RAPID|MINER

USER MANUAL

rapid-i
REPORT THE FUTURE

RapidMiner 5.0

Manual

Rapid-I
www.rapid-i.com

Contents

1 Motivation and Fundamental Terms

In this chapter we would like to give you a small incentive for using data mining and at the same time also give you an introduction to the most important terms. Whether you are already an experienced data mining expert or not, this chapter is worth reading in order for you to know and have a command of the terms used both here and in RapidMiner.

1.1 Coincidence or not?

Before we get properly started, let us try a small experiment:

- Think of a number between 1 and 10.
- Multiply this number by 9.
- Work out the checksum of the result, i.e. the sum of the numbers.
- Multiply the result by 4.
- Divide the result by 3.
- Deduct 10.

The result is 2.

Do you believe in coincidence? As an analyst you will probably learn to answer this question in the negative or even do so already. Let us take for example what is probably the simplest random event you could imagine, i.e. the toss of a coin.

"Ah" you may think, "but that is a random event and nobody can predict which side of the coin will be showing after it is tossed". That may be correct, but the fact that nobody can predict it does in no way mean that it is impossible in principle. If all influence factors such as the throwing speed and rotation angle, material properties of the coin and those of the ground, mass distributions and even the strength and direction of the wind were all known exactly, then we would be quite able, with some time and effort, to predict the result of such a coin toss. The physical formulas for this are all known in any case.

We shall now look at another scenario, only this time we can predict the outcome of the situation: A glass will break if it falls from a certain height onto a certain type of ground. We even know in the fractions of the second when the glass is falling: There will be broken glass. How are we able to achieve this rather amazing feat? We have never seen the glass which is falling in this instant break before and the physical formulas that describe the breakage of glass are a complete mystery for most of us at least. Of course, the glass may stay intact "by chance" in individual cases, but this is not likely. For what it's worth, the glass not breaking would be just as non-coincidental, since this result also follows physical laws. For example, the energy of the impact is transferred to the ground better in this case. So how do we humans know what exactly will happen next in some cases and in other cases, for example that of the toss of a coin, what will not?

The most frequent explanation used by laymen in this case is the description of the one scenario as "coincidental" and the other as "non-coincidental". We shall not go into the interesting yet nonetheless rather philosophical discussions on this topic, but we are putting forward the following thesis:

The vast majority of processes in our perceptible environment are not a result of coincidences. The reason for our inability to describe and extrapolate the processes precisely is rather down to the fact that we are not able to recognise or measure the necessary influence factors or correlate these.

In the case of the falling glass, we quickly recognised the most important characteristics such as the material, falling height and nature of the ground and can already estimate, in the shortest time, the probability of the glass breaking by analogy reasoning from similar experiences. However, it is just that we cannot do with the toss of a coin. We can watch as many tosses of a coin as we like; we will never manage to recognise the necessary factors fast enough and extrapolate them accordingly in the case of a random throw.

So what were we doing in our heads when we made the prediction for the state of the glass after the impact? We measured the characteristics of this event. You could also say that we collected data describing the fall of the glass. We then reasoned very quickly by analogy, i.e. we made a comparison with earlier falling glasses, cups, porcelain figurines or similar articles based on a similarity measure. Two things are necessary for this: firstly, we need to also have the data of earlier events available and secondly, we need to be aware of how a similarity between the current and past data is defined at all. Ultimately we are able to make an estimation or prediction by having looked at the most similar events that have already taken place for example. Did the falling article break in these cases or not? We must first find the events with the greatest similarity, which represents a kind of optimisation. We use the term "'optimisation"' here, since it is actually unimportant whether we are now maximising a similarity or the sales figures of one enterprise or any other - the variable concerned, so similarity here, is always optimised. The analogy reasoning described then tells us that the majority of glasses we have already looked at broke and this very estimation then becomes our prediction. This may sound complicated, but this kind of analogy reasoning is basically the foundation for almost every human learning process and is done at a staggering speed.

The interesting thing about this is that we have just been acting as a human data mining method, since data analysis usually involves matters such as the representation of events or conditions and the data resulting from this, the definition of events' similarities and of the optimisation of these similarities.

However, the described procedure of analogy reasoning is not possible with the toss of a coin: It is usually insufficient at the first step and the data for factors such as material properties or ground unevenness cannot be recorded. Therefore we cannot have these ready for later analogy reasoning. This does in no way mean however that the event of a coin toss is coincidental, but merely shows that we humans are not able to measure these influence factors and describe the process. In other cases we may be quite able to measure the influence factors, but we are not able to correlate these purposefully, meaning that computing similarity or even describing the processes is impossible for us.

It is by no means the case that analogy reasoning is the only way of deducing forecasts for new situations from already known information. If the observer of a falling glass is asked how he knows that the glass will break, then the answer will often include things like "every time I have seen a glass fall from a height of

more than 1.5 metres it has broken". There are two interesting points here: The relation to past experiences using the term "always" as well as the deduction of a rule from these experiences:

If the falling article is made of glass and the falling height is more than 1.5 metres, then the glass will break.

The introduction of a threshold value like 1.5 metres is a fascinating aspect of this rule formation. For although not every glass will break immediately if greater heights are used and will not necessarily remain intact in the case of lower heights, introducing this threshold value transforms the rule into a rule of thumb, which may not always, but will mostly lead to a correct estimate of the situation. Instead of therefore reasoning by analogy straight away, one could now use this rule of thumb and would soon reach a decision as to the most probable future of the falling article. Analogy reasoning and the creation of rules are two first examples of how humans, and also data mining methods, are able to anticipate the outcome of new and unknown situations.

Our description of what goes on in our heads and also in most data mining methods on the computer reveals yet another interesting insight: The analogy reasoning described does at no time require the knowledge of any physical formula to say why the glass will now break. The same applies for the rule of thumb described above. So even without knowing the complete (physical) description of a process, we and the data mining method are equally able to generate an estimation of situations or even predictions. Not only was the causal relationship itself not described here, but even the data acquisition was merely superficial and rough and only a few factors such as the material of the falling article (glass) and the falling height (approx. 2m) were indicated, and relatively inaccurately at that.

Causal chains therefore exist whether we know them or not. In the latter case we are often inclined to refer to them as coincidental. And it is equally amazing that describing the further course is possible even for an unknown causal chain, and even in situations where the past facts are incomplete and only described inaccurately.

This section has given you an idea of the kind of problems we wish to address in this book. We will be dealing with numerous influence factors, some of which can only be measured insufficiently or not at all. At the same time there are

often so many of these factors that we risk losing track. In addition, we also have to deal with the events which have already taken place, which we wish to use for modelling and the number of which easily goes into millions or billions. Last but not least, we must ask ourselves whether describing the process is the goal or whether analogy reasoning is already sufficient to make a prediction. And in addition, this must all take place in a dynamic environment under constantly changing conditions - and preferably as soon as possible. Impossible for humans? Correct. But not impossible for data mining methods.

1.2 Fundamental Terms

We are now going to introduce some fundamental terms which will make dealing with the problems described easier for us. You will come across these terms again and again in the RapidMiner software too, meaning it is worth becoming acquainted with the terms used even if you are an experienced data analyst.

First of all we can see what the two examples looked at in the previous section, namely the toss of a coin and the falling glass, have in common. In our discussion on whether we are able to predict the end of the respective situation, we realised that knowing the influence factors as accurately as possible, such as material properties or the nature of the ground, is important. And one can even try to find an answer to the question as to whether this book will help you by recording the characteristics of yourself, the reader, and aligning them with the results of a survey of some of the past readers. These measured reader characteristics could be for example the educational background of the person concerned, the liking of statistics, preferences with other, possibly similar books and further features which we could also measure as part of our survey. If we now knew such characteristics of 100 readers and had the indication as to whether you like the book or not in addition, then the further process would be almost trivial. We would also ask you the questions from our survey and measure the same features in this way and then, for example using analogy reasoning as described above, generate a reliable prediction of your personal taste. “Customers who bought this book also bought...”. This probably rings a bell.

1.2.1 Attributes and Target Attributes

Whether coins or other falling articles or even humans, there is, as previously mentioned, the question in all scenarios as to the characteristics or features of the respective situation. We will always speak of **attributes** in the following when we mean such describing factors of a scenario. This is also the term that is always used in the RapidMiner software when such describing features arise. There are many synonyms for this term and depending on your own background you will have already come across different terms instead of "attribute", for example

- Characteristic,
- Feature,
- Influence factor (or just factor),
- Indicator,
- Variable or
- Signal.

We have seen that description by attributes is possible for processes and also for situations. This is necessary for the description of technical processes for example and the thought of the falling glass is not too far off here. If it is possible to predict the outcome of such a situation then why not also the quality of a produced component? Or the imminent failure of a machine? Other processes or situations which have no technical reference can also be described in the same way. How can I predict the success of a sales or marketing promotion? Which article will a customer buy next? How many more accidents will an insurance company probably have to cover for a particular customer or customer group?

We shall use such a customer scenario in order to introduce the remaining important terms. Firstly, because humans are famously better at understanding examples about other humans. And secondly, because each enterprise probably has information, i.e. attributes, regarding their customers and most readers can therefore relate to the examples immediately. The attributes available as a minimum, which just about every enterprise keeps about its customers, are for example geographical data and information as to which products or services the customer has already purchased. You would be surprised what forecasts can be

made even from such a small amount of attributes.

Let us look at an (admittedly somewhat contrived) example. Let us assume that you work in an enterprise that would like to offer its customers products in future which are better tailored to their needs. Within a customer study of only 100 of your customers some needs became clear, which 62 of these 100 customers share all the same. Your research and development department got straight to work and developed a new product within the shortest time, which would satisfy these new needs better. Most of the 62 customers with the relevant needs profile are impressed by the prototype in any case, although most of the remaining participants of the study only show a small interest as expected. Still, a total of 54 of the 100 customers in the study said that they found the new product useful. The prototype is therefore evaluated as successful and goes into production - now only the question remains as to how, from your existing customers or even from other potential customers, you are going to pick out exactly the customers with whom the subsequent marketing and sales efforts promise the greatest success. You would therefore like to optimise your efficiency in this area, which means in particular ruling out such efforts from the beginning which are unlikely to lead to a purchase. But how can that be done? The need for alternative solutions and thus the interest in the new product arose within the customer study on a subset of your customers. Performing this study for all your customers is much too costly and so this option is closed to you. And this is exactly where data mining can help. Let us first look at a possible selection of attributes regarding your customers:

- Name
- Address
- Sector
- Subsector
- Number of employees
- Number of purchases in product group 1
- Number of purchases in product group 2

The number of purchases in the different product groups means the transactions in your product groups which you have already made with this customer in the

past. There can of course be more or less or even entirely different attributes in your case, but this is irrelevant at this stage. Let us assume that you have the information available regarding these attributes for every one of your customers. Then there is another attribute which we can look at for our concrete scenario: The fact whether the customer likes the prototype or not. This attribute is of course only available for the 100 customers from the study; the information on this attribute is simply unknown for the others. Nevertheless, we also include the attribute in the list of our attributes:

- *Prototype positively received?*
- Name
- Address
- Sector
- Subsector
- Number of employees
- Number of purchases in product group 1
- Number of purchases in product group 2

If we assume you have thousands of customers in total, then you can only indicate whether 100 of these evaluated the prototype positively or not. You do not yet know what the others think, but you would like to! The attribute "prototype positively received" thus adopts a special role, since it identifies every one of your customers in relation to the current question. We therefore also call this special attribute **label**, since it sticks to your customers and identifies them like a brand label on a shirt or even a note on a pinboard. You will also find attributes which adopt this special role in RapidMiner under the name "label". The goal of our efforts is to fill out this particular attribute for the total quantity of all customers. We will therefore also often speak of **target attribute** in this book instead of the term "label". You will also frequently discover the term goal variable in the literature, which means the same thing.

1.2.2 Concepts and Examples

The structuring of your customers' characteristics by attributes, introduced above, already helps us to tackle the problem a bit more analytically. In this way we ensured that every one of your customers is represented in the same way. In a certain sense we defined the type or **concept** "customer", which differs considerably from other concepts such as "falling articles" in that customers will typically have no material properties and falling articles will only rarely buy in product group 1. It is important that, for each of the problems in this book (or even those in your own practice), you first define which concepts you are actually dealing with and which attributes these are defined by.

We implicitly defined above, by indicating the attributes name, address, sector etc. and in particular the purchase transactions in the individual product groups, that objects of the concept "customer" are described by these attributes. Yet this concept has remained relatively abstract so far and no life has been injected into it yet. Although we now know in what way we can describe customers, we have not yet performed this for specific customers. Let us look at the attributes of the following customer for example:

- *Prototype positively received: yes*
- Name: *Müller Systemtechnik GmbH*
- Address: *Meisenstr. 7, Böblingen*
- Sector: *Sector*
- Subsector: *Pipe bending machines*
- Number of employees: $>$ *1000*
- Number of purchases in product group 1: *5*
- Number of purchases in product group 2: *0*

We say that this specific customer is an **example** for our concept "customer". Each example can be characterised by its attributes and has concrete values for these attributes which can be compared with those of other examples. In the case described above, Müller Systemtechnik GmbH is also an example of a customer who participated in our study. There is therefore a value available for our target

attribute "prototype positively received?". Müller Systemtechnik was happy and has "yes" as an attribute value here, thus we also speak of a **positive example**. Logically, there are also negative examples and examples which do not allow us to make any statement about the target attribute.

1.2.3 Attribute Roles

We have now already become acquainted with two different kinds of attributes, i.e. those which simply describe the examples and those which identify the examples separately. Attributes can thus adopt different roles. We have already introduced the role "label" for attributes which identify the examples in any way and which must be predicted for new examples that are not yet characterised in such a manner. In our scenario described above the label still describes (if present) the characteristic of whether the prototype was received positively.

Likewise, there are for example roles, the associated attribute of which serves for clearly identifying the example concerned. In this case the attribute adopts the role of an identifier and is called **ID** for short. You will find such attributes identified with this role in the RapidMiner software also. In our customer scenario, the attribute "name" could adopt the role of such an identifier.

There are even more roles, such as those with an attribute that designates the weight of the example with regard to the label. In this case the role has the name **Weight**. Attributes without a special role, i.e. those which simply describe the examples, are also called **regular** attributes and just leave out the role designation in most cases. Apart from that you have the option in RapidMiner of allocating your own roles and of therefore identifying your attributes separately in their meaning.

1.2.4 Value Types

As well as the different roles of an attribute there is also a second characteristic of attributes which is worth looking at more closely. The example of Müller Systemtechnik above defined the respective values for the different attributes, for example "Müller Systemtechnik GmbH" for the attribute "Name" and the value "5" for the number of past purchases in product group 1. Regarding the attribute "Name", the concrete value for this example is therefore random free text to a

certain extent; for the attribute “number of purchases in product group 1” on the other hand, the indication of a number must correspond. We call the indication whether the values of an attribute must be in text or numbers the **Value Type** of an attribute.

In later chapters we will become acquainted with many different value types and see how these can also be transformed into other types. For the moment we just need to know that there are different value types for attributes and that we speak of value type **text** in the case of free text, of the value type **numerical** in the case of numbers and of the value type **nominal** in the case of only few values being possible (like with the two possibilities “yes” and “no” for the target attribute). Please note that in the above example the number of employees, although really of numerical type, would rather be defined as nominal, since a size class, i.e. “> 1000” was used instead of an exact indication like 1250 employees.

The following table will give you an overview of all value types supported by RapidMiner:

Value type	**RapidMiner name**	**Use**
Nominal	nominal	Categorical non-numerical values, usually used for finite quantities of different characteristics
Numerical values	numeric	For numerical values in general
Integers	integer	Whole numbers, positive and negative
Real numbers	real	Real numbers, positive and negative
Text	text	Random free text without structure
2-value nominal	binominal	Special case of nominal, where only two different values are permitted
multi-value nominal	polynominal	Special case of nominal, where more than two different values are permitted
Date Time	data_time	Date as well as time
Date	date	Only date
Time	time	Only time

1.2.5 Data and Meta Data

We want to summarise our initial situation one more time. We have a **Concept** "customer" available, which we will describe with a set of **Attributes**:

- *Prototype positively received? Label; Nominal*
- Name: *Text*
- Address: *Text*
- Sector: *Nominal*
- Subsector: *Nominal*
- Number of employees: *Nominal*
- Number of purchases in product group 1: *Numerical*
- Number of purchases in product group 2: *Numerical*

The attribute "Prototype positively received?" has a special **Role** among the attributes; it is our **Target Attribute** here. The target attribute has the **Value Type Nominal**, which means that only relatively few characteristics (in this case "yes" and "no") can be accepted. Strictly speaking it is even binominal, since only two different characteristics are permitted. The remaining attributes all have no special role, i.e. they are **regular** and have either the value type **Numerical** or **Text**. The following definition is very important, since it plays a crucial role in a successful professional data analysis:

This volume of information which describes a concept is also called meta data, since it represents data via the actual data.

Our fictitious enterprise has a number of **Examples** for our concept "customer", i.e. the information which the enterprise has stored for the individual attributes in its customer database. The goal is now to generate a prediction instruction from the examples for which information is available concerning the target attribute, which predicts for us whether the remaining customers would be more likely to receive the prototype positively or reject it. The search for such a prediction instruction is one of the tasks which can be performed with data mining.

However, it is important here that the information for the attributes of the individual examples is in an ordered form, so that the data mining method can access it by means of a computer. What would be more obvious here than a table? Each of the attributes defines a column and each example with the different attribute values corresponds to a row of this table. For our scenario this could look like in table 1.1 for example.

We call such a table an **Example Set**, since this table contains the data for all the attributes of our examples. In the following and also within RapidMiner we will use the terms **Data, Data Set** and **Example Set** synonymously. A table with the appropriate entries for the attribute values of the current examples is always meant in this case. It is also such data tables which have lent their name to data analysis or data mining. Note:

***Data** describes the objects of a concept, **Meta Data** describes the characteristics of a concept (and therefore also of the data).*

Most data mining methods expect the examples to be given in such an attribute value table. Fortunately, this is the case here and we can spare ourselves any further data transformations. In practice however this is completely different and the majority of work during a data analysis is time spent transferring the data into a format suitable for data mining. These transformations are therefore dealt with in detail in later chapters.

1.2.6 Modelling

Once we have the data regarding our customers available in a well-structured format, we can then finally replace the unknown values of our target attribute with the prediction of the most probable value by means of a data mining method. We have numerous methods available here, many of which, just like the analogy reasoning described at the beginning or the generating of rules of thumb, are based on human behaviour. We call the use of a data mining method **model** and the result of such a method, i.e. the prediction instruction, is a **model**. Just as data mining can be used for different issues, this also applies for models. They can be easy to understand and explain the underlying processes in a simple manner. Or they can be good to use for prediction in the case of unknown situations. Sometimes both apply, such as with the following model for example, which a data mining method could have supplied for our scenario:

Prototype positively received?	*Name*	*Address*	*Sector*	*Subsector*	*Number of employees*	*Number of purchases group 1*	*Number of purchases group 2*	...
yes	Müller Systemtechnik GmbH	Meisenstr. 7, Böblingen	Sector	Pipe bending machines	> 1000	5	0	...
?	Meier Papier	Taubenweg 6, Coburg	IT	Telecommunications	600-1000	3	7	...
no	Schulze & Nagel	Amselallee 5, Homberg	Trade	Textiles	< 100	1	11	...
...	...	...	...	...	...	...	...	...

Table 1.1: An example scenario

"If the customer comes from urban areas, has more than 500 employees and if at least 3 purchases were transacted in product group 1, then the probability of this customer being interested in the new product is high."

Such a model is can be easily understood and may provide a deeper insight into the underlying data and decision processes of your customers. And in addition it is an operational model, i.e. a model which can be used directly for making a prediction for further customers. The company "Meier Papier" for example satisfies the conditions of the rule above and is therefore bound to be interested in the new product - at least there is a high probability of this. Your goal would therefore have been reached and by using data mining you would have generated a model which you could use for increasing your marketing efficiency: Instead of just contacting all existing customers and other candidates without looking, you could now concentrate your marketing efforts on promising customers and would therefore have a substantially higher success rate with less time and effort. Or you could even go a step further and analyse which sales channels would probably produce the best results and for which customers.

In the following chapters we will focus on further uses of data mining and at the same time practise transferring concepts such as customers, business processes or products into attributes, examples and data sets. This will train the eye to detect further possibilities of application tremendously and will make analyst life much easier for you later on. First though, we would like to spend a little time on RapidMiner and give a small introduction to its use, so that you can implement the following examples immediately.

2 Design of Analysis Processes with RapidMiner

The analysis of large volumes of data with data mining methods is generally regarded as a field for specialists. The latter create more or less complex analysis processes with often shockingly expensive software solutions for predicting the imminent handing in of notices or the sales figures of a product for example. The economic benefit is obvious, and so it was thought for a long time that the use of data mining software products was also associated with high software license costs and the support often necessary due to the complexity of the subject matter. Probably no later than when the open source software RapidMiner was developed could anybody seriously doubt that software solutions for data mining did not have to be expensive or difficult to use.

The development of RapidMiner was begun under the name "Yet Another Learning Environment" (YALE) at the chair for artificial intelligence of Dortmund University, under the direction of Prof. Dr. Katharina Morik. The software became more and more mature as time went on; more than half a million downloads have been recorded since development started in 2001. Among the many thousands of users there were also many enterprises which were looking for a partner for services and projects with the appropriate data mining competence. The developers of RapidMiner responded to this need and created the enterprise Rapid-I, which is responsible for the further development and maintenance of the software today. In the course of the enterprise's establishment, the YALE software was renamed to RapidMiner in line with its new meaning. So RapidMiner and the enterprise behind it, Rapid-I, are on good footing: Rapid-I came fourth in

the national Start-Up competition “start2grow” and won first prize in Europe’s handsomely-rewarded IT competition “Open Source Business Award”. RapidMiner itself has already been chosen as the most-used open source data mining solution on the well-known data mining portal “KDnuggets” for the third time in a row - and RapidMiner also performed more than well on the whole with a close second place among the over 30 other solutions, including proprietary solutions.

2.0.1 Flexibility and Functionality

But what exactly makes RapidMiner the world’s leading open source data mining software? According to an independent comparative study by the Chemnitz University of Technology, which was presented during the international Data Mining Cup 2007 (DMC-2007), RapidMiner does the best among the most important open source data mining tools both in terms of technology and applicability. This also reflects the focus of the development work which has always been put on a user-friendly combinability of the latest as well as established data mining techniques.

This combining gives RapidMiner a high flexibility when defining analysis processes. As we will see in the following, *processes* can be produced from a large number of almost randomly nestable *operators* and finally be represented by so-called operator trees or by a process graph (*flow design*). The process structure is described internally by XML and developed by means of a graphical user interface. In the background, RapidMiner constantly checks the process currently being developed for syntax conformity and automatically makes suggestions in the case of problems. This is made possible by so-called *meta data transformation*, which transforms the underlying meta data as early as at the design stage in such a way that the form of the result can already be foreseen and solutions can be identified in the case of unsuitable operator combinations (*quick fixes*). In addition, RapidMiner offers the analyst the possibility of defining *breakpoints* and of therefore inspecting virtually every intermediate result. Successful operator combinations can be pooled into *building blocks* and are therefore available again in later processes.

Thus the processes of RapidMiner combine the power of development environments, as known from programming languages, with the simplicity of visual programming. The modular approach also has the advantage that even internal

analysis processes can be examined in the greatest detail and utilised. Analysts can therefore also look into the individual partial steps of a cross-validation for example or equally evaluate the effect of pre-processing – which is typically not possible with other solutions and often results in error estimations being too optimistic.

RapidMiner contains more than 500 operators altogether for all tasks of professional data analysis, i.e. operators for *input* and *output* as well as *data processing (ETL)*, *modelling* and other aspects of data mining. But also methods of *text mining*, *web mining*, the automatic sentiment analysis from Internet discussion forums (*sentiment analysis*, opinion mining) as well as the *time series analysis* and - *prediction* are available to the analyst. In addition, RapidMiner contains more than 20 methods to also *visualise* high-dimensional data and models. Moreover, all learning methods and weighting factors of the Weka Toolbox have also been completely and smoothly integrated into RapidMiner, meaning that the complete range of functions of Weka, which is equally widespread in research at the moment, also joins the already enormous range of functions of RapidMiner.

2.0.2 Scalability

In October 2009, Version 4.6 of RapidMiner was released and the fully revised Version 5.0 was finally released at the end of 2009. The direction taken becomes more than clear in these two versions: in addition to great functionality, the main focus is on optimisation with regard to scalability, even to large data volumes. One of the main characteristics of RapidMiner has always been a concept similar to that of relational databases, which makes different *views* on data sources possible. This concept has been further refined by RapidMiner and now offers the possibility of combining a number of such views in such a way that the data is transformed *on-the-fly* and data copies become largely unnecessary. In this way RapidMiner achieves a memory usage which is often much lower in comparison and can, assuming that RapidMiner and the analysis processes are configured appropriately, even make handling several 100 million data sets child's play.

Further innovations like the improved lift charts of RapidMiner support the optimisation of direct mailing and marketing campaigns, churn reduction, the increase of customer retention and the cost-benefit optimised acquisition of customers. Extended pivotings, new aggregation functions, an extensive date and time han-

dling, the simplified function-based design of new attributes, optimised wizards among other things for the automatic optimisation of data mining process parameters as well as new visualisations with zooming and panning also make improved analyses and data transformations possible and make running the programme a whole lot easier in addition. The most substantial innovations of the new Version 5 of RapidMiner is however the complete revision of the graphical user interface, which now also shows the explicit data flows instead of just the operator tree and in addition, on the basis of the repository now integrated, also supports meta data transformation during the design time.

2.0.3 A Question of Format

A further focus of RapidMiner is high connectivity to the most varied of data sources such as Oracle, IBM DB2, Microsoft SQL Server, MySQL, PostgreSQL and Ingres, the access to Excel, Access and SPSS files as well as numerous other data formats. Together with the hundreds of operators for data pre-processing, RapidMiner can therefore also be used, apart from data analysis, for data integration and transformation (ETL) with outstanding results.

And the user even has a choice of different formats for the software itself. RapidMiner is available in the free RapidMiner Community Edition on the one hand, which can be downloaded from the website at any time and free of charge, and in the Enterprise Edition, which combines the advantages of the free Community Edition with a complete professional support with guaranteed response times.

2.1 Installation and First Repository

Before we can work with RapidMiner, you of course need to download and install the software first. You will find it in the download area of the developer Rapid-I's website:

http://www.rapid-i.com

Download the appropriate installation package for your operating system and install RapidMiner according to the instructions on the website. All usual Windows versions are supported as well as Macintosh, Linux or Unix systems. Please note that an up-to-date Java Runtime (at least version 6) is needed for the latter.

If you are starting RapidMiner for the first time, you will be asked to create a new repository. We will limit ourselves to a local repository on your computer first of all - later on you can then define repositories in the network, which you can also share with other analysts:

Figure 2.1: Create a local repository on your computer to begin with the first time you start the program.

For a local repository you just need to specify a name (alias) and define any directory on your hard drive. You can select the directory directly by clicking on the folder icon on the right. It is advisable to create a new directory in a convenient place within the file dialog that then appears and then use this new directory as a basis for your local repository. This repository serves as a central storage location for your data and analysis processes and will accompany you in the near future.

2.2 Perspectives and Views

After choosing the repository you will be welcomed into the so-called *Welcome Perspective*(Fig. 2.3).

The lower section shows current *news* about RapidMiner, if you have an Internet connection. The list in the centre shows the *analysis processes recently worked*

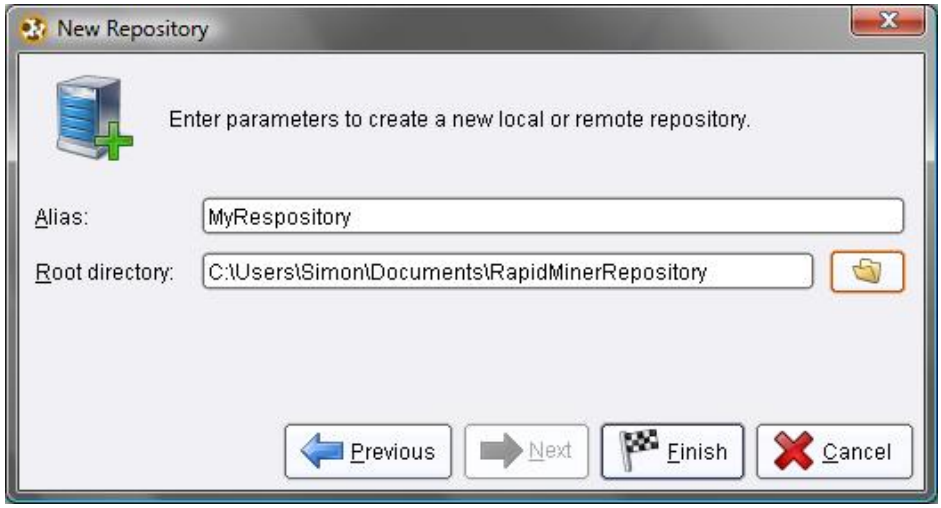


Figure 2.2: Definition of a new local repository for storing your data and analysis processes. It is advisable to create a new directory as a basis.

on. This is practical if you wish to continue working on or execute one of these processes. You can open a process from this list to work on or execute it simply by double clicking. The upper section shows typical actions which you as an analyst will perform frequently after starting RapidMiner. Here are the details of these:

1. *New:* Starts a new analysis process. First you must define a location and a name within the process and data repository and then you will be able to start designing a new process.

2. *Open Recent:* Opens the process which is selected in the list below the actions. Alternatively, you can also open this process by double-clicking inside the list. Either way, RapidMiner will then automatically switch to the Design Perspective.

3. *Open:* Opens the repository browser and allows you to select a process to be opened within the process Design Perspective.

4. *Open Template:* Shows a selection of different pre-defined analysis processes, which can be configured in a few clicks.

5. *Online Tutorial:* Starts a tutorial which can be used directly within Rapid-

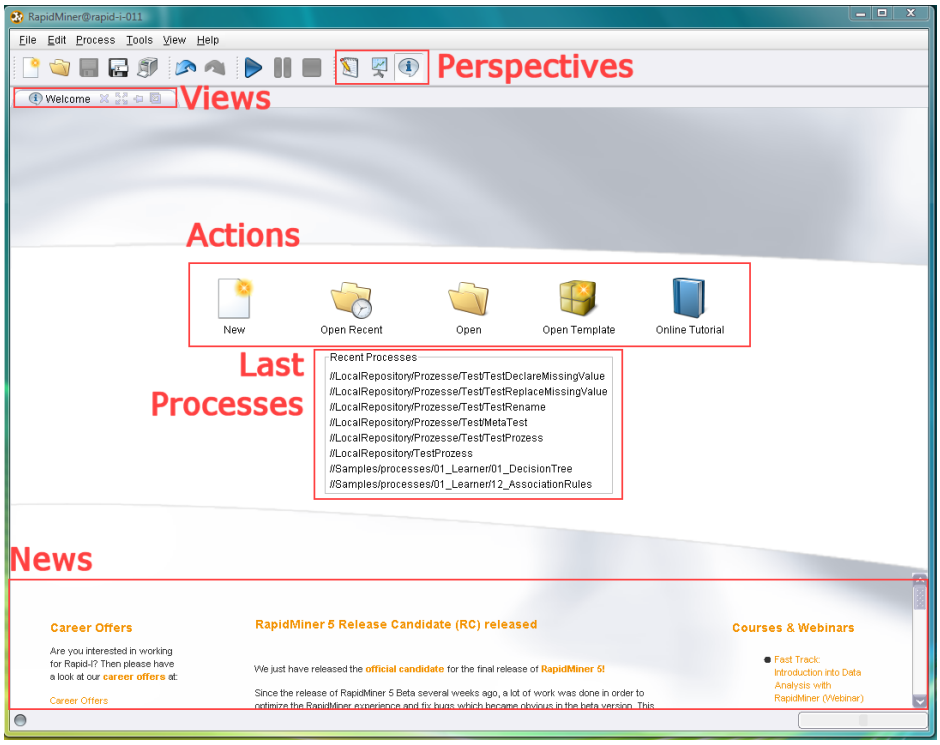


Figure 2.3: Welcome Perspective of RapidMiner.

Miner and gives an introduction to some data mining concepts using a selection of analysis processes. Recommended if you have a basic knowledge of data mining and are already familiar with the fundamental operation of RapidMiner.

At the right-hand side of the toolbar inside the upper section of RapidMiner you will find three icons which switch between the individual RapidMiner perspectives. A *perspective* consists of a freely configurable selection of individual user interface elements, the so-called *views*. These can also be arranged however you like.

In the Welcome Perspective there is only one view, one preset at least, namely the welcome screen, which you are looking at now. You can activate further views by accessing the “View” menu:

In the subitem “Show View” you will find all available views of RapidMiner.

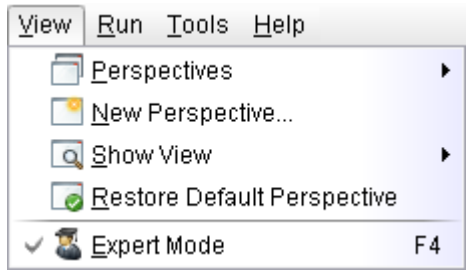


Figure 2.4: View menu.

Views which are now visible in the current perspective are marked with a tick. Activate a further view by making a selection, for example the view with the name “Log”. You will now see in Fig. 2.5 that a second view with this name has been added in the Welcome Perspective.

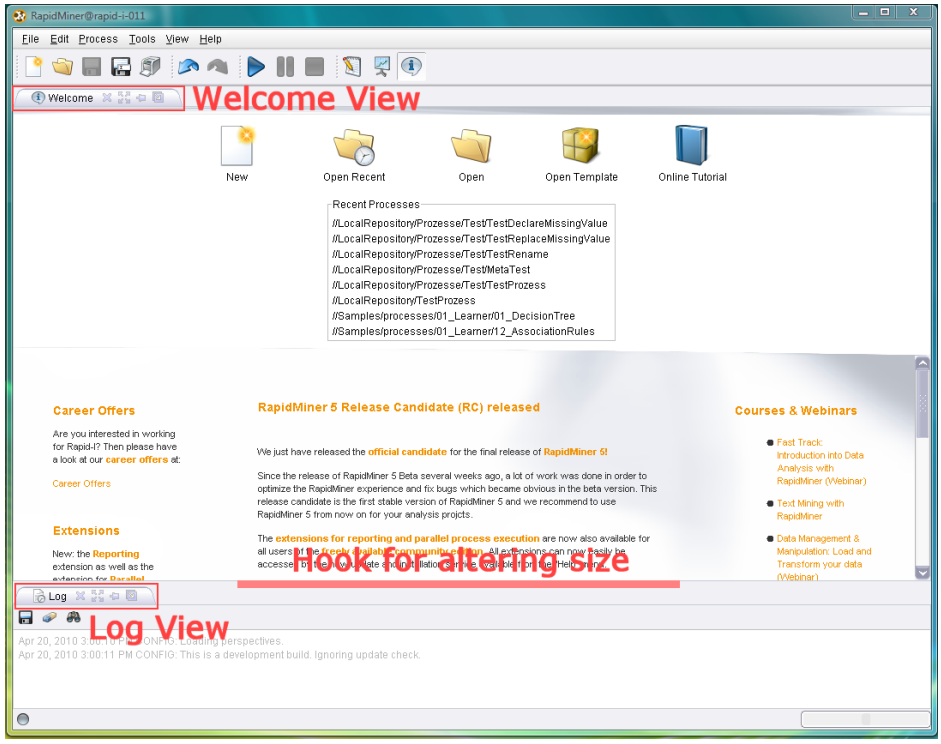


Figure 2.5: Size changes between views

You will now see the familiar Welcome View and the new Log View at the bot-

tom. If you now move the mouse into the highlighted area between them the cursor changes shape and indicates that you can change the sizes of the views by dragging, so by holding the mouse button down and moving the cursor. Feel free to try it out.

As already suggested, you can also change the position of the views as you like. In order to do this, simply move the cursor onto the name area of the view and drag the view to another position. The position in which the view would be arranged after releasing the mouse button is highlighted by a transparent grey area:

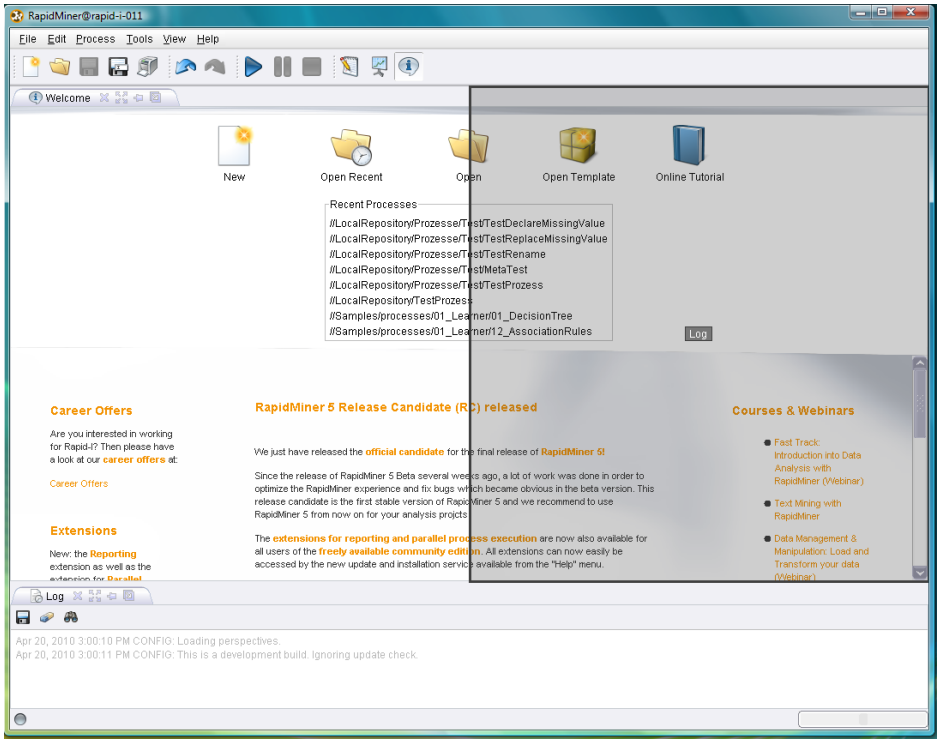


Figure 2.6: dragging the lower Log View to the right-hand side and highlighting the new position.

You can combine individual views in this way to create several file cards, meaning that only one is ever visible. Or you can drag the Log View from below to the right-hand area, so that the division now runs vertically and no longer horizontally. You can even undock a view completely and move it outside the RapidMiner window. If you would like to see a view in full for a short time, then

you can maximise a view and minimise it again later on. This is also done if you double click on the name area of a view. Each view offers you the following actions:

Figure 2.7: Actions for views

The following actions are possible for all RapidMiner views among others. The other actions should be self-explanatory:

1. *Close:* Closes the view in the current perspective. You can re-open the view in the current or another perspective via the menu "View" – "Show View".

2. *Maximize:* Maximises the view in the current perspective. Can also be done by double-clicking on the name area.

3. *Minimize:* Minimises the view in the current perspective. The view is displayed on the left-hand side of the perspective and can be maximised again or looked at briefly from there.

4. *Detach:* Detaches the view from the current perspective and shows it within its own window, which can be moved to wherever you want.

Now have a little go at arranging the two views in different ways. Sometimes a little practice is required in order to drop the views in exactly the desired place. It is worthwhile experimenting a little with the arrangements however, because other settings may make your work far more efficient depending on screen resolution and personal preferences.

Sometimes you may inadvertently delete a view or the perspective is unintentionally moved into particularly unfavourable positions. In this case the "View" menu can help, because apart from the possibility of reopening closed views via "Show View", the original state can also be recovered at any time via "Restore Default Perspektive".

In addition, you will also have the option here of saving your own perspectives under a freely selectable name ("New Perspective...") as well as of switching between the saved and pre-defined perspectives.

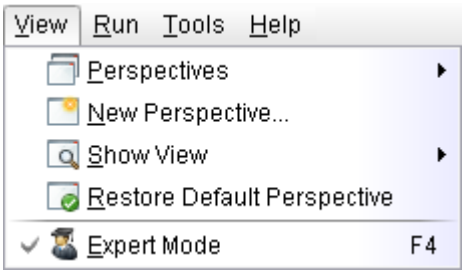


Figure 2.8: View menu

2.3 Design Perspective

As already mentioned at the beginning, you will find an icon for each (pre-defined) perspective within the right-hand area of the toolbar:

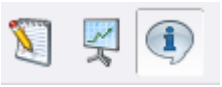


Figure 2.9: Toolbar Icons for Perspectives

The icons shown here take you to the following perspectives:

1. *Design Perspective:* This is the central RapidMiner perspective where all analysis processes are created and managed.

2. *Result Perspective:* If a process supplies results in the form of data, models or the like, then RapidMiner takes you to this Result Perspective, where you can look at several results at the same time as normal thanks to the views.

3. *Welcome Perspective:* The Welcome Perspective already described above, which RapidMiner welcomes you with after starting the program.

You can switch to the desired perspective by clicking inside the toolbar or alternatively via the menu entry "View" – "Perspectives" followed by the selection of the target perspective. RapidMiner will eventually also ask you automatically if switching to another perspective seems a good idea, e.g. to the Result Perspective on completing an analysis process.

Now switch to the Design Perspective by clicking in the toolbar. It will be dealt with in detail in this chapter. The Result Perspective will then be the topic of a later chapter. You should now see the following screen:

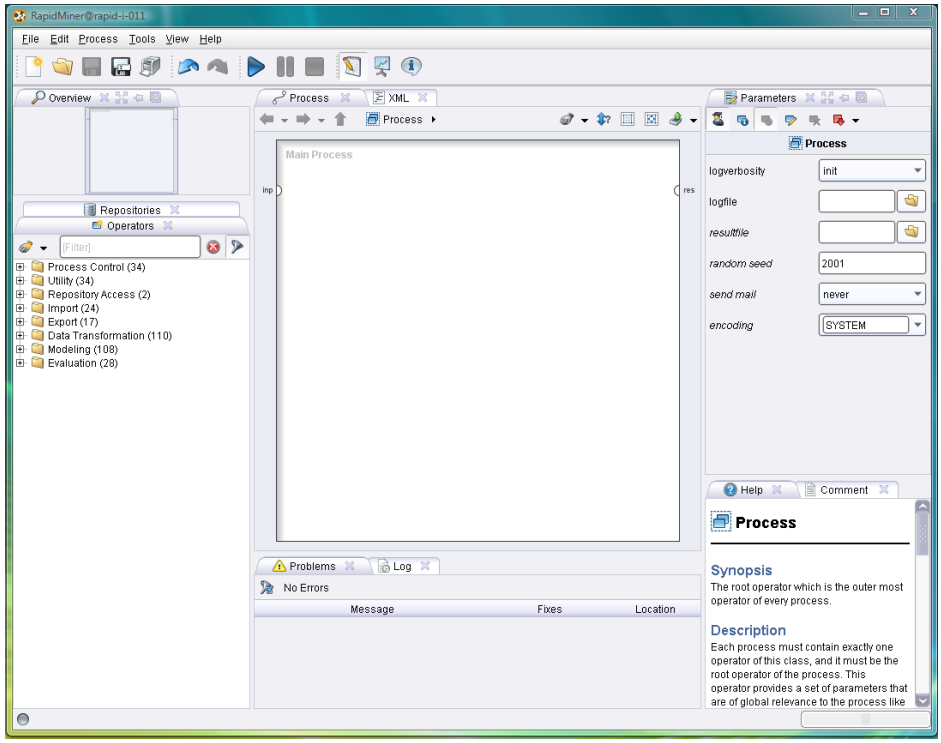


Figure 2.10: Design Perspective of RapidMiner

Since the Design Perspective is the central working environment of RapidMiner, we will discuss all parts of the Design Perspective separately in the following and discuss the fundamental functionalities of the associated views.

2.3.1 Operators and Repositories View

There are two very central views in this area, at least in the standard setting, which are described in the following.

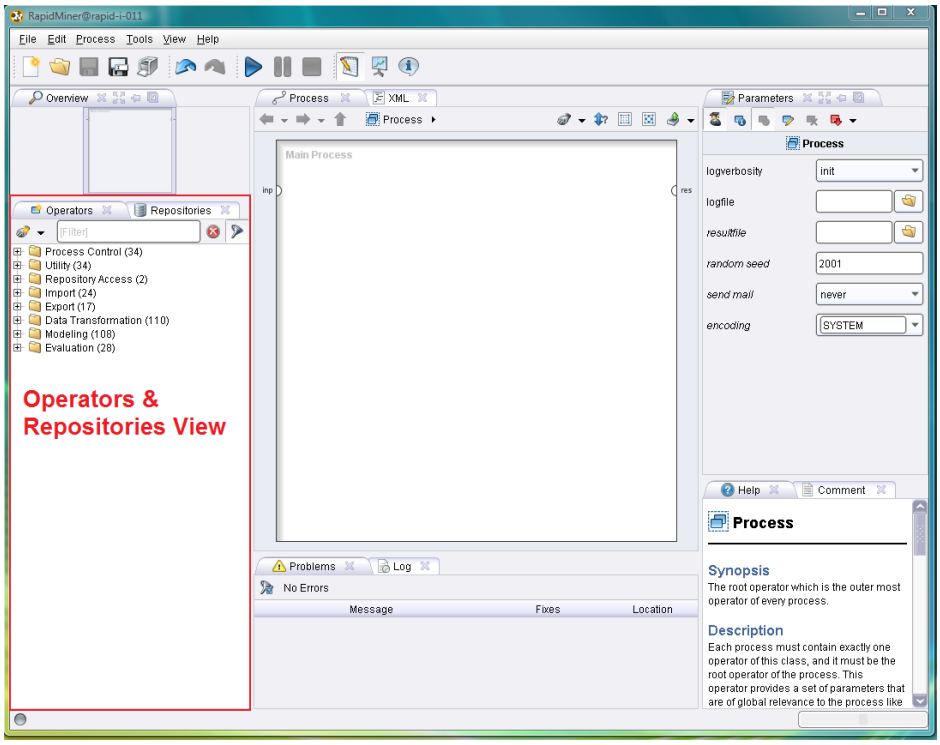


Figure 2.11: Design Operators of RapidMiner

Operators View

All work steps (operators) available in RapidMiner are presented in groups here and can therefore be included in the current process. You can navigate within the groups in a simple manner and browse in the operators provided to your heart's desire. If RapidMiner has been extended with one of the available extensions, then the additional operators can also be found here.

Without extensions you will find at least the following groups of operators in the tree structure:

- *Process Control:* Operators such as loops or conditional branches which can control the process flow.

- *Utility:* Auxiliary operators which, alongside the operator "Subprocess" for grouping subprocesses, also contain the important macro-operators as well

as the operators for logging.

- *Repository Access:* Contains the two operators for read and write access in repositories.
- *Import:* Contains a large number of operators in order to read data and objects from external formats such as files, databases etc.
- *Export:* Contains a large number of operators for writing data and objects into external formats such as files, databases etc.
- *Data Transformation:* Probably the most important group in the analysis in terms of size and relevance. All operators are located here for transforming both data and meta data.
- *Modelling:* Contains the actual data mining process such as classification methods, regression methods, clustering, weightings, methods for association rules, correlation and similarity analyses as well as operators, in order to apply the generated models to new data sets.
- *Evaluation:* Operators using which one can compute the quality of a modelling and thus for new data e.g. cross-validations, bootstrapping etc.

You can select operators within the Operators View and add them in the desired place in the process by simply dragging and dropping. You can choose whether new operators are to be directly connected with existing operators as suitably as possible on the basis of the available meta data information or not. Just select the plug symbol on the left-hand side of the toolbar of the view and define whether outgoing and/or incoming connections are to be created automatically. Otherwise you have to connect the operator yourself.

Figure 2.12: Actions and filters for the Operators View

In order to make the work as easy for you as possible, the Operators View also supports another filter besides, which can be used to search for parts of the operator name or the complete operator name. Just enter the search word into the filter field. As soon as there are less than 10 search hits altogether, the tree

is opened up to reveal all search hits. This means you do not need to navigate through the complete hierarchy each time. Clicking on the red cross next to the search field erases what is currently entered and closes up the tree again.

tip: Professionals will know the names of the necessary operators more and more frequently as time goes on. Apart from the search for the (complete) name, the search field also supports a search based on the initial letters (so-called camel case search). Just try "REx" for "Read Excel" or "DN" for "Date to Nominal" and "Date to Numerical" – this speeds up the search enormously.

Repositories View

The repository is a central component of RapidMiner which was introduced in Version 5. It serves for the management and structuring of your analysis processes into projects and at the same time as both a source of data as well as of the associated meta data. In the coming chapters we will give a detailed description of how to use the repository, so we shall just say the following at this stage.

reference: Since the majority of the RapidMiner supports make use of meta data for process design, we strongly recommend you use the repository, since otherwise (for example in the case of data being directly read from files or databases) the meta data will not be available, meaning that numerous supports will not be offered.

2.3.2 Process View

The Process View shows the individual steps within the analysis process as well as their interconnections. New steps can be added to the current process in several ways. Connections between these steps can be defined and detached again. Finally, it is even possible to define the order of the steps in this perspective. But first things first.

2.3.3 Operators and Processes

Working with RapidMiner fundamentally consists in defining analysis processes by indicating a succession of individual work steps. In RapidMiner these process

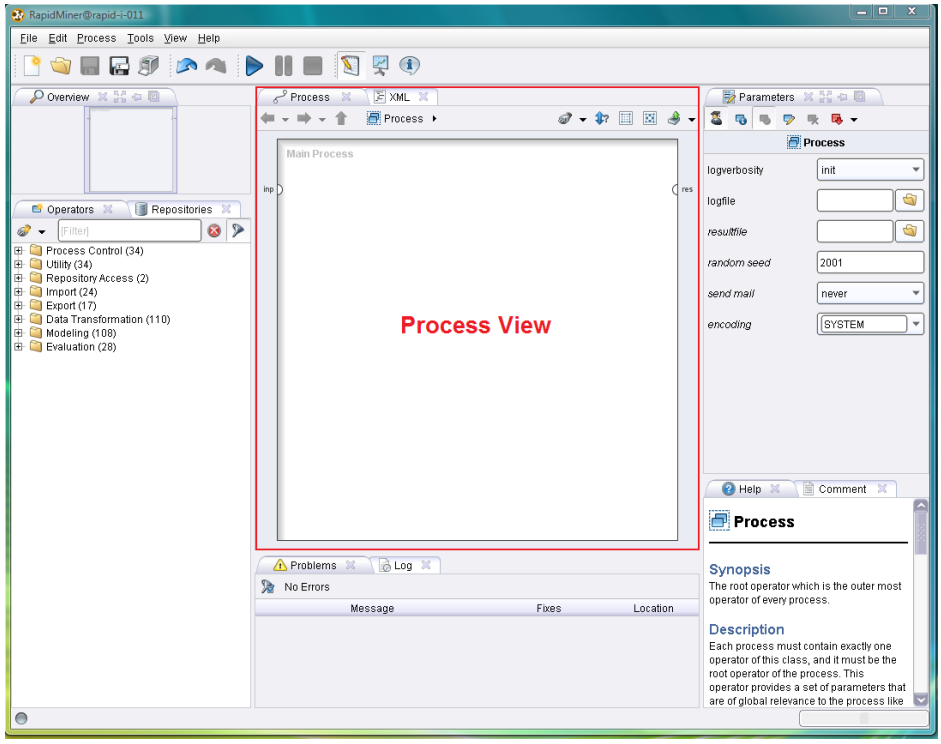


Figure 2.13: In the Process View the components of RapidMiner, the so-called operators, are connected

components are called **operators**. An operator is defined by several things:

- The description of the expected inputs,
- The description of the supplied outputs,
- The action performed by the operator on the inputs, which ultimately leads to the supply of the outputs,
- A number of parameters which can control the action performed.

The inputs and outputs of operators are generated or consumed via **ports**. We will see that an operator is represented in RapidMiner by a module in the following form:

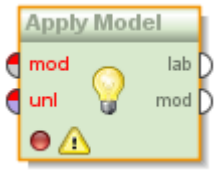


Figure 2.14: An operator can be connected via its input ports (left) and output ports (right).

Such an operator can for example import data from the repository, a database or from files. In this case it would have no input ports, although it would have a parameter at least specifying the location of the data. Other operators transform their inputs and return an object of the same type. Operators that transform data belong in this group. And other operators still consume their input and transform it into a completely new object: many data mining methods come under this category and supply a model for the given input data for example.

The colour of the ports indicates the input type a port must be supplied with. For example, a bluish colour indicates that an example set is required. If the upper half and the name of the port are red, then this indicates a problem. This problem is easy to see for the operator above: it is not connected and the input ports still need a connection to a suitable source.

Output ports are white if the result is unclear or cannot (yet) be supplied in the current configuration. As soon as all necessary configurations are complete, i.e. all necessary parameters are defined and all necessary input ports connected, then the output ports are coloured according to their type.

Figure 2.15: Status indicators of operators

But it is not only the ports that can visualise their status by means of different status indicators, but also the complete operator. These are given from left to right by a:

- *Status light:* Indicates whether there is a problem like parameters that have

not yet been set or unconnected input ports (red), whether the configuration is basically complete but the operator has not yet been implemented since then (yellow) or whether everything is OK and the operator has also already been implemented successfully (green).

- *Warning triangle:* Indicates when there are status messages for this operator.
- *Breakpoint:* Indicates whether process execution is to be stopped before or after this operator in order to give the analyst the opportunity to examine intermediate results.
- *Comment:* If a comment has been entered for this operator, then this is indicated by this icon.
- *Subprocess:* This is a very important indication, since some operators have one or more subprocesses. It is shown by this indication whether there is such a subprocess. You can double click on the operator concerned to go down into the subprocesses.

If several operators are interconnected, then we speak of an **analysis process** or **process** for short. Such a succession of steps can for example load a data set, transform the data, compute a model and apply the model to another data set. Such a process may be as follows in RapidMiner:

Such processes can easily grow to several hundred operators in size in RapidMiner and spread over several levels or subprocesses. The process inspections continually performed in the background as well as the process navigation aids shown below ensure that you do not lose track and that you define correct processes, even for more complex tasks.

Inserting Operators

You can insert new operators into the process in different ways. Here are the details of the different ways:

- Via drag&drop from the Operators View as described above,
- Via double click on an operator in the Operators View,

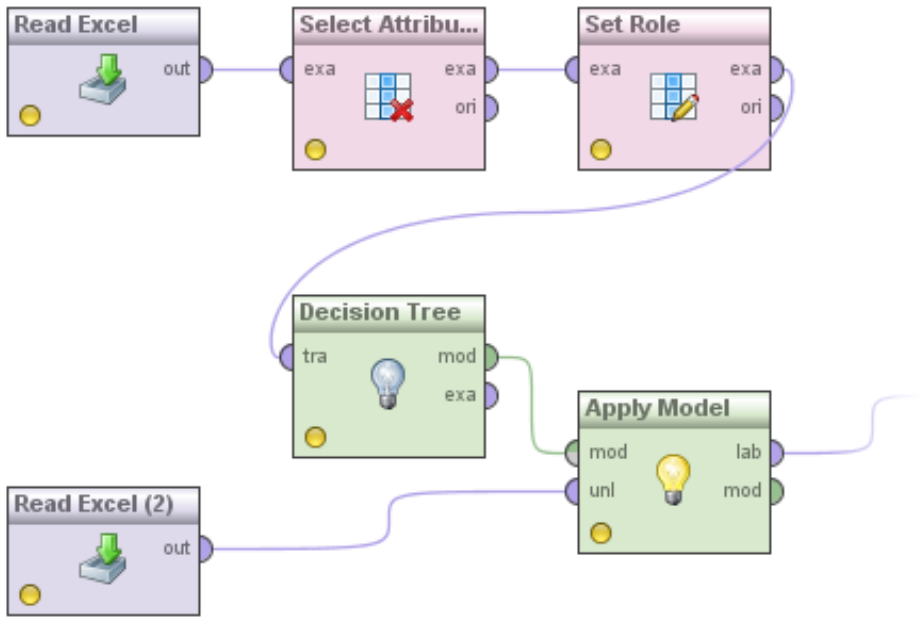


Figure 2.16: An analysis process consisting of several operators. The colour coding of the data flows shows the type of object passed on.

- Via dialog which is opened by means of the first icon in the toolbar of the Process View,
- Via dialog which is opened by means of the menu entry "Edit" – " New Operator..." (CTRL-I),
- Via context menu in a free area of the white process area and there via the submenu"New Operator" and the selection of an operator.

Figure 2.17: Actions in the Process View

In each case new operators are, depending on the setting in the Operators View, either automatically connected with suitable operators, or the connections have to be made or corrected manually by the user.

Connecting Operators

After you have inserted new operators, you can interconnect the operators inserted. There are basically three ways available to you, which will be described in the following.

Connections 1: Automatically when inserting

If you have activated the option for automatic connection under the plug symbol in the Operators View, then RapidMiner will try to connect the operator with suitable output ports after inserting. If, for example, the new operator has an input port which requires an example set, then RapidMiner will try to find an operator that could already produce such an example set. If there is only one option, then this choice is clear and the operator is connected. If there are several options however, RapidMiner will try to select the option which is the closest on the left above the current mouse position. The associated operator is marked with a frame and a shadow. In this way you can set the course for a correct connection as early as during insertion.

Tip: It is recommended that you activate the option for automatic connection for the input ports at least. Even if the connection algorithm based on the meta data occasionally creates a wrong connection, you still save yourself a lot of work for all cases where the correct connection is automatically recognised.

Connections 2: Manually

You can also interconnect the operators manually and this is even necessary for more complex processes. In order to do this, click on an output port. You will now draw an orange strand. Click on an input port in order to connect the selected output port with this input port. In order to cancel the process, hold the mouse still and click using the right-hand mouse button. The orange strand will disappear and you can continue working as normal.

Connections 3: Fully automatically

Sometimes numerous operators are already in a (sub)process and are not yet connected. In this case the options “Auto-Wire” and “Re-Wire” can serve you well, which are hidden behind the plug symbol in the Process View directly beside the icon for opening the dialog for a new operator. This works particularly well if a relatively sequential approach was taken when the process was created and

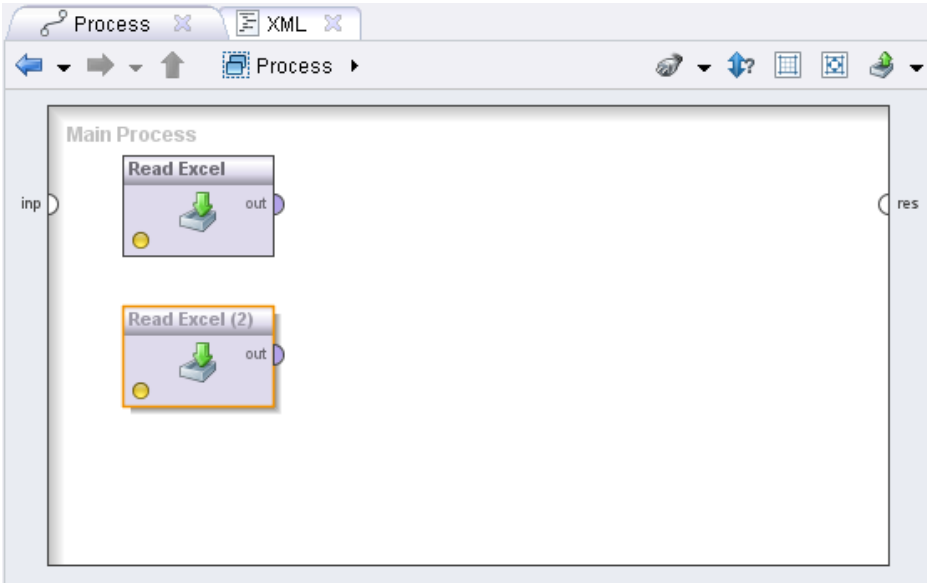


Figure 2.18: The second operator is highlighted during the dragging process (frame plus shade) and is preferably connected with the new operator if the latter is now dropped and expects an example set.

the operators were properly lined up one behind the other, i.e. the previous operator was always marked by a frame and shadow during insertion. It is always wise however to perform a manual examination following the fully automatic connection since unintended connections can occur, especially in the case of more complex processes.

Selecting Operators

On order to edit parameters you must select an individual operator. You will recognise the operator currently selected by its orange frame as well as its shadow.

If you wish to perform an action for several operators at the same time, for example moving or deleting, please select the relevant operators by dragging a frame around these.

In order to add individual operators to the current selection or exclude individual operators from the current selection, please hold the CTRL key down while you

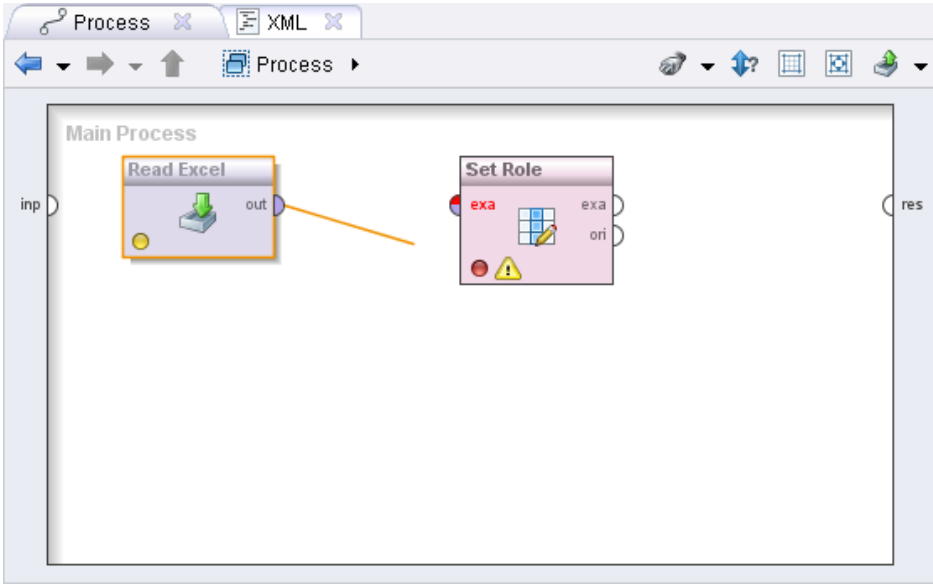


Figure 2.19: Click on an output port in order to connect, right click to cancel.

click on the relevant operators or add further operators by dragging a frame.

Moving Operators

Select one or more operators as described above. Now move the cursor onto one of the selected operators and drag the mouse while holding down the button. All selected operators will now be moved to a new place depending on where you move the mouse.

If, in the course of this movement, you reach the edge of the white area, then this will be automatically enlarged accordingly. If you should reach the edge of the visible area, then this will also be moved along automatically.

Deleting Operators

Select one or more operators as described above. You can now delete the selected operators by

- Pressing the DELETE key,

- Selecting the action “Delete” in the context menu of one of the selected operators,
- By means of the menu entry “Edit” – “Delete”.

Deleting Connections

Connections can be deleted by clicking on one of the two ports while pressing the ALT key at the same time. Alternatively, you can also delete a connection via the context menu of the ports concerned.

Navigating Within the Process

If we look again at the toolbar for the Process View, then we can see that we have only made use of the two actions on the left so far. In this section we will discuss the following four elements: the arrow pointing left, the arrow pointing right, the arrow pointing upwards and the navigation bar (breadcrumb).

Figure 2.20: Actions in the Process View

The individual actions:

1. *Arrow pointing left:* Returns to the last place of editing in a similar way to the navigation which is familiar from internet browsers. Individual steps can also be skipped via the pop-up menu.
2. *Arrow pointing right:* Returns to the most recent editing places in the history in a similar way to the navigation which is familiar from internet browsers. Individual steps can also be skipped via the pop-up menu.
3. *Arrow pointing upwards:* Leave the current subprocess and return to the greater process.
4. *Navigation bar:* The navigation bar shows the way from the main process to the current subprocess via all levels passed through. Clicking once on

one of the operators will show the process concerned. You can navigate further downwards using the small arrows pointing right.

In order to therefore descend into a subprocess, you need to double click on an operator with the subprocess icon at the bottom on the right. In order to go a level up again, you can navigate upwards using the arrow. The current path is shown by the navigation bar, which can alternatively be used to navigate in both directions.

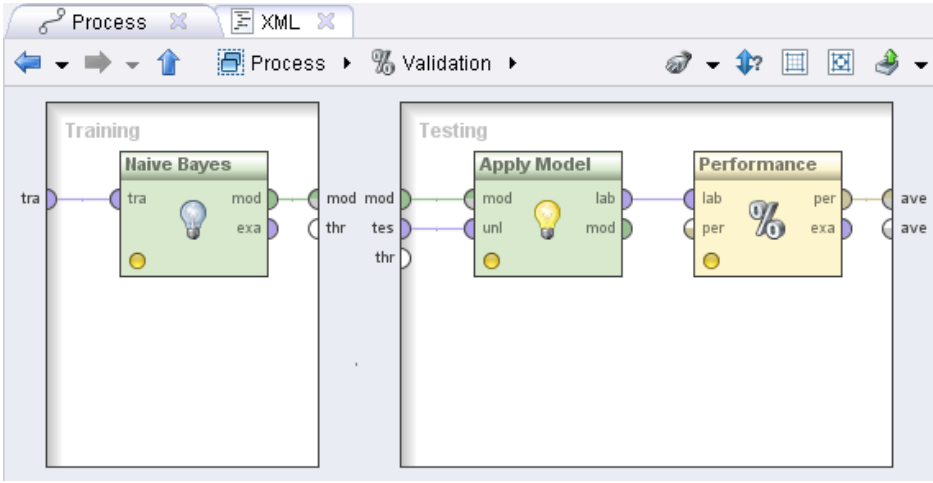


Figure 2.21: A subprocess named "Validation" which can be left again using the arrow pointing upwards or via the navigation bar.

Defining the Execution Order

In nearly all cases, RapidMiner succeeds in automatically determining the correct execution order of the operators. In order to do this, RapidMiner uses the connection information and the fact that an operator, the result of which is to be used by another operator, must obviously be executed before the latter.

However, there are cases where the order cannot be automatically defined such as completely parallel subprocesses or where the automatic order is not correct, for example because a macro must first be computed before it can be used as a parameter in a later operator. But there are also other reasons that often play a big part, such as more efficient data handling or an exact order desired for execution (for reporting for example).

For this purpose, RapidMiner offers an elegant method for indicating the order of the operators and even for editing the execution order comfortably. Please click on the double arrow pointing upwards and downwards with the question mark in the toolbar of the Process View and switch to the view for order definition. Instead of the icon for each operator, the number of its execution will now be shown. The transparent orange line connects the operators in this order, as shown in Fig. 2.22.

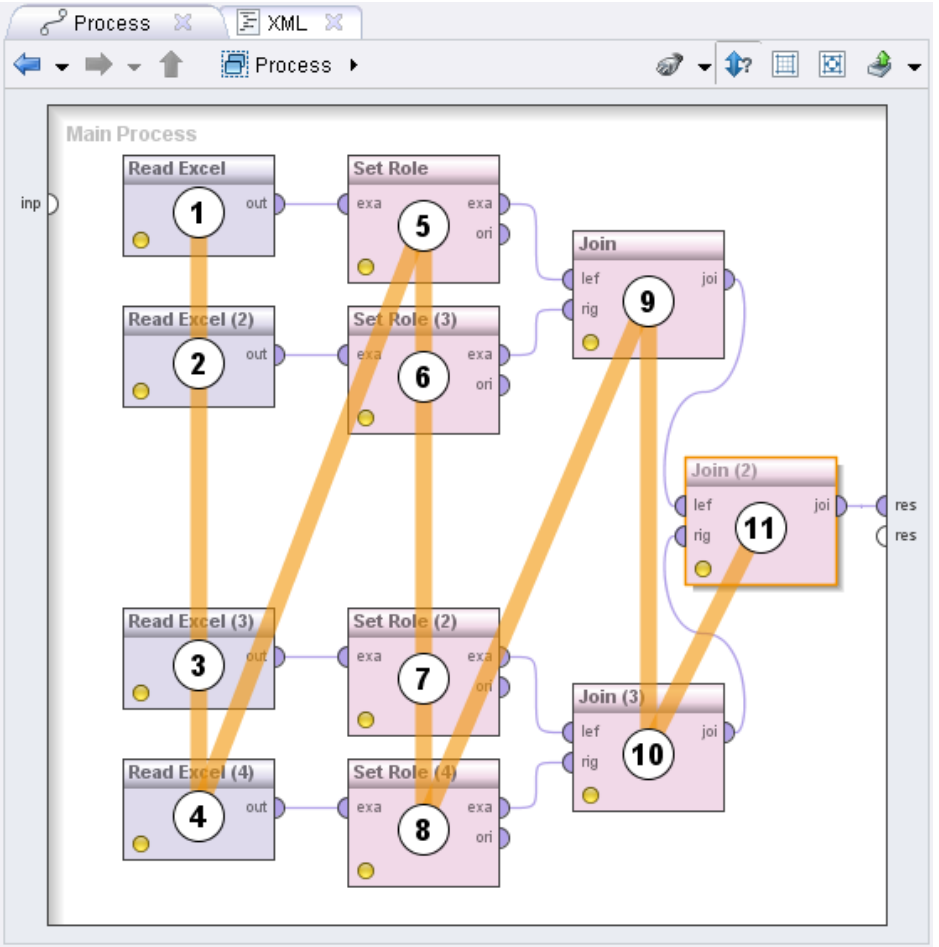


Figure 2.22: Representation of the execution order. This order is unfavourable however, since more data sets have to be handled at the same time.

In order to change such an order, you can click anywhere on an operator. The path leading to this operator can now not be changed, but clicking again to

choose an operator that comes after the selected one will attempt to change the order in such a way that the second operator is executed as soon as possible after the first. While you move the mouse over the remaining operators, you will see the current choice in orange up to this operator and in grey starting from this operator. A choice that is not possible is symbolised by a red number. You can cancel a current selection by right-clicking. In this way you can, as shown in Fig. 2.23, change the order of the process described above to the following with only a few clicks.

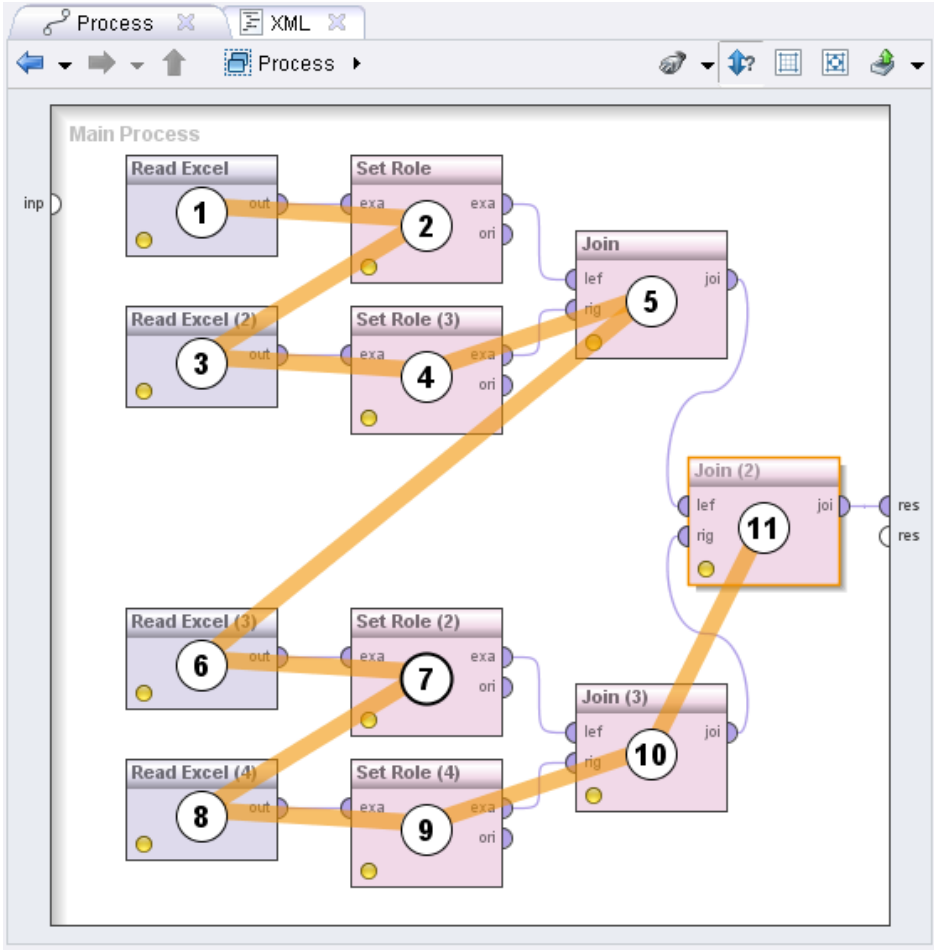


Figure 2.23: New order after some changes.

2.3.4 Further Options of the Process View

After having discussed nearly all options of this central element of the RapidMiner Design Perspective, we will now describe the remaining actions in the toolbar, which can be seen in Fig. 2.24, as well as further possibilities of the Process View.

Figure 2.24: Actions in the Process View

The three icons on the right-hand side of the Process View toolbar perform the following actions:

1. *Automatic arrangement:* Rearranges all operators of the current process according to the connections and the current execution order.

2. *Automatic size:* Changes the size of the white working area in such a manner that all operators currently positioned have just enough space. This is particularly practical for automatic downsizing
(size optimisation).

3. *export:* The current process perspective can both be printed and be exported to PDF and other formats.

2.3.5 Parameters View

Fig. 2.25 shows the Parameters View of RapidMiner.

Numerous operators require one or several parameters to be indicated for a correct functionality. For example, operators that read data from files require the file path to be indicated. Much more frequently however, parameters are not absolutely necessary, although the execution of the operator can be controlled by indicating certain parameter values and, in the case of modelling, also frequently be optimised.

After an operator offering parameters has been selected in the Process View, its parameters are shown in the parameter view. Like the other views, this view also

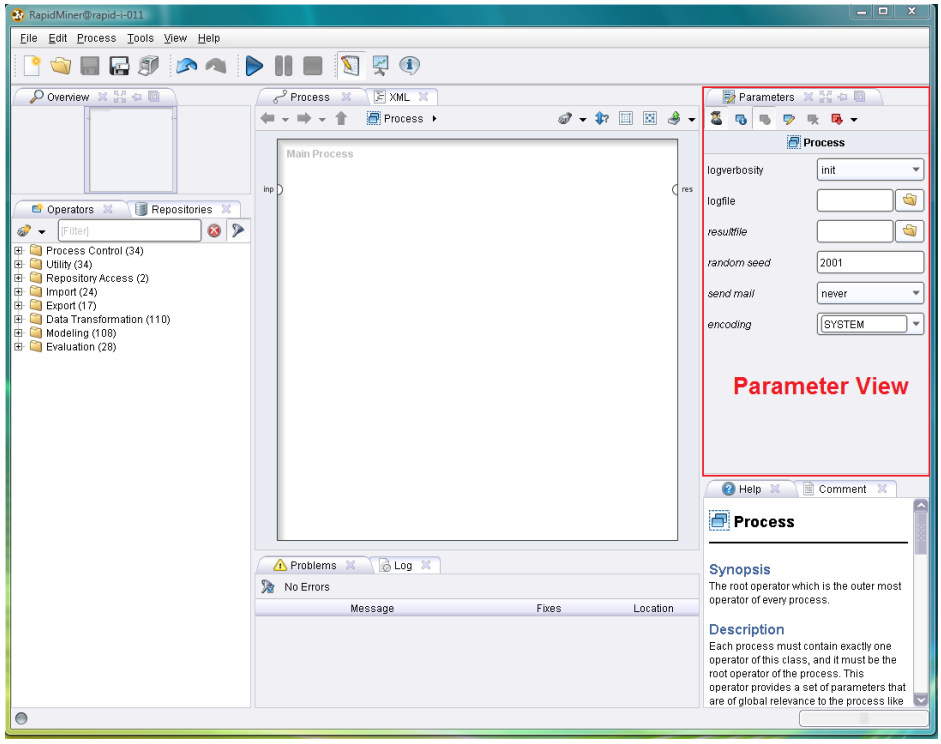


Figure 2.25: Parameters of the currently selected operator are set in the parameter view.

has its own toolbar which is described in the following. Under the toolbar you will find the icon and name of the operator currently selected followed by the actual parameters. Bold font means that the parameter must absolutely be defined by the analyst and has no default value. Italic font means that the parameter is classified as an expert parameter and should not necessarily be changed by beginners to data analysis.

Please note that some parameters are only indicated when other parameters have a certain value. For example, an absolute number of desired examples can only be indicated for the operator “sampling” when “absolute” has been selected as the type of sampling.

The actions of the toolbar refer, just like the parameters, to the operator currently selected. The details of these are:

1. *operator info:* Display of some fundamental information on this operator such as expected inputs or a description. This dialog is also displayed by pressing F1 after selection, via the context menu in the Process View as well as via the menu entry “Edit” – “Show Operator Info...”.

2. *Enable/Disable:* Operators can be (temporarily) deactivated. Their connections are detached and they are no longer executed. Deactivated operators are shown in grey. Operators can also be (de)activated within their context menu in the Process View as well as via the menu entry “Edit” – “Enable Operator”.

3. *Rename:* One of the ways to rename an operator. Further ways are pressing F2 after selection, selecting “Rename” in the context menu of the operator in the Process View as well as the menu entry “Edit” – “Rename”.

4. *delete:* One of the ways to delete an operator. Further ways are pressing DELETE after selection, selecting “delete” in the context menu of the operator in the Process View as well as the menu entry “edit” – “delete”.

5. *Toggle Breakpoints:* Breakpoints can be set here both before and after the execution of the operator, where the process execution stops and intermediate results can be examined. There is also this possibility in the context menu of the operator in the Process View as well as in the “Edit” menu. A breakpoint after operator execution can also be activated and deactivated with F7.

6. *Flag as Dirty:* Sets the status of the operator again so that it will always be executed if process execution is repeated.

7. *expert mode:* The icon on the far right switches between expert mode and beginner mode. Only in the expert mode are all parameters shown; in the beginner mode the parameters classified as expert parameters are not shown.

2.3.6 Help and Comment View

Figure 2.26: The parameters of the operator "nominal to date".

Operators View

Each time you select an operator in the Operators View or in the Process View, the help window within the Help View shows a description of this operator. These descriptions include

- A short synopsis which summarises the function of the operator in one or a few sentences,
- A detailed description of the functionality of the operator,
- A list of all parameters including a short description of the parameter, the default value (if available), the indication as to whether this parameter is an expert parameter as well as an indication of parameter dependencies.

Comment View

Unlike Help, the Comment View is not dedicated to pre-defined descriptions but rather to your own comments on individual steps of the process. Simply select an

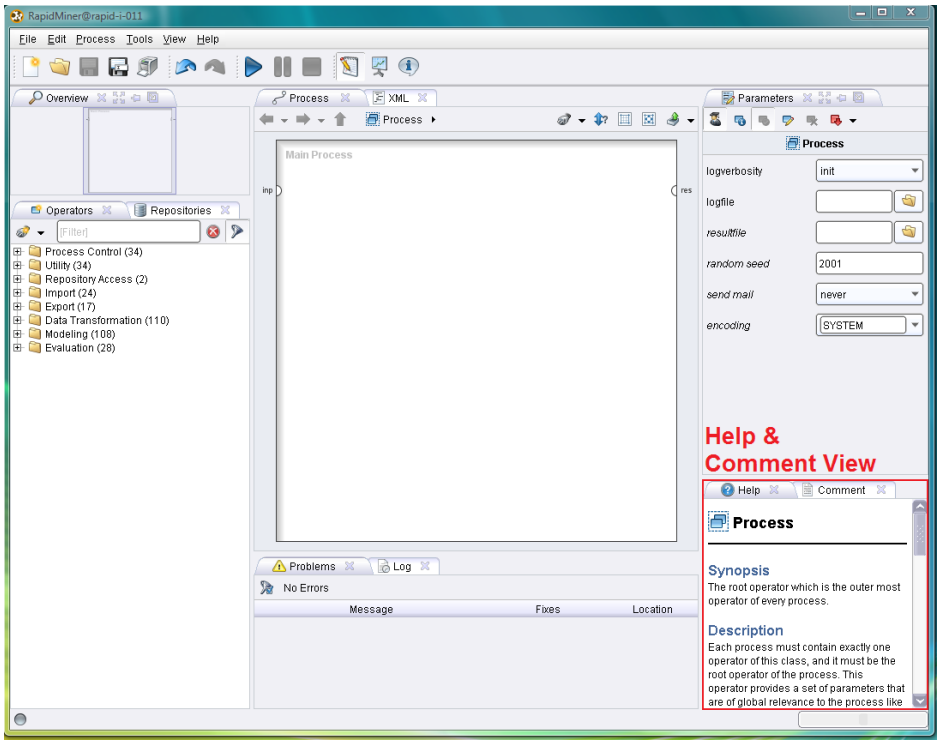


Figure 2.27: Help texts are shown both for operators currently selected in the Operators View and for those from the Process View shown.

operator and write any text on it in the comment field. This will then be saved together with your process definition and can be useful for tracing individual steps in the design later on. The fact that a comment is available for an operator is indicated by a small text icon at the lower edge of the operator.

2.3.7 Overview View

Particularly in the case of extensive processes, the white work area will no longer be sufficient and will be enlarged either via the context menu of the Process View, by means of the key combinations of CTRL and the arrow pointing left, right, upwards and downwards or simply by dragging an operator to the edge. In this case however, the entire work area will no longer be visible at the same time and navigation within the process will be made more difficult. In order to

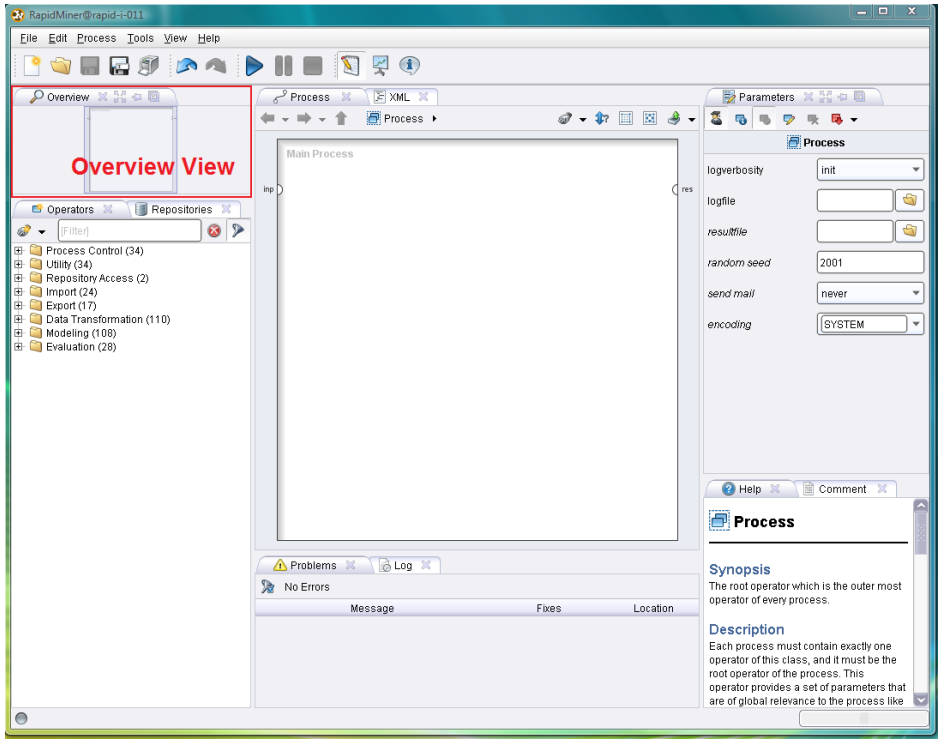


Figure 2.28: Keep track by using the Overview View.

improve the overview and provide a comfortable way of navigating at the same time, RapidMiner offers the Overview View, which shows the entire work area and highlights the currently displayed section with a small box:

You will see that the section moves within the Process View when scrolling - now using the scrollbar or simply by dragging an operator to the edge of the section. Or you can simply drag the highlighted area in this overview to the desired place and the Process View will adjust automatically.

2.3.8 Problems and Log View

Fig. 2.30 shows the Problems and Log View of RapidMiner.

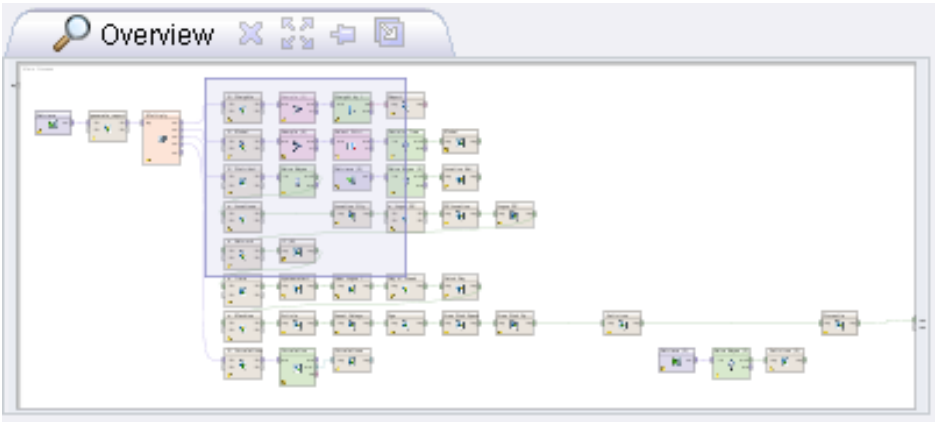


Figure 2.29: The Overview View shows the entire process and highlights the visible section.

Problems View

A further very central element and valuable source of help during the design of your analysis processes is the Problems View. Any warnings and error messages are clearly indicated in a table here (Fig. 2.31).

In the first column with the name "Message" you will find a short summary of the problem. In this case the data mining method "Gaussian Process" is not able to handle polynomial (multivalued categorical) attributes. The last column named "location" shows you the place where the problem arises in the form of the operator name and the name of the input port concerned. Please also consider what is on the right-hand side of the Problems View toolbar. You can activate a filter with this, meaning that only the problems of the operator currently selected are indicated. This is extremely practical for larger process with several error sources.

A considerable innovation of RapidMiner 5 however is the possibility of also suggesting solutions for such problems and of implementing them directly. These solution methods are called **quick fixes**. The second column gives an overview of such possible solutions, either directly as text if there is only one possibility of solution or as an indication of how many different possibilities exist to solve the problem. In the example above there are two different possibilities for handling the second problem. But why is this solution suggestion called "quick fix"? Just

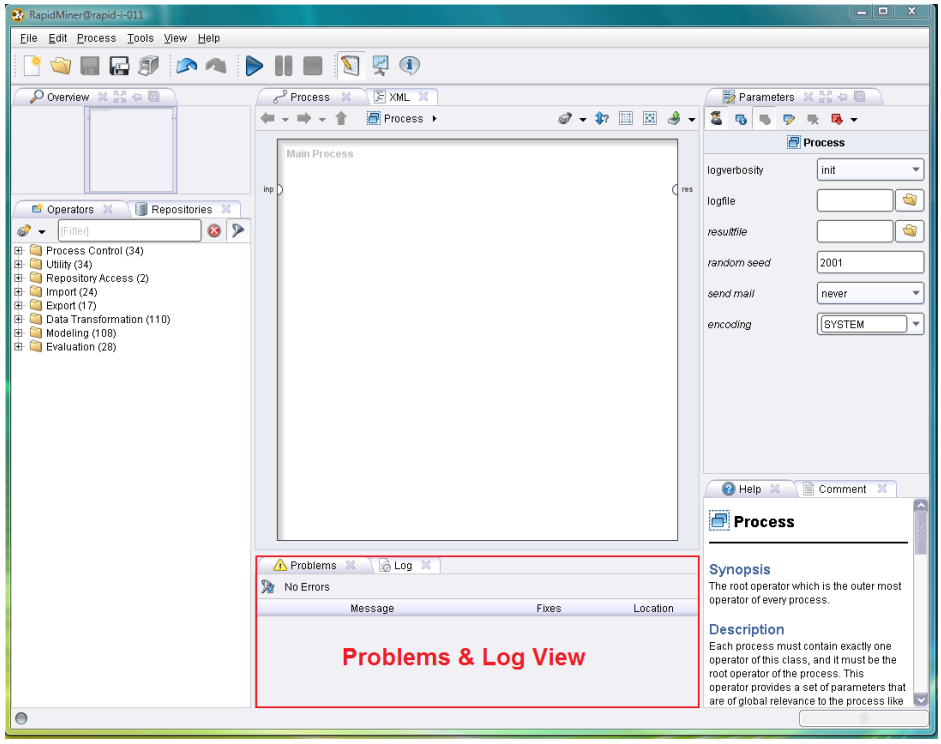


Figure 2.30: The table in the Problems View clearly indicates all (potential) problems in the design and also gives tips for the solution in many cases (quick fixes). You will find further information in the Log View.

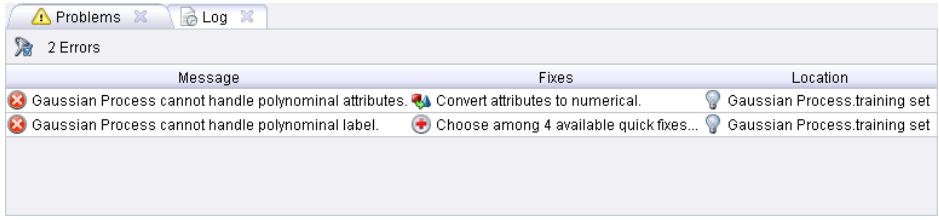


Figure 2.31: Representation of all current problems.

try double clicking on the relevant quick fix field in the table in such a case. In the first case the solution suggestion would be directly executed and a relevant operator automatically configured and inserted in such a way that the necessary pre-processing is performed.

In the second case with several possibilities of solution a dialog would appear asking you to select the desired solution method. In this case too, once one of the possibilities is selected, one or more necessary operators would be configured and inserted in such a way that the problem no longer arises. In this way you can recognise problems very early on and, with just a few clicks, very comfortably eliminate them as early as during the design process.

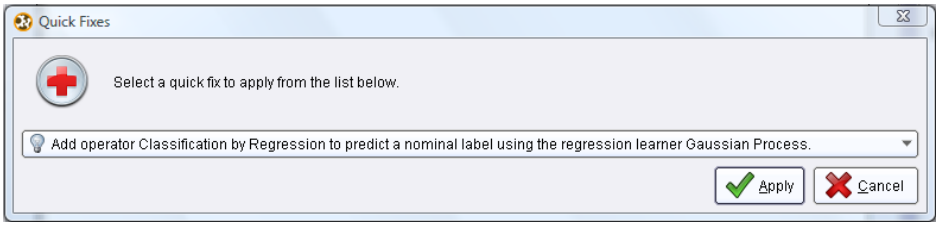


Figure 2.32: Selection dialog in the case of several possible quick fixes.

Note: The determination of potential problems as well as the generation of quick fixes are among the functions of RapidMiner that are dependent on meta data being supplied correctly. We strongly recommend you use the repository, since otherwise (e.g. in the case of direct reading of data from files or databases) the meta data will not be available and these supports will therefore not be offered.

Log View

During the design, and in particular during the execution of processes, numerous messages are written at the same time and can provide information, particularly in the event of an error, as to how the error can be eliminated by a changed process design.

You can copy the text within the Log View as usual and process it further in other applications. You can also save the text in a file, delete the entire contents or search through the text using the actions in the toolbar.

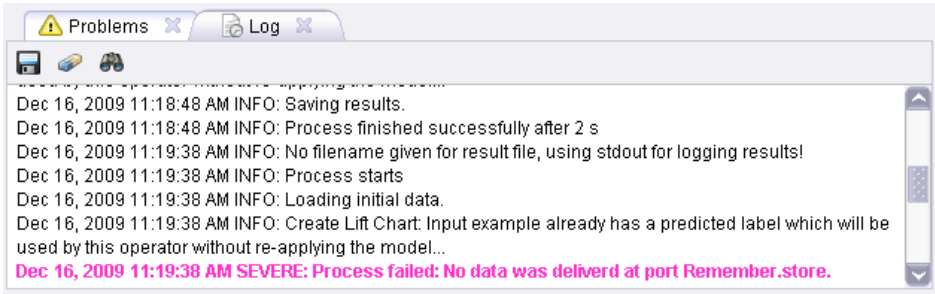


Figure 2.33: Further information, particularly on process execution and in the event of an error, can be found in the Log View.

3 Execution of Analysis Processes with RapidMiner

We became acquainted with the fundamental elements of the graphical user interface of RapidMiner in the last chapter, such as perspectives and views, and discussed the most important aspects of the Design Perspective of RapidMiner. We would now like to make use of the new possibilities in order to define and execute an initial simple analysis process. You will soon realise how very practical it is that, with RapidMiner, you do *not* need to perform the process again for every change in order to determine the effect of the change. But more on that later.

3.1 Creating a New Process

Whether you now select the action "New" from the Welcome Perspective, the "New" icon on the far left-hand side of the main tool bar of RapidMiner or the associated entry in the "File" menu: A new analysis process is created in each case which you can work on in the following. Before you can start however, the "Repository Browser" (Fig. 3.1) appears and asks you to give a storage location for your new process.

Simply select a repository and a location, i.e. a directory where you would like to save the new process. New directories can be created via the context menu of repository entries or even of the repository itself. After you have selected the location, give your process a name and confirm your choice with "Ok".

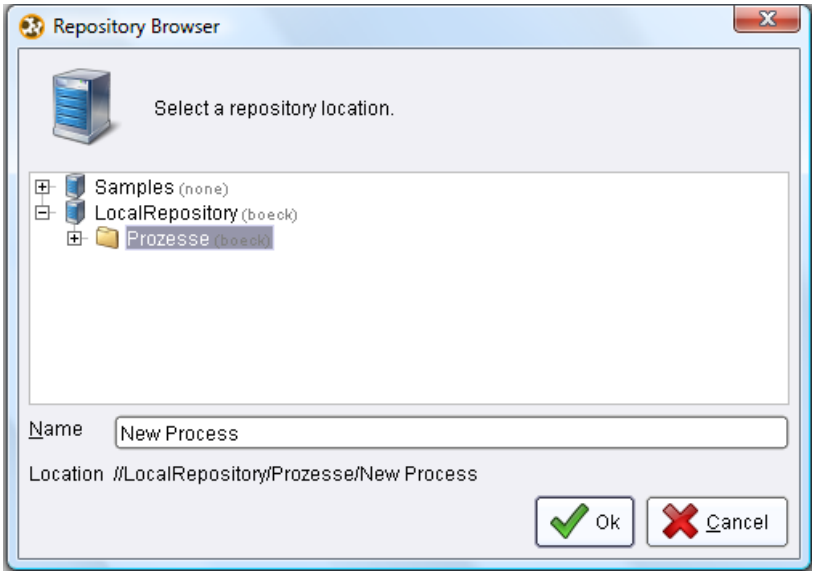


Figure 3.1: The Repository Browser serves for the selection of repository entries or storage locations similarly to the file dialogs known from operating systems.

Tip: You can also create a new process without generating an entry in the repository by closing the repository browser with "Cancel". This is not recommended however, since the remaining repository entries, like those for data, are defined in relation to the process. This makes executing the process on servers in the network easier as well as forwarding it to other analysts or copying it for other computers. We recommend always creating a repository entry also for new processes.

In principle, you are completely free in how you structure your repository. In the context menu of the entries in the repository browser and also in the repository view you will find all necessary entries for the administration of your data and processes, as you can see them in Fig. 3.2.

The details of the actions are as follows:

1. *Store Process Here:* Stores the current process in the given location,
2. *Rename:* Renames the entry or the directory,
3. *Create Folder:* Creates a new directory in this place,

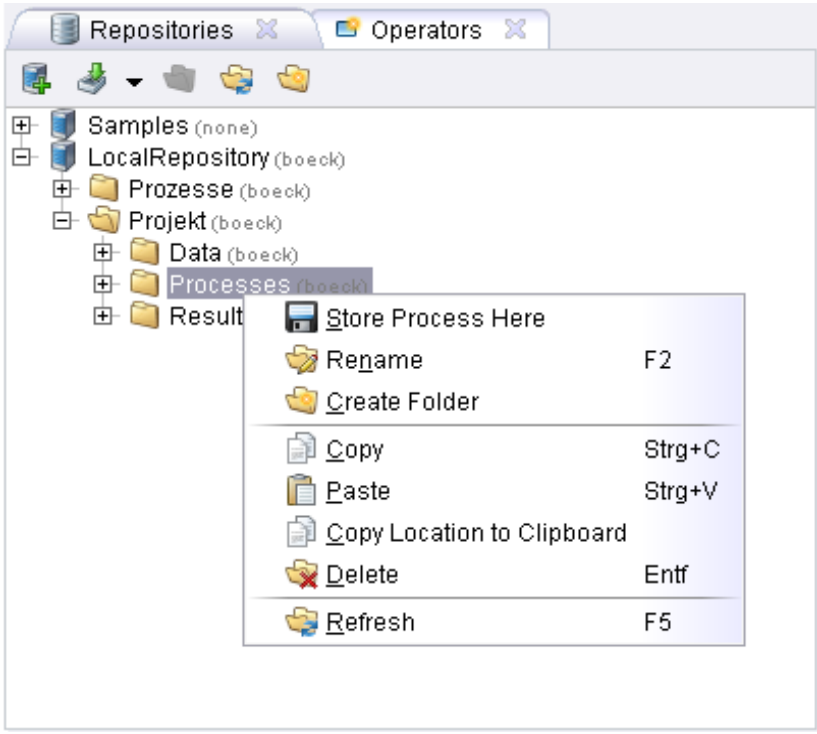


Figure 3.2: The context menu of the repository entries, both in the repository browser and in the repository view, offers all necessary options for administration.

4. *Delete:* Deletes the selected repository entry or directory,

5. *Copy:* Copies the selected entry for later insertion in other places,

6. *Paste:* Copies a previously copied entry to this place,

7. *Copy Location to Clipboard:* Copies a clear identifier for this entry to the clipboard, meaning you can use this as a parameter for operators, in web interfaces or the like,

8. *Refresh:* Updates the display.

It is recommended that you create new directories in the repository for individual analysis projects and name these accordingly. It will never hurt to structure further within the projects, e.g. structuring into further subdirectories for project-

specific data, different phases of data transformation and analysis or for results. A repository could thus have the following structure for example:

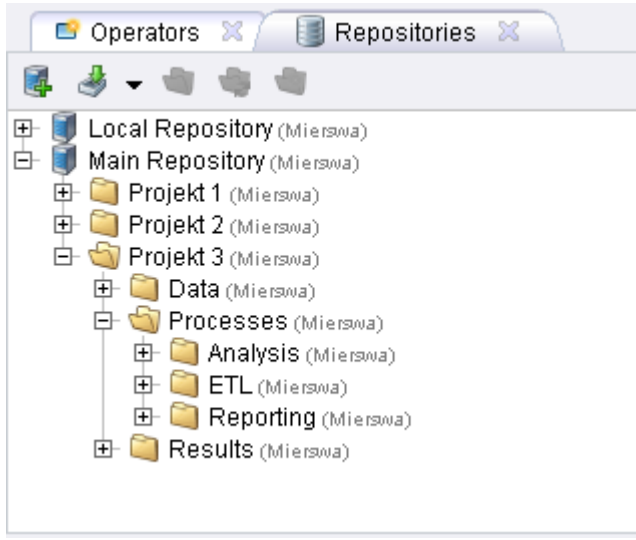


Figure 3.3: A repository with structuring into projects and each of these structured according to data, processes and results.

3.2 The First Analysis Process

After you have defined the location and name of the process, RapidMiner automatically switches to the Design Perspective and you can start with the process design. In later chapters we will talk in detail about how you can load data in RapidMiner and store it in your repository. In this section however, the basic execution of processes is more important to us and we will therefore wait a short while before analysing real data.

As long as you have not changed the selection and positions of the individual views for the Design Perspective, your screen should more or less look like the one in 3.4.

We will now begin our new process starting with the generating of data which we can work on. As already said: We will see in later chapters how we can use data from the repository or even import it directly from other data sources such as

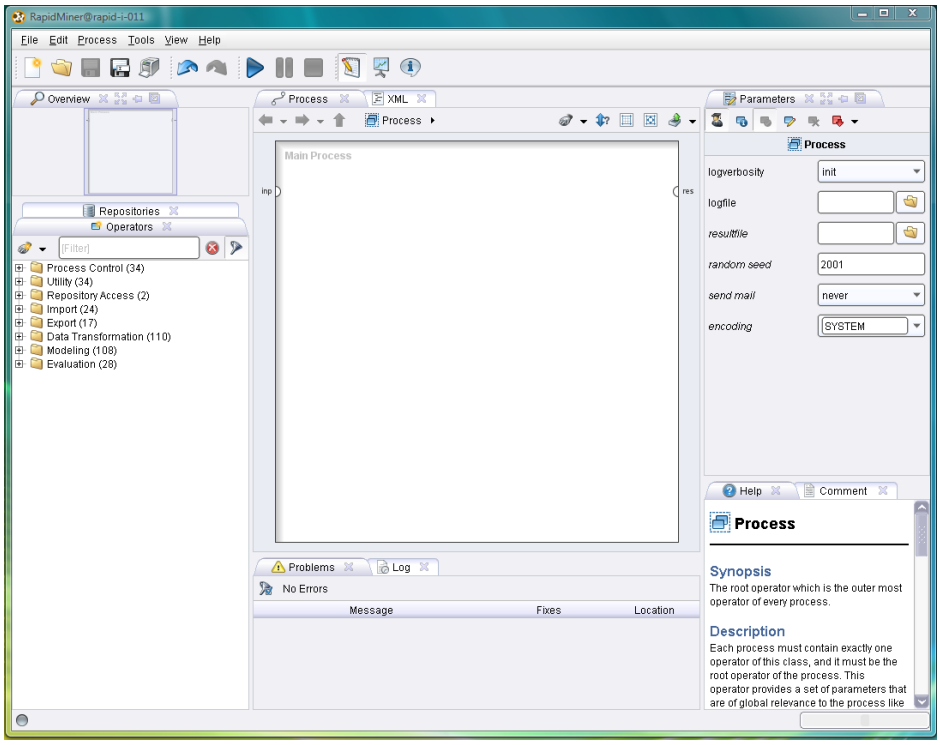


Figure 3.4: The preset Design Perspective immediately after the creation of a new process.

databases or files by using operators. But for the moment we will put this aside and generate a small synthetic data set.

Please now expand the group "Utility" in the Operators View and then "Data Generation". The numbers in brackets next to the individual groups indicate the number of operators for this group. You should now see several operators that can be used for generating an artificial data set. This includes the operator "Generate Sales Data". Now drag this operator onto the white area while holding down the mouse button and release it there. The operator will be inserted and also directly connected depending on the automatic connection setting in the Operators View. If this does not happen, you can manually connect the output port of the new operator with the first result port of the entire process on the right-hand side of the white work area. Alternatively, it would of course have also been possible to insert the operator by means of the New Operator Dialog, as described in the previous chapter. Either way, the result ought to look roughly

like this:

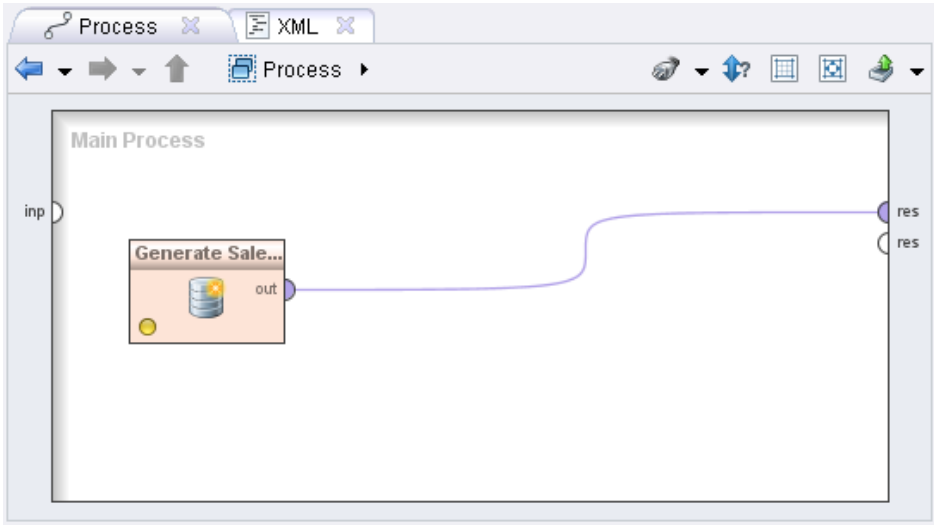


Figure 3.5: An initial and admittedly very simple process, which generates some data and displays the result in the Result Perspective.

As you have surely noticed, the full name of this operator, "Generate Sales Data", is too long and is cut off after the first few letters. Move the mouse onto this operator and stay there for a few moments. The name will now be shown in full in a small animation. Of course, you could also rename the operator and give it a shorter name, although you would then miss out on the fancy animation:

Figure 3.6: Long names are shown if the cursor remains still on an operator for a while.

As you can see, the status indicator of the operator at the bottom left-hand side is yellow. This means that the operator has not produced any errors, but has also not yet been successfully executed. So you have only fully configured the operator thus far, but this does in no way mean it has been directly executed. You can easily see that from the fact that the status indicator then turns green. Had you not noticed that you have already configured the operator? Indeed, the

configuration was very simple in this specific case: It was not at all necessary to set any operator parameter. A red status indicator and entries in the Problems View would have indicated such a configuration need.

3.2.1 Transforming Meta Data

We will now deal with one of the most fascinating aspects of RapidMiner, namely the ability to compute the output of an operator or process beforehand and to even do this during the design time, so without having to load the actual data or even perform the process. This is made possible by the so-called *meta data transformation* of RapidMiner.

Of course, each operator defines the way in which the received input data is transformed. This is its task at the end of the day. The special thing about RapidMiner however is that this can not only be done for actual data but also for the meta data about this data. This is typically much less voluminous than the data itself and gives the analyst an excellent idea of which characteristics a particular data set has. The meta data in RapidMiner essentially equates to the concept descriptions we discussed previously. It contains the attribute names of the example set as well as the value types and the roles of the attributes and even some fundamental statistics.

So much for the theory, but what does the meta data look like in practice i.e. in RapidMiner? In RapidMiner the meta data is provided at the ports. Just go over the output port of the recently inserted operator with the cursor and see what happens:

A tooltip will appear which describes the expected output of the port. First the name of the operator and of the port followed by the kind of meta data. In this case we are dealing with the meta data of an example set. The number of the examples can also be inferred (100) as well as the number of the attributes (8). Then there comes a description of the path the object would have taken through the process during an execution. In this case the path has only one station, i.e. the port of the generating operator. However, the most important part of the meta data (at least for an example set) is the table which describes the meta data of the individual attributes. The individual columns are:

1. *role:* The role of the attribute. If nothing is indicated then it is a regular

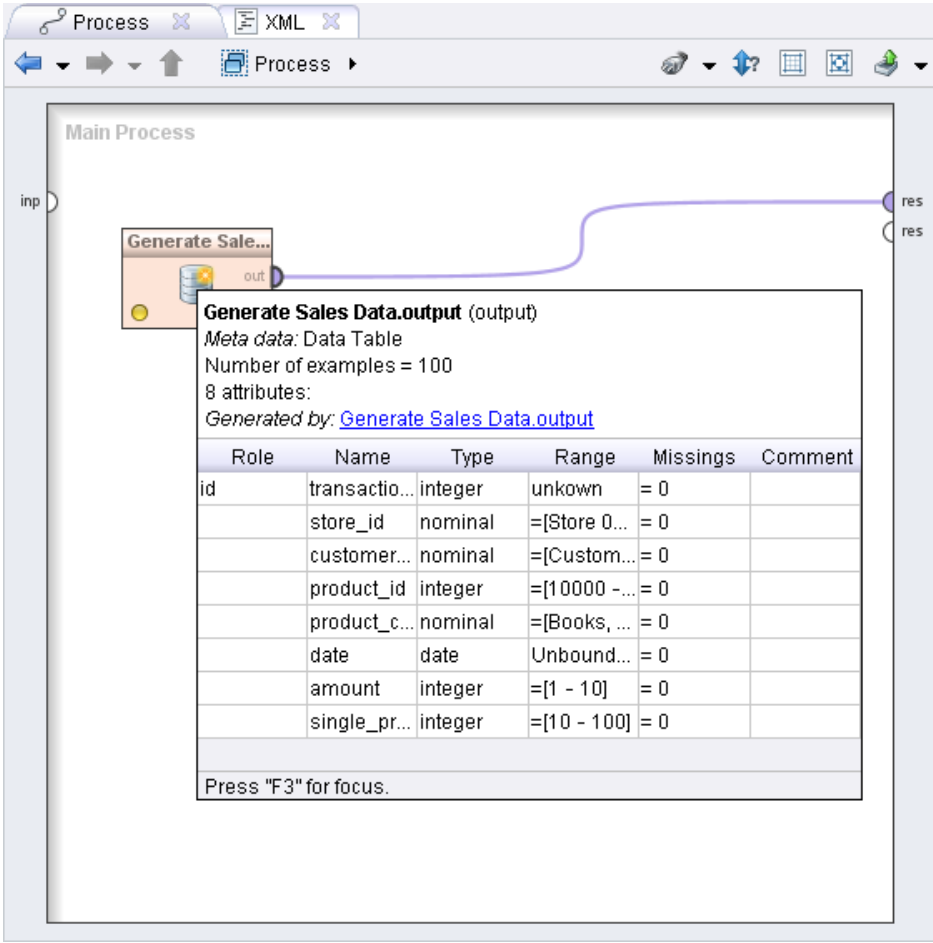


Figure 3.7: The meta data of the output port of the operator "Generate Sales Data".

attribute,

2. *name:* The name of the attribute,

3. *type:* The value type of the attribute,

4. *range:* The value range of the attribute, so the minimum and maximum in the case of numerical attributes and an excerpt of possible values in the case of nominal attributes,

5. *missings:* The number of examples where the value of this attribute is unknown.

Tip: There are such tooltips of a higher complexity in several places in RapidMiner, also for the operator descriptions for example, which are shown as a tooltip in the Operators View. You can take time to read the tooltip and also adjust it in terms of size if you press key F3 beforehand.

Please note that the meta data can often only represent an estimation and that an exact indication is not always possible. This is explained by the fact that parts of the meta data are unknown or can only be indicated inaccurately, e.g. with the indication "<100 Examples" for the number of the examples. Nevertheless, the meta data is a valuable source of help both for forthcoming design decisions and for the automatic recognition of problems as well as the suggestions for their solution i.e. quick fixes.

Back to our example. Trained analysts will recognise straight away that the data must be so-called transaction data, where each transaction represents a purchase. We gave the following attributes for our example set:

- *transaction_id:* Indicates a clear ID for the respective transactions,
- *store_id:* Indicates the store where the transaction was made,
- *customer_id:* Indicates the customer with whom the transaction was made,
- *product_id:* Indicates the ID of the product bought,
- *product_category:* Indicates the category of the product bought,
- *date:* Indicates the transaction date,
- *amount:* Indicates the number of objects bought,
- *single_price:* Indicates the price of an individual object.

If we look at the last two attributes first, then we see that, whilst the number and the individual price of the objects are given within the transaction, the associated total turnover however is not. Next we therefore want to generate a new attribute with the name "total_price", the values of which correspond to the product from amount and single price. For this we will use a further operator named "Generate

Attributes", which is located in the group "Data Transformation" – "Attribute Set Reduction and Transformation" – "Generation". Drag the operator behind the first operator and connect the output port of the data generator with the input port of the new operator and connect the output port of the latter with the result output of the total process. The screen ought to then look roughly like in Fig. 3.8:

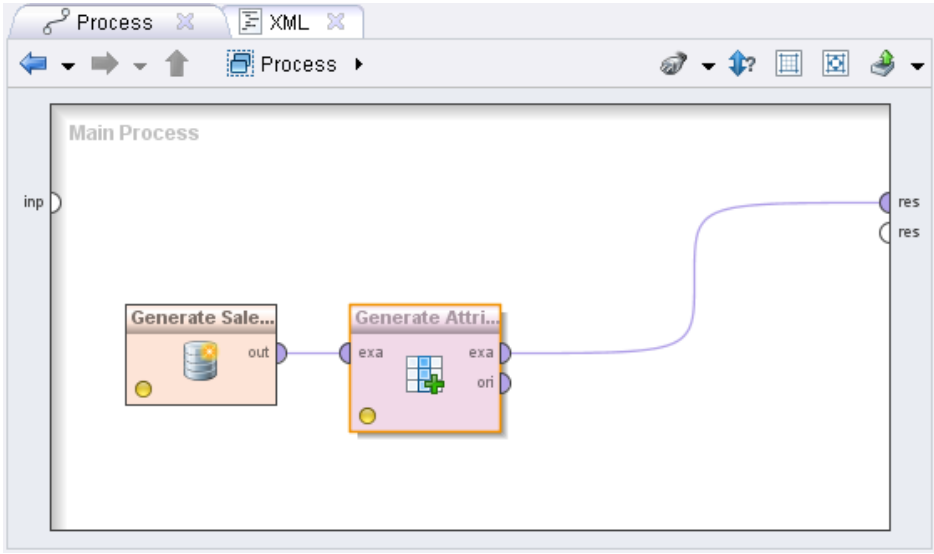


Figure 3.8: The data is generated first and then a new attribute is produced.

Tip: Instead of dragging an operator into the Process View and reconnecting the ports, you can also drag the operator onto an existing connection. If you move the cursor position exactly onto the connection, the latter will be highlighted and the new operator will be inserted directly into the connection.

Even if this process would work now, which is visible from the yellow status indicators and the empty Problems View, then the second operator would not compute anything without a further configuration and the final result would only be the same as that after the first operator. We therefore choose the new operator "Generate Attributes" and select it in this way. The display in the parameter view changes accordingly and the parameters of this operator are shown. The substantial parameter has the name "function descriptions" and is configured on the associated button with one click, as can be seen in Fig. 3.9:

After you have pressed the button with the name "Edit List (0)", a dialog will

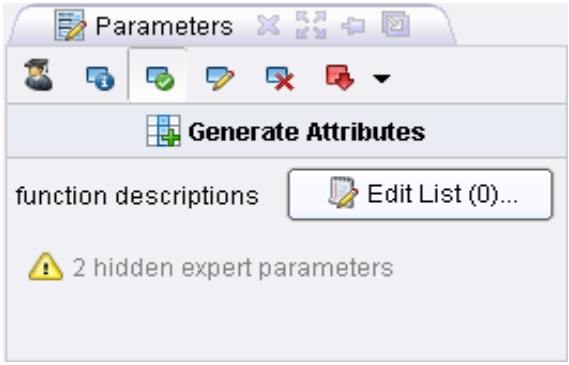


Figure 3.9: The parameters of the operator "Generate Attributes".

open giving you the opportunity to enter the desired computation in Fig. 3.10.

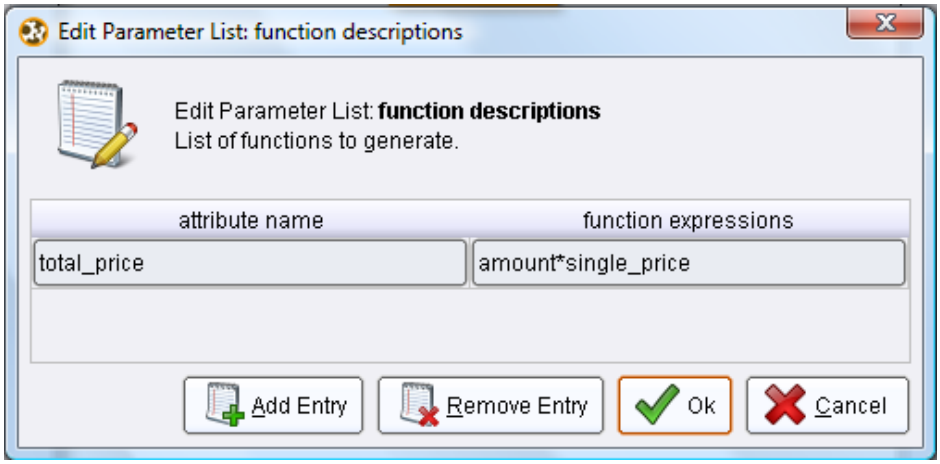


Figure 3.10: Computation of the new attribute "total_price" as a product of "amount" and "single_price".

You can add further entries in such lists of individual parameters with the two actions "Add Entry" and "Remove Entry" and also delete selected entries. The names of the desired parameters are in the table heading. Add a row, enter the name of the new attribute on the left and enter the function on the right which computes this new attribute. In this case it is simply the product of two other attributes. Confirm your input with "Ok" and the dialog will close. The button that says "Edit List" ought to now show a "1" in brackets, meaning that you can see how many entries the parameter list has and therefore in this case how many

new attributes are generated also.

We can now observe what effect the addition of the operator “Generate Attributes” has on the meta data. RapidMiner has already transformed the meta data in the background and you can see the new meta data as a tooltip via the output port of the operator (Fig. 3.11).

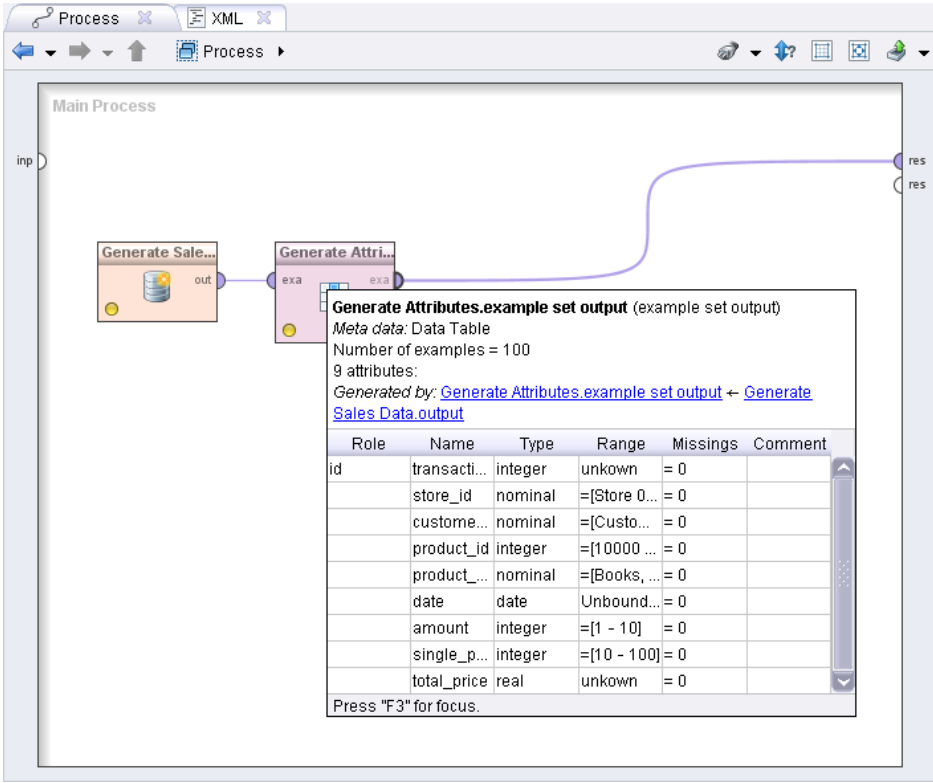


Figure 3.11: The meta data contains the complete path of the object and remains, with the exception of the newly added attribute “total_price”, unchanged.

It is easy to see in the line “Generated by” that the last thing the object stemmed from now is the operator “Generate Attributes” and was previously the operator “Generate Sales Data”. In addition, hardly anything has changed - both the number of the examples and the eight original attributes have stayed the same. However, a ninth attribute has been added: Our newly defined attribute “total_price” can also now be found in the table.

And our process has still not been executed, as you can see just by looking at the status indicators which are still yellow. You may now ask yourself: "And? So I know the result beforehand and without process execution. What do I get from that?". Well, rather a lot. You can now see at a glance what a particular operator or (sub)process is doing with the input data. Since the meta data is also considerably smaller than the complete data sets, this examination can also be performed much faster than on the complete data. This way you get feedback in the shortest time as to whether there is a problem which may make further data transformation necessary and not only after an analysis process lasting several hours has aborted with an error. And last but not least, RapidMiner can continue processing the information from the meta data and continue supporting you in the design of the process, e.g. through only all attributes that are still available (and newly generated) being displayed on the graphical user interface while attributes are being filtered.

Now try the following for example: Open the group "Data Transformation" – "Attribute Set Reduction and Transformation" – "Selection" and drag the operator named "Select Attributes" into the process - ideally directly onto the connection after the last operator. Remember that the connection must be highlighted before you drop the operator, but then it will be correctly reconnected immediately. You should have now defined the process as in Fig. 3.12.

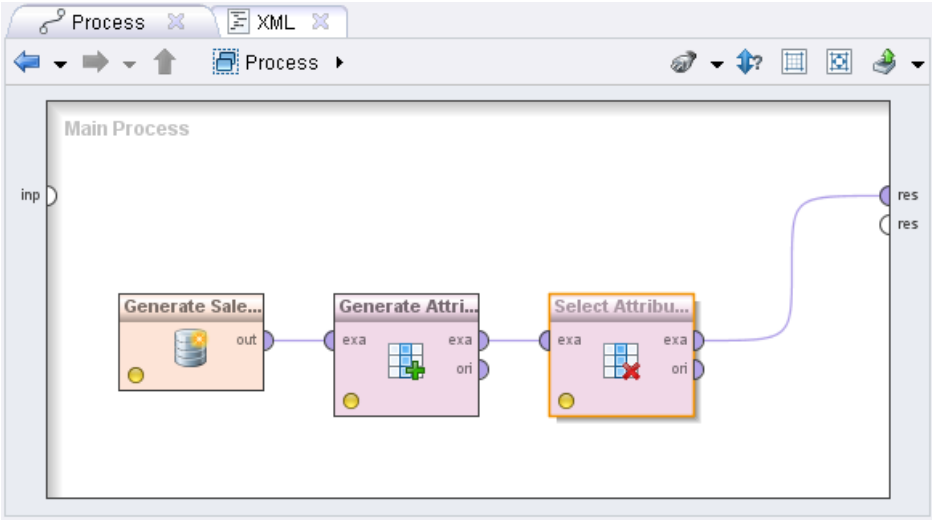


Figure 3.12: Generation of data, generation of a new attribute, selection of a subset of attributes.

Select the new operator and select the option “subset” in its parameters for the parameter “attribute filter type”. Please note that a further parameter named “attributes” has now appeared. This is in bold, so you would need to define it before you could perform the process. You can also see this from the red status indicator of the operator as well as from the entry in the Problems View. You could now choose the quick fix in the Problems View by double-clicking or simply configure the parameter “attributes”: Again by clicking on a button, this time the one that says “Select Attributes...”. The parameters should be like in Fig. 3.13.

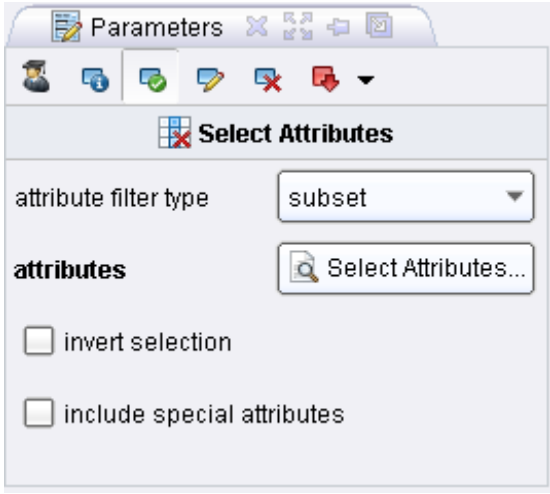


Figure 3.13: The parameter “attributes” only appears if “subset” has been chosen as the filter type.

Now press the button that says “Select Attributes...” and select the attributes “product_category”, “store_id” and “total_price” from the list in the dialog (Fig. 3.14) that appears either by double-clicking or by pressing the button in the centre with an arrow pointing to the right.

Did you notice? The new attribute “total_price”, which had so far only been computed within the meta data transformation, was already ready for you to select here - without you ever having executed the process. If you examine the meta data at the output port again, you will see that only the three selected attributes are left plus the transaction ID, which also has a special role (that of ID) and was therefore not affected by the selection. Since we would like to remove this ID too, select the option “include special attributes” in the parameters of

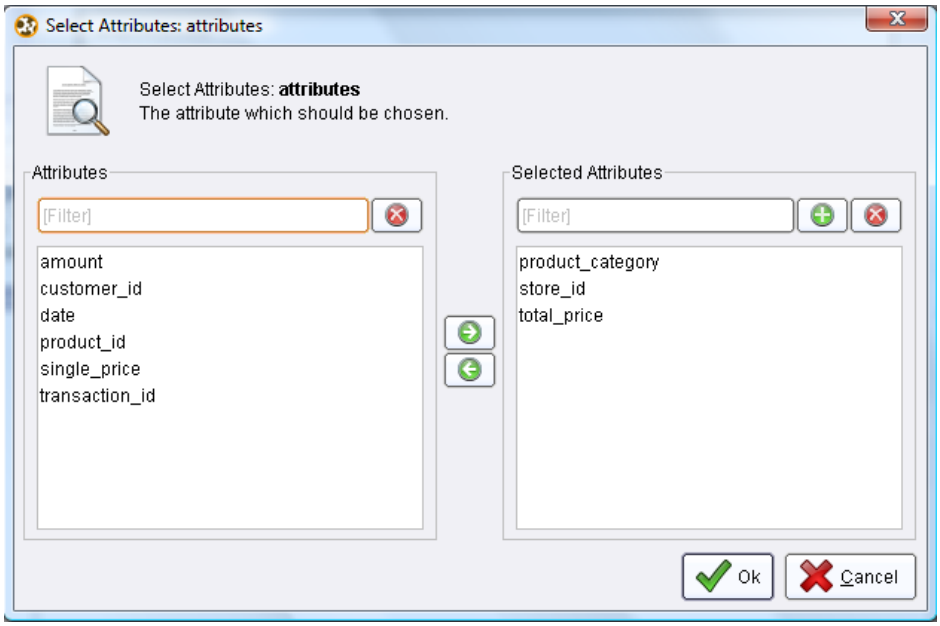


Figure 3.14: Individual attributes or subsets can be selected or even deleted with the operator "Select Attributes".

the operator "Select Attributes" and examine the meta data again: Now only the three desired attributes are left. You can find out the effects of these and all other parameters in the description of parameters in the Help View and also in the operator reference.

Tip: It is a basic rule of RapidMiner that operators from the group "Data Transformation" are usually only executed on regular attributes, so on those without a special role. However, the operators offer an option called "include special attributes" for this, meaning that the changes are also applied to those with a special role.

3.3 Executing Processes

Now we are ready and want to execute the process just created for the first time. The status indicators of all operators should now be yellow and there should be no entries in the Problems View. In such a case it should be possible to execute

our process consisting of the three operators (for generating data, computing the total turnover for each transaction and filtering attributes) without any problems.

You have the following options for starting the process:

1. Press the large play button in the toolbar of RapidMiner,
2. Select the menu entry “Process” – “Run”,
3. Press F11.

Figure 3.15: The play button starts the process, you can stop the process in between with the pause button and stop aborts the process completely.

While a process is running, the status indicator of the operator being executed in each case transforms into a small green play icon. This way you can see what point the process is currently at. After an operator has been successfully executed, the status indicator then changes and stays green - until for example you change a parameter for this operator: Then the status indicator will be yellow. The same applies for all operators that follow. This means you can see very quickly which operators a change could have an effect on.

The process defined above only has a short runtime and so you will hardly have the opportunity to pause the running process. In principle however you can briefly stop a running process with the pause symbol e.g. in order to see an intermediate result. The operator currently being executed is then finished and the process is then stopped. You can recognise a process that is still running but currently paused by the fact that the colour of the play icon changes from blue to green. Press the play button again to continue executing the process further.

If you do not wish to merely pause the process but to abort it completely, then you can press the stop button. Just like when pausing, the operator currently being executed is finished and the process fully aborted immediately after. Please note that you can switch to the Design Perspective immediately after aborting the process and make changes to processes, even if the execution of the current operator is being finished in the background. You can even start further processes and do not need to wait for the first process to be completed.

Note: It was explained above that the operator being executed is always completed if you abort. This is necessary to ensure a sound execution of operators. However, completing an operator may need much more time in individual cases and also require other resources such as storage space. So if when aborting very complex operators you can see this taking hours and requiring additional resources, then your only option is to restart the application.

3.3.1 Looking at Results

After the process was terminated, RapidMiner should have told you that new results are available and asked you whether to go to the Result Perspective. If this was not the case, then you probably did not connect the output port of the last operator with one of the result ports of the process on the right-hand side. Check this and also check for other possible errors, taking the notes in the Problems View into consideration (Fig. 3.16).

Feel free to spend a little time with the results. Since the process above has not yet performed any modelling but only transformed data, the result only consists of an example set. You can look at the meta data of this data set and try out the table plus some of the visualisations in the plot view. In the next chapter we will talk in detail about the possibilities of the Result Perspective. If you wish to return to the Design Perspective, then you can do this at any time using the switching methods you are familiar with.

Tip: After some time you will want to switch frequently between the Design Perspective and the Result Perspective. Instead of using the icon or the menu entries, you can also use keyboard commands F8 to switch to the Design Perspective and F9 to switch to the Result Perspective.

3.3.2 Breakpoints

Meta data transformation is a very powerful tool for supporting the design of analysis processes and making it much more comfortable. There is simply no longer the necessity to perform the process more often than needed for test purposes during design. In fact, the expected result can already be estimated on the basis of the meta data. Thus meta data transformation and propagation ought to revolutionise the world of data analysis a little: instead of having to perform

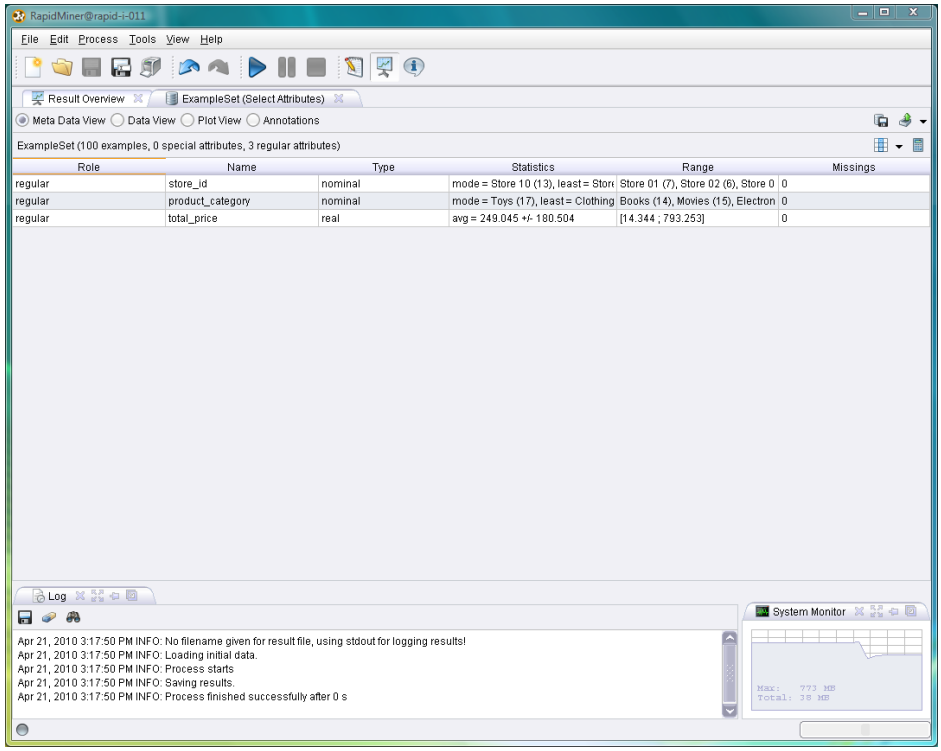


Figure 3.16: After a process has been successfully executed you can look at the results in the Result Perspective.

each step separately as before in order to configure the next operator, the results of several transformations can now be foreseen directly without any execution. This is of course an enormous breakthrough, in particular for the analysis of large data sets.

Nevertheless the necessity arises in some cases of going beyond the meta data and seeing a specific result in all its details. When the design is running it is usually no problem to place the desired (intermediate) result at a result port of the process and to execute the process very simply. The desired results are then shown in the Result Perspective. But what can you do if the process has already finished being designed and all output ports are already connected? Or if the intermediate result is deep inside an intricate subprocess? There is of course a sophisticated solution in RapidMiner for this too, which does not make any process redesign necessary. You can simply insert a so-called breakpoint by selecting one of the options "Breakpoint Before" or "Breakpoint After" from the context menu of an operator, as shown in Fig. 3.17.

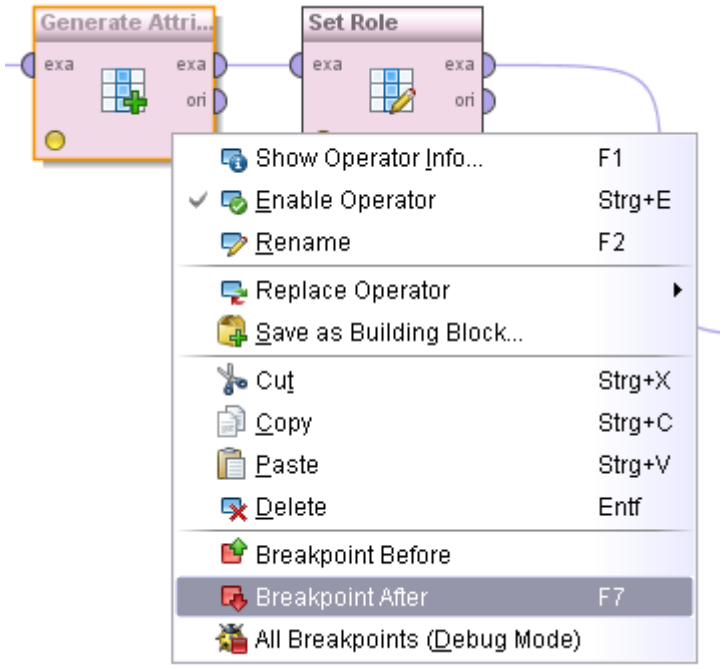


Figure 3.17: You can stop the process cycle using breakpoints and examine intermediate results.

If a breakpoint was inserted after an operator for example, then the execution of the process will be interrupted here and the results of all *connected* output ports will be indicated in the Result Perspective. This means you can look at these results without having to make further changes to the process design. A breakpoint before an operator functions similarly to a breakpoint after an operator: In this case the process will be interrupted before the execution of this operator and the objects next to the *connected* input ports of this operator are indicated. The fact that a breakpoint is next to an operator is indicated by a small red symbol at the lower edge of the operator (Fig. 3.18).

Figure 3.18: A breakpoint is defined before or after this operator.

Tip: The use of "Breakpoint After" in particular is relatively frequent, which is why this action also has a keyboard shortcut. You can add a breakpoint after the operator currently selected or remove all breakpoints currently present by pressing key F7.

Depending on whether you have configured RapidMiner accordingly, RapidMiner automatically switches to the Result Perspective in the case of a breakpoint and shows the intermediate results. Alternatively, you can simply switch to the Result Perspective yourself. You can see that you are in a breakpoint at this time and not for example at the end of the process by looking at two indicators: First of all, the status indicator in the bottom left-hand corner of the main window of RapidMiner shows a red light, i.e. a process is running but is not being actively executed at present. If no process at all were running at present, then this indication would just be grey. The second indicator for a breakpoint is the play symbol which is now green instead of blue:

Figure 3.19: The green play symbol indicates that the process is currently in a breakpoint and can continue being executed if pressed.

The process can now be started again simply by pressing the green play symbol and continue being executed until complete or until the next breakpoint. You can of course abort the process completely as usual by pressing stop.

4 Displaying Data and Results

In the previous sections we have seen how the graphical user interface of RapidMiner is built up and how you can define and execute analysis processes with it. At the end of such a process the results of the process can then be indicated in the Result Perspective. Switch now to this Result Perspective by clicking once in the toolbar. This will be dealt with in detail within this chapter. Depending on whether you have already produced representable results, you should now see, in the original settings at least, roughly the screen shown in 4.1.

Otherwise, you can recreate this preset perspective under "View" – "Restore Default Perspective" as always. The Result Perspective is the second central working environment of RapidMiner alongside the Design Perspective already discussed. We have already discussed the Log View at the bottom and the repository on the top right. In this chapter we will therefore focus on the remaining components of the perspective.

4.1 System Monitor

The system monitor, which you will find in the preset perspective at the bottom on the right, is a simple memory monitor which gives you an overview of the memory currently in use. Although RapidMiner already tries (by way of numerous measures such as going without data copies and using views instead) to reduce the memory requirement, data analysis remains a field with a high memory requirement in many cases. The memory monitor indicates the maximum memory available in RapidMiner ("Max") and the maximum memory usable at

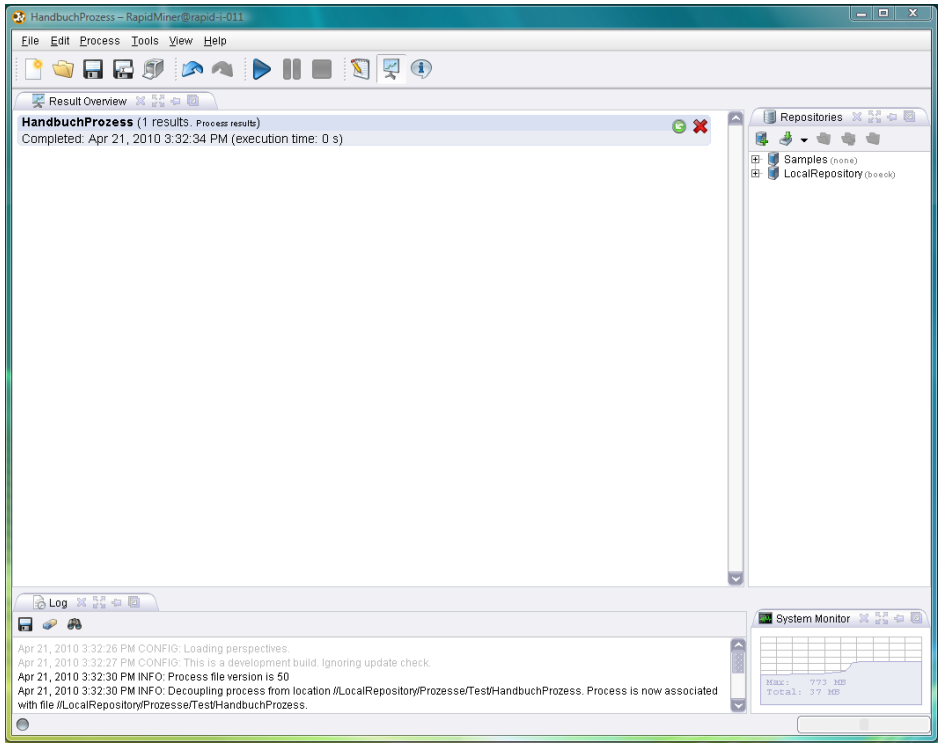


Figure 4.1: Result Perspective of RapidMiner.

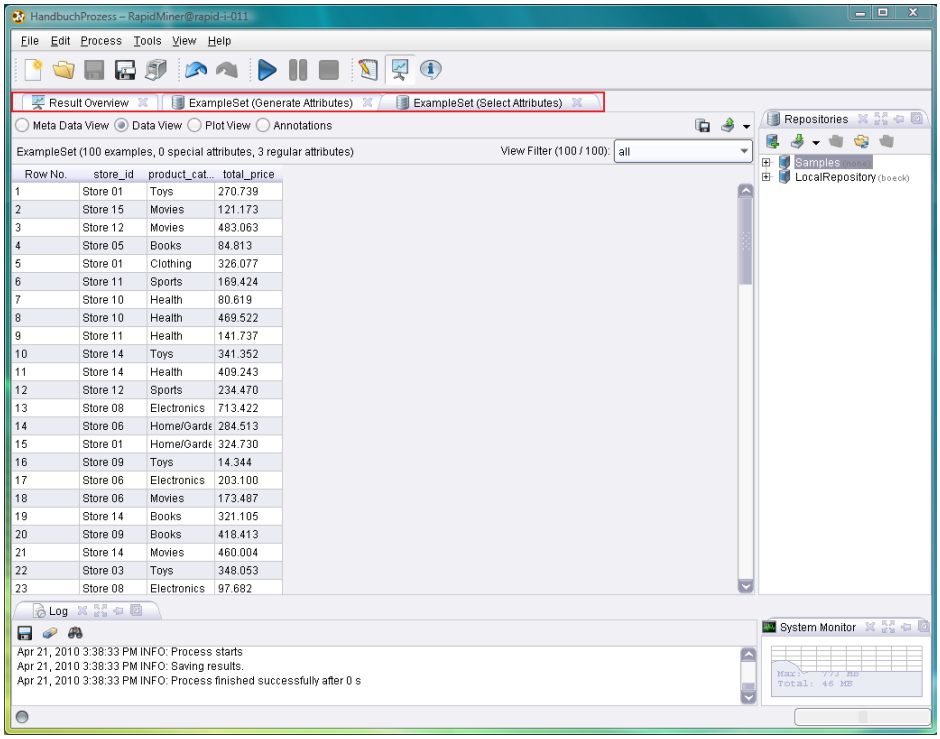
present ("Total"). The latter corresponds to the upper line of the monitor and can be increased as high as the absolute maximum "Max" if necessary. This is done automatically and when possible, only if necessary. If the memory monitor is completely filled, then the quantity indicated in "Total" is used. If this is just as high as "Max", then RapidMiner is at the absolute limit and would have to abort the process in the case of even greater memory requirement.

It is often possible to still perform such a process through skilful pre-processing, batch treatment, use of views or skilful memory management within RapidMiner. This is a field for specialists however and is therefore not part of this user manual.

4.2 Displaying Results

We have already seen that objects which are placed at the result ports at the right-hand side of a process are automatically displayed in the Result Perspective after the process is complete. The large area on the top left-hand side is used here, where the Result Overview is also already displayed, which we will discuss at the end of this chapter.

Each currently opened and indicated result is displayed as an additional tab in this area:



Row No.	store_id	product_cat...	total_price
1	Store 01	Toys	270.739
2	Store 15	Movies	121.173
3	Store 12	Movies	483.063
4	Store 05	Books	84.813
5	Store 01	Clothing	326.077
6	Store 11	Sports	169.424
7	Store 10	Health	80.619
8	Store 10	Health	469.522
9	Store 11	Health	141.737
10	Store 14	Toys	341.352
11	Store 14	Health	409.243
12	Store 12	Sports	234.470
13	Store 08	Electronics	713.422
14	Store 06	Home/Garde	284.513
15	Store 01	Home/Garde	324.730
16	Store 09	Toys	14.344
17	Store 06	Electronics	203.100
18	Store 06	Movies	173.487
19	Store 14	Books	321.105
20	Store 09	Books	418.413
21	Store 14	Movies	460.004
22	Store 03	Toys	348.053
23	Store 08	Electronics	97.682

Figure 4.2: Each open result is displayed as an additional tab in the large area on the top left-hand side.

Strictly speaking, each result is also a view, which you can move to anywhere you wish as usual. In this way it is possible to look at several results at the same time. You can of course also close individual views, i.e. tabs, by clicking once on the cross on the tab. The other functionalities of views, such as maximisation

through double-clicking etc., are also completely available to you here.

If you have not deactivated the question, RapidMiner will ask you on completion of a process whether the old results are to be closed before the new results are displayed. It ultimately depends on your personal preference whether you wish to always leave old results open for the purpose of comparability and close them manually. However, thanks to the Result Overview already mentioned, this additional work hardly seems necessary and so we rather recommend you have the old results closed automatically in order to improve the overview and rule out confusion.

4.2.1 Sources for Displaying Results

There are several sources from which you can have results displayed. We will present all ways to you in the following:

1. Automatic Opening

We have already seen that the final results of a process, i.e. objects which are supplied to the result ports on the right-hand side in the process, are displayed automatically. The same also applies for the results at connected ports in the case of a breakpoint. This is definitely the most frequently used and also recommended option for displaying results. You can simply collect at the result ports all process results that you wish to see at the end of an analysis process and they are all shown together in the tabs of the Result Perspective.

2. Results from Repositories

The second option for displaying results is loading results from one of your repositories. You can do this via the context menu of a repository entry or simply by double-clicking on an entry. Of course, this process is not only recommended for reviewing results, but also for comparing with earlier results.

3. Results from Ports

A third possibility for looking at results and even intermediate results is displaying results which are still at ports. RapidMiner tries to store the results, which were supplied by individual operators, at the relevant ports for a while longer. If there are still results at a port, these can be selected and looked at via the context menu of the port:

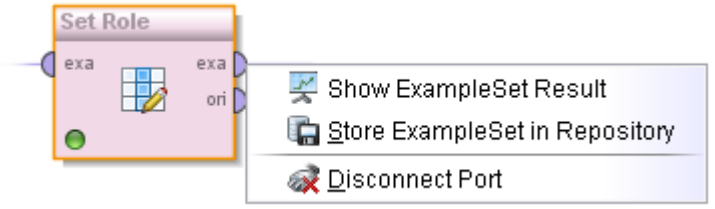


Figure 4.3: Display of results which are still at ports.

You may know this approach from other data analysis tools: You add an operator, execute it and indicate the results via the context menu or via special operators for this. Even if this approach seems intuitive and easy-to-use for small data sets, we urge you to avoid this method, since it will lead to problems no later than during the analysis of large data sets. In this case a copy of the data would have to be held ready at each port so that this result can be provided later on. RapidMiner goes a completely different way here; a way which also promises greater success in the long term: The meta data is transformed and propagated by the process and data is only made available where absolutely necessary. This kind of RapidMiner analysis thus combines the interactivity allowed by established meta data with simple process definition for the analysis of data sets, including large ones.

Note: RapidMiner has a sophisticated memory management here. As already mentioned above, results are kept at the ports "for a while longer". These results are deleted as soon as the memory of RapidMinder or other programs necessary for this is required. This means: Results can disappear from the ports and then no longer be available for visualisation. This is one of the reasons for the efficiency of RapidMiner and for this reason we also recommend the automatic display via connected ports as described above, since the provision of results is guaranteed here.

4.3 About Data Copies and Views

The fact that no unnecessary data copies are created is sometimes a source for confusion. This applies in particular for the second possibility of displaying results mentioned above i.e. via the context menu of ports. Let us assume you have a data set and add an operator for normalisation. In its presetting, the normalisation operator changes the underlying data. Even if you look at the data set at a port which is before the normalisation in the process flow, but chronologically after the normalisation was already performed, then the data at the port will have also changed beforehand. This behaviour should actually be quite clear - as previously mentioned, no copy of the data was created either and the same data set was changed further. And yet this "strange" behaviour of "uncontrolled data changes" leads to confusion from time to time.

However, you do have two ways of influencing this behaviour:

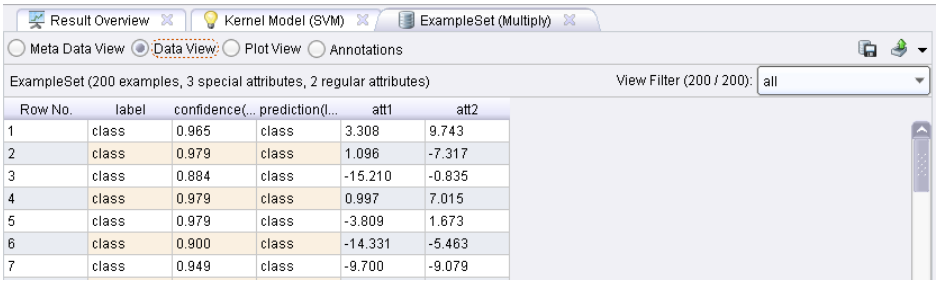
1. *Use of views*: Numerous operators for data transformations offer a parameter "create view", which, instead of causing a change to the data, merely causes a further view to be put on the data, which changes the data on-the-fly, so during data access. These computations do then not affect previous ports or even ports in other, parallel strands of the process.

2. *Explicit copies*: Especially for smaller data sets, the combination of the operators "Multiply" with " Materialize Data" can be a way out. You as an analyst can explicitly define your wish to have a copy of the data by first multiplying the reference to the data set by means of "Multiply" and then explicitly re-creating both virtual data sets as tables by means of "Materialize Data".

No analyst will seriously do this much work just to be able to access the results via the ports. But such interconnections can arise from time to time, even in parallel strands of processes, and then be resolved depending on the size of the data set by means of views or even explicit copies.

4.4 Display Formats

However the results got into the Result Perspective, each result is displayed within its own file card. And in addition, there are other different ways of displaying a large number of results, which are also referred to as views within RapidMiner:



Row No.	label	confidence(...	prediction(l...	att1	att2
1	class	0.965	class	3.308	9.743
2	class	0.979	class	1.096	-7.317
3	class	0.884	class	-15.210	-0.835
4	class	0.979	class	0.997	7.015
5	class	0.979	class	-3.809	1.673
6	class	0.900	class	-14.331	-5.463
7	class	0.949	class	-9.700	-9.079

Figure 4.4: The views "Meta Data View", "Data View" (currently shown) und "Plot View" exist for a data set.

For data sets for example there are three views, i.e. the display of meta data and statistics ("Meta Data View"), the display of the data itself ("Data View") as well as the display of different visualisations ("Plot View"). In the example above you can see the data view of a data set in the form of a table. Besides such tables, further standard display formats are available, which we would like to explain in the following.

Please note beforehand that all views share two common buttons on the top right-hand side: the left-hand icon serves for storing this result in the repository and the second serves for different forms of exporting the result, for example through printing or exporting into a graphic file.

4.4.1 Text

The most fundamental form of visualisation is that in text form. Some models as well as numerous other results can be displayed in textual form. This is typically done within the so-called "Text View", which you can select (if there are several views for this object) using the buttons directly underneath the tab.

In RapidMiner you can always highlight such texts with the mouse and copy onto the clipboard with CTRL + C. The results are then available in other applications

also. You can also highlight longer texts completely by clicking on the text area followed by CTRL + A and then copy.

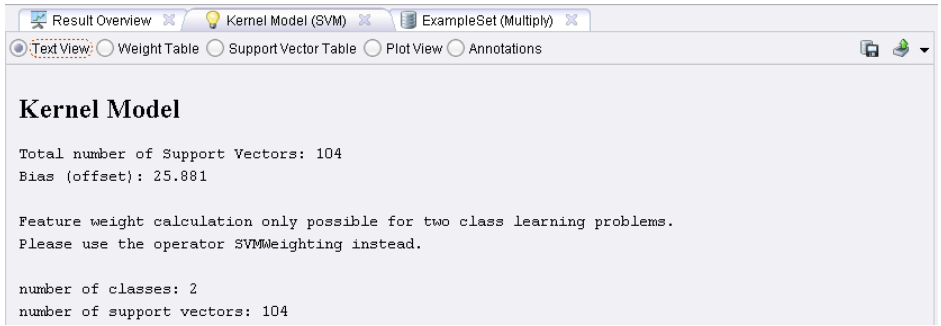


Figure 4.5: Some models such as rule sets are displayed in textual form. Numerous other objects also offer a display in form of a readable text.

4.4.2 Tables

One of the most frequent display formats of information within RapidMiner is tabular form. This is hardly surprising for a software solution with the primary goal of analysing data in tabular structures. However, tables are not only used for displaying data sets, but also for displaying meta data, weightings of influence factors, matrices like the correlations between all attributes and for many more things. These views frequently have the term "Table" in their name, especially if confusions are to be feared. Otherwise, such tables are simply referred to with terms like "Data View" or "Meta Data View".

Colour Schemes

Nearly all tables in RapidMiner use certain colour codings which enhance the overview. With data sets for example, the lines are shown alternately in different colours. Attributes with a special role are given a light yellow background here and regular attributes a light blue one:

This colour coding is also transferred to the meta data: attributes with special roles also have a consistently light yellow background and regular attributes have an alternating light blue and white background. However, this colour scheme can be completely different for other objects, like in Fig. 4.7. In the case of a

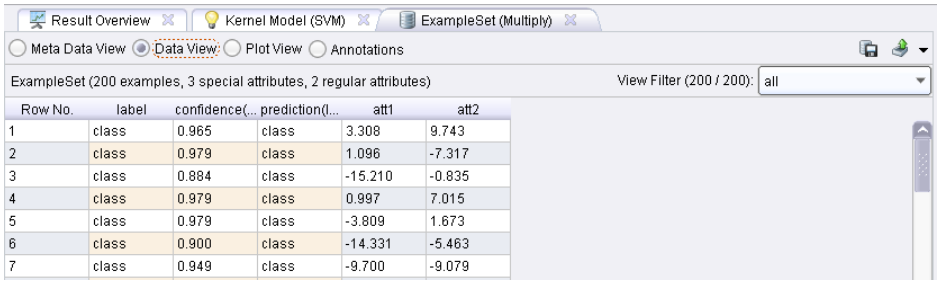


Figure 4.6: Colour codings and alternating line backgrounds make navigation easier within tables.

correlation matrix for example, individual cells can also be coloured: the darker they are, the stronger the correlation between these attributes.

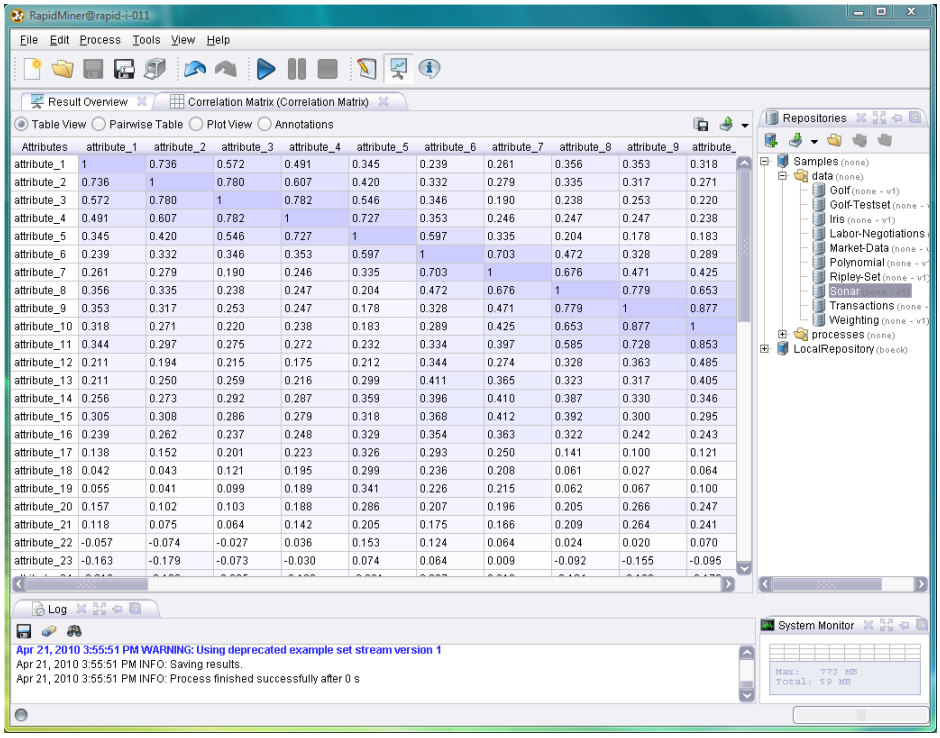
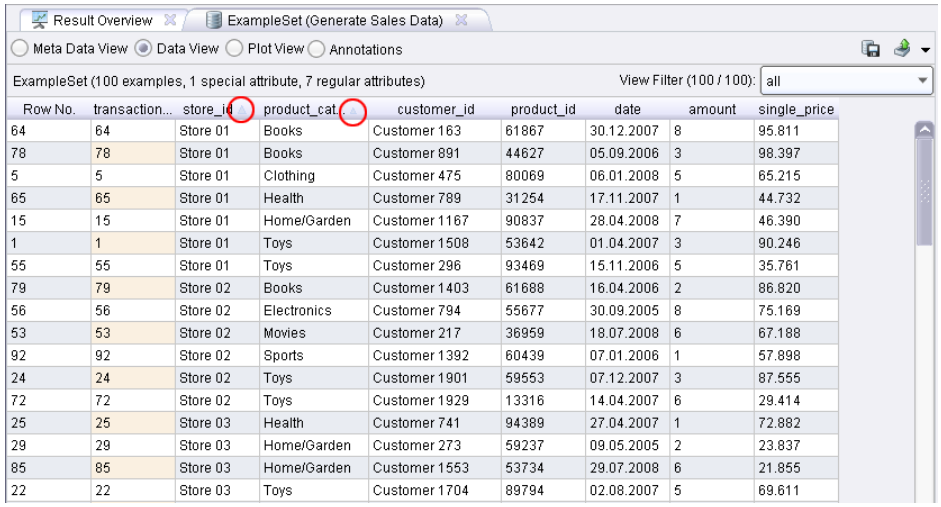


Figure 4.7: Tables in RapidMiner often indicate interesting information with colours. In this case darker backgrounds highlight stronger correlations between attributes.

Sorting

Most tables can be sorted in RapidMiner with a simple click. Move the cursor roughly into the centre of the column heading and click on the heading. A small triangle will now indicate the sorting order. A further click will change the sorting order and a third click will deactivate the sorting again.

You can sort also according to several columns at the same time, i.e. sort by one column first and then by up to two further columns within this sorting. In order to do this, start by sorting the first column and sort in the desired order. Now press and hold the CTRL key while you add further columns to the sorting. In the following example we have sorted the transactions according to the ID of the store first of all, and then by the category of the article. The order of the columns within this sorting is symbolised by triangles of different sizes ranging from large to small (Fig. 4.8).



Row No.	transaction...	store_id	product_cat...	customer_id	product_id	date	amount	single_price
64	64	Store 01	Books	Customer 163	61867	30.12.2007	8	95.811
78	78	Store 01	Books	Customer 891	44627	05.09.2006	3	98.397
5	5	Store 01	Clothing	Customer 475	80069	06.01.2008	5	65.215
65	65	Store 01	Health	Customer 789	31254	17.11.2007	1	44.732
15	15	Store 01	Home/Garden	Customer 1167	90837	28.04.2008	7	46.390
1	1	Store 01	Toys	Customer 1508	53642	01.04.2007	3	90.246
55	55	Store 01	Toys	Customer 296	93469	15.11.2006	5	35.761
79	79	Store 02	Books	Customer 1403	61688	16.04.2006	2	86.820
56	56	Store 02	Electronics	Customer 794	55677	30.09.2005	8	75.169
53	53	Store 02	Movies	Customer 217	36959	18.07.2008	6	67.188
92	92	Store 02	Sports	Customer 1392	60439	07.01.2006	1	57.898
24	24	Store 02	Toys	Customer 1901	59553	07.12.2007	3	87.555
72	72	Store 02	Toys	Customer 1929	13316	14.04.2007	6	29.414
25	25	Store 03	Health	Customer 741	94389	27.04.2007	1	72.882
29	29	Store 03	Home/Garden	Customer 273	59237	09.05.2005	2	23.837
85	85	Store 03	Home/Garden	Customer 1553	53734	29.07.2008	6	21.855
22	22	Store 03	Toys	Customer 1704	89794	02.08.2007	5	69.611

Figure 4.8: Sorting was first performed in this table in ascending order according to the attribute "store_id" and then according to product category within the store ID blocks, also in ascending order.

Note: Sorting can be time-consuming. It is therefore deactivated for large tables, so that no sorting is started inadvertently and the program cannot be used in this time. You can set the threshold value at which sorting is deactivated in the settings under "Tools" – "Preferences".

Moving Columns

You can change the order of columns in most tables by clicking on the column heading and dragging the column to a new position while holding down the mouse button. This can be practical if you wish to compare the contents of two columns with one another in large tables.

Adjusting Column Widths

You can adjust the width of columns by holding the cursor over the area between two columns and changing the width of the column to the left of the separation area while holding down the mouse button. Alternatively, you can also double-click on this gap, which causes the width of the column to the left of the gap to be automatically adjusted to the necessary minimum size. Last but not least, you can also hold the CTRL key down when you double-click on a gap, causing the size of all columns to be adapted automatically.

Tip: You should note this combination (CTRL + double click on a gap in the column heading area) so that you can quickly adjust column widths.

Actions in the Context Menu

In most tables you can open a context menu with further actions by right-clicking on a table cell. The details of these actions are:

1. *Select Row:* Selecting a line,
2. *Select Column:* Selecting a column,
3. *Fit Column Width:* Adjusting the width of the selected column,
4. *Fit all Column Widths:* Adjusting all column widths,
5. *Equal Column Widths:* Using same standard width for all columns,
6. *Sort by Column (Ascending):* Sorting by this column in ascending order,
7. *Sort by Column (Descending):* Sorting by this column in descending order,

8. *Add to Sorting Columns (Ascending):* Adding to the sorting columns (ascending),

9. *Add to Sorting Columns (Descending):* Adding to the sorting columns (descending),

10. *Sort Columns by Names:* Reordering columns by sorting the column headings in alphabetical order,

11. *Restore Column Order:* Restoring the original column order.

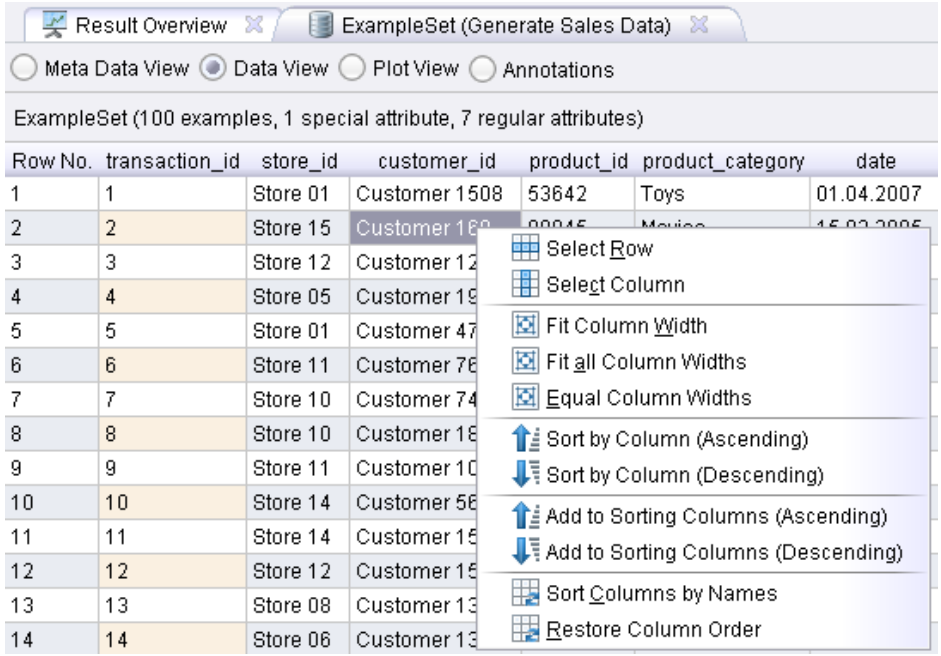


Figure 4.9: Actions like selecting lines or columns, sorting contents by columns or adjusting column widths are available in a context menu.

Copying Table Contents

Just as with the text view above, you can also highlight individual cells within tables using the mouse or highlight the complete table by clicking in the table and using CTRL + A. Actions are also available in the context menu for highlighting whole lines or columns. You can then copy the selected area onto the clipboard

by means of CTRL + C and paste it into other applications. Please note that the table structure stays as it is, if for example you paste into applications such as Microsoft Excel which support tabular data.

4.4.3 Plotters

One of the strongest features of RapidMiner are the numerous visualisation methods for data, other tables, modells and results offered in the "Plot View".

Configuring Plotters

The structure of all plotters in RapidMiner is basically the same. There is a configuration area on the left-hand side, which consists of several familiar elements:

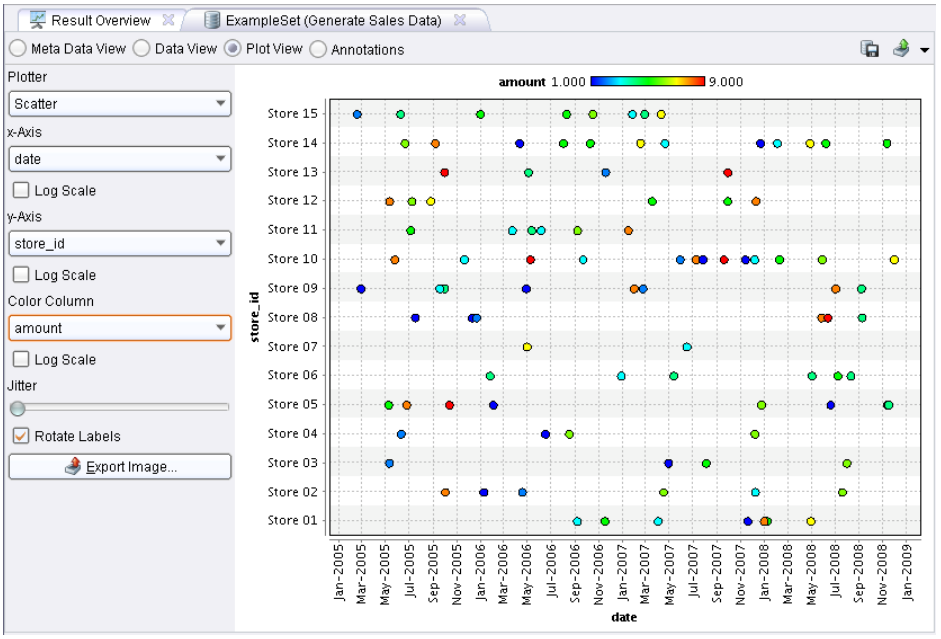


Figure 4.10: Visualisation of a data set and the plotter configuration on the left-hand side.

The most important setting can be found right at the top and corresponds to the type of visualisation. More than 30 different 2D, 3D and even high-dimensional

visualisation methods are available for displaying your data and results. In the image above you will see a plot of the "Scatter" type. Depending on the type of plotter selected, all further setting fields change. With a scatter plot for example, you indicate the attributes for the x axis and for the y axis and can use a third attribute for colouring the points. You can do further things specific to the scatter plot, such as indicate whether the axes are to be scaled logarithmically.

Tip: The "Jitter" function is very helpful, especially for data sets which do not only contain numbers but also nominal values. You indicate whether and how far the points are to be moved away from their original position in a random direction. You can therefore make points, which would otherwise be covered by other points, easily visible.

Many plotters also allow further display configurations, e.g. whether the text at the x axis is to be rotated so that long texts can still be read. Play around a little with the settings and various possibilities and you will soon be familiar with the numerous possibilities for visualisation.

Tip: You can change the colours used in the settings under "Tools" – "Preferences".

Changing the Plotter Type

The selection of the plotter type significantly defines which parameters you can set. In Fig. 4.11 you can see an example of a "Bars Stacked" type plotter. Instead of the different axes you now set attributes according to which the data is to be grouped (here: "store_id") and which attribute is to be used for defining the stacks (here: "product_category"). The height of the bars then corresponds to the sum (here: "Aggregation" is on "sum") of the attribute defined as value column (here: "amount").

Computing Visualisations

Last but not least, it is to be mentioned here that there are other visualisations which are so complex that they must be computed especially. These visualisations, such as a Self-Organizing Map (SOM), then offer a button named "Calculate" with which the computation and visualisation shown in Fig. 4.12 can be started.

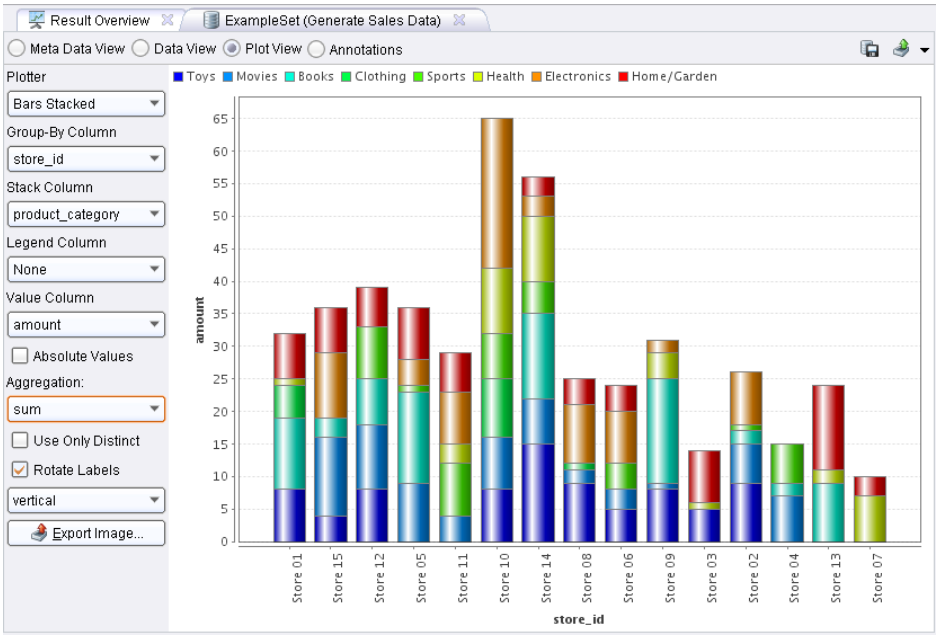


Figure 4.11: Change of the plotter configuration depending on the plotter type.

4.4.4 Graphs

Graphs are a further display format which are found relatively frequently in RapidMiner. Graphs basically mean all visualisations which show nodes and their relationships. These can be nodes within a hierarchical clustering or the nodes of a decision tree as in Fig. 4.13.

Graphs like that of the decision tree above are mostly referred to as a "Graph View" and are available under this name.

Zooming

Using the mouse wheel, if there is one, you can zoom in on and out from the graphs. Alternatively, you also have two buttons on the top left-hand side of the configuration area with which you can increase and reduce the zoom level of your graph.

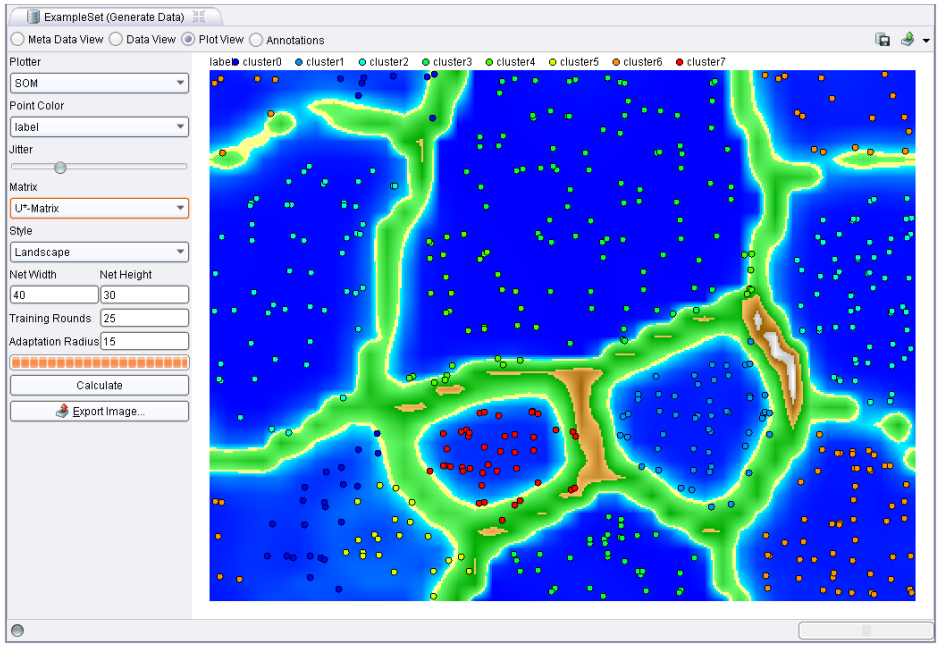


Figure 4.12: Complex visualisations such as SOMs offer a "Calculate" button for starting the computation. The progress is indicated by a bar.

Mode

Two fundamental navigation methods are available in the graph, which are also called modes:

1. *Moving*: The mode for moving the graph is selected by pressing the left-hand button in the mode box. In this case you can move the section of the graph while holding down the left mouse button in order to view different areas of the graph in detail.

2. *Selecting*: The mode for selecting individual nodes is selected by pressing the right-hand button in the mode box. Now you can select individual nodes by clicking on them or, while holding down the mouse button, define a selection box in a free area for several nodes at the same time. By holding down the SHIFT key you can add individual nodes to the selection or exclude them from the selection. Nodes that are currently selected can be moved while holding down the mouse button.

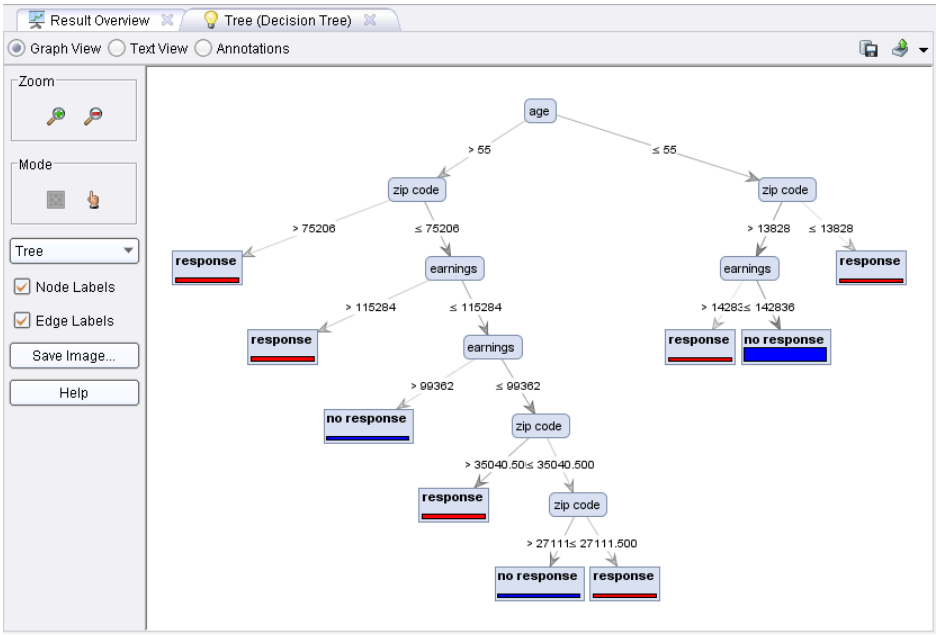


Figure 4.13: A decision tree in a graph view.

You will find further notes on handling graphs in these two modes in the help dialog, which is shown if you click on the button "Help" in the configuration area of the graph.

Further Settings

You can set whether the captions for nodes and edges are to be shown or not. The most important setting, not necessarily for trees but for other graphs, is the choice of a suitable layout, which can be made in the selection box directly underneath the mode box. The different algorithms have different strengths and weaknesses and you usually have to try and see which display gives the best result for the graph at hand.

4.4.5 Special Views

Alongside the views text, table, plotter and graph described above, there are also occasionally further display components, which are rarer however and which

should be self-explanatory. For Frequent Itemsets for example there is another special kind of table or graph for the related association rules.

4.5 Result Overview

We already mentioned the Result Overview at the beginning, which can always be found as a kind of placeholder in the place where the remaining results are also indicated:

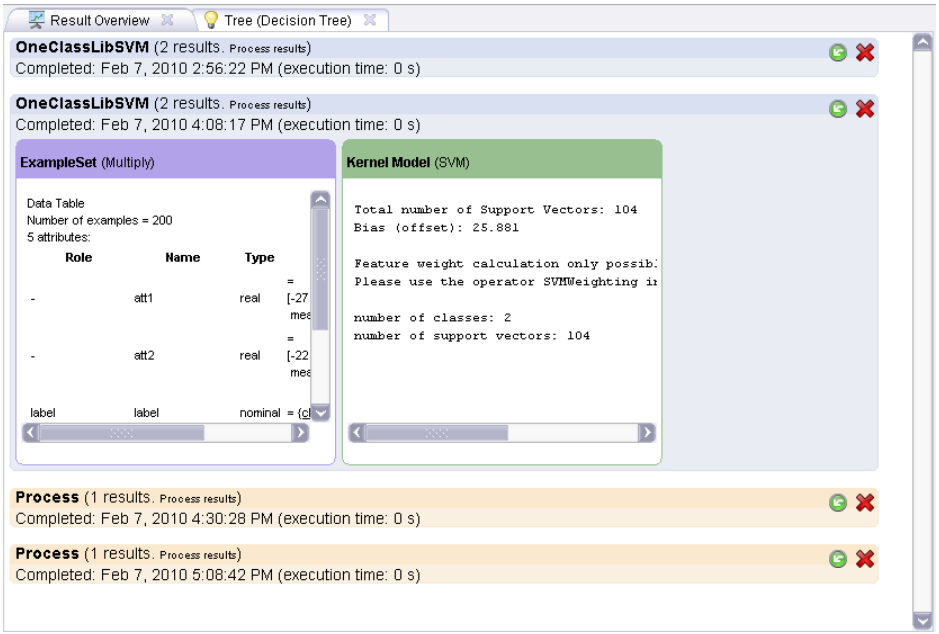


Figure 4.14: The Result Overview indicates the results of the last analysis processes.

The Result Overview serves as a compact overview of all process executions of the current RapidMiner session. Each two-line entry consists of the name of the process, the number of the results as well as information on when the process was completed and how long it ran for. Each block displaying results of the same process have an alternate colouring.

You can see a detailed view of the results by clicking on an entry. In the case above the result consists of an example set and an SVM model. A further click

on the entry will close it again. You can of course also open several entries at the same time and compare the results comfortably by doing so.

Two actions are available for each entry on the top right, i.e.

1. To restore the process belonging to an entry in this form and
2. To delete the entry from the Result Overview.

In addition, you have the option to delete the complete overview in the context menus of the Overview and of the individual entries.

Note: If you decide to close the Result Overview, RapidMiner will warn you that no more results will be shown in this perspective. We therefore urge you not to close the Result Overview or to at least leave one Result Overview open in a perspective.

5 Managing Data: The Repository

Tables, databases, collections of texts, log files, websites, measured values – this and the like is at the beginning of every data mining process. Data is prepared, converted, merged, and at the end you will receive new or differently represented data, models or reports. In this chapter you will find out how to handle all these objects with RapidMiner.

5.1 The RapidMiner Repository

As soon as your collection of processes and associated files exceeds a certain size, you will see it is wise to organise these in a consistent and structured manner. One possibility is the organisation of projects on file level. Files are grouped into projects and a directory is created in each case for output data, intermediate results, reports, etc.

While creating organised project structures is sensible, using the normal file system is recommended only in the rarest cases and is hardly sufficient for the needs of a data mining solution. Different reasons such as confidentiality or limited storage space can make creating files on the local computer impossible. If a process created on the local computer is to be executed on a remote server, this requires manual interventions like copying the process and adapting paths. The collaborative creation of processes, manipulation of data and evaluation of results requires an external rights and version administration. Files stored in different formats require the correct setting of parameters such as separators and coding for each new loading. Intermediate results and process variants soon grow to a

considerable number, meaning that one can lose track easily. Loading and looking at data in order to regain an overview requires a loading process that may be lengthy or even the running of an external application. Annotations of files which can make this easier are not supported by normal file systems.

RapidMiner's answer to all these problems is the repository, which takes up all data and processes. Although data can also be introduced into processes from outside the repository, which is necessary for the execution of ETL processes for example, using the repository offers a number of advantages which you will not want to miss:

- Data, processes, results and reports are stored in locations indicated relative to one another in a mechanism that is transparent for the user.
- Opening or loading the files requires no further settings. Data can be opened, looked at or incorporated into the process with a single click. You will get an overview of the stored data, its characteristics and remarks made by yourself at any time without having to open the file separately.
- All input and output data plus intermediate results are annotated with meta information. This ensures the consistency and integrity of your data and makes validating processes possible at the time of development as well as the provision of context-sensitive assistants.

The repository can either be on a local or shared file system or be made available by the external RapidMiner analysis server named RapidAnalytics. The following picture shows the repository view, which displays the content of the repository. RapidMiner provides a set of example processes and example data which you will find in the repository initially created. Some of these can be seen in the Fig. 5.1.

5.1.1 Creating a New Repository

In order to be able to use the repository, you must first create one. RapidMiner asks you to do this when it is started for the first time. You can later add further repositories by using the first button in the toolbar of the repository view. The following pictures show the simple procedure. If you do not have the RapidAnalytics analysis server, select the first option to create a local repository and then choose Next. Now give your repository a name and choose a directory for it to be created in. Close the dialog with Finish. You can now use your

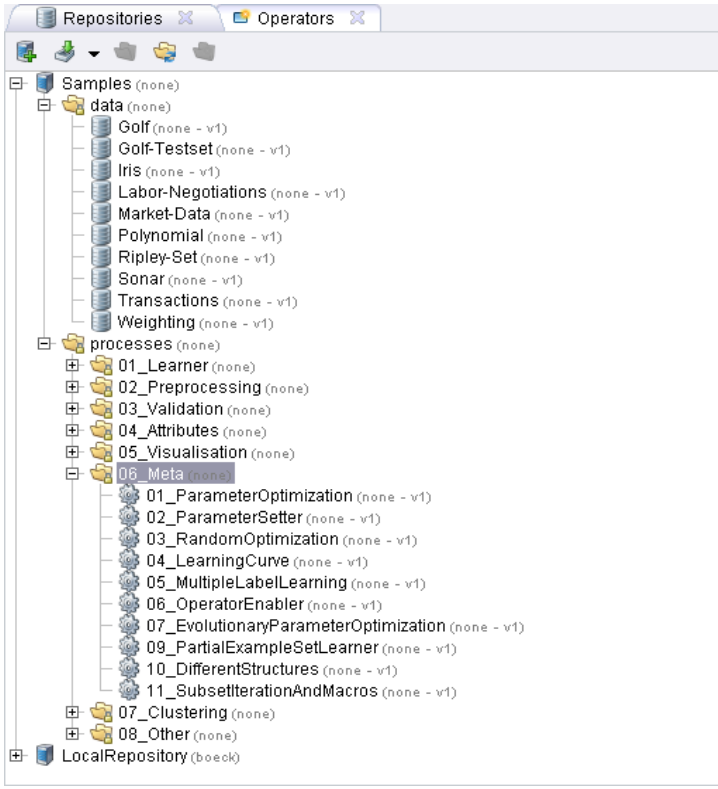


Figure 5.1: The repository view with an opened example directory.

repository.

5.2 Using the Repository

It makes sense to use a uniform directory structure for projects, for example a project folder with the name of the project and a folder each for processes, input data and results. All examples in this manual follow this structure. You can create directories using the context menu in the repository view or using the button in the toolbar at the top of this view.

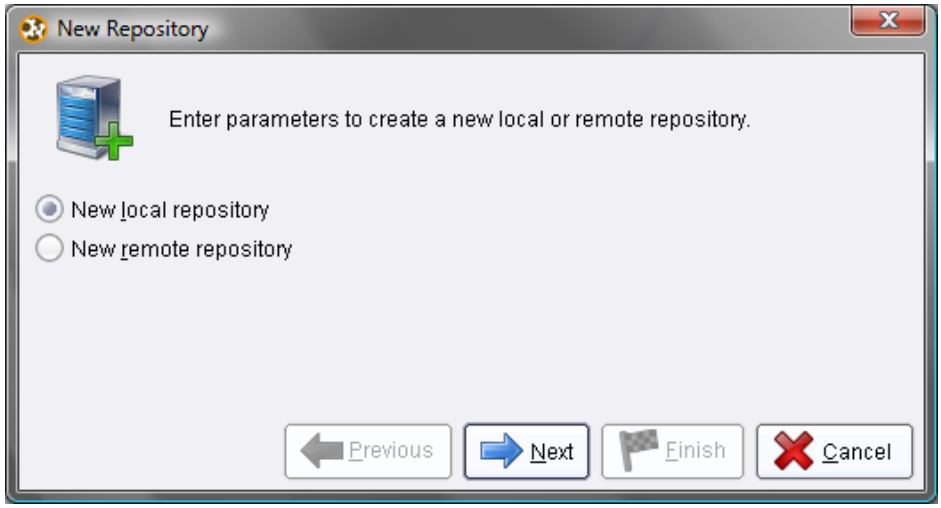


Figure 5.2: You can use a repository on a shared RapidAnalytics analysis server or select a local repository.

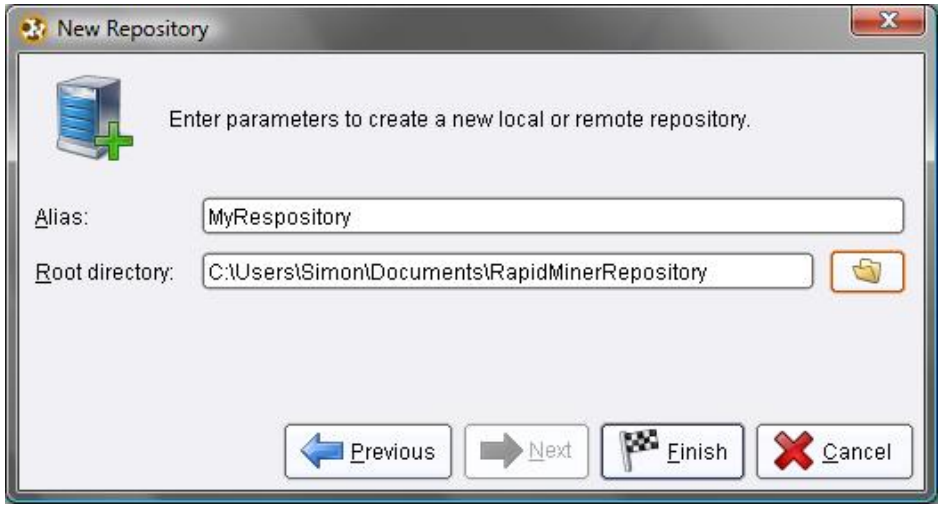


Figure 5.3: RapidMiner asks for the name and directory for a newly created local repository.

5.2.1 Processes and Relative Repository Descriptions

Before discussing in the following sections how you can store data and processes in the repository and access these again, we would like to first give some fundamental

tips on referencing these objects within the repository. You can store processes in the repository by selecting the entry "Store Process" in the context menu or by selecting the appropriate entry in the "File" menu. In the latter case, the repository browser opens, where you can indicate the location for storing the process. After a process has been stored in the repository, all references to repository entries set as parameters of operators are resolved in relation to the location of the process. What does that mean? Entries in the repository are designated as follows:

//RepositoryName/Folder/Subfolder/File

The two slashes at the beginning indicate that the name of a repository will follow first. Then further folder names and finally a file name. We call such details *absolute*. In the following description

/Folder/Subfolder/File

the repository designation is missing at the front. This description is therefore *repository-relative*. It refers to the file described in the same repository, where the process in which this description is used is located. The slash at the front indicates an absolute path description. If this is also missing, the description *relative* is resolved:

../RelativeFolder/File

designates for example a file in the folder "RelativeFolder" which we reach by moving up ("..") a directory from the file containing the current process and looking for the folder "RelativeFolder" there. So if the process is located for example in the file

//MyRepository/ProjectA/Processes/ProcessB,

this description leads to

//MyRepository/ProjectA/RelativeFolder/File.

Note: The descriptions above probably sound more complicated than they really are in practice. As long as, before anything else, you define a location within the repository for each new process and then simply use the repository browser for all operator parameters requiring an entry in the repository, RapidMiner will ensure, fully automatically, that relative data is always used as far as possible.

This makes repository restructuring and making copies for other users easier in particular, which would be difficult with absolute descriptions.

5.2.2 Importing Data and Objects into the Repository

There are numerous ways to import data and other objects such as models into the repository. We will now describe the most important ones.

Importing Example Sets with Wizards

If you have data in a certain format and wish to use it in a RapidMiner process, so-called *wizards* are available to you for many file formats and databases. A wizard is a dialog which guides you step by step through the loading process. With all wizards you can assign certain meta data such as attribute types, ranges of values and roles for the individual columns. In the upper area of the repository you will find an icon which starts the appropriate wizard for the selected file type. You will find the same action in the "File" menu of RapidMiner. Finally, there is another particularly simple way to import files: Simply drag the file to be imported into the repository while holding down the mouse button. If possible, an appropriate wizard will then be started.

The Operator "Store"

If you have an ETL process or another process, the result of which you would like to store in the repository, you can do this by integrating the operator "Store" into your process.

Using the operator "Generate Data", the example process in this picture generates a data set, which is to be stored in the repository. The "Store" operator only has one parameter, "repository_location". If you press the button with the folder next to this parameter, you will get a dialog in which you can first assign a folder in the repository and then a name for the data set. If you execute the process, you will see that a new entry will appear in the repository containing the generated data set. The store operator is therefore particularly useful for data integration and transformation processes which are to be performed automatically or regularly, for example within the process scheduler of the RapidAnalytics

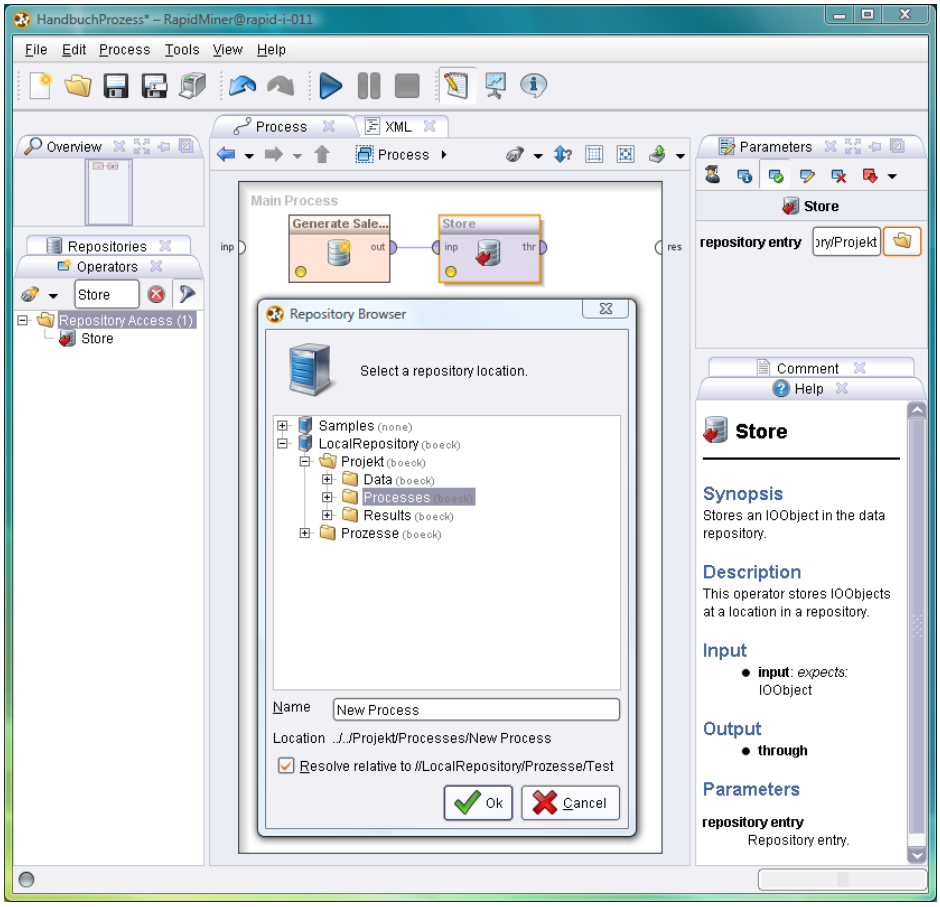


Figure 5.4: The operator "Store" can be used to store any data and objects in the repository. The dialog shows the repository browser so that the storage location can be specified and appears in the parameters of the operator if the "Directory" button is clicked on.

server. Using the wizard as described above is definitely the more frequently used way to ensure a one-off and fairly interactive integration of data.

Note: You can not only connect data sets with the store operator, but also models and all other RapidMiner objects. You can therefore also store any results in your repository.

Importing other formats by means of operators

The repository stores data sets in a format which contains all data and meta data needed by RapidMiner. Your data will probably be in another format at the beginning: CSV, Excel, SQL databases, etc. As described above, you can transfer these files into your repository. However, RapidMiner can also import numerous other formats within processes. You will find operators for this in the group "Import". However, caution is required when using these operators: The availability of meta data is not guaranteed for these operators, which can lead for example to processes that assume the existence of certain attribute values only noticing any errors in the runtime of the process. Nevertheless, using these file formats is sometimes not avoidable, e.g. for the regular execution of ETL processes. The goal of these processes should be however to transfer the data into the repository with a subsequent store operator so that it can be used by the actual analysis processes that follow.

The operators of the "Import" group have numerous parameters tailored to the respective format. Please see the respective operator documentation for their description.

Storing Objects from the Result or Process View

After you have executed a process, the Result Perspective with the tab of the same name is presented to you in the basic setting. On the right-hand side of its toolbar there is a button with which you can store the result currently selected in the repository. A dialog will also appear here allowing you to select a folder and a name.

If your process contains intermediate results which are not (or no longer) indicated in the Result Perspective, you can also store these from the Process View. In order to do this, click on a port where data is present using the right-hand mouse button. This is the case at the output ports of all operators that have already been executed. You will recognise this from the darker colour and an appropriate entry in the context help. You select the menu entry "Store in Repository" here to store the object. Please bear in mind however that the data at the ports may be released again after some time in order to save memory and is therefore not guaranteed to stay at the ports for any amount of time. Please also see the explanations in the previous chapter.

5.2.3 Access to and Administration of the Repository

Once you have imported your data into the repository you can use it in your processes with the retrieve operator. You can drag the operator from the Operators View into the process as usual and define the parameter for the repository entry there. But it gets easier still: Simply drag an entry in the repository (e.g. a data set) onto the Process View using the mouse. A configured operator with a reference to this entry will now be inserted automatically here. If the entry is an object, a new operator of the "Retrieve" type will be created and configured accordingly. If the repository entry is a process however, then a new operator of the "Execute Process" type will be created and its parameter will automatically refer to the selected process from the repository.

You will find further ways of accessing the repository by right-clicking once on entries in the repository. You will be familiar with these possibilities from the file management of your computer. These actions are also available via the toolbar of the repository view and are largely self-explanatory:

1. *Store Process here:* Stores the current process to the location indicated,
2. *Rename:* Renames the entry or the directory,
3. *Create Folder:* Creates a new folder here,
4. *Delete:* Deletes the selected repository entry or directory,
5. *Copy:* Copies the selected entry so it can be pasted in other places later on,
6. *Paste:* Copies a previously copied entry to this place,
7. *Copy Location to Clipboard:* Copies a clear identifier for this entry onto the clipboard, meaning you can use this as a parameter for operators, in web interfaces or the like,
8. *Open Process:* If you have selected a process, the current process will be closed and the selected one loaded,
9. *Refresh:* If the repository is located on a shared file system or if you use the RapidMiner analysis server RapidAnalytics, meaning data can be changed

at the same time by other users, you can refresh the view of the repository with this.

5.2.4 The Process Context

We have already used the output ports of the process on the right-hand side of the Process View previously e.g. in order to make the results of the process visible in the Result Perspective. In addition to the output ports of the process there are also input ports, which you will find on the left-hand side of the Process View. We have never connected these before. This is not even worthwhile in the basic setting, at least not for the sources, because the process itself then has no input. Connecting the inner sinks does have an effect however: All objects which arrive at a sink at the end of the process are presented in the Result Perspective as the result of the process.

These input and output ports of the process have a further function however. A typical process begins with a set of retrieve operators, which are followed by a set of processing operators, and ends with a set of store operators. You can avoid having to create these operators by using the Context View, which you will find in the "View" menu. Fig. 5.5 shows this Context View.

In the Context View you have the possibility of placing data from a repository at the entry ports and of writing outputs back into the repository. You can give such an indication for each port. This has two advantages:

- You can forget about the operators for Retrieve and Store, which often makes your process somewhat clearer.
- Using the context is also practical for testing processes which are to be integrated by means of the operator "Execute Process": The data at this operator will overwrite the values defined in the process context.

5.3 Data and Meta Data

Apart from the actual data, RapidMiner also stores other information in the repository: Data *about* the data, so-called meta data. Such meta data is available for each type of object, and it can be particularly useful for models and data sets.

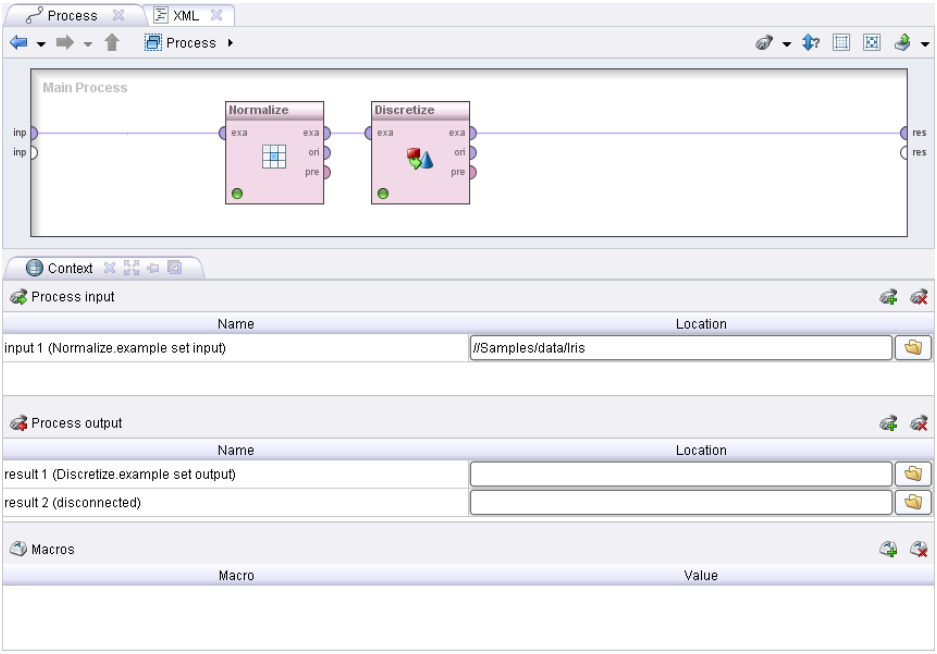


Figure 5.5: The process context. For "Input" you indicate repository entries which are to serve as an input of the process and be placed at input ports of the process. For "Output" you indicate where the results in the repository are to be saved to.

The meta information stored for data sets includes for example:

- The number of examples,
- The number of attributes,
- The types, names and roles of the attributes,
- The ranges of values of the attributes or some fundamental statistics,
- plus the number of missing values per attribute.

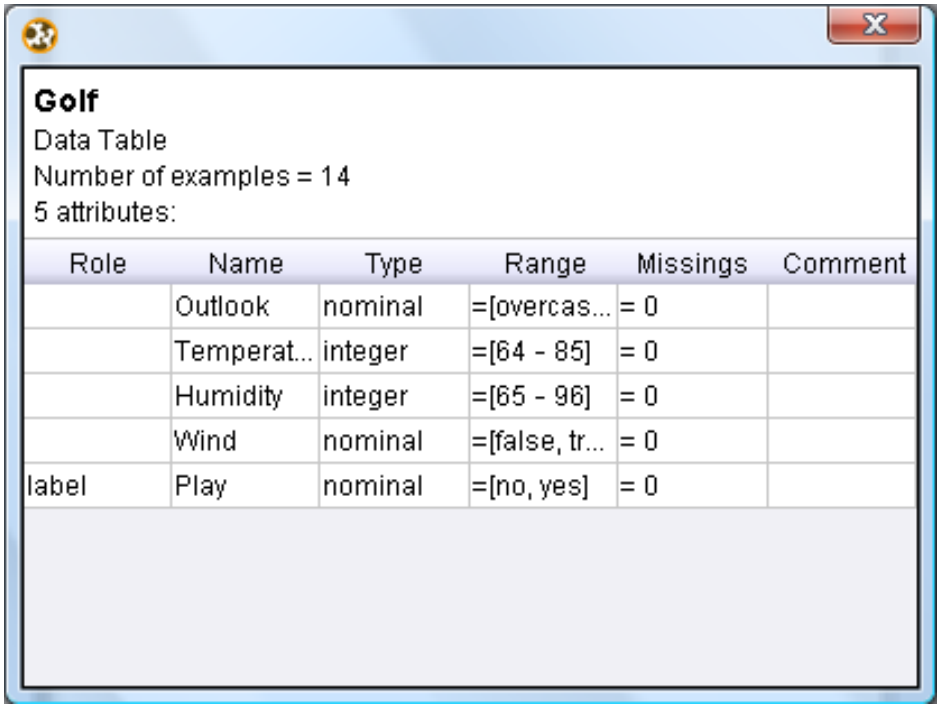
This information can be seen in the repository without loading the data set beforehand, which can take some time depending on size. Simply move the cursor over a repository entry and stay on the entry for a few seconds: The meta data will be presented to you in the form of a so-called tooltip. Unlike in other programs, this help information is much more powerful than normal: By pressing key F3 you can turn such a tooltip into a proper dialog, which you can move around and change in size as you wish. In addition, these RapidMiner tooltips are also able to include elements other than textual information with the meta data, such as tables for example.

Please note that the meta information does not necessarily have to be available immediately. You may have to first initiate the loading of the meta data by clicking once on a link within the tooltip. Doing this means that, should the tooltips of the repository entries be inadvertently looked at, the possibly quite large meta data is prevented from having to be loaded immediately causing RapidMiner to slow down.

Tip: Hold the cursor over a repository entry for a short time in order to look at the meta data or load it first. If the entry is an intermediate result for example, you can easily recognise what pre-processing has already taken place.

The following picture shows what the meta data for the golf data set from the example directory provided with RapidMiner looks like. First you will see that the data set contains 14 examples ("Number of examples") and 5 attributes ("Number of attributes"). The attribute with the name "Outlook" is nominal and takes the three values "overcast", "rain" and "sunny". The attribute "Temperature" on the other hand is numerical and takes values ranging from 64 to 85 - given in Fahrenheit of course. Finally, the attribute "Play" is nominal again, but still has a special role: It is marked as "label". The role is in italics and is given before

the attribute name.



Golf
Data Table
Number of examples = 14
5 attributes:

Role	Name	Type	Range	Missings	Comment
	Outlook	nominal	=[overcas...	= 0	
	Temperat...	integer	=[64 - 85]	= 0	
	Humidity	integer	=[65 - 96]	= 0	
	Wind	nominal	=[false, tr...	= 0	
label	Play	nominal	=[no, yes]	= 0	

Figure 5.6: The meta data of the golf data set from the example directory of the "Sample" repository provided with RapidMiner. You will find the data set named "Golf" in the "data" directory in this repository.

5.3.1 Propagating Meta Data from the Repository and through the Process

You have already seen that the meta data described above accompanies the actual data on its way through the RapidMiner process as early as when you create the process. As previously mentioned, it is however absolutely necessary for this meta data propagation and transformation that you are able to manage the data in a RapidMiner repository and obtain the meta data from this. For this reason we would like to remind you of and underline the necessity of using the repository for data and process management in order to provide support during process design.

In this section we will carry out a further example for the design of a process,

only this time we will revert to a data set from the RapidMiner repository. We will therefore now carry out the complete process for the first time, from the retrieval of the data right through to the creation of the results. Of course, this process would typically be preceded by importing data into the repository using one of the methods presented above, but in this case we shall skip this step and simply use one of the data sets already provided by RapidMiner instead.

Load for example the provided data set Iris using a retrieve operator, by simply dragging the entry concerned (in the same directory as the golf data set already used above) into the Process View. Do not execute the process yet though. Insert a normalise operator and connect its input with the output of the retrieve operator. Set the parameter "method" to "range transformation". In this setting, the operator serves for re-scaling numerical values so that the minimum is currently 0 and the maximum is currently 1. Select an individual attribute which you wish to apply this transformation to, e.g. the attribute "a3". For this purpose, set the filter type "attribute filter type" to "single" and select the attribute "a3" at the parameter "attribute". Now go over the output port of retrieve first with the mouse and then over the upper output port of the normalise operator. In both cases you will see the meta data of the Iris data set. You will notice however that the meta data of the selected attribute have changed: The range of values of "a3" is now normalised to the interval [0,1] after the transformation. Or to be more precise: The range of values of a3 *would in the case of execution* be normalised to the interval [0,1].

Insert a further operator, the operator "Discretize by Frequency". Connect this with the normalise operator. Set the parameter "range name type" to "short" and this time select another attribute, for example "a2", with the same mechanism as above. Now go over the output port of the new operator with the mouse and observe how the meta data has changed: The selected attribute is now no longer numerical but nominal and takes the values "range1" and "range2": The discretization operator breaks the numerical range of values apart at a threshold value and replaces values below this value with "range1" and values above this value with "range2". The threshold value is automatically chosen so that there is the same number of values above as below.

If you wish to have the values divided up into more than two ranges of values, adjust the parameter "number of bins" accordingly. You can see the process and the indicated meta data in the following picture:

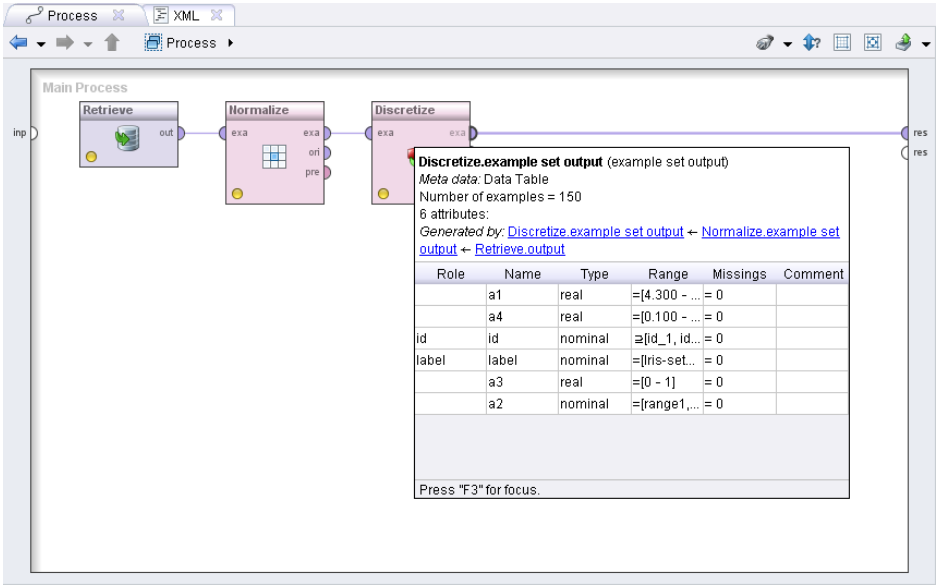


Figure 5.7: Meta data transformation in RapidMiner.

You are surely wondering why the parameter "range name type" had to be set to "short". See for yourself and set it to "long". If you execute the process, you will see that the nominal values now give more information: They additionally contain the limits of the intervals created. This is handy, but insignificant for the process. The information on the interval limits is not available however as long as the discretization has not actually been performed. Therefore it cannot be considered for the meta data display at the time of process development. It can then only be indicated in the meta data that the range of values of the discretized attribute is a superset of the empty set i.e. that it is not empty. This means that the meta data is not fully known. So in this case we can say virtually nothing at all about the expected meta data, except that the set of nominal values is a superset of the empty set. A trivial statement, but one that is nevertheless correct. The meta data cannot be fully determined in all cases as early as at the time of development. This is generally the case whenever the meta data is dependent on the actual data as it is here. In this case RapidMiner tries to obtain as much information as possible about the data.

Rapid-I GmbH
Stockumer Str. 475
D-44227 Dortmund
Tel.: +49 (0) 231 425 786 90
E-Mail: contact@rapid-i.com
www.rapid-i.com