St. Luke's Life Science Institute

**Evidence-Based Medicine and Nursing Workshop**

(Aug 2005)

# Material for the hands-on section

Janez Demsar

Faculty of Computer and Information Science,
University of Ljubljana, Slovenia

This text vaguely follows the CRISP-DM standard (www.crisp-dm.org). It is adapted for the purposes of medical data mining and for the use at the workshop, so the selection of methods and their descriptions is limited in width and depth. The text is includes general guidelines and advices, and exercises for the hands-on section. Most exercises use the Cleveland "heart disease" data (Robert Detrano, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation, downloaded from the UCI ML repository, and edited by Janez Demsar). Other exercises are only for students who brought their own data.

# I. Data loading, checking and preprocessing

## 1. Formatting the data

Typical data mining applications can load data from spreadsheet programs (e.g. Excel), databases (SQL), and various tab- or comma-delimited formats. Orange's support for Excel is being developed and SQL is available only on scripting level for now. Orange Canvas, which we are going to use, supports tab-delimited formats and proprietary formats of several popular machine learning programs.

1. See the documentation for Orange's data formats, particularly tab-delimited files (in Orange Reference Guide, see the page Data Structures / Loading and Saving Data / Tab-delimited files). Observe the file heart_disease.tab with a text editor or Excel. (If you installed Python and Orange at the standard location, the data is in directory c:\Python23\lib\site-packages\orange\doc\datasets\.)

2. If you have brought your own data, convert it to a suitable format for Orange.

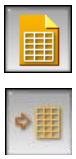
## 2. What are we modeling?

We shall focus on supervised attribute-based data: each data instance is described by a set of attributes (independent variables), and a class (dependent variable). In medicine, the features describe the patient and the class is what we try to predict, e.g. the presence or absence of the disease. If the value to be predicted is continuous, we call this regression. Regression is closer to statistics, but less supported by machine learning and visualization. In statistics we often convert categorical (discrete) variables and outcomes to continuous, while in machine learning we often do the opposite.

3. We shall here limit ourselves to classification data. If you brought your own data set and have a continuous variable to predict, please convert it to a discrete by choosing a suitable cut-off point. (You can use two cut-off points to split your outcomes into three intervals, but this will make the problem harder. For our exercises, I recommend two intervals only.) You can leave the descriptive features continuous.

## 3. Basic examination of the data

4. Load the data into Orange and see whether it looks OK.



5. Check how the missing values are coded. Orange should show them as question marks. People are often sloppy and use several different notations (NA, 99, ?, a blank) in the same file.

6. Check for spelling errors. Orange (and many other tools) is case-sensitive: you should not write the same values sometimes with small and sometimes with capital letters.

7. Get to know the attributes. What do they mean? Is their meaning used consistently? Are the values used consistently?

8. Edit the names and values of the features if necessary. In most data mining packages there is no reason for coded values (e.g. chest pain type being 0 for typical angina, 1 for the atypical, 2 for asymptomatic and 3 for non-anginal, or 0 and 1 for female and male). This only makes the data, graphs and models less readable.

9. Check the attributes. Verify the data types – numerical, discrete, binary – are all types correct? Fix that, if needed (this also goes for those using the heart disease data set, there may be some errors in it, too). For each attribute, observe the distributions, the averages, deviations, value ranges. Are there any obvious mistypes like heart rates of over five hundred? Are there any heavy outliers? If you are an expert on that data, you can observe individual outliers and try to determine if they are true or not, and what to do with them. Do the distributions correspond to the expected ones, e.g. is the average age such as is generally believed for the particular disease? Is the sample a good representative of the modeled population? E.g. if the data contains five times as many female as male, but this is not the property of the modeled population, we have a problem.


## 4. Select the data

We usually need to select a subset of variables and of data instances. It is very important that this step is well-documented and the rationales are given for all decisions.

- Decide which attributes are really important. If you take too many, you are going to have a mess. If you remove too many, you can miss something important.

- You can remove attributes with too many missing values. These will be problematic for many data mining procedures, and are not very useful any way, since they don't give you much data. Also, if they weren't measured when your data was collected, think whether they are going to measure it when your model is used.

- If you have more attributes with a similar meaning, it is often beneficial to keep only one – unless they are measured separately and there is a lot of noise, so you can use multiple attributes to average out the noise. (You can decide later which one you are keeping.)

- If you are building a predictive model, verify which attributes will be available when the prediction needs to be made. Remove the attributes that are only measured later or those whose values depend upon the prediction (for instance, drugs that are administered depending upon the prediction for a cancer recurrence).

Selection of attributes can be done either inside a tool (Orange) or in the file. I recommend using Orange to select the attributes (and the class attribute, if needed), but save the data in a file and use that file later. This way you will not forget which attributes you used in the study.

How do we select attributes?

- Manually. If you know the data, you will be able to tell the useful form the rest.

- Using automated feature selection procedures, data- (chi-square; information gain, ReliefF) or method-based (stepwise logistic regression; wrapper-based selection)

10. If you brought your own data, decide whether there are any attributes or instances which you don't want to use, and remove them. For the sake of the exercise, I recommend keeping a smaller subset of around 10 attributes.


As for data instances, you should – depending on the methods you intend to use – remove the instances with too many missing values, and the instances that for some reason don't belong to your sample, such as patients with side complications unrelated to the disease observed, or obvious outliers. This step should be particularly well-justified and explained in the report.

Instances are sometimes removed or weighted to balance the class distributions. This needs to be done in problems where we are interested in finding the instances of a very small group, such as people with a rare disease. Balancing will decrease the classification accuracy, but possibly increase the sensitivity. Doing this requires a lot of experimenting and tweaking the arguments of the methods.

11. If you have instances with many missing values, remove them as well, but not before formalizing (and writing down) a selection criteria. I recommend doing it outside Orange, although you can use the below widgets as well.

## 5. Data Preprocessing

Any manipulation of the data which is based on the statistical properties of the data should be done without using the data instances intended for testing the derived models. For instance, if missing values are replaced by the averages, the averages should be computed from the instances used for modeling only.

**Missing data.** Knowing why it is missing can help dealing with it. Here are a few things we can do:

- Ask the owner of the data or a domain expert if he can provide either per-instance or general values to impute. Maybe the data was not collected since its value was known (e.g. pulse rate was not recorded when it was normal). Maybe the missing values can be concluded from the other attributes (blood loss was not recorded when there were no wounds – so we should impute 0) or inapplicable (the reason of death is missing, but another attribute tells that the patient survived).

- Leave them missing; this can limit a choice of data mining procedures you can use.

- Impute constant values, such as “the worst” or “the best” possible, or the most common value (for discrete attributes) or the average/median (for the continuous).

- Predict the missing values from other values. You can use small classification trees for that purpose. (Basically, you construct predictive models to predict the “independent” variables.)

My personal experience: keep it simple, try the solutions in the order above. The optimal solution is if an expert can help you, while prediction from other values may work well in theory but is too chanceful in practice.

12. If you brought your own data, think what to do in that regard. If there are not too many missing values, leave them alone for the time being.

**Categorization (discretization).** Some data mining methods are unable to treat numerical attributes. If we want to use them, we must convert them into categorical by defining one or more cutoff points. How do we find them?

- Manually. Use established standards, common sense ... You can also decide based on the data (use a point which yields two intervals with the greatest chi-square difference), but don’t use the test data while doing this (see the remark at the beginning)!

- Simple data-based. For instance, Orange can categorize a variable by splitting its range into a prescribed number of intervals with equal number of examples.

- More complex, like Fayyad-Irani categorization for supervised data. This one can also detect and remove (seemingly) redundant attributes.

Again, the simplest solution usually works best. If there are some established cutoff point(s) for the feature, compare them to what simple techniques would give. If they are similar, use the established ones. Otherwise, you can try both and see which works better. The complex methods may work well in theory, but in practice they are quite often inferior to the simple ones.

13. Instruct Orange to categorize the data, once to four quartiles and once using entropy-based Fayyad-Irani discretization. If this is your data - do you find the proposed intervals sensible? (We shall, however, use continuous attributes in our course; if you have your own data, you don't need to categorize it unless you want to.)


**"Continuization".** This is the opposite of categorization and is required by many statistical methods, such as the popular logistic regression. Each categorical variable is replaced by one or more numerical, usually with values of either 0 or 1. Many packages perform this operation automatically, without the need for user intervention. We will not need this in the today's course.

**Simplifying the variables.** You can often achieve better results by merging several values of a discrete variable into a single one. Sometimes you may also want to simplify the problem buy merging the class values. The data set that we use originally had five values (the severity of the disease) but is traditionally used with a binary outcome (disease present or not).

**Construction of derived variables.** Sometimes it is useful to combine several features into one, e.g. presence of neurological, pulmonary and cardiovascular problems into a general "problems". There can be various reasons for doing this, the simplest being a small proportion of patients which each single disease. More complicated are features that need to be combined because of interactions, e.g. being woman aged around 50 is a special condition (because of menopause), so if menopause is a factor in a disease observed, it is worth combining these two features into a single one. There is a whole area of machine learning that deals with automatic feature construction; this topic is too advanced and too specific, so we will not include it in this course.

# II. Exploration and Visualization

In this phase we use various statistical and visual techniques to get a deeper insight into the data. We shall focus on the latter. Some of the relations which we shall explore using visual devices are more the domain of statistics (for instance, correlations between features), so tools like SPSS or R are more suitable for formal analysis of these relations, while Data Mining tools typically offer easier visual inspection than statistical packages.

In graphs that do not show individual instances (e.g. represented as points) but only agglomerated results, like Sieve Diagrams and Mosaic Plots, you should carefully check the number of instances and take them as an intuitive measure of the relation's significance.

14. Observe the distributions of values for each feature and the corresponding probability of a target class. What are the effects of individual values? Are they consistent with your or a domain expert's prior knowledge? If not, why? (Contradiction can often be explained by small samples.)

For the heart disease data, check:

- what is the effect of age,
- who among the patients in this study is at a greater risk – women or men,
- explain the seemingly peculiar effect of the cholesterol,
- which kind of chest pain is the most dangerous.

15. Use scatter plot to observe the relations between features. Are there any correlated values, values that exclude each other ...

For heart disease data:

- interpret the graph showing the relation between the maximal achieved heart rate and age,
- observe the graph with exercise induced angina and gender
- do you know for a (medical? sociological?) explanation for the seeming relation between gender and the number of major vessels colored?

16. In the scatterplot, you can use VizRank for finding interesting projections. (It will not find anything really interesting on the heart disease data, but you can try it on your own data, if you brought it.)

17. Use the Sieve Diagram to observe the relation between the chest pain and exercise induced angina. You can also look for other interesting relations (use Calculate chi squares). Note two things: Sieve diagram cannot show continuous features, but it can show the class on one axis.

18. Plot a Sieve Multigram and explore the strongest discovered relations using the Sieve Diagram.

19. Try to find some interesting relations between pairs of features and the outcome using the Mosaic Display. (Hint: try and comment thal vs. chest pain or exercise induced angina vs. gender, or exercise induced angina and slope peak exc. ST.) You can also try more than two attributes (try exc. ind. angina, slope peak and gender).

20. Find some interesting projections using RadViz and VizRank. Do general positions of the classes agree with what you know/expect?

21. You can also try whether FreeViz gives a good separation and sensible results for your data. (RadViz does not correctly treat multivalued discrete attributes: this is one of the cases where you need to continuize them. The widget "Continuize" is one of the widget in the experimental section "Others".)

22. PolyViz is similar to RadViz, can in principle give better separations, but is more complicated to interpret. Try whether it tells you anything useful about the data which you haven't learned from other visualizations.

23. Now for a test of understanding and skill: use any widgets you find suitable to answer the following questions:
    - Females have relative little exercise induced angina – this is more typical for men, and this combination (male and exercise induced angina) is not good. True or not?
    - Are there any features in which women and men strongly differ in this data?
    - What kind of chest pain is much more typical for patients with thal=reversable defect than for other patients? Is having this kind of chest pain a good or a bad sign for them?
    - There are kinds of check pain which seem to be a good indication in a combination with normal thal, but less so (or even bad) with abnormal thal. Which are these? How many patients are there in the latter groups?

The relations you discover like this can only be statistically tested on a separate data set. As mentioned in the lecture, you shouldn't test too many hypotheses. Alternatively to statistics, you can judge about a found relation based on its unexpectedness and the number of patients for which it holds.

This phase can help you reconsider your selection of attributes and remove some if you have too many. You shouldn't be too radical here, though: simple visual analysis cannot always show the potential use of the attributes in a complex model, such as a classification tree.

# III. Modeling

We need modeling for two purposes:

- to gain new knowledge between the predicted value and the predictors
- to design a predictive model for use in practice

The first aim requires that the induced model is understandable and interpretable by human. This therefore excludes the black-box models like neural networks, support vector machines and k-nearest neighbours. The second purpose can be served by black-box models, but they are usually unable to present the arguments for the prediction and are thus less desired.

Statistical models are more rigid and make various assumptions about the data. Machine learning is more relaxed, but gives no guarantee about the validity of the results. Any kind of modeling must therefore be followed by evaluation of the results using separate data sets and domain background knowledge.

In medicine we are often also considered about the probabilities of the outcomes. Most modeling techniques can be extended to give such probabilities.

# 1. Linear models

Linear models are models that are specified by a hyperplane in the features space which separates the instances of two (or more) classes. The distance from the plane can be used for predicting probabilities of classes. Examples of such methods are logistic regression, naive Bayesian classifier and linear support vector machines. They differ in the way they define the optimal hyperplane. We shall only observe the first two, logistic rergession and naive Bayesian classifier.

A common feature of linear models is that the prediction is made by summing the evidence provided by individual features. Each value of each feature can be assigned a certain number of points for or against the target class, and the sum of corresponding values for each patient can be translated into probabilities. This way of thinking is supposedly close to how physicians subconsciously sum up the evidence.

**Logistic regression** comes from statistics. All features must be numeric; the binary can easily be treated as numeric, while the multivalued need to be converted by a suitable method – usually first into binary and then treated as numeric. Logistic regression doesn't handle unknown values and can only predict binary outcomes. Various implementations of logistic regression can, however, solve these problems automatically, without user's intervention.

**Naive Bayesian classifier** is a method from machine learning that is to some degree equivalent to logistic regression. The resulting model can be written in the same form as that of logistic regression if the features and the outcome are binary. It can handle unknown values and multivalued outcomes.

Mathematically, naive Bayesian classifier is in general equivalent to a sum of univariate logistic regression models. From this stems the most important difference between the two methods: naive Bayesian classifier is susceptible to correlated features. If some features are correlated, the probability predictions can be badly calibrated, yet it seems that class predictions themselves do not suffer considerably. Each individual feature's impact on the prediction, however, is computed correctly.

On the other hand, logistic regression does not suffer for malcalibrated probability prediction because it can compensate for the correlated features. The side-effect is, however, that the features are not treated individually so the estimates for the impact of each single feature cannot always be trusted (since this is a statistical method, it of course provides the confidence intervals which are in such cases considerably wider).

Both models (as well as other linear separators) can be presented in form of a nomogram. There are some minor differences between the nomograms for logistic regression and naive Bayesian classifier which stem from the differences in the two methods.

24. Construct a naive Bayesian classifier for the heart disease data set and observe the corresponding nomogram. (Align the features to the zero point and use 2D representation for continuous attributes.)

- Which two features have (in overall) the greatest impact on the outcome?
- What are the worst and the best predictors, considering a good outcome?
- Is it better to be a female or be young (e.g. 30 years)?
- Which forms of slope peak exc. ST predict good outcome?
- Which forms of chest pain predict a good outcome? Order them in order of importance and use the Distributions widget to check if the two widgets agree.[1]
- Does atypical anginal pain have a similar effect on the prediction as the typical anginal pain? Or is it more similar to some other value of this attribute? If we decide to use naive Bayesian model, we can merge the feature values with very similar effects to a single feature (e.g. have only two or three different types of chest pain instead of two). This would be useful when we have too few instances with each particular value. Find the right widget to check whether this is also the case here.
- The younger you are, the better: while the prior probability of the bad outcome is 46%, the probability of such outcome for younger patients is smaller and for older patients it's higher. Which is the "neutral age" at which the age does not change probability?
- Observe the curve for cholesterol: it is bent. What does a bent curve mean? Is it true in this case, or is it just a random effect? (Hint: instruct the Nomogram widget to show confidence intervals and a histogram with a max size of 30.) Can you imagine circumstances in which we would expect a bent curve?

25. If you have brought your own data, try a similar exercise with it. How well do results correspond to your (and to the general) knowledge about the domain?

26. Draw a logistic regression nomogram. Which of the above questions can you answer? Which of the questions you can't and why? Which answers are the same for naive Bayesian classifier and for logistic regression?[2]

27. If you have your own data as well, see what the logistic regression model for it looks like. Which of the two models makes more sense?

28. Imagine some random data for a patient and make a prediction for her or him, using both models. Do they differ? Why – which of the values you chose have the most different effect on the prediction? (Also try this with your data if you have it.)

---

[1] Should they? Are the probabilities that the naive Bayesian classifiers use the same as those shown by the Distributions widget? If you think the answer is yes, check the curves for the continuous attributes. Are they the same in both widgets?

[2] Orange seems to have a (new) problem with using a stepwise selection of attributes for logistic regression on some kinds of data. Please leave this option unchecked.

29. Using a naive Bayesian nomogram, investigate whether predictive factors for women are the same as for men. For a more drastic example, observe the difference in nomograms for groups of patients with different values of thal. Observe the relation between thal and the features whose importance changes most drastically in the scatter plot.


## 2. Classification trees

Induction of classification trees is a part of the golden repertoire of machine learning methods. The learning algorithm recursively divides the data into ever smaller subsets, using the most "informative" feature as a split criterion. The procedure stops when a subset is too small or when it is pure enough (ideally, all instances belong to the same class). The result of the algorithm is a tree with split criteria (e.g. age>50 or gender=male) in internal nodes and class predictions and probabilities in the leaves. We usually also post-prune the tree using a procedure which goes from the leaves to the root and for each split estimates whether the classification accuracy would be better with or without it. It the latter is the case, the split is removed and the node becomes a leaf. (Details of these algorithms are beyond the scope of this course.)

When making predictions, an instance is assigned to a leaf through checking its values for the criteria on the path.

Classification trees can work with numeric and discrete (binary or multivalued) features, they can handle multivalued outcomes. A similar method, regression trees, constructs trees for predicting numeric outcomes.

Trees should not be too large. What is "too large" depends upon the complexity of the domain, but the upper limit for medical domains is probably around 10 splits. On the other side, trees with as few as 3–4 splits often make considerably good predictions.

The advantage of classification trees is a clear representation of the model. They are also (usually) able to make predictions using only a small number of features, thus not a lot of things need to be measured. Trees are not useful for examining the effects of a large number of features, since only a few are actually used in the model. Since all instances that are classified to the same leaf are predicted the same probability, classification trees have rather low AUCs due to a high number of ties and can also be hard to fit for the desired sensitivity and specificity (see the section on model evaluation).

30. Familiarize yourself with the options for constructing decision trees. Induce a tree in which you want the features to be binarized (e.g. values of multivalued features are always separated into two groups; the optimal separation is computed for each split separately). We want a minimum of 10 instances in each leaf and don't want to split the nodes with less than 15 instances or nodes in which the majority class

reached 95 %. Leaves should be recursively merged and m for pruning should be set to 15.

Connect the tree to the Classification Tree Viewer widget and inspect it. If you brought your own data set, decide whether the tree makes sense to you or not. Try changing the options and see how they affect the result. (Please don't set the Attribute Quality Estimation to ReliefF since it temporary doesn't work.)

31. Try Classification Tree Viewer 2D instead of the textual Classification Tree Viewer. Observe the visualization options, e.g. Baseline for Line Width or Tree Node Color.

Also, when the tree is constructed, Orange allows us to select instances from one tree node (leaf or an internal node) and analyze its instances using any visualization or modeling method.

32. Connect the Attribute Statistics widget to the output of the tree viewer (one of the two above). Choose an attribute, e.g. thal. Now click on the nodes of the tree (I suggest you first try the root and then its children) and observe how the distributions change.


33. Take a scatter plot, feed the entire data set to the slot Examples and the data from the tree viewer to the slot Example Subset. Now you can select nodes from the tree and see where the corresponding examples lie in the scatter plot. For beginning, try to visualize the same attributes which the tree split criteria is based on.

34. In a similar fashion, you can connect any other widget that accepts data to the output of the tree viewer. You can even connect learners. For instance, try connecting a naive Bayesian learner and the nomogram widget to the output of the tree viewer. Now arrange the screen so that the tree viewer and the nomogram are visible; click on the nodes of the tree and observe the nomogram.




In the previous section, the nomogram showed the importance of features for the entire sample (and, hopefully, the entire population). With this combination of widgets we can observe the importance of features for the patients that belong to individual tree node.

Orange offers a rather unique functionality: manual construction of classification trees aided by the whole set of visualization widgets. It is also possible to combine the manual and automatic construction of the trees: trees can be constructed automatically and then refined (usually pruned) manually, according to our background knowledge. Or, we can construct the first few nodes manually and then see what continuation is proposed by Orange. We can keep it if we like it, or discard it if we don't.

35. Try it. Connect the manual tree builder with various visualization widgets and construct your own tree. If you do this on a problem you

really know (that is, if you brought your own data set), then you will probably first let the algorithm build the entire tree automatically, and then manually prune or replace some nodes.

### 3. CN2 Rules

This is another popular machine learning method. It constructs a model in form of if-then rules; the if-part gives a set of conditions and the then-part is the prediction. CN2 rules have been extended to handle many different feature types beyond the numeric and discrete.

36. Construct and observe a list of CN2 Rules. You can connect the output of the viewer widget to various visualization widgets and explore the properties of the subgroups discovered the algorithm. If you brought your own data set (or when you use this algorithm in practice), you can easily check whether the rules make sense or not.


## IV. Model evaluation

The golden rule of all science – from physics to medicine – is that a hypothesis should never be tested on the same data from which they were formulated. For models found by machine learning algorithms this is especially important since these algorithms are nothing more then automatic procedures for finding the most probable hypothesis, the hypothesis that is the most "provable" with the data.[3] It is easy to show that machine learning algorithms are quite capable of finding the statistically very significant hypotheses even on completely random data – unless the hypotheses are verified on a separate data set.

You have three basic options regarding the model evaluation:

- Treat the model as a hint about the relations in the data. This doesn't allow for any stronger conclusions than "the model suggests that the age is an important predictor for female, less so for male". You can then go on and look for deeper medical explanation for the phenomenon; you can, for instance, discover that the actual problem is not the age itself, but the menopause. With such a setup, the model cannot be the "end product", but only a hunch for you.

- Use a repetitive sampling schema such as cross-validation, leave-one-out or bootstrap. The most popular in machine learning is cross-validation, which divides the data into a prescribed number of folds (usually 10), and repeats ten tests in which

[3] The situation with statistical models is somewhat different: they make certain assumptions about the nature of the data (like normality of distributions of feature values or their independence) and can based on that assess the statistical validity of the model. While this is sounder than the ad-hoc statistical procedures, the problem is that the assumptions usually cannot be verified.

one fold is left out, the others are used for learning and the one the was left out is used for testing. The problem with cross-validation is that it can (and often does) build ten quite different models, so the computed score does not measure the quality of a certain model but the quality of the modeling algorithm. When you afterwards use the complete data set to construct a model for publication and use, you can only hope that its performance is going to be the same.

- Use a separate test data set. You can simulate a real-world situation and, if you have the data that was collected over multiple years, put the last year away for a while, use the rest of the data for training and then check how the constructed models perform on the last year's data. The other possibility is to randomly select a certain percentage of the instances (usually 30%) and set them aside to be used for testing.[4]

The right way for doing predictive data mining in medicine – if you want to publish a (tested) model – is to first set a part of the data aside (we shall refer to this as the "test data", while the remaining data is "training data"). You should do that before looking at the data. Then you can test various modeling algorithms, subsets of features, subsets of instances etc. For validating them, you can use the cross validation on the train data alone (that is, the cross validation will split the training data into multiple folds etc.). Finally, when you decide for the optimal subset of features and the modeling algorithm (or, possibly, a few of them), use the whole training data to induce the final model. You then publish this model and assess its performance on the test data. This way of doing things is most likely to get your paper published without raising methodological concerns.

37. Split the data in proportion 30:70 and show the two subsets in two Data Table widgets. Check whether the numbers of examples are correct (the number of examples is shown at the upper left corner).

When you do it for real, you should save both tables into separate files to ensure the reproducibility of your results. Try it now!

Orange provides the widgets for the two operations: one that can (beside other things) be used for splitting the data into the test and train data set, and the other that can be used for cross-validation.

There are several measures for evaluation of models:

- classification accuracy (CA) is the proportion of correctly classified instances in test data or the average over the test folds in cross-validation;
- sensitivity is the proportion of correctly recognized positive examples and specificity is the proportion of correctly recognized negative examples;

[4] As a side note, the "heart disease" data set was compiled in Cleveland, but the models were tested on the data from Budapest, Zurich and Basel, to check whether this kind of exchange of models between institutions is possible. They discovered that the model tended to be biased towards different classes in different institutions, but the overall experiment was successful.

- concordance index (CI) is the probability that from a pair of two examples – one positive and one negative – the algorithm would correctly recognize which is which (i.e. assign the positive one a greater probability of being positive). Concordance index is equivalent to the area under ROC curve (see below) and to the Wilcoxon-Mann-Whitney statistics;

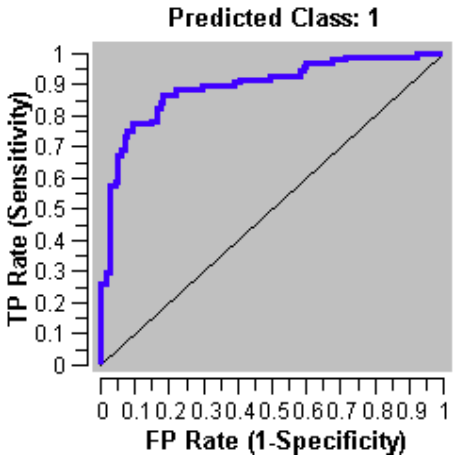
- Brier score is the average square distance between the correct and the given prediction of probabilities. For a contrast from the above measure, lower Brier scores are better than higher.

Brier score is the least known of the four, but useful since it measures the calibration of probability predictions. CI measures the discriminative power and the method's ability to correctly "order" the examples; for instance, in triage the model with the highest CI would perform well with respect to taking the most critically injured first.

38. Take the four learning algorithms we learnt about (naive Bayesian classifier, logistic regression, classification trees and CN2), the Majority classifier (which always classifies into the majority class, thus providing a baseline), and test them using cross-validation on training data. Based on these results alone, which algorithm would you use if the goal is to be able to decide which patients of a pair is at a greater risk? (For this exercise, there is a complete schema at the end of this text, but first try to solve it yourself.)

Sensitivity and specificity depend upon the chosen cut-off point: if we classify an instance as positive if its probability of being positive is above 20% (instead of 50 %), we will increase the sensitivity at the cost of lower specificity. Since the two measures are related, it only makes sense to observe them at the same time. The ROC (receiver-operator characteristic) curve plots 1–specificity (also known as the false positive rate) against sensitivity (true positive rate).

In the ROC curve on the right, the diagonal represents the basic behavior (a random classifier) and the blue line is the curve for ROC for logistic regression using cross-validation on the training data set. For instance, if we want a sensitivity of 0.7 (that is, if our goal is to detect 70% of the patients at the high risk), the specificity will be around 1–0.0=0.95. The optimal curve is the one that would go through the upper left corner.




The area under that curve is equal to the concordance index. The optimal curve thus has a concordance index of 1.

39. If we are choosing between naive Bayesian classifier, logistic regression and classification tree, and we want to achieve a minimal

false positive rate at the requested 90 % true positive rate: which classifier are we going to choose and what is the expected false positive rate? (You can use the schema from the previous exercise: connect the ROC widget to the output of Test Learners and disable the unneeded curves in the ROC widget's tab "General".)

There are other curves similar to ROC, for instance the lift chart. It originates from marketing: if we sort the potential customers by classifier's probability predictions and contact them in that order, the curve shows how the number of "positive customers" rises with the number of contacted customers (actually, the x-axis shows the probability predicted by the classifier, not the number of customers contacted). In medical terms, this can be the number of the patient for which we detect the disease as a function of the threshold probability.

40. Imagine that we are conducting a study for which we need a group of 70 positive instances. To find them, we are going to take all the patients which the naive Bayesian classifier predicts as positive with a probability higher than a certain threshold. Find the correct threshold, using only the training data!


41. If we use the threshold found in the previous exercise, how many of the test data set instances with the predicted probability above this threshold are positive? Since this data set is smaller, we do not expect the same number, 70, but only 30 instances of the positive class. (Try solving the problem yourself, and then look at the schema at the end of this document for a hint.)

Calibration curve tells us how well the classifiers probabilities are calibrated. Ideally, if classifier predicts 40 % chance of a certain outcome, than 4 out of 10 such patients should have this outcome. The calibration curve is a curve with the predicted probability on x-axis and the true probability on y-axis. The optimal curve goes diagonally from the bottom left to the upper right corner.

42. Which of the classifiers we play with has the best calibrataion?


Almost invariably, logistic regression achieves the best calibration since the method itself is designed to optimize the calibration – that's how logistic regression works. If it is the probability prediction you care about, and if you have not problems with the shortcomings of logistic regression, this is the method of choice.

Beside the statistical evaluation of classifiers, it is often instructive to observe which patients get misclassified. If we are constructing the model manually or the modeling method allows for other kinds of manual interventions, this can help us enhance the model.

43. Who has a better chance to be correctly classifier using naive Bayesian classifier – men or women? (Hint: use two Attribute Statistics widgets, one to show the distribution of gender in the entire data set and one in

the misclassified instances. Then compare the number to see the proportion of misclassified women/men.)

44. In excercise 35 we manually constructed a classification tree. Construct it again, using only the training data set. While doing it, observe the performance of the model in the Test Learners widget, and try to determine which patients get misclassified using other widgets, such as the Scatter plot.

When modeling is finished, the resulting models can be tested on the left-out test data set.

45. Take the tree from the previous exercise, a logistic regression learner, naive Bayesian learner and automatically constructed tree. Now test all models on the test data set: use Test Learners, but give it the training data to the slot Data and test data to Separate Test Data. In the widget itself, check “Test on test data”. Use another Test Learners widget to see the results of the models on a cross-validation on the train data set. How different are the numbers?

# V.Model Deployment

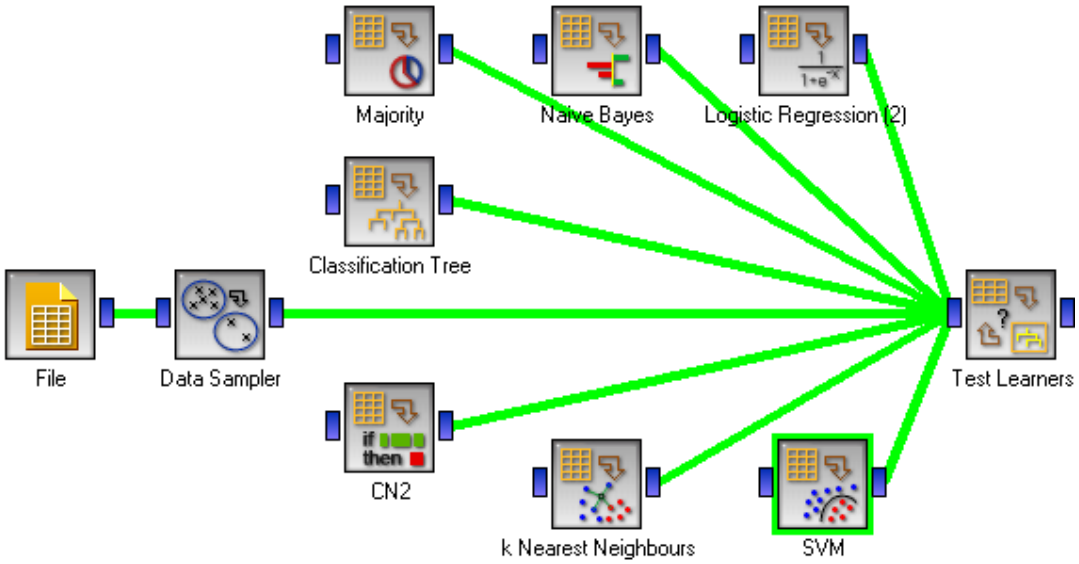
Data mining is a pointless exercise if its results are not used. If the modeling was done only to gain some new insights in the problem, then statistical tests on independent data and publishing a paper or a report is all we need. If the goal was to induce a predictive model, then the model should be put in a form that is useful in practice.

Classification trees are easy to understand and can be given to physicians in a printed form. We can also transform them to a set of rules, similar to the popular “Wizards” in the modern computer interfaces. CN2 rules already come in form of rules, but since the rules can overlap, they are less useful for manual use.

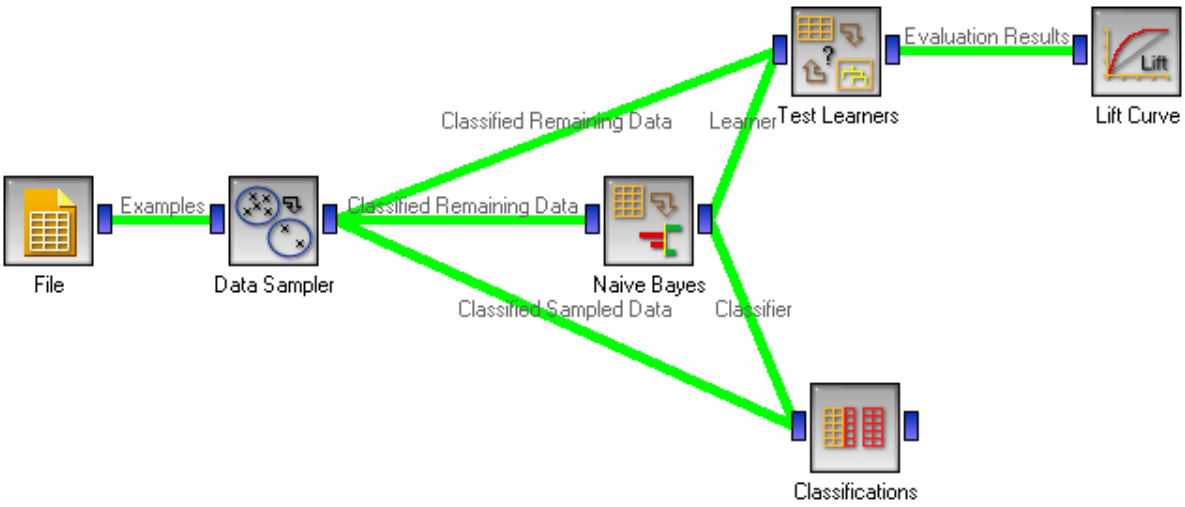
Naive Bayesian classifiers and logistic regression models can be published in form of nomograms. If the points are given in -100 to 100 scale, not as odds ratios, the points for individual features can easily be summed and converted to the probability prediction manually. The physician who uses nomograms can also “explain” the decision in terms of features which “vote” for and against a certain outcome.

If the number of features used for making prediction is small enough (less than 5) and the features are binary or at most ternary, the model can be represented with a lookup table which gives the prediction for each specific case. In such cases, these models are easier to use than trees, rules or nomograms, but they offer no explanation for the decision.

The abovementioned methods, as well as all other predictive models, even the black-box neural networks and support vectors, can be deployed by programming a user interface that run on a handheld or a mobile. Programs that run on a standalone machines are less suitable, especially in situations like field triage. However, web-based expert models are also becoming increasingly popular.




Schema for exercise 38




Schema for exercise 41. Note that the Naive Bayes widget outputs a learner (an algorithm, not a model!) to the Test Learners and a classifier, trained on the training data to the Classifications. The latter widget gets and shows the test data instances as classified by the classifier trained on the train data. To answer the question, we should instruct Classifications to show probabilities for the positive class and sort the instances according to this column. Then we can simply count the positive instances.