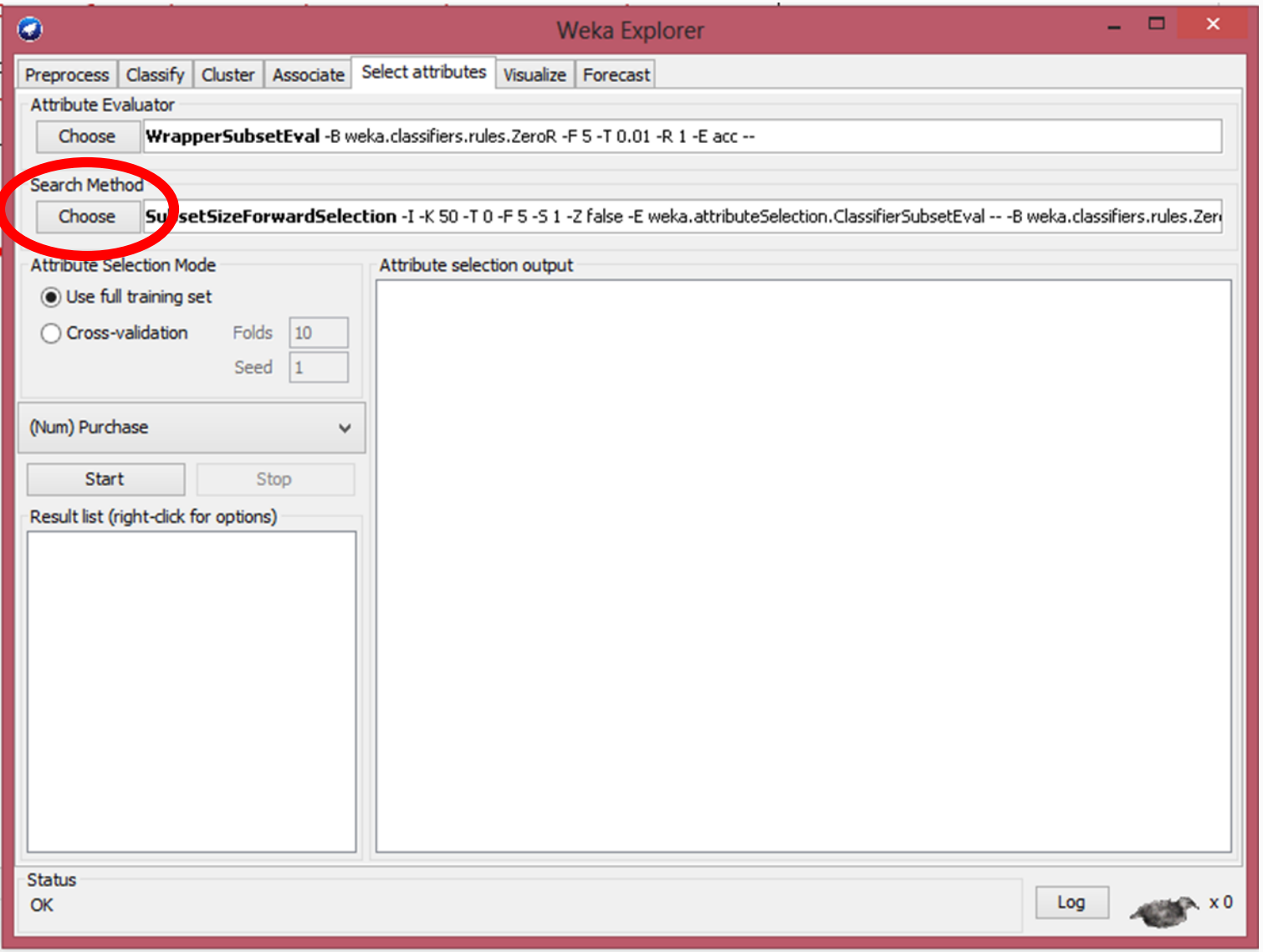
Select and configure ML algorithm…

Accuracy (default discrete classes), RMSE (default numeric), AUC, AUPRC, F-measure (discrete class)

Number of folds to use to estimate subset accuracy



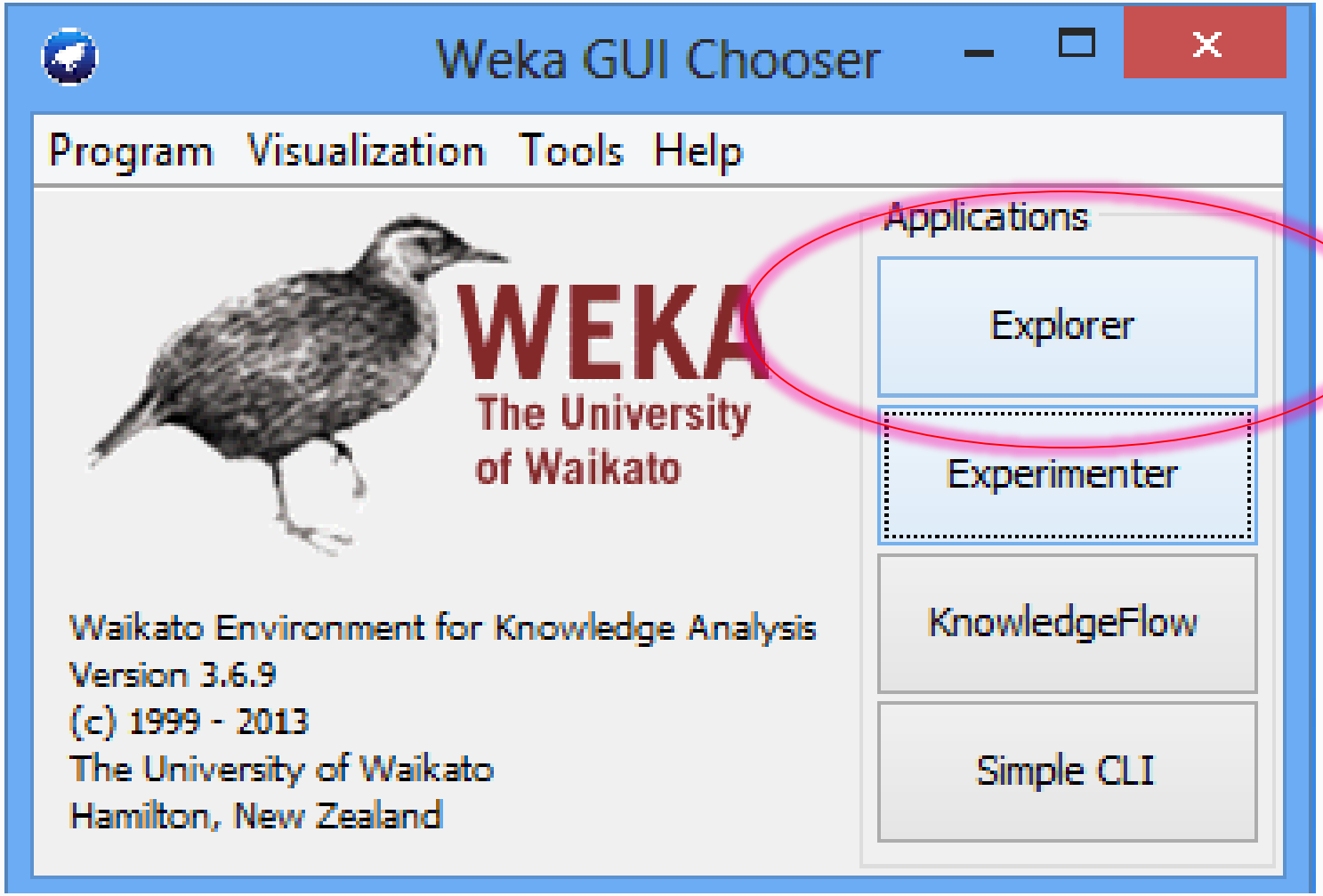
Search Method – interface to specify search algorithm such as BestFirst, ExhaustiveSearch, GeneticSearch, GreedyStepwise, etc.

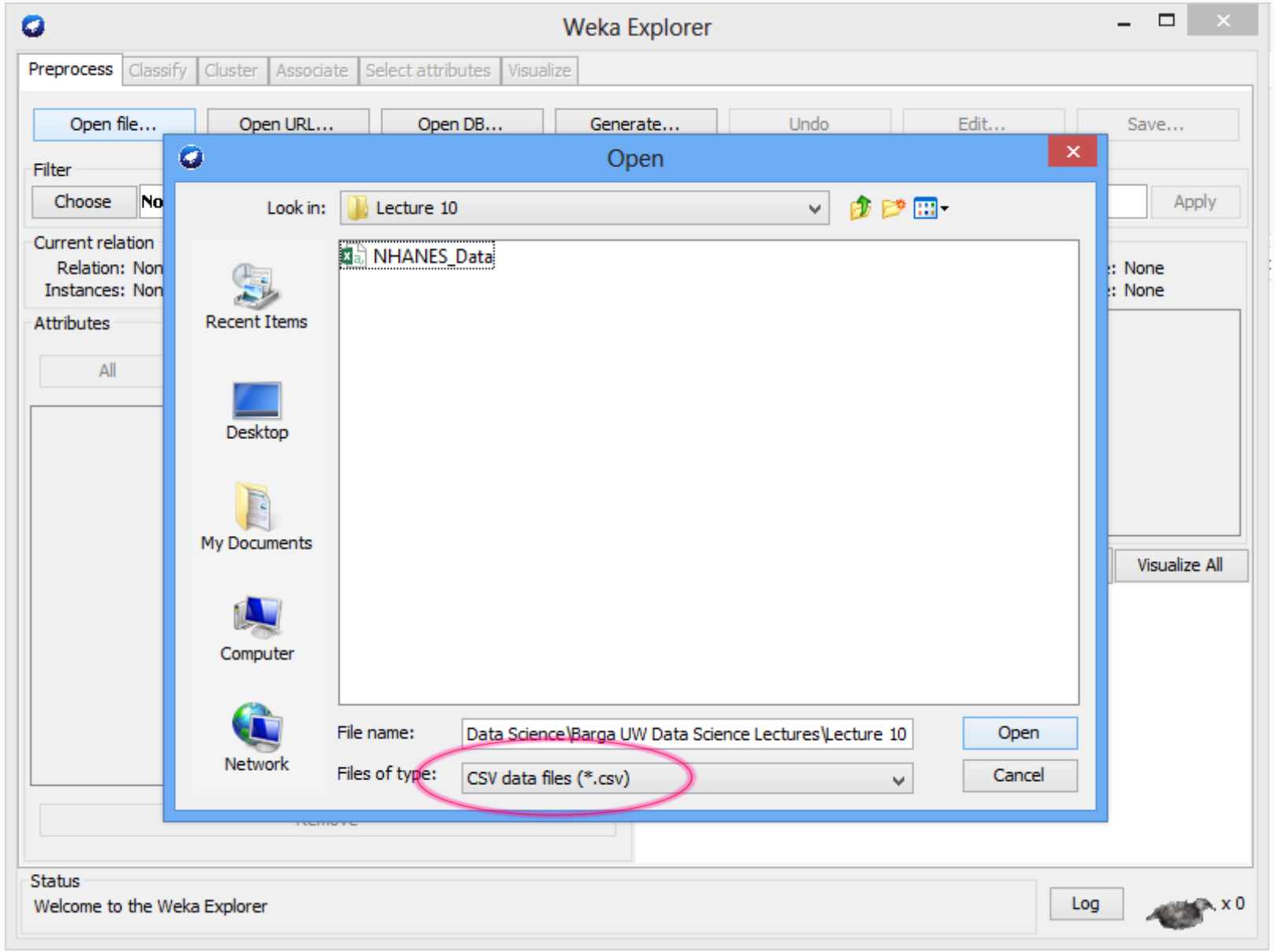


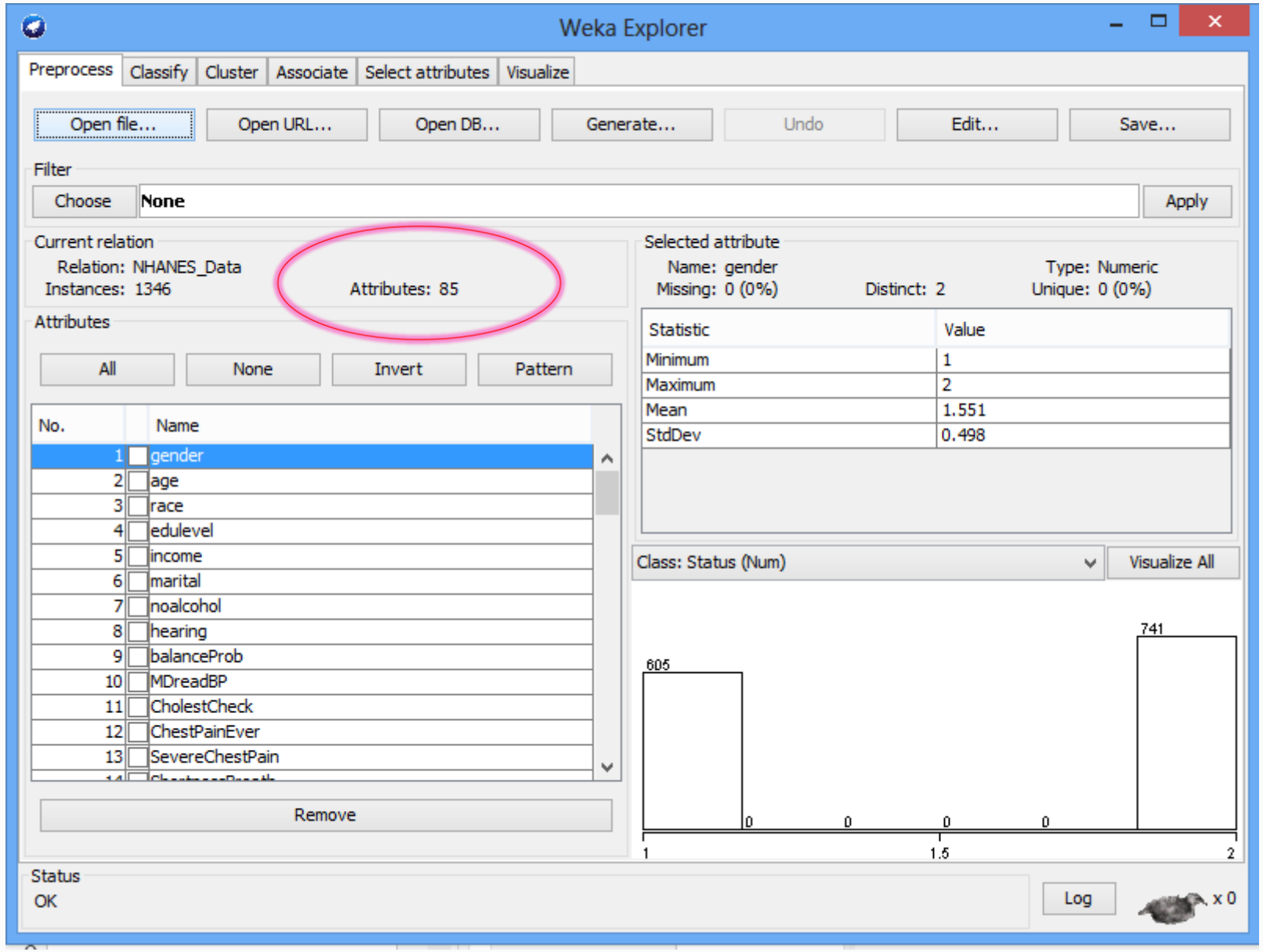
BestFirst: Default search method, it searches the space of descriptor subsets by greedy hill-climbing augmented with a backtracking facility. The BestFirst method may start with the empty set of descriptors and searches forward (default behavior), or starts with the full set of attributes and searches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

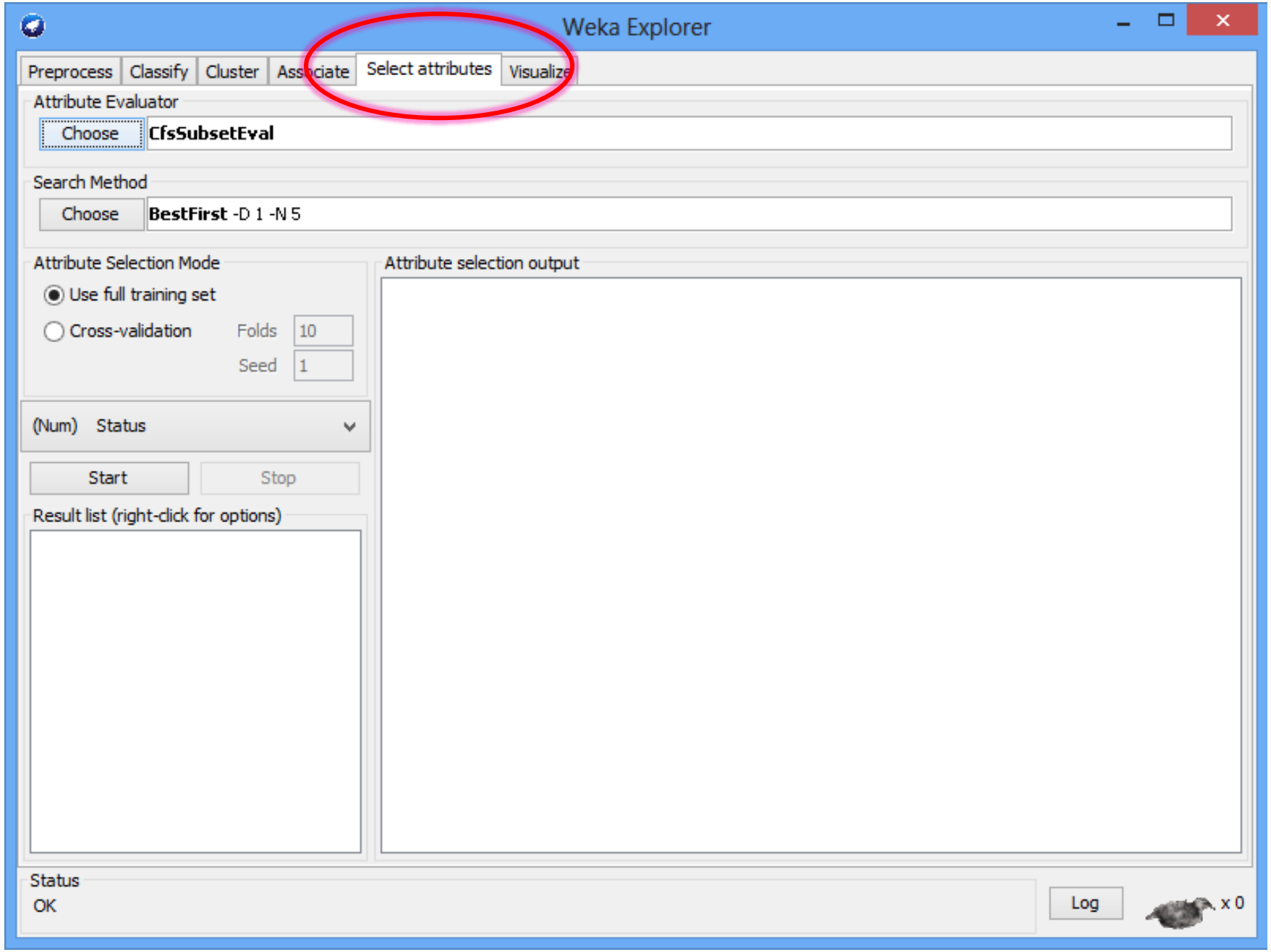
Other options include:

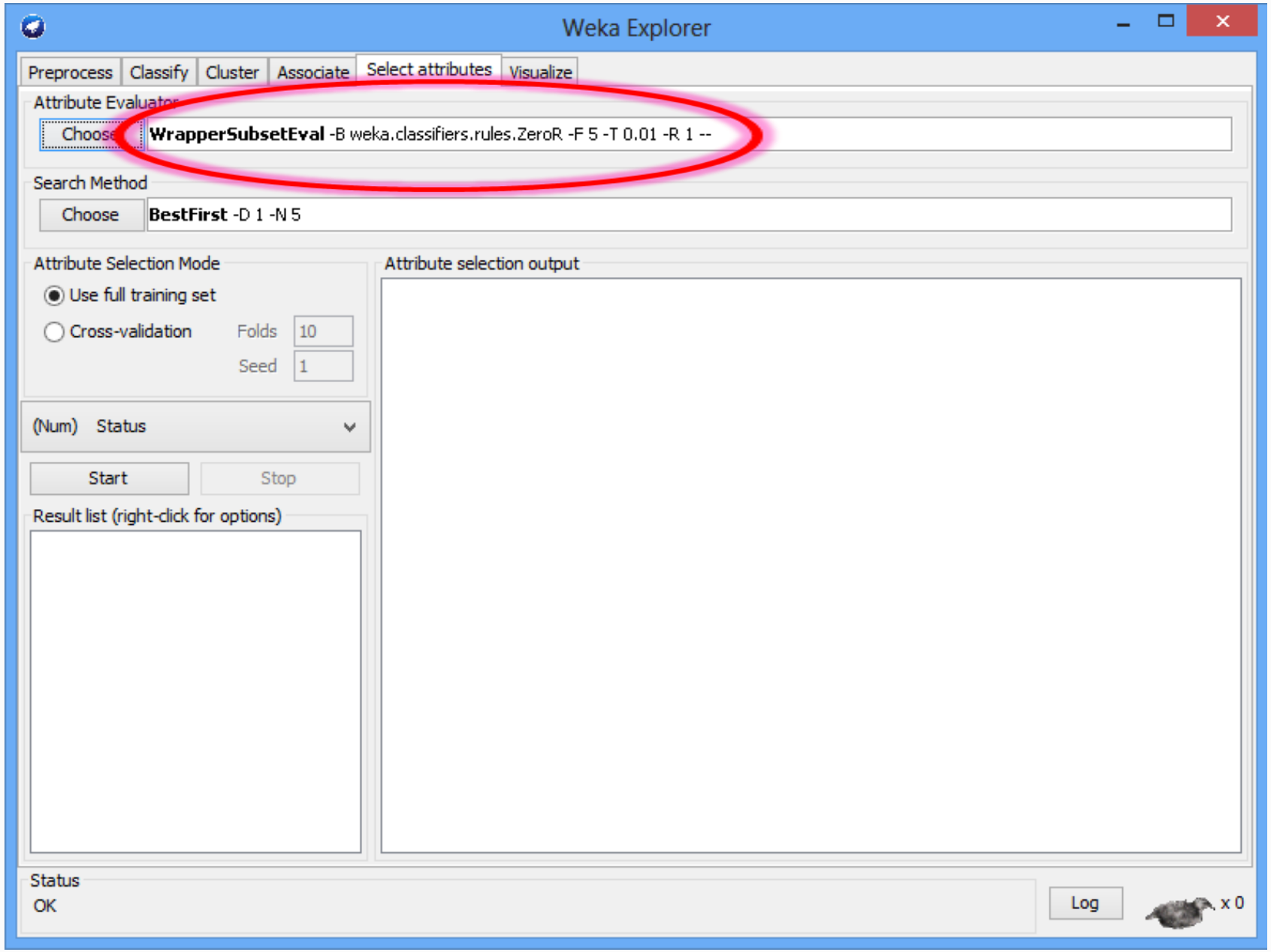
- GreedyStepwise;
- EvolutionarySearch;
- ExhaustiveSearch;
- LinearForwardSearch;
- GeneticSearch (could take hours)

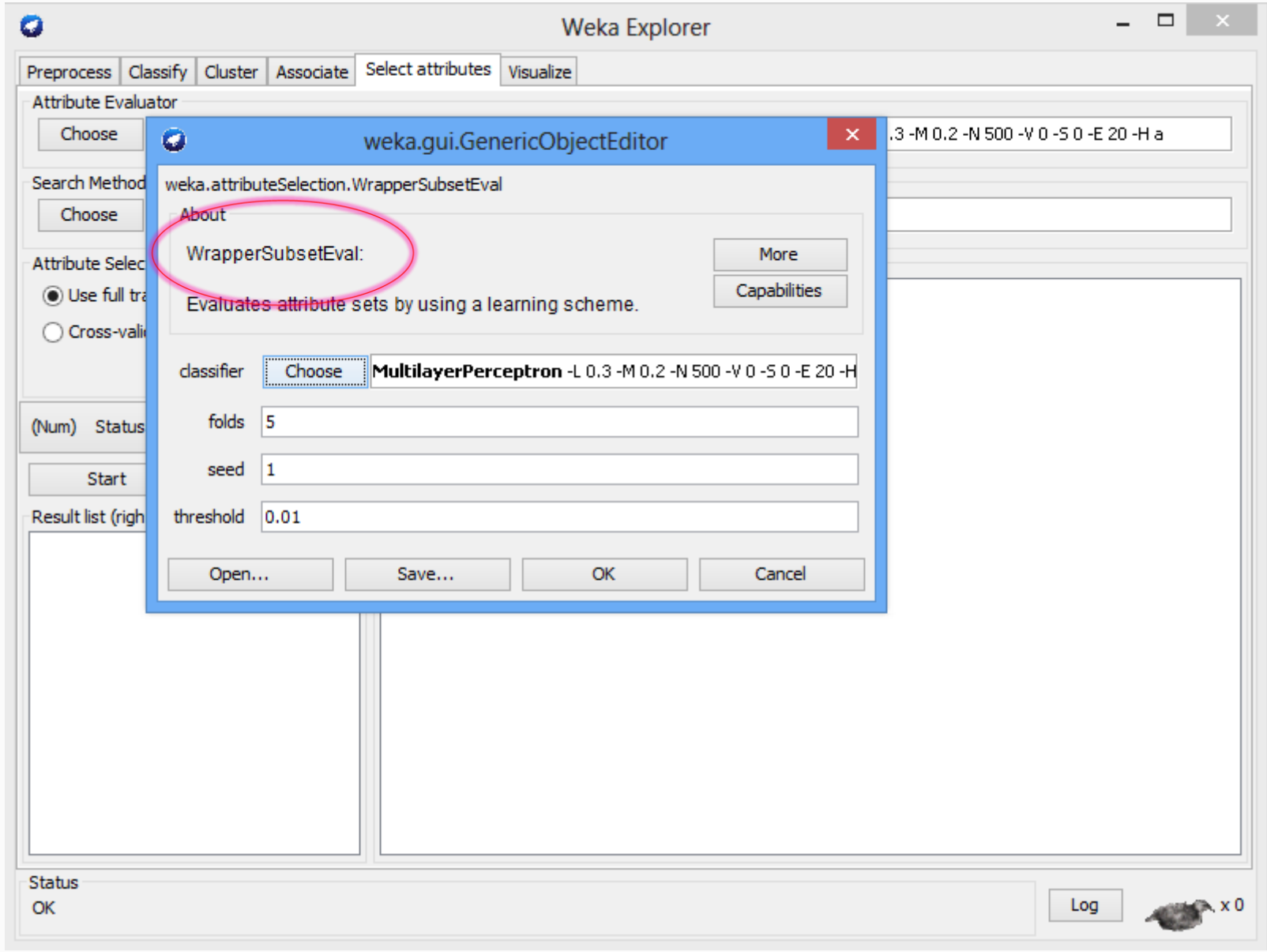

Weka GUI Chooser
Program Visualization Tools Help
WEKA
The University of Waikato
Applications
Explorer
Experimenter
KnowledgeFlow
Simple CLI
Waikato Environment for Knowledge Analysis
Version 3.6.9
(c) 1999 - 2013
The University of Waikato
Hamilton, New Zealand


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
Apply
Current relation
Relation: Non
Instances: Non
Attributes
All
: None
: None
Visualize All
Status
Welcome to the Weka Explorer
Log
x 0
Open
Look in:
Lecture 10
NHANES_Data
Recent Items
Desktop
My Documents
Computer
Network
File name:
Data Science\Barga UW Data Science Lectures\Lecture 10
Files of type:
CSV data files (*.csv)
Open
Cancel

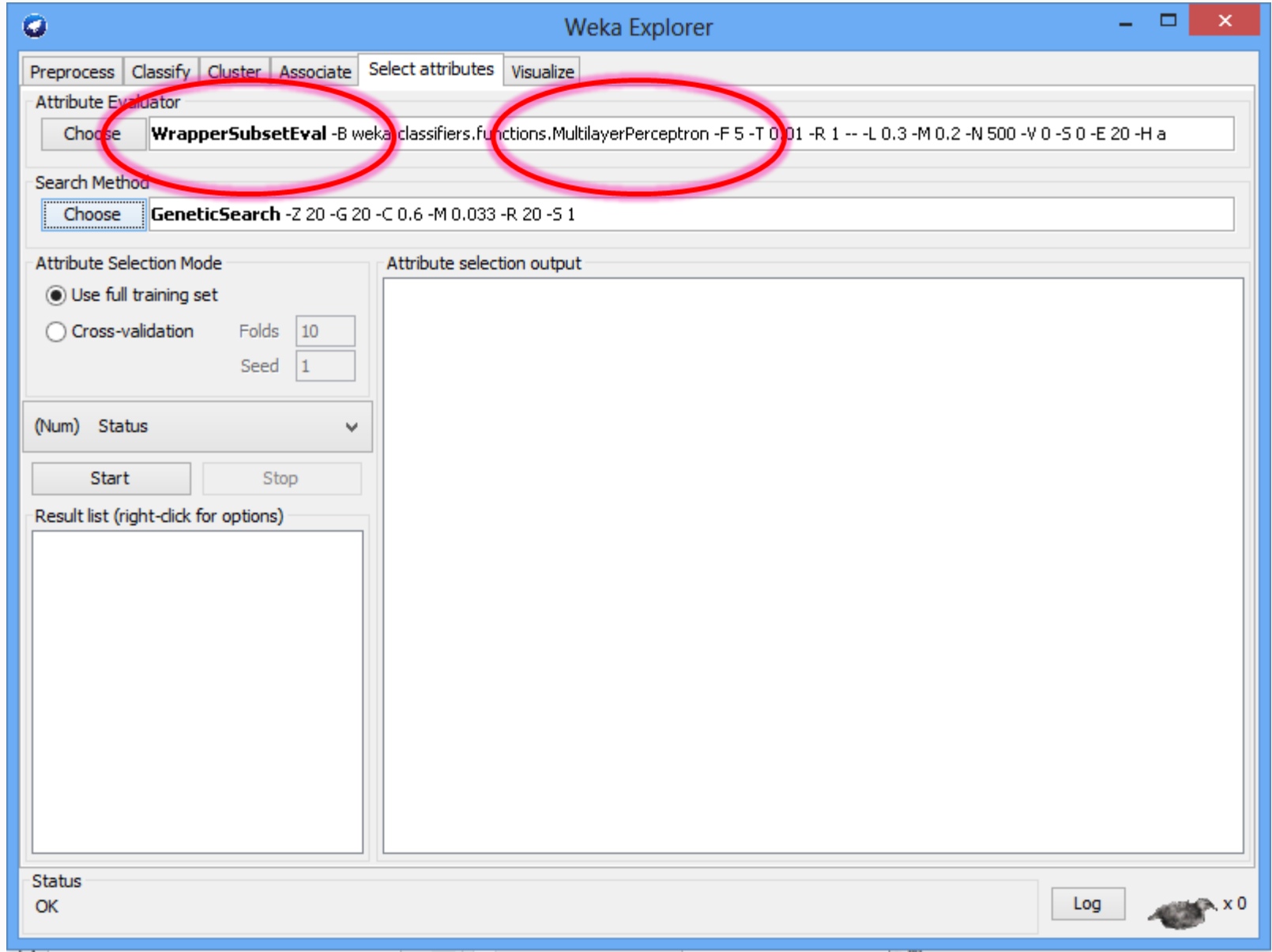

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: NHANES_Data
Instances: 1346
Attributes: 85
Selected attribute
Name: gender
Type: Numeric
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)
Statistic
Value
Minimum
1
Maximum
2
Mean
1.551
StdDev
0.498
Attributes
All
None
Invert
Pattern
No.
Name
1 gender
2 age
3 race
4 edulevel
5 income
6 marital
7 noalcohol
8 hearing
9 balanceProb
10 MDreadBP
11 CholestCheck
12 ChestPainEver
13 SevereChestPain
Remove
Class: Status (Num)
Visualize All
741
605
0
0
0
0
1
1.5
2
Status
OK
Log
x 0

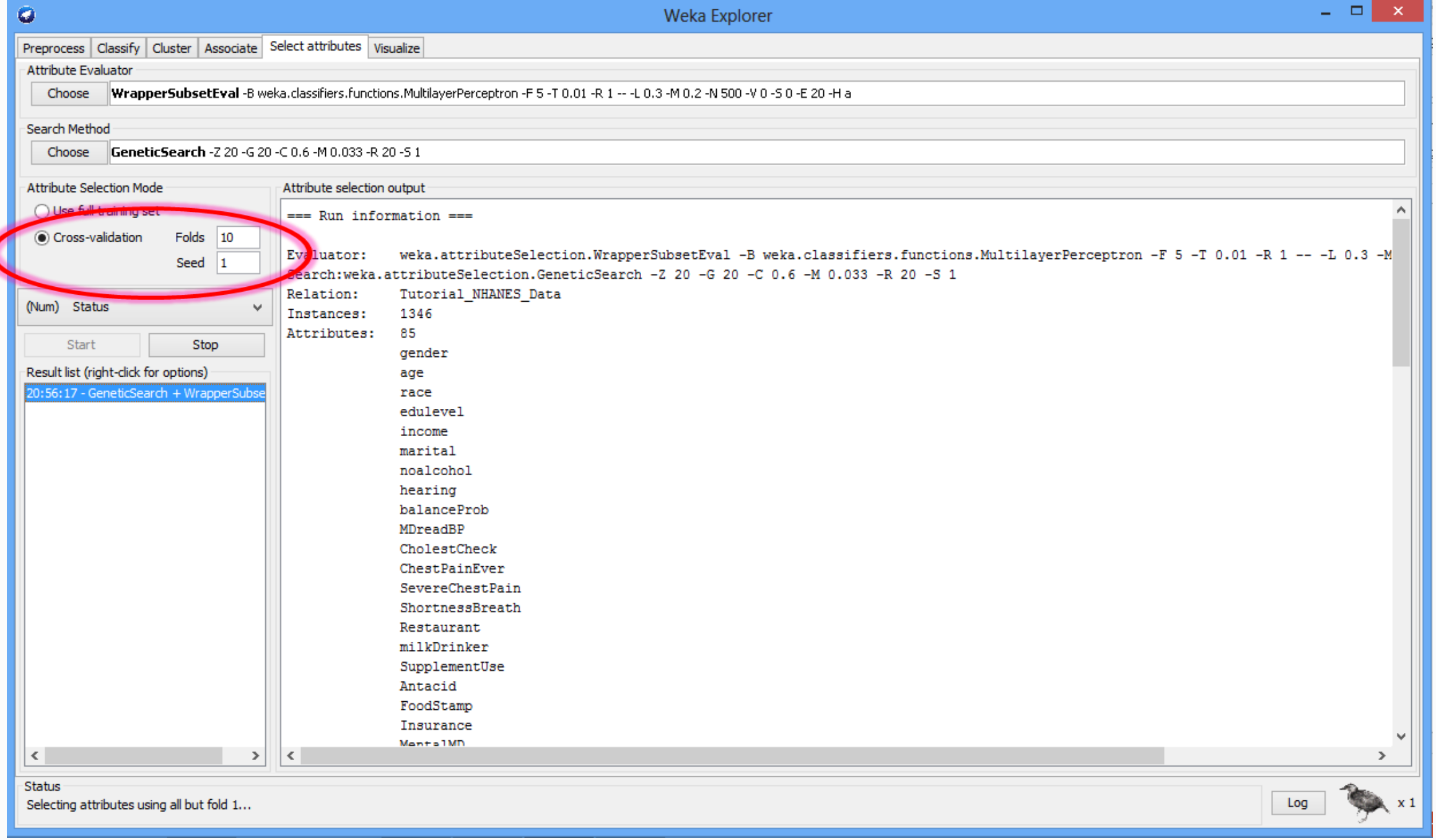

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
CfsSubsetEval
Search Method
Choose
BestFirst -D 1 -N 5
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
Attribute selection output
(Num) Status
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
WrapperSubsetEval -B weka.classifiers.rules.ZeroR -F 5 -T 0.01 -R 1 --
Search Method
Choose
BestFirst -D 1 -N 5
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
Attribute selection output
(Num) Status
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Search Method
Choose
Attribute Selec
Use full tra
Cross-vali
(Num) Status
Start
Result list (righ
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.attributeSelection.WrapperSubsetEval
About
WrapperSubsetEval:
Evaluates attribute sets by using a learning scheme.
More
Capabilities
classifier
Choose
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H
folds
5
seed
1
threshold
0.01
Open...
Save...
OK
Cancel

This will take several hours to complete (~6+)…




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
WrapperSubsetEval -B weka.classifiers.functions.MultilayerPerceptron -F 5 -T 0.01 -R 1 -- -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Search Method
Choose
GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Num) Status
Start
Stop
Result list (right-click for options)
20:56:17 - GeneticSearch + WrapperSubse
Attribute selection output
=== Run information ===
Evaluator: weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.functions.MultilayerPerceptron -F 5 -T 0.01 -R 1 -- -L 0.3 -M
Search:weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1
Relation: Tutorial_NHANES_Data
Instances: 1346
Attributes: 85
gender
age
race
edulevel
income
marital
noalcohol
hearing
balanceProb
MDreadBP
CholestCheck
ChestPainEver
SevereChestPain
ShortnessBreath
Restaurant
milkDrinker
SupplementUse
Antacid
FoodStamp
Insurance
Status
Selecting attributes using all but fold 1...
Log
x 1

10 minute break…

Course Project

Data Science

Deriving Knowledge from Data at Scale

That's all for tonight….

Nov 23rd, 2015