UNIVERSITÀ DEGLI STUDI DI PISA

Facoltà di Economia

Facoltà di Scienze, Matematiche, Fisiche e Naturali

Corso di laurea specialistica in Informatica per l'economia e per l'azienda

TESI DI LAUREA

FREE OPEN SOURCE SOFTWARE FOR BUSINESS INTELLIGENCE

RELATORE

Prof. Salvatore RUGGIERI

Candidato

David BALABAN

ANNO ACCADEMICO 2008-09

SUMMARY

Free Open Source Software (FOSS) has recently grown, becoming a significant part of the IT market. We use the word “FOSS” to refer to software under a license which grants the right to access the source code and use, study, and change the software. We must not confuse FOSS with “non-commercial software”: antonyms of FOSS are “closed” and “proprietary” software.

The first purpose of this paper is to maintain an unbiased position. The analysis begins with a general overview of the FOSS world and then moves focus to business intelligence: during the last years, several tools have finally entered the market, becoming actual competitors to proprietary software.

Although FOSS still needs to grow, a large number of companies are already deploying or at least testing some FOSS solutions. In addition, the research world has shown interest providing several market surveys and software analyses.

After illustrating the selection criteria used, the paper describes the most interesting FOSS tools for each of the following business intelligence subcategories: database management systems (DBMS), data integration tools, analytical tools and business intelligence suites.

In addition, the FOSS data mining solutions RapidMiner and KNIME are evaluated and tested on a set of data. Although the two programs are not as widespread as the proprietary data mining tools, they can be considered actual competitors to the proprietary software.

INDEX

1 INTRODUCTION

1.1 Background Information and Problem Statement

Free (*libre*) Open Source Software (also known as FOSS) is a phenomenon which is changing the dynamics of the software market. FOSS solutions are entering all the computer science areas, providing every kind of software from the smallest tools to whole operative systems.

Although propriety software is still considered the primary choice for commercial software adoption, in the last decade FOSS have grown exponentially and managed to gain recognition from both the academic and the industrial world.

The rising popularity of some FOSS projects makes it interesting to study the FOSS world. Although many researchers have already analysed several FOSS, the literature is often produced by the same people who are involved in the FOSS projects and, as a result, the documents can be biased. The first intention of this paper is, therefore, to complete an impartial analysis of the FOSS world.

In addition to the generic overview of Free Open Source Software, we illustrate an original inspection of the world of business intelligence FOSS, often referred to as Open Source Business Intelligence (OSBI). Business intelligence refers to a wide selection of tools created to answer many different needs. This paper suggests a classification for business intelligence FOSS and shows, for each group, the background of the more successful projects.

Finally, the host company DataMinds is interested in discovering new opportunities for adopting business intelligence FOSS, or more precisely, data mining FOSS. After discussing the recent development of the FOSS market and analysing the most popular business intelligence FOSS solutions, we illustrate a detailed evaluation of RapidMiner and KNIME, two upcoming data mining FOSS.

The main indications which guide this research are given by PhD Salvatore Ruggieri, teacher from “Università degli Studi di Pisa”, and MSc Kim Lillesøe, DataMinds partner manager.

DataMinds, located in Aarhus (DK), delivers surveys, data mining and business intelligence solutions, especially within the human resources and marketing area. The company has supported this research providing data for the analysis, in order to make a software evaluation on the selected FOSS tools.

The paper is meant to be both a contribution to the analysis of the FOSS phenomenon and a document which will lead the host company to choose a possible adoption of business intelligence FOSS solutions.

1.2 Review of the Literature about FOSS and Business Intelligence

The economic studies about FOSS have been helpful to understand the reasons why FOSS has grown so much in the last years. These documents have shown the scenarios where the software has been used [Bitzer 07a] and the motivations leading programmers to work on FOSS projects [Bitzer 07b].

Reports about the usage of FOSS can be found in [Daffara 09] [Wang 01]. Further information about FOSS solutions can be found in the surveys which analyse a list of tools and illustrate their characteristics [Chen 07] [Golfarelli 09] [Thomsen 05] [Thomsen 09]. These papers lists the more popular FOSS tools and indicate some criteria to analyse them.

[Thomsen 05] gives an overview of the more interesting business intelligence FOSS solutions present in the market at that time, giving a short description of each tool. [Thomsen 09] is an updated version of the previous survey. The evaluations illustrated in these papers are not based on data gathered by configuring and running the software. The analyses are based on

information collected from the official websites of the FOSS projects and from their documentation, mailing lists and forum. The authors mention the necessity of a deeper evaluation in future papers where few business intelligence FOSS solutions should be configured and tested in terms of development time, ease-of-use, features and problems.

[Golfarelli 09] shows an analysis similar to [Thomsen 05] and [Thomsen 09], but focusing on business intelligence FOSS platforms. The author gives a generic overview of the current development of the main business intelligence FOSS suites, comparing them with the corresponding proprietary software.

The survey illustrated in [Chen 07] focuses on data mining FOSS solutions. This study was conducted a couple of years ago and does not include the upgrades of the new versions of the tools.

We made large use of statistical reports about FOSS adoption, especially those which focus on the business intelligence solutions. [Madsen 09], [Richardson 09], [Rexer 07], [Rexer 08] and [KDNuggets 09] are the main sources used. Advice advise

Finally we must not forget to mention the huge amount of documentation offered by the main FOSS communities. The tools evaluation, illustrated in this paper, is supported by – but not based on – information gathered from the documents provided with the software.

In addition to the literature produced by the communities, the Internet contains several pages about the analysis of FOSS solutions, and further more philosophic speculations regarding the subject. However, these documents are often written by authors who are pro or against FOSS, or more precisely, their opinion is often biased by personal economic interests in the market. These sources, therefore, have been carefully used in the paper, in order to maintain the wanted unbiased position.

It should be emphasised that neither the author, DataMinds or Università degli Studi di Pisa is involved in any business intelligence FOSS project and the

paper is, therefore, not written in order to recommend certain tools instead of others.

1.3 Thesis Content

After introducing the origins and purposes of the thesis, the paper is organised as follows.

Chapter 2 contains a short history of “Open Source” and “Free Software”. The aim is to give an overview of the most relevant communities and people, who are pioneers and founders of the FOSS movements. In addition, we go through the main licenses used for FOSS solutions in Section 2.4.1.

The analysis continues in Chapter 3 where the first sections explore the key issues concerning FOSS, subdividing them into technical features (Section 3.1), economic features (Section 3.2) and philosophic features (Section 3.3). After analysing the main issues of generic FOSS, we move focus to the business intelligence FOSS. The background history of these tools is analysed in Section 3.5 while, based on statistical reports, their actual status is described in Section 3.6.

“Business intelligence” is a generic expression which gathers many fields and areas of application. Chapter 4 suggests groups to subdivide the business intelligence tools into: we describe each of these groups in a separate section and list the main FOSS solutions for each of them.

In Chapter 5 we focus our attention on the group of business intelligence tools used for data mining. The analysis provides a description of the selection process (Section 5.1) and of the criteria used to evaluate the selected data mining tools (Section 5.2). Afterwards each software review is shown in a separate section: RapidMiner in Section 5.3 and KNIME in Section 5.4.

2 INTRODUCTION TO FREE SOFTWARE, OPEN SOURCE SOFTWARE

OSS, FS, FOSS, or FLOSS?

"The enemy is proprietary software." [1]
Free Software Foundation

Currently, many different names are used to refer to "free software", but the common ideologies are similar. In the next sections we discuss the different definitions and study which are the more relevant communities involved in free software.

The idea of Free Software can be dated back to when computers were tools for research, firstly in universities. Software was freely passed around, and programmers were paid for the act of programming, not for the programs themselves. Only later on, when computers reached the business world, programmers began to support themselves by restricting the rights to their software and charging fees for each copy [Perens 09a].

We must not confuse "Open Source software" and "Free software" with "non-commercial software". Antonyms of OSS/FS are "closed" and "proprietary" software.

Figure 2.1 is a modified version of the diagram by Chao-Kuei [GNU 08a]. The picture shows the different categories of software we will describe in the following sections.

[1] *[GNU 09h]*

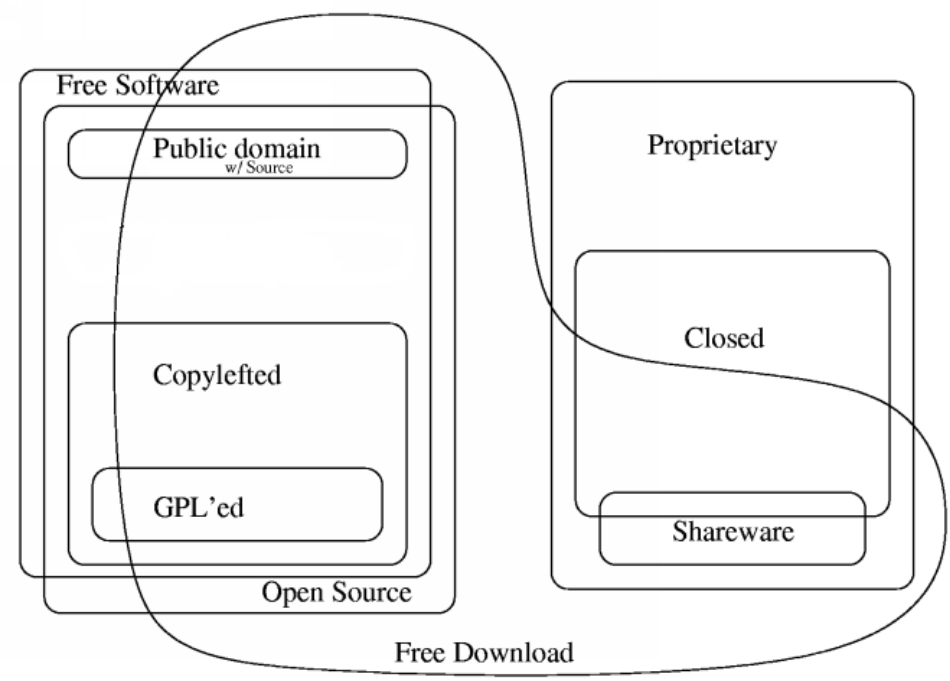

Image 2.1 (Modified version of diagram by Chao-Kuei)

2.1 Free Software Foundation (FSF): Users' Four Freedoms (1986)

"Free software is a matter of liberty, not price.
To understand the concept, you should think of free as in free speech, not as in free beer."[2]
Richard Stallman

The Free Software Foundation (FSF) was founded in October 1985, initially to raise funds to help develop GNU (the name "GNU" is a recursive acronym that stands for "GNU's Not Unix") [GNU 09a]. The Free Software Foundation has become the principal organisational sponsor of the GNU project [GNU 09b].

The GNU project was conceived in 1983 by Richard Stallman as a response to the fact that almost all software became proprietary in the 1980s. The

[2] *[GNU 09d]*

intention was to bring back the software sharing spirit he had experienced in the IT community in the MIT (Massachusetts Institute of Technology) Artificial Intelligence Lab during the previous years; the same spirit which was the base for computer user groups from the previous decades (e.g. SHARE in IBM Corporation in the 1950s and DECUS in Hewlett-Packard Company in the 1960s) [GNU 09c] [Stallman 08].

The manifesto of the Free Software Foundation can be summarised in the "Users' Four Freedoms" document written by Richard Stallman in 1986 [GNU 09d]:

> Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it means that the program's users have the four essential freedoms:
>
> - The freedom to run the program, for any purpose (freedom 0).
> - The freedom to study how the program works, and change it to make it do what you wish (freedom 1). Access to the source code is a precondition for this.
> - The freedom to redistribute copies so you can help your neighbor (freedom 2).
> - The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits (freedom 3).
>
> Access to the source code is a precondition for this.

The word "free", however, is ambiguous and it created some problems for the Free Software Foundation. "Free" was meant to be associated with

"freedom", in respect to the users' essential freedoms. Anyway it is still often mistaken as "without cost, payment, or charge". Richard Stallman coined the popular slogan "free as in free speech, not as in free beer" to overcome the poor choice of the word and to support the Free Software Foundation definition [GNU 09d].

The Free Software Foundation invites to usage and spread of the GNU General Public License, preferably the most recent version, GNU GPLv3. People are anyway invited to send their own licence to have it analysed and perhaps approved by the Free Software Foundation [GNU 09e]. Several licenses are listed and commented by the Free Software Foundation which divides them into three groups: GPL-Compatible Free Software Licenses, GPL-Incompatible Free Software Licenses and Non-Free Software Licenses.

2.2 Debian's Social Contract with the Free Software community (1997)

"We are Software In The Public Interest,
producers of the Debian GNU/Linux system.
This is the 'social contract' we offer to the free software community." [3]
Bruce Perens

Richard Stallman wrote the Four Freedoms document and distributed it around the MIT campus, but it took several years before the document made its way out into the public world. Before that, in 1997, Bruce Perens decided to write and virtually sign a "social contract" with the Free Software community [Perens 09b]. Perens wanted to define what "free" meant in the project he was leader of back then: the Debian GNU/Linux Distribution [Perens 97].

[3] *[Perens 97]*

This social contract is the stepping stone for Free Software and it has been used to write the Open Source Definition in 1998 [Perens 09a].

2.3 Open Source Initiative: Open Source Definition (1998)

"The Open Source Definition is a bill of rights for the computer user." [4]
Bruce Perens

"Why not call it, as we used to, free software?
[…] the real reason for the re-labeling is a marketing one." [5]
Open Source Initiative

A part of the Free Software community, led by Bruce Perens and Eric Raymond, split from the main group in 1998 to create the Open Source Initiative. The Debian Free Software Guidelines were used as the basis for the Open Source Definition [Perens 09a].

The ambiguity of the word "free" is the main reason why the two groups split. The word "free" has been used in many ways and with different meanings in the IT world: "free" can be used for software which costs no money, but with closed source code; software placed in the public domain is "free" from copyright restrictions; some software called "free" can be covered by copyright and a license agreement [OSI 06b].

The Open Source Initiative is a marketing program for free software. "Free Software" does not have a unique definition and it has, therefore, been misunderstood by business people, who mistake the desire to share with anti-commercialism [Moglen 00]. The community wants to promote the Free

[4] *[Perens 09a]*
[5] *[OSI 06c]*

Software changing the label to the name coined by Eric Raymond: “Open Source Software”.

2.4 Free Software vs. Open Source, a final comment

“When I say ‘Open Source’, I mean the same thing as ‘Free Software’.” [6]
Bruce Perens

The two communities had many disputes after the split in 1998, but they argued mainly about philosophic issues [GNU 09f]. The media, however, labelled Richard Stallman and Eric Raymond as direct competitors, rather than people who shared the same basic ideas. The real difference, however, was that Stallman had conceived the GNU General Public License as a political manifesto as well as a software license, while the Open Source Initiative had a focus which was centred around marketing.

Despite the fundamental definition differences, the list of GPL-Compatible licenses and OSI Certified licenses are for the most part the same. The Free Software Foundation stresses the philosophic differences arguing that the word “open” never refers to “Freedom”. At the same time, it admits that even if it is not exactly the same class of software, the differences between the two categories are small: nearly all free software is open source, and nearly all open source software is free [GNU 09g].

We will use the acronym FOSS (Free and Open Source Software) to refer to the type of software characterised by the properties we have analysed in this chapter.

[6] *[Perens 09c]*

2.4.1 Which licenses are used for FOSS?

The Free Software Foundation provides and recommends the GNU General Public License (GPL). The GPL is a "copylefting[7]" license which is designed to prevent the code from becoming proprietary. Once distributed, anyone is allowed to use the program and modify it, but the license deny linking the code to proprietary software.

If a developer wants to allow the linking of proprietary software to her code but keep it free, she can use the additional permissions listed in the GNU Lesser (or Library) General Public License (LGPL). This license is primarily intended for code libraries; like the GPL, LGPL-licensed software cannot be changed and made proprietary, but the LGPL does permit proprietary programs to link to the library [LGPL 07].

The two licenses, however, are not covered by any FOSS license, neither GPL or LGPL: "Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed."

Finally the MIT (aka X11) and BSD licenses let anyone do almost anything with the code except sue the authors. An OSS/FS program using a "non-copylefting" license, like the MIT or BSD licenses, can be extended by anyone and also included into proprietary software [Wheeler 09].

The Open Source Initiative provides a list of OSI Certified licenses which respect the Open Source Definition: this is one of the weapons used to achieve the purpose of the Open Source community, that is "to identify and lessen or remove issues caused by license proliferation" [OSI 10]. The license proliferation phenomenon is a big thread for the FOSS communities since many of the licenses are legally incompatible with other free and open source licenses and others could contain loopholes [OSI 07]. To avoid the license proliferation, the

[7] *Copyleft: "Copyleft is a way of using of the copyright on the program. It doesn't mean abandoning the copyright; in fact, doing so would make copyleft impossible. The word 'left' in 'copyleft' is not a reference to the verb 'to leave' — only to the direction which is the inverse of 'right'." [GNU 08b]*

Open Source Initiative promotes the OSI Certified licenses and organises them in groups, analysing all the details [OSI 10].

3 FROM FOSS TO OSBI

Is FOSS a potential threat to proprietary software?

" 'How do you make money with Free Software?'
was a very common question just a few years ago.
Today, that question has evolved into
'What are successful business strategies
that can be implemented on top of Free Software?' " [8]
Carlo Daffara

The FOSS communities have attracted the interest of economists, especially after the success of some FOSS like the operating system "Linux", the "Apache Web Server" or the web browser "Firefox". One of the reasons for the success is that FOSS tools are considered to be developed by highly qualified and motivated individuals, who keep the tools up to date [Bitzer 07b] [Chen 07]. Indeed, FOSS developers are open to new ideas, often becoming pioneers in their sector.

There are, nevertheless, lots of people who are still wondering what the actual benefits of FOSS are. We will try to address some of the main concerns, also listed in [Doshi 06], subdividing the analysis into three categories: technical, economic and philosophic. The chapter continues focusing on the benefits and risks in the adoption of FOSS.

After introducing the Free (*libre*) and Open Source Software world, the analysis will move the attention to FOSS solutions for Business Intelligence.

Business intelligence FOSS (sometimes abbreviated to OSBI, "Open Source Business Intelligence") have been gathering momentum in recent years and have finally entered the business intelligence market [Chen 07], becoming big competitors for the proprietary tools [Richardson 09].

We will try to understand the reasons why business intelligence FOSS solutions have become so popular and why they are a potential threat to well

[8] *[Daffara 09]*

known proprietary software, such as IBM SPSS Modeler (formerly Clementine) and SAS BI.

3.1 Key technical issues

The capitalistic market is not used to handling products with characteristics like FOSS. Since FOSS does not follow the usual rules, people are puzzled by this new kind of products: is it safe to rely on FOSS? Are they supported tools? Secure?

The simplest answer is an overview of the current market: is FOSS used by companies? If a big percentage of companies are using FOSS, it means that these tools are reliable enough to work with them. The approach is quite pragmatic and not scientific at all, but it can be a good indicator. We could get surprised reading that about 86% of Fortune 1'000 companies are deploying or testing FOSS, and that a similar percentage is found in Europe [Daffara 09].

Many big companies are working with FOSS and thus relying on them. Anyway it is still common to think that paid programmers are better than FOSS programmers. [Bitzer 07a] and [Daffara 09] have analysed the phenomenon and assured that FOSS programmers are not less trustful then paid workers: the driving intrinsic incentives, defined by [Bitzer 07a] as "doing something because it is inherently interesting, enjoyable or challenging", are in many cases more effective than monetary compensation. At the same time, as we will see in the next sections, it is actually not true that all FOSS programmers are not paid.

People can work for FOSS projects and have revenues from them. Most large scale FOSS project are related to companies that provide paid-for support, similarly to proprietary software projects [Daffara 09]. We must underline the importance of the technical support and of the community for any FOSS solutions: the former is often the major revenue for companies selling FOSS, the

latter is the developers and users group which grows with the software. These two subjects – support and FOSS communities – will be mentioned often in the analysis.

Finally, are FOSS solution secure? Commercial software supporters state that FOSS cannot be secure. The main motivation is the actual open source of FOSS solutions: everybody can access to the source and for this reason, they are considered more vulnerable to exploitation [Jones 04]. In contrast, FOSS supporters claim that FOSS means increased security: open source code is in the public view and it will therefore be exposed to extreme scrutiny, with problems being found and fixed instead of being kept secret until the wrong person discovers them [GNU 06]. Both the analyses seem reductive. [Perens 09a], however, shows a deeper study of the security problem and demonstrates that FOSS are not more vulnerable than proprietary software.

3.2 Key economic issues

"Yes, we can live on open-source software."
Open Source Initiative

We have analysed some of the technical issues in the last section. Now we should focus on the main economic key question: is there any money to be made on FOSS? The question has a double answer: first of all we have to focus on the programmers working on FOSS projects, second we have to mention the companies selling them.

Many people are discussing about the implications of FOSS usage for the programmers: the FOSS world is convinced that it would not hurt the programmers, while some of them are still concerned about their future job market. The confusing volunteer perception plays a decisive role in the issues:

although volunteer contributions are a significant part of large scale projects, around 50% of developers have received a financial compensation for working on FOSS projects [Daffara 09]. The compensation is either directly paid to the developers or to the community in order to improve the project. Therefore, many programmers could end up changing their job or redefining it, more than being unemployed [OSI 06c].

Many companies are currently involved in FOSS projects and some of these companies depend completely on them – RedHat is a popular example. This would be hardly possible if there was no money to be made on FOSS. Once again we can see that the focus is on support and maintenance: the real revenues in FOSS project are from the services – training, subscription, start-up consultancy – rather than from licenses [Giogia 08]. We can see that companies use FOSS to fuel the market for selling a separate product or service, such as providing support and installation services (e.g. Linux Distributions) or using the software as a stepping stone to sell a higher end product or service (e.g. Jaspersoft). Finally some FOSS projects are made to avoid or share costs: if many not-competitor developers need a product, it makes sense to share development costs (e.g. X Window System and the Apache web server).

Finally, most companies adopting FOSS report significant cost savings, which can be directly transferred to external professional services or incorporated as additional profit margin [Daffara 09]. We will analyse this point in Section 3.4.

3.3 Key philosophic issues

This section is a short analysis of all the philosophic speculation on FOSS. The aim is to identify the more important issues without getting lost into pro or anti-FOSS propaganda.

FOSS is not hostile to intellectual property, but it actually tries to prevent appropriation of code without giving back contributions or credit. This is one of the reason why many developers prefer FOSS licenses, like the GPL, to other licenses [Daffara 09]. Furthermore, an author using a copyleft policy is not completely against copyright: using a copyleft policy means to impose some of the copyright restrictions. The alternative to copyright or copyleft would be dropping the work into the public domain, where no copyright restrictions are imposed.

We already discuss the economic reasons which drive the programmers to work on FOSS projects. Now we must focus on the philosophic part of the subject. The literature shows different motives to program FOSS: [Bitzer 07a] explains it as a mixture of extrinsic motives (i.e. expecting a separable outcome) and intrinsic motives (i.e. doing something because it is inherently interesting, enjoyable or challenging). The possibility to gain a future wage or signalling your own product are good examples of extrinsic motives, while some intrinsic motives could be the actual need for a specific software, the fun in coding and releasing the product as a gift to the community or to the world. Using a metaphor from [Prasads 01]: “each programmer contributes a brick and each gets back a complete house in return”.

The group identification is the key point to understand FOSS communities, especially to understand the “gift culture”. The capitalistic goal is to maximise the profits, while many FOSS developers are focusing on enhancing their reputation and creating value by technology innovation [Bitzer 07a].

There would be more philosophy issues to discuss, but the analysis would be external to the purposes of this paper. Next section will conclude the first section of the chapter, summarising the benefits and risks that companies have in the adoption of FOSS tools.

3.4 Benefits and Risks in the adoption of FOSS

"In many cases, the question is 'when' to focus on open-source alternatives to traditional closed-source solutions, not 'if' you should focus on them." [9]
Gartner Group

After answering some of the main issues about FOSS, we can analyse the benefits and risk in the adoption of FOSS and point out the cost-saving and cost-raising characteristics of the FOSS development process.

Section 3.2 showed the economic aspects for programmers and companies "selling" FOSS solutions, while this paragraph focuses on the economic issues for companies "adopting" FOSS. To do it, we must answer this simple question: is cheaper adopting FOSS than proprietary software? The simple question needs a difficult answer: at first glance it is easy to say that it is always better to get a software for free, but a deeper analysis will show that it is not always true.

Although the direct cost of FOSS solutions is often zero, resulting particularly appealing to customers, we must extend the analysis to the involved indirect costs: development, technical support, and maintenance efforts can weight lots upon the total cost [Bitzer 07b] [Wang 01]. Consequently, FOSS is a good solution for those companies which are big enough or have such a strong developing team to be able to have internal development, support and maintenance. This way the indirect costs can be minimised.

The zero direct cost lets any company try the solution. A large indirect cost, on the other hand, cuts off all those companies which need large external support and maintenance, and cannot afford it. Some small organisations, already priced out of proprietary solutions, might think that open source is also beyond their reach because they do not have enough staff to evaluate and deploy open source solutions [Damiani 09].

[9] *Gartner Group, Hype Cycle for Open-Source Software, 2005.*

As already mentioned, the community are important for FOSS projects: these groups formed by customers, researchers, teachers and students, are the main driver of the FOSS projects evolution [Giogia 08].

The FOSS world seems optimistic about the future of FOSS solutions, primarily because of the big communities which grew behind the major projects. It claims that FOSS solutions will (*a*) develop much faster than commercial ones, because they are not constrained by compatibility problems and rigid or obsolete architectures [Golfarelli 09] and (*b*) evolve based on real life needs and worldwide market demands rather than on finances sales targets or a marketing agenda [Selim 09].

Once again the community identification, is the gist of this analysis: a FOSS user is not just a customer, but can become a motivated active member in the development of the project.

The other side of the coin is the big major risk for FOSS communities to split into groups, each following a different project. This phenomenon is even more frequent when there are no liable coordinating institutions deciding which path the project should follow. Even if the new ideas could be quickly implemented by the splitting groups, thus creating new technology innovation, this scenario could easily lead to redundant development, incompatible standards, and extra costs of attracting programmers [Bitzer 07b].

We could use the term “pure community-based” FOSS for those solutions which are not linked to a supporting company. A community based tool is, therefore, mainly developed inside its own community. Pure community-based FOSS arises major problems, such as the obvious lack of professional help, and could even suffer of delays in supply and under-provision [Bitzer 07a]. Some literature is also sceptic about the product maturity and selection [Damiani 09]. We already discussed this topic in the previous sections: the issue is minor or bigger depending on the software category.

Finally it is important to mention the importance of the license and its limitations [Wang 01]: no company should consider a software, with a license that is in contrast with the project goals.

3.4.1 Focusing on Business Intelligence

The description of benefits and risks in the adoption of FOSS is also valid for business intelligence FOSS solutions: a good example is the importance of the communities, which are often the main factor for the success of the FOSS tools. More accurately, a FOSS community increases product visibility and offers an extended pool expertise to support the software [Bondur 09].

However, we must underline the investigations which need to be read from a different point of view. For instance, the economic analysis for a FOSS solution is altered by the extremely high direct cost of the business intelligence proprietary solutions.

The number of proprietary business intelligence solutions, already large and in the last period increased by the FOSS tools, makes the software evaluation a long and time consuming process. More precisely, a company interested in implementing a business intelligence tool has to spend a long time identifying all the possible solutions and comparing their characteristics.

The process, however, is made easier for FOSS tools [Kemp 09]: any company can download and evaluate any FOSS without negotiating a trial license, like it would be necessary for proprietary software. The phenomenon is registered by several reports [KDNuggets 09] [Rexer 07] [Rexer08] [Richardson 09], which show that FOSS are often evaluated by companies already used to business intelligence tools.

In other words, the selection process is different from the “classic” method used to evaluate and select the “best quality-price ratio” proprietary solution.

The selection of the FOSS business intelligence solution should be manly imposed by (*a*) the license the software goes under, (*b*) the support required and (*c*) the financial capabilities of the user company.

The license can often dictate the choice of the software: for example, a license like GPL, which let anyone work on the open source, results inappropriate if the company intends to embed the software into something to redistribute under a proprietary license.

To overcome this issue, the most of the commercial FOSS companies have developed and adopted a “dual licensing” policy: usually the software can be downloaded for free, under a GPL license, while a proprietary license edition can be negotiated. This method leads to a double return for the developer company, which can have its efforts supported by a FOSS community and receive an income from selling support contracts and proprietary licenses [Damiani 09].

Second, the customer cannot choose without taking into consideration how much support it needs. Some FOSS solution are developed completely by a community which does not offer commercial support. In these cases it can be dangerous to base your business on software with no support contract available [Kemp 09].

Finally, companies should base the decision on their financial capabilities. Small and medium enterprises are often cut out from the market by the high direct costs in the acquisition of proprietary business intelligence software. In addition to this hurdle, it is usually difficult to calculate the return of investment for business intelligence adoption. FOSS with zero direct cost can create a new market bringing analytics to places and people who are not used to business intelligence tools.

The FOSS world claims that organisations that are open to using business intelligence FOSS tools can save around half of their annual business intelligence budget [Selim 09]. It is hard to say if this results take into consideration all the costs involved in business intelligence adoption. As mentioned before, indirect costs can be surprisingly high and often the major expense for FOSS solutions.

3.5 Short History of FOSS Business Intelligence Tools

The business intelligence world has always been dominated by proprietary software from big companies: Oracle Hyperion, SAP Business Object and IBM Cognos are just some of the many business intelligence proprietary software in the market nowadays.

The first business intelligence FOSS were small solutions covering just a specific need and focused especially on the data warehouse process. The best example is the development of MySQL, the popular open source database management system. Further examples can be Octupus for Extract Transform and Load (ETL) application, Mondrian for On Line Analytical Processing (OLAP) servers and JPivot for OLAP clients.

It took several years before the FOSS world could offer a stable and complete Business Intelligence solution [Ruffatti 08]. In the last five years the business intelligence FOSS tools have finally been recognised on the market as direct competitors to the proprietary software: a good example is the Gartner report [Richardson 09], which names the FOSS Pentaho among the most popular business intelligence solutions. More precisely, 6% of organisations already using business intelligence solutions planned to begin using Pentaho in the next 12 months [Richardson 09].

Business intelligence FOSS initiatives are even expected to grow in number and quality, with companies investing less money in proprietary software and gradually converting to open source solutions [Laplante 09]. Gartner expects the adoption of FOSS BI tools to double every year and to grow five-fold through 2012 [Bitterer 09]. The scenario seems too extreme, but it shows once again how FOSS are growing in the business intelligence scenario.

3.6 Market Trends: willing to try and resistance in changing

As shown by several statistical studies, many companies do not want to change the adopted business intelligence system, but show interest on new tools, often trying extra both proprietary and FOSS business intelligence solutions [KDNuggets 09] [Rexer 07] [Rexer 08] [Richardson 09].

Few managers would accept to change the business intelligence system adopted by their company especially if, during the previous years, the company had to spend a high amount of money in the acquisition of the software. In addition, companies are reluctance to adopt business intelligence platforms that do not come from their enterprise application vendor [Richardson 09] and consequently change the support provider. Finally, the survey in [Richardson 09] shows that 61% of firms already using business intelligence systems do not plan to begin using business intelligence from new vendors in the next year.

However, respondents from [Richardson 09] still show a tendency to purchase multiple platforms, which indicates some continued space for independent BI specialists. [Rexer 08] reports an average usage of 5.4 tools per user in 2008.

Finally we can say that company are reluctant to extra expenses, but they are open to trying new solutions for free. As we said in section 4.2, this is a key point to reach the market for those FOSS tools with zero direct cost.

4 BUSINESS INTELLIGENCE FOSS TOOLS

What is on the market?

The chapter focuses on the most interesting business intelligence FOSS available on the market. Before listing and describing the different solution, we must suggest some subcategories to split the business intelligence tools into: business intelligence is such a wide subject that it would make no sense comparing tools from different subcategories. For example, database management systems and analytical tools, both considered business intelligence programs, are in two completely different markets.

The analysis starts with the FOSS which deals with data storage (i.e. DataBase Management Systems, alias DBMS). Section 5.2 deals with data integration FOSS tools, while Section 5.3 describes the different analytical FOSS tools: simple reporting solutions are listed in Section 5.3.1, OLAP systems in Section 5.3.2 and data mining tools in Section 5.3.3. Finally the chapter introduces the business intelligence FOSS suites, which are described in Section 5.4. For each group, some interesting FOSS are listed and their distinctive characteristics are described.

4.1 DataBase Management Systems (DBMS) FOSS tools

A DataBase Management System (DBMS) is a group of software that let the user control a database: the user can save and extract information and, more generally, work on the stored data. The purposes of the adoption of DBMS can vary: some projects need small amount of data in order to create a statistical report, while others could need huge data sources to feed a data mining model [Grossman 09].

There are several database management systems in the market, both proprietary and FOSS solutions. The following list contains some of the most popular DBMS FOSS tools:

- Infobright + EnterpriseDB
- MySQL
- PostgreSQL

Infobright project, formerly known as Brighthouse, is subdivided into the community portal Infobright.org and the company Infobright.com. The DBMS is one of the few solutions created in order to maximise the data warehousing capabilities. More precisely, Infobright integrates MySQL into a Column-Orientated Relational DBMS.

In 2009, Infobright became an actual partner of MySQL [MySQL 10]: the DBMS provides some functionalities that MySQL is missing, becoming one of the more interesting solutions for analytic and business intelligence projects.

MySQL, probably the best example of FOSS, is often used by the FOSS world to defend the quality and possible success of FOSS solutions. MySQL is a Relational DBMS used for any kind of project. It is the most used FOSS DBMS in the market nowadays: [Madsen 09] shows that MySQL is used in 75% of the projects which adopt a FOSS DBMS. This popular tool is also used in some big businesses, like the massive Internet websites YouTube, Flickr, and Wikipedia [MySQL 07].

MySQL can go under the terms of the GNU GPLv2. In contrast, several proprietary licenses have been used to integrate MySQL into commercial, not-FOSS systems.

In 2008, MySQL has been bought by Oracle. The market operation raised many discussions, especially by FOSS supporters sceptic about the future of the project.

PostgreSQL is another popular FOSS solution in the market, supported by the big Internet community “PostgreSQL.org”. The software is an Object Relational DBMS which goes under a BSD license.

The company EnterpriseDB provides commercial support and maintenance for the FOSS DBMS. In addition, the company created an homonymous FOSS DBMS (i.e. EnterpriseDB) which is modelled on PostgreSQL and adds extra features, such as the Oracle PL/SQL (Procedural Language/Structured Query Language) compatibility.

[Madsen 09] registered a 44% usage of PostgreSQL; an extra 10% can be add for the version implemented by EnterpriseDB.

18% of the participants to the survey [Madsen 09] use a FOSS solution as main DBMS. The systems managed to reach such a high value, thanks to the experience and reputation gathered in the long time activity. Indeed, the DBMS solutions have been in the market for years, contrary to the young age of the solutions in the other FOSS categories,

FOSS database management systems, however, need to improve some of their features. The main problem are (*a*) low performance in the query execution and (*b*) poor embedded data integration functionalities [Madsen 09]. In addition, the most of the DBMS are still in lack of data warehouse specific features: just few solutions, like Infobright and Hadoo, have grown in the last years [KDD 0].

4.2 Data Integration FOSS tools

Business intelligence projects can need huge amount of data, often gathered from different sources. Before working on them, the analyst needs to merge the data in a unique source.

Different approaches have been used to solve this issue. Two of the most popular are the creation of a data warehouse using Extract Transform Load (ETL) techniques and the adoption of Enterprise Information Integration (EII) tools.

Extract Transform Load (ETL) normally refers to the set of functionalities used to create a data warehouse: the data is (*a*) extracted from different databases, (*b*) transformed into a unique standard and (*c*) loaded into the data warehouse. Thus a copy of the data is always available for analysis purposes: the data warehouse is an element independent from the transactional systems.

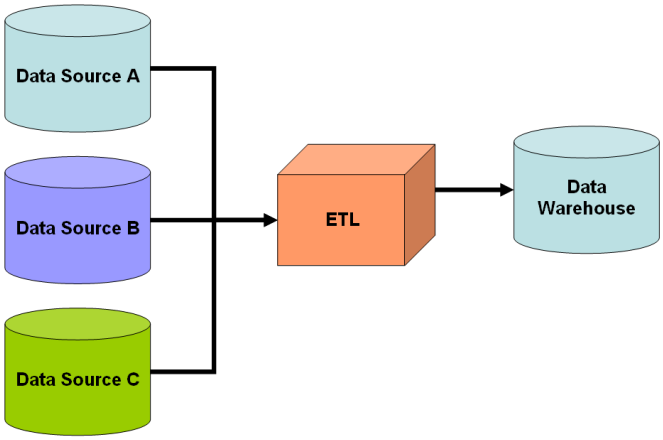

Image 4.1 (Data Warehouse configuration)

Enterprise Information Integration (EII), on the other hand, provides the user with a "virtual data warehouse". The data warehouse does not actually exist, but the EII tool offer the same functionalities: the user writes a query to access the virtual source and the data integration program subdivides the query into partial queries, one per each connected data source. Each data source processes the request and sends the results to the link "wrapper", which transforms them in order to create a homogenous view for the user.

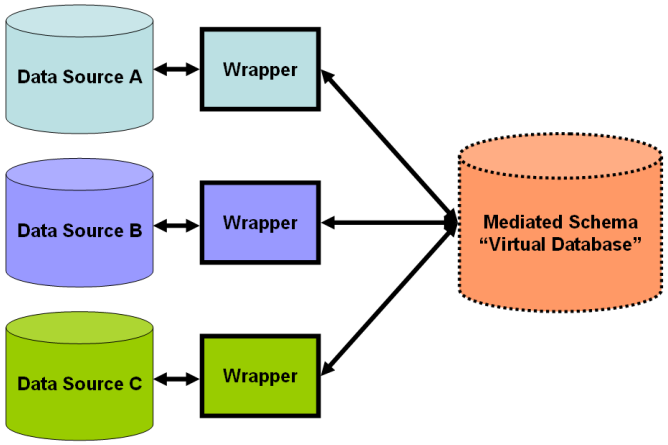

Image 4.2 (Data Integration configuration)

A deeper discussion about the characteristics, benefits and pitfalls of the two techniques is beyond the goals of this research.

The following list contains the most interesting data integration FOSS:

- DataCleaner
- Jitterbit
- Pentaho Data Integration (formerly Kettle)
- Talend Open Studio + Talend Open Profiler

DataCleaner is a rather young FOSS with high focus on data quality. The young has gained popularity in the last years: [Madsen 09] registers a 8% usage in the data integration market.

Jitterbit is the data integration FOSS solution offered by the homonymous supporting company. The downloadable edition of the tool, called Jitterbit Community, is primarily tested and supported by the FOSS community. It goes under the Jitterbit Public License, which is a modified version of the OSI certified Mozilla Public License v1.1.

Pentaho Data Integration, built into the Pentaho business intelligence suite, is the most popular ETL FOSS tool in the market [Madsen 09]. The data integration system is the further development of the project formerly called

Kettle, currently supported by Pentaho Corporation. Pentaho Data Integration is not the only tool of the Pentaho business intelligence suite that we will mention in this chapter.

Talend is the first data integration FOSS to have entered the Gartner Magic Quadrant [Friedman 09]. In addition, the Gartner report from November 2009 shows interest in the FOSS tools and defines “visionary” the approach to the market by the FOSS supporting companies. The authors from Gartner are, however, keen to add that the FOSS model is not proved to be stronger than traditional methods yet.

Talend supporting company offers a big selection of data management solutions, both FOSS, under the GNU GPL, like Talend Open Studio and Talend Open Profiler or “subscription based products”, such as Talend Integration Suite. The latter solutions, which go under a dual licensing policy, extend the basic FOSS with extra functionalities.

In the list of the most popular data integration FOSS in the market, the combination of the two FOSS solutions by Talend are placed as second [Madsen 09]. As we have seen, Talend is also mentioned in the Gartner Magic Quadrant, where each solutions is described through a list of strengths and cautions [Friedman 09].

4.3 Analytical FOSS tools

Business intelligence analytical tools can be subdivided into three levels [Albano 09]: (*a*) reporting tools offer the minimum functions to facilitate the decision making process, (*b*) On Line Analytical Processing systems let the user access the data using a multidimensional approach, (*c*) data mining tools aim to discover knowledge using automatic and semiautomatic models. We will focus on each of these levels in the following three sections.

4.3.1 Reporting FOSS tools

The ability to create a report is one of the most popular business intelligence functionalities. Whatever is the study field, the user needs to document the data analysis in order to display the results to the interested audience. This process is powerfully assisted by business intelligence tools, which consent to create human readable reports with an automatic, or at least semi-automatic, method.

The reporting tools are often implemented into a larger business intelligence solution. A good example is the BIRT-project module into the data mining tool KNIME.

The following solutions are some of the more interesting FOSS reporting tools currently available on the market:

- JasperReports
- OpenReports
- Pentaho Report Designer (formerly JFree Reports)

One of the oldest reporting FOSS solutions is JasperReports. The project started in 2001 and it is currently supported by the company JasperSoft and the Internet community “JasperForge.org”. The reporting tool, which goes under a GNU Lesser General Public License version 3, is included into the business intelligence suite JasperSoft developed by the homonymous supporting company.

JasperReports creates reports which can be saved using several file formats. The reports can also be showed in some Java enabled applications, such as J2EE or Web applications. An extended proprietary edition of the reporting tool, called JasperReport Professional, is offered by JasperSoft.

Not to be confused with OpenReport, OpenReports (“OReports.com”) is a web reporting solution that provides browser based reporting functionalities. The FOSS project, supported by “SourceForge.net”, includes four of the most

popular FOSS reporting engines: JasperReports, JFreeReport, JXLS, and Eclipse BIRT. The tool, based on Java technology, is license under the GNU GPL.

OpenReports Professional is an extended version of the FOSS solution, which goes under a proprietary license. The professional edition offers some extra functionalities, like user dashboards, alerts, drilldown charts and report usage statistics. The list price for OpenReports Professional is approximately $500 per server.

JFreeReport is a Java library with report generator functions, which goes under the GNU Lesser General Public License. It can send output to the screen and the printer, or export to text, CSV, HTML, XML, Excel, and Acrobat PDF files.

JFreeReport attracted enough interest that, in January 2009, Pentaho developing team decided to base on it the development of the reporting tools embedded into the business intelligence suite. Pentaho Corporation is currently offering the two FOSS solutions (i.e. Pentaho Reporting and Pentaho Report Designer) and the related enterprise edition.

4.3.2 OLAP FOSS tools

On Line Analytical Processing (OLAP) tools offer an approach to data sources, which is innovative compared to the one used in the “classic” On Line Transactional Processing (OLTP) tools. OLTP are system used to manage a transactional database, while OLAP solutions highly perform on complex queries in order to facilitate the user to run multidimensional data analysis.

OLAP can be defined as the second level of analysis, which overcome the simple reporting solutions we mentioned in the previous section. Many systems, however, combine the two methods: a good example is OpenReports which extend the reporting tool whit OLAP functionalities, using the Mondrian OLAP server and the JPivot Java library.

Many different models have been developed for OLAP solutions. We mention some of the more popular in the following list: (*a*) Multidimensional OLAP (MOLAP) systems use an optimised multi-dimensional array to storage the data; (*b*) Relational OLAP (ROLAP) systems adopt a schema structure coherent with the relational database they work on; (*c*) Hybrid OLAP (HOLAP) systems combine the two previous solutions. Apart from these main methods, the literature mentions Web-based OLAP (WOLAP), Desktop OLAP (DOLAP) and Real Time OLAP (RTOLAP).

Before focusing on the OLAP client side solutions, we must mention the most popular OLAP FOSS servers:

- Mondrian (ROLAP)
- Palo (MOLAP)

Mondrian, recently included into Pentaho project, is the most popular ROLAP server FOSS. The system, developed in Java, goes under the Eclipse Public License (EPL).

Palo project is supported by the Jedox company, which offers a FOSS edition of the system under the GNU General Public License version 2. Additionally, the user can acquire support from the company or buy the proprietary Palo Premium Edition. The latter system, which rises above OLAP functions, must be compared with the other business intelligence suites.

The following solutions are the most popular OLAP FOSS clients:

- JPivot
- JPalo
- OpenOLAP

The Java library JPivot provides a frontend OLAP table to the Mondrian ROLAP engine. The popular combination Mondrian-JPivot is used by many systems, such as JasperAnalysis, OpenReports and Pentaho, in order to provide OLAP functionalities.

JPalo is a group of tools developed by the supporting company Tensegrity Software. These FOSS solutions are created to manage a Palo OLAP server and analyse the data through table and cube browsing.

Palo Client and Palo Web Client are the main solutions: Palo Client has extra functionalities to model the data structure, while Palo Web Client is the web-based application to interface the server. Finally, the Palo Client functions can also be integrated in the FOSS OpenOffice: the add-on let the user create a personalised dashboard into the desktop application.

OpenOLAP is a FOSS system which integrates both ROLAP and MOLAP functions in PostgreSQL. The developing team is currently migrating the solution to make it also work with MySQL.

4.3.3 Data Mining FOSS tools

Data mining tools integrate different operations into a system in order to extract new and useful knowledge from large amounts of data. This process, normally facilitated by a clear and intuitive user interface, can be applied to many business problems and, therefore, adopted in many industries [Chen 07].

Chapter 6 illustrates the selection of the more interesting data mining FOSS solutions and the evaluation processes of the chosen tools. In the current section we must describe the two following relevant systems, which have not been evaluated:

- R project
- Weka

The R project, based on the R programming language developed at the University of Auckland, New Zealand, is a system environment for statistical analysis which goes under the GNU General Public License. Many tools have implemented the R statistical engine in order to offer extra functionalities and, especially, a more user friendly interface.

Weka (Waikato Environment for Knowledge Analysis) is the FOSS data mining suite developed in Java at the University of Waikato, New Zealand. The original WEKA project goes under the GNU General Public License, but the software has been integrated in several other business intelligence tools, such as Pentaho business intelligence suite. Since 2006, Weka has been part of the Pentaho project, which provides community resources for Weka users.

The University of Waikako, which is still the main developer of Weka, has carefully documented the software: [Witten 05] is a complete guide to the program and a general guide to data mining, while [Hall 09] illustrates the recent new developments for the project.

WEKA and the R project are both interesting solutions, widely used for data mining and statistical purposes. They are also embedded into the two evaluated tools (i.e. KNIME and RapidMiner), and especially for this reason, we decided to exclude them from the evaluation process.

4.4 Business Intelligence FOSS Suites

A business intelligence system with, at least, data integration and reporting functions is called business intelligence suite. The property business intelligence solutions, such as IBM Cognos, are normally business intelligence suites. In

contrast, FOSS tools are often restricted to a smaller set of functions. In the last years, however, some FOSS projects have started collecting and integrating tools for each of the different business intelligence categories.

Many of the tools mentioned in the last sections are integrated into the following most popular business intelligence FOSS suites:

- Pentaho BI Suite
- Jaspersoft BI Suite
- Spago BI

Currently, Pentaho BI Suite and Jaspersoft BI Suite are the biggest business intelligence FOSS projects. Both the solutions are provided by their supporting companies, which are strongly involved in the developing. SpagoBI, one of the main projects by the SpagoWorld community, was founded and is currently managed by the Italian company Engineering. Engineering is also a "strategic member" and co-founder of the FOSS community OW2 Consortium.

Since 2004, the year of the foundation, Pentaho has grown considerably. The third version of the project was released in May 2009 and new updates are frequently available.

Pentaho Corporation is the supporting company which offers both an enterprise and a community based edition. The main FOSS projects involved in Pentaho BI Suite are (*a*) Kettle for the data integration module, (*b*) Mondrian for data analysis, (*c*) Pentaho Reporting, which includes also the formerly known JFree Reports, and (*d*) Weka for data mining tasks.

JasperSoft project started in 2004 when the Panscopic project developing team joined Teodor Danciu, creator of the popular JasperReport library. The general overview of JasperSoft is similar to what we mentioned about Pentaho: both the solutions are subdivided into minor modules and are completely managed by their supporting companies, which offers both an enterprise and a community edition. In addition, both the community are still undersised.

"JasperForge.org" is the FOSS community website for the JasperSoft project. The main projects embedded into the business intelligence suite are (*a*) the data integration tool JasperETL, (*b*) the reporting library JasperReports, (*c*) the reporting and OLAP analysis server JasperServer, and (*d*) the report designer tool iReports. A module to integrate data mining functionalities is missing.

The business intelligence suite SpagoBI is supported by SpagoWorld, the FOSS initiative founded by the Italian company Engineering. SpagoBI can be considered a "pure" FOSS solution since it has a community edition only, under the GNU Lesser General Public License. Engineering is the main developer of the project and provides support, consulting and training for the system.

All the SpagoWorld project are hosted by the partner OW2 Consortium. OW2 Consortium, formed in 2007 from ObjectWeb Consortium and Orientware, is a global FOSS community which aims to "the development of open-source distributed middleware, in the form of flexible and adaptable components" [OW2 09].

5 DATA MINING FOSS

Data mining is a subject which overlaps several study areas. This characteristic can be traced in the different origins of data mining tools: mathematics and statistic studies (R project and Rattle), biology research groups (Databionic ESOM Tools), the pharmaceutics world (KNIME), and unsurprisingly computer science researchers (AlphaMiner, KEEL, Orange, RapidMiner, Weka) are all parts in the development of this kind of software.

In addition, the academia is strongly involved in the creation and management of these projects. Indeed, universities are a big source for data mining FOSS solutions.

The aim of this chapter is to document the selection and evaluation of the more interesting data mining FOSS solutions. This analysis is not meant to decide which of the selected solution is the best one, but to describe the positive and negative aspects of each of them.

Section 6.1 and 6.2 analyse the criteria used in the selection and evaluation processes, while the other sections show the results for RapidMiner (Section 6.3) and KNIME (Section 6.4).

5.1 Selection Process: which tools are the more interesting solutions?

As we mentioned in Section 5.4, many data mining FOSS are available on the Internet these days. However, we will focus only on FOSS which can be used for commercial purposes. It is behind the scopes of this research the evaluation of the solutions developed only for non-profit and research purposes.

After identifying all the data mining FOSS available on the Internet, the first stage of the selection process creates a smaller group of tools, which are consistent with the purposes of the research. The following selection criteria were used in the process: (*a*) the user must be allowed to use the software for commercial purposes, (*b*) the FOSS community behind the tool has to be active and robust and (*c*) the project must be running and healthy. This last point is measured through the date of the last released version of the software and the frequency of updates.

The selecting factors have cut off all the outdated tools, the ones with no community support and those with a license which avoid commercial usage. The remaining data mining FOSS went through a second stage, where we isolated the tools which resulted popular, standalone or innovative.

The following list contains the solutions that we considered the most interesting data mining FOSS available on the Internet:

- Databionic ESOM Tools
- KNIME
- Orange
- RapidMiner (formerly known as Yale)
- Rattle (based on Project R)
- Weka

Finally we chose to configure RapidMiner and KNIME on our local machine in order to make a deeper analysis.

RapidMiner is the most popular data mining FOSS these days [KDNuggets 09] [Rexer 08]. Rapid-I, the company which manages the RapidMiner project, has just released the new software version 5, which upgrade an already successful product, i.e. RapidMiner 4. We will describe the analysis of RapidMiner in Section 6.3.

Section 6.4 illustrates the results of the evaluation of KNIME. This product has attracted the interest of many authors: the literature focuses much attention on this tool, giving impressive reviews and underlining its innovative characteristics [Berthold 09] [Chen 07].

The analysis is based on our experience and on the information from the literature. The software testing results are compared with those gathered from the most popular data mining proprietary solution: “Clementine 12”.

5.2 Evaluation Process: which points to focus the attention on?

The traditional quality evaluation models cannot be applied to Free and Open Source Software (FOSS), as they cannot be tuned to evaluate both the software and the community as a whole [Samoladas 08].

The evaluation process must include the analysis of the tools’ functionalities and the background of the FOSS project. We cannot forget the influence of the FOSS community and how important is the role of the company which manages the project and provides support for the software.

Although we are aware of the fact that there are several FOSS quality models, such as OSMM, QSOS, or OpenBRR, we decided to base the evaluation on the criteria listed in the following sections. Those quality models are meant to be used for automatic and semiautomatic evaluations, while this research is based on a user analysis. Once again, the aim of this paper is the description of the positive and negative aspects for each of the selected solutions, and not a ranking list made to show which tool is the best one.

We initially investigated the literature about user analysis of FOSS solutions [Chen 07] [Golfarelli 09] [Thomsen 09] [Wang 01] and derived a new evaluation criteria group from the literature. Table 5.1 lists the key characteristics used to

analyse RapidMiner and KNIME, and summarises the criteria used in the analyses from the literature.

A. Software related characteristics				
A.1 Functionalities	**Chen [10]**	**Golfar. [11]**	**Thom. [12]**	**Wang [13]**
Integration with different techniques	✓	✓		
Integration with data sources and other products	✓	✓		✓
Integration with user's platform	✓	✓	✓	✓
Extensibility	✓			

A.2 Performance	**Chen**	**Golfar.**	**Thom.**	**Wang**
Data pre-processing capability	✓			
Algorithms' performance	✓			
Stability-reliability	✓			✓

A.3 User Friendly	**Chen**	**Golfar.**	**Thom.**	**Wang**
Usability	✓	✓		
Data and model visualisation	✓			

B. Community and supporting company related characteristics				
	Chen	**Golfar.**	**Thom.**	**Wang**
License	✓	✓	✓	✓
Community	✓		✓	
Support			✓	✓
Documentation			✓	

Table 5.1 (Evaluation criteria used in the paper and in the literature)

[10] [Chen 07]
[11] [Golfarelli 09]
[12] [Thomsen 09]
[13] [Wang 01]

[Chen 07] illustrates the analysis of twelve data mining FOSS. The paper focuses the attention on the software related characteristics, but does not consider other important criteria, such as support and documentation. The list suggested in [Chen 07], however, contains the core criteria used in our analysis.

Some extra criteria have been added in order to complete the analysis. The characteristics related to the community and the supporting company, which were missing in [Chen 07], are especially used in the FOSS evaluation illustrated in [Thomsen 09]. The survey, which describes the more interesting FOSS for each business intelligence area, analyses the FOSS solutions using both criteria common to all the business intelligence software and area-specific characteristics. The authors, in particular, underline the important roles covered by the community and the supporting company. Unfortunately, [Thomsen 09] does not focus on the data mining FOSS tools and, therefore, we could only inherit the few criteria which refer to all the business intelligence software categories.

Another business intelligence FOSS survey is illustrated in [Golfarelli 09], which describes three of the more interesting business intelligence suites in the market. Although the three FOSS projects are well introduced, the paper lacks many important criteria necessary to make a deep and exhaustive analysis.

Finally, [Wang 01], one of the first paper to list the criteria to consider in a FOSS analysis, describes an easy evaluation model and the detailed instructions to apply it. We could use only few of the criteria from [Wang 01] since the paper does not focus on either business intelligence or data mining solutions.

The data mining FOSS evaluation illustrated in this paper uses the criteria listed in Table 5.1. A detailed description of the criteria used to study the software characteristics can be found in Section 5.2.1, while Section 5.2.2 focuses on the criteria related to the supporting company and the community.

5.2.1 Software Related Characteristics

FUNCTIONALITIES: which technical features are supported?

- **Integration with different techniques.**

As we already mentioned, data mining is a discipline which overlaps several areas. In order to suite the large number of problems, many data mining algorithms have been created.

Some developers tend to upgrade the largest number of techniques in their data mining solution and use the number of algorithms for marketing purposes. This value, however, is not indicative, because many algorithms could share the same purpose and thus solve the same kind of problem.

A good data mining system have to offer and integrate the smallest group of algorithms able to solve any kind of problem. Moreover, the tool has to provide techniques for all the different steps of the analysis, from the pre-processing to the modelling and evaluation.

- **Integration with data sources and other products.**

There are many ways the user can provide data to data mining tools: she can create a flat file or arrange the data in a specific file format, she can load the data in a database or even create a data warehouses. It is easier to use a data mining tool which can access as many different data sources as possible.

Consequently, the data are often handled by another software before being used by the data mining tool. Standards and protocols are important in order to maximise the interoperability between systems – either FOSS or proprietary software [Chen 07]. A good data exchange can save the meta-data information, gathered in one tool, and make them available in another software. Finally, the interoperability can be extended to model exchange.

The Predictive Model Markup Language (PMML), considered the most used data mining standard, is developed by the vendor led consortium Data Mining

Group (DMG). The more relevant companies involved in data mining support this group and the PMML standard [DMG 10].

The PMML uses a XML format, which guarantees platform independency. A rich set of features is developed in order to let the user create a standardised report of the used statistical models, data mining processes, data preparation functions, and post-processing elements [Grossman 09]. The user can, therefore, take full advantages of the software interoperability [Pechter 09].

Finally, another level of product integration is the "backward compatibility": the new version of the software has to be able to open and handle those documents created with the previous software releases. If this is not possible, the user could end up losing hours of work to recreate them.

- **Integration with user's platform.**

A final note about integration is about the user's platform. The data mining solution is not valuable if it cannot be used, at least, on the most popular operative systems. We should, therefore, consider which hardware and software platforms the tools can be used with [Thomsen 09].

- **Extensibility.**

"Good extensibility means easy integration of new methods" [Chen 07].

A good software should be easily extendable. To facilitate the user in customising and integrating the tool into any technical environment, it is important to implement the FOSS solution in a popular and easy language, such as Java [Wang 01].

In addition, the architecture should allow an easy, preferable automatic, installation of new features and patches provided by the community or the supporting company.

PERFORMANCE

- **Data pre-processing capability.**

Data miners spend a large proportion of their time working on data pre-processing [Chen 07]: more than a third of their time is spent accessing and preparing data [Rexer 08]. You cannot use the data if you do not make them ready to be manipulated and if you do not understand them completely. Even if ETL (Extract Transform and Load) functions are not part of data mining, a system with good pre-processing capability facilitates this step and, therefore, can more easily solve the problem.

- **Algorithms' performance.**

Data mining projects often need to handle big amount of data, up to several terabytes [Madsen 09]. Naturally, the bigger size of the data source you have, the longer it takes to get the results. Unfortunately the processing time often grows exponentially to the size of the data.

We will test the FOSS solutions using a dataset of 302'900 records with 28 classes. The proprietary data mining software Clementine 12 will be used to compare the results.

- **Stability-reliability.**

No user would ever base her efforts on a program which cannot be consider robust and highly reliable. The stability must be proved through a large number of applications, case studies, evaluation and reviews [Wang 01]: it takes time and efforts to build a good reputation.

Another important factor is the frequency in the release of bug fixing upgrades. It is a problem to find a bug, but it is even a bigger problem to have to wait long before having it fixed.

USER FRIENDLY

- **Usability.**

The learning curve of a good data mining system should not be too steep, in order to help the beginners. A user friendly system facilitates the usage for the users, both beginners and experts, since it provides a logic approach to the tools. A Graphical User Interface (GUI) is usually preferred to a command line interface.

The usability can be recalled even in the work of developers and system administrators: a good data mining solution should be easy to install and manage [Golfarelli 09]. This can be achieved through a sensible development, good guides and help manuals.

- **Data and model visualisation.**

An important user friendly aspect for data mining tools is the visualisation of data and models. In view of the fact that the user could need to handle huge amount of data, it is better to have functions which facilitate the analysis and interpretation of data. This can be applied to all the levels of data mining, from the pre-processing to the result evaluation.

5.2.2 Community and Supporting Company Related Characteristics

- **License.**

As we said in Section 2.4.1, the license is an important characteristic for a FOSS solution. Even if a license can be hard to interpret, the user should spend some time going through it to understand what she is allowed to do with the

FOSS tool. This is especially important if the user's intention is to integrate the software in other systems, since some licenses are not compatible with others.

- **Community.**

A community with an active environment (forum, wiki, mailing list, etc) can increase the software adoption and facilitate its usage and, in addition, a FOSS supported by a valuable community is more often updated [Chen 07]. Finally a constructive community is often a great source of new ideas and functionalities to develop for the system.

- **Support.**

Once again we have to mention the support: the commercial usage of a FOSS solution cannot be based just on the community. It needs a company which can provide professional support, at reasonable price [Wang 01]. At the same time the supporting company can offer – and sell – maintenance and customised development of new tools to integrate into the FOSS solution.

- **Documentation.**

To augment the value of the FOSS system, the supporting company, or the community, should write a complete guide for the software. Additional interesting resources would be the APIs, training instructions and any other document which could help the user to use and extend the software.

5.3 RapidMiner

RapidMiner, formerly known as Yale (Yet Another Learning Environment), has been the data mining FOSS most used for commercial purposes in the last years [KDNuggets 07] [KDNuggets 08] [KDNuggets 09]. The latest survey considers RapidMiner the second most popular data mining tool after Clementine, taking into consideration both proprietary solutions and FOSS. In addition, the software is one of the most mentioned solutions in the literature about data mining FOSS.

Image 5.2 (RapidMiner Logo)

The software has been developed by the Artificial Intelligence Unit of University of Dortmund since 2001. The first version of RapidMiner was released in 2002 and since 2004 the open-source edition has been hosted by the website SourceForge.net. The German company Rapid-I is currently managing the project and providing commercial services for RapidMiner users.

The 4.5 version and the 5 beta version were released in October 2009. Version 4.5 is an upgrade of the old RapidMiner 4: this software was very successful and drew extra attention to data mining FOSS. The beta version of RapidMiner 5th edition, which is stable and reliable enough to be used, has a different GUI implementation and offers an even broader set of functions. The "RapidMiner 5 Release Candidate", released in December 2009, fixed the – mostly minor – bugs.

The algorithm library from Weka is fully integrated: all learning schemes and other functionalities from the Weka learning environment are available and can be used like all other RapidMiner operators. However, the support for these

nodes is poor: RapidMiner developers do not offer any direct help and pass the buck to the Weka community.

Finally it is important to mention the extra functionalities offered by RapidMiner: the data mining system has evolved in a wider business intelligence system, integrating ETL, OLAP and reporting tools.

5.3.1 Software Related Characteristics

FUNCTIONALITIES: which technical features are supported?

- **Integration with different techniques.**

RapidMiner offers enough operators to cover some of the following major data mining issues, such as decision trees, classification rules, regression, deviation detection, clustering, association rule discovery and sequential pattern discovery. The user can select between around 75 modelling algorithms accordingly to her needs. Additional nodes let the user choose the favourite methods for the validation and evaluation processes.

The user must often try different solutions for the same problem, for example, whenever it is necessary to choose the best performing algorithm. In these situations, the selected model node can be easily replaced with another one from the same group, without modifying the flow structure. This is possible since similar nodes have the same inputs and outputs.

For each data mining category, the Weka algorithms are integrated and grouped in a separate folder. Unfortunately they often work in a slightly different way from the other RapidMiner operators, making it harder to implement them into the flow.

As we mentioned in the introduction of the chapter, the ETL, OLAP and reporting functions are implemented through a big number of operators. These

nodes are well integrated into the system and cover the most important functionalities required by the common user.

All in all, RapidMiner has many nodes, although some of them are quite similar. The developers approach is to augment the number of nodes and keep the nodes' tuning settings as simple as possible. Dissimilar is the approach used by Clementine and KNIME, which offer less nodes, but a bigger selection of options.

- **Integration with data sources and other products.**

RapidMiner can import data from CSV, ARFF and XLS file formats, which are some of the most used for data storing. The software can also handle some proprietary file formats which are inherited from popular data handling solutions: good examples are the SPSS and the C4.5 formats. Additional data sources are web pages, documents stored in ASCII, PDF, HTML and XML format, audio data and time series data.

The user can adjust the node settings for the data importing, in order to keep the meta-data information. If this usually successful process would unfortunately fail, the user can easily fix the problem using the "Data Transformation" functions.

Although the number of data export functions is less extensive, RapidMiner is still able to save the data using the more important file formats. The connection with several different kinds of database is also supported.

In our analysis, we successfully imported and exported data from CSV and XLS file, and tested the connection with databases stored in MySql and Microsoft Access.

Finally, RapidMiner has its own import/export object, called "repository". The data saved into a repository is enriched with robust meta-data. These objects work much better than any other RapidMiner import/export object since they are perfectly integrated into the flows. Any other node can access the meta-data stored in the repository and use the information to populate its own

settings. The user, therefore, is facilitated in the selection of attributes, values and types.

A part from data and meta-data, RapidMiner can handle extra data, such as models, attributes, parameters, threshold and performance results. Any of them can be saved and loaded using specific nodes. These functions are helpful to migrate data from one flow to another.

One of the most interesting future RapidMiner implementation is the support of the Predictive Model Markup Language (PMML). The standard is not integrated yet, but it is expected for the first quarter 2010 at the latest [Rapid-I 10b]. In the meanwhile Rapid-I has been elected "Observer Member" for the Data Mining Group [DMG 10], which is the developing group for the PMML standard.

Backward compatibility is also supported by RapidMiner: the new version 5 can import the flow models created with the precedent release, through the XML structure used to store the process information. RapidMiner 4 saves the flow directly into a XML files, while RapidMiner 5 can import and export the flow using the XML or use a new file extension "rmp".

- **Integration with user's platform.**

RapidMiner 5 can be installed on Microsoft Windows platforms using one of the two automatic executable installer, created to support both 32bit and 64bit systems. Users with different platforms need to install a Java Runtime Environment, version 5 or higher, on their system. Afterwards, the RapidMiner system can be installed on the Java platform.

- **Extensibility.**

Some official extensions are available on RapidMiner webpage. Among them, we already mentioned the Weka extension, which offers all the algorithm implemented in the popular data mining solution.

Additional data types can be used extending RapidMiner with (*a*) the "Text" plugin, which is meant to facilitate the textual data handling, (*b*) the "Value Series" methods meant to extract data automatically from series data, such as audio files, (*c*) the "Data Stream" extension which provides extra operators for data stream mining and finally (*d*) the "Conditional Random Field" plugin which provides some basic operators for named entity recognition [Rapid-I 10b].

We successfully tested the installation of all RapidMiner official extension on the beta version 5. The Release Candidate improved the extensibility function creating an automatic installation for the official extensions. Unfortunately there are still some problems with this upgrade, which make more complicated to install the extensions.

Finally, as we already mentioned, RapidMiner is developed in Java. The usage of this popular coding language facilitate the users who want to extend the software with custom functions or integrate it into the proprietary tool.

PERFORMANCE

- **Data pre-processing capability.**

RapidMiner offers a robust group of ETL functions which can pre-process the data before analysing and creating the model. The number of nodes, grouped in the "Data Transformation" folder and subdivided accordingly to their function, is big enough to solve any issue. The user can find any relevant data handling function, such as type conversion, filtering, sorting or name, role and values modification.

We found it easier to pre-process the data within other software (e.g. Microsoft Access or SQL Server) before importing the data in RapidMiner, primarily because we already had major experience in those data handling tools. The RapidMiner ETL functions, however, have been helpful, especially for minor

changes or to get the data ready for testing different algorithms on the same data set.

As we will discuss in the next sections, RapidMiner has a strong data visualisation tool, which is particularly handy for those analyses associated with the CRISP "Data Understanding" process [CRISP 10].

- **Algorithms' performance.**

For all the several RapidMiner nodes we tried, the FOSS system performed well, often even better than the proprietary Clementine 12. The testing flows were analogous in all the systems and the nodes were tuned with similar settings.

RapidMiner results take into consideration the time used to store the data in a repository. Even if RapidMiner can create a model on data read directly from a file, it is easier to handle data stored in a repository. Hence we decided to include this extra time to measure the performance.

The results we obtained with RapidMiner are definitely positive. The performance, however, is much lower if the flow contains a Weka algorithm (more precisely, we tried to use the node "M5P" to create a model tree).

- **Stability-reliability.**

The tests run on RapidMiner did not shown any major problem, apart from some minor bugs in the beta version. The Candidate Release have fixed the bugs, and although we experienced some problems with the automatic installation of the extensions, RapidMiner 5 can be considered a stable version

RapidMiner is considered the most popular data mining FOSS solution [KDNuggets 09] [Rexer 08]. Many analyst, who decided to rely on RapidMiner and adopt it as primary tool, considered it mature enough to support their work. Another positive indicator for the stability of the system is the high percentage of users who are satisfied by RapidMiner and intend to keep on using it [Rexer 09].

Further information about RapidMiner can be found in the literature: [Chen 07] illustrates some data mining FOSS solutions and considers the system one of the best tool in the market.

RapidMiner seems to be a robust solution which is appreciated by users and researchers. In addition, the supporting company Rapid-I frequently releases new patches, which augment the reliability of the software.

USER FRIENDLY

- **Usability.**

RapidMiner has significantly improved the usability in the last version: the tree graphical interface, used till RapidMiner 4, has been replaced with a flow chart, which facilitates the beginner users, especially for intricate processes.

In addition, the Graphical User Interface is built in order to keep all the important elements always available on the screen (Figure 6.2).

The process is shown in the bigger section, in the middle of the window. The user can structure the process creating a flow: a rectangle is created for each node, while a line is drawn to connect the nodes which exchange information. The colours are used wisely to underline the different functions of nodes and connections, in order to guide the user in the creation of a correct flow. The "Overview" tab is helpful to navigate in a extended process which overlaps the size of the "Process" tab.

The node options are listed in the "Parameters" tab, which is always visible in the right side of the window. For each option, the user can read an explanatory text in the "Help" tab. The fact that the "Parameters" tab, used to access the node's options, is always visible makes it much faster and user-friendly than the approach used by Clementine or KNIME (i.e. opening a window containing the options by double-clicking on the node).

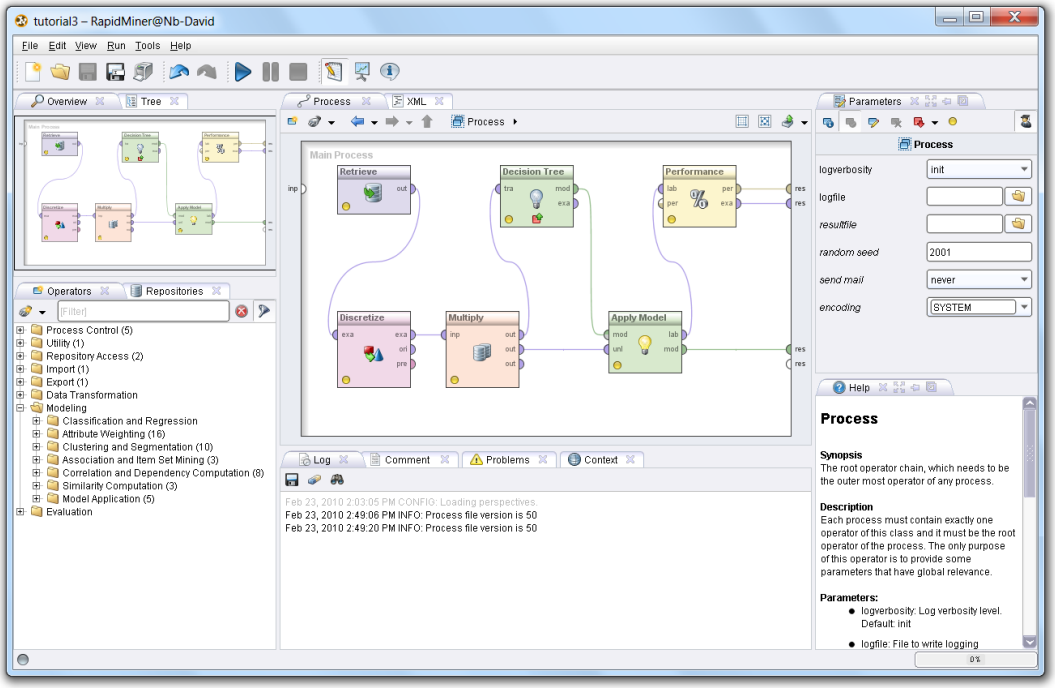

Image 5.3 (RapidMiner Graphical User Interface)

The “Repositories” tab list all the RapidMiner sources available for the user. The selected repository can be dragged and dropped into the “Process” tab in order to create a “Retrieve” node and link it to that source.

The “Operators” tab has a similar structure: it contains a list of all the nodes available in RapidMiner, subdivided in groups in a four levels hierarchy structure. The eight main groups are: (*a*) process control, (*b*) utility, (*c*) repository access, (*d*) import, (*e*) export, (*f*) data transformation, (*g*) modelling and (*h*) evaluation. A search function, inserted into the “Operators” tab, facilitate the user to locate a specific node.

The nodes, however, are so numerous that it is not easy to organised them in a logical hierarchy. The current structure is fine, but it could be improved in order to optimised the node selection process.

Each node in the “Operators” tab is described with a short text which appears in a popup when the mouse is over it. These descriptive texts, particularly helpful for the beginners, should be written better. Moreover, it

would be helpful to extend the popup help function to other elements, such as the folders in the hierarchy list or the settings in the “Parameters” tab.

The “Log”, “Problems” and “System Monitor” tabs are a great help to keep the user aware of what is happening. In details: the first one is a timed list of all the events; the second shows the errors in the current process and suggests possible solutions; the third is a screen of the memory used by RapidMiner. All these important tools are extra functionalities, which are actually missing in the popular SPSS Clementine 12.

Finally, it looks like RapidMiner inherited the Weka nodes without taking care of some important aspects like usability. The settings for the Weka nodes, shown in the “Parameters” tab, are noted with very unclear abbreviations which make the nodes inherited from the Weka extensions almost unusable.

- **Data and model visualisation.**

RapidMiner has an interesting visualisation tool which can plot the data in many different charts. This tool, called “Plot View”, is clear and easy to use even for beginner users.

Once the data table is loaded into the “Result Overview” page, the user can read a summary of the data, information about the meta-data, or analyse the data through the “Plot View”.

The data can be explored using around thirty different plots. For each kind of graph, a list of settings facilitates the data visualisation and comprehension. Additional options let the user edit the colours and shaped used in the chart, in order to underline the interesting results.

The models visualisation is similar and equally good. It is also possible to plot multiple models, or results, in the same window. This is particularly important to understand better the results obtained in a loop-process[14].

[14] Loop-process: process where the same group of nodes is applied to the data recursively.

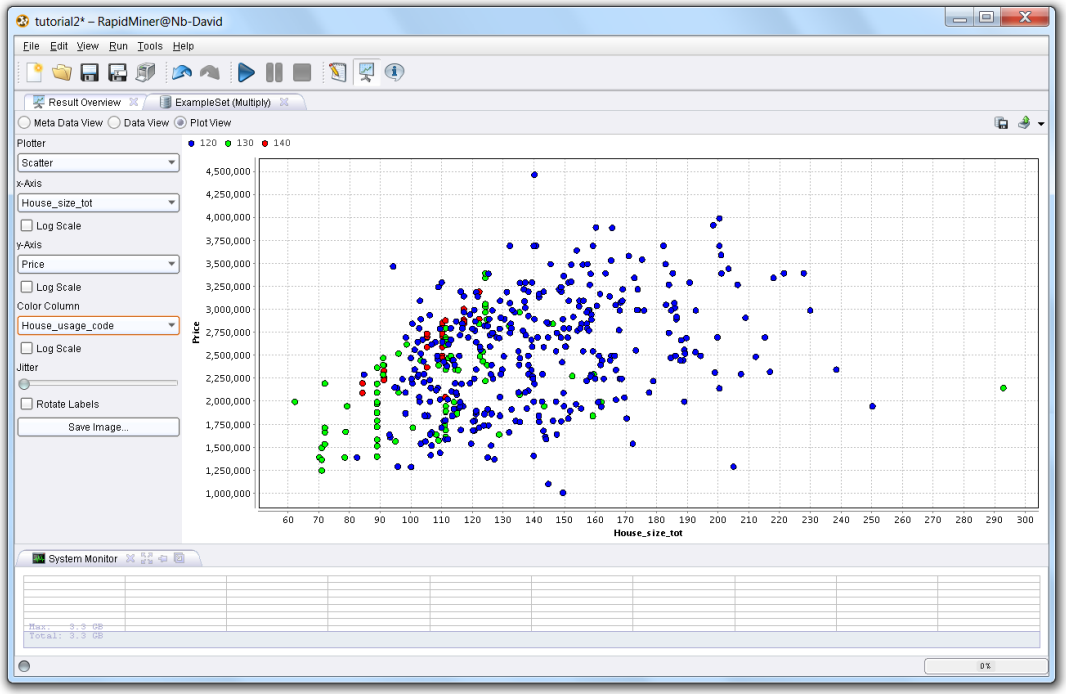

Image 5.4 (RapidMiner "Plot View" Tool)

Finally, the visualisation tool could be further improved with extra functions to export the results: for example, a reporting function could gather the results selected by the user and save them in a text document.

5.3.2 Community and Supporting Company Related Characteristics

- **License.**

RapidMiner follows a dual licensing policy: the community edition is under a GNU Affero General Public License v.3, while the enterprise edition has a proprietary license.

The GNU Affero General Public License (AGPL) v.3 is almost identical to the GNU GPL v.3, apart from the Affero clause which deals with software running over a computer network. The Free Software Foundation recommends the GNU

AGPL for any software that will be deployed as a network service. The license is also approved by the Open Source Initiative.

The user can decide to download and use RapidMiner without paying, using the community edition, or buy a proprietary license to embed RapidMiner into her own software or extend the system for a customer.

Despite some other FOSS software under a dual license policy, both the community and the proprietary versions are exactly the same.

- **Community.**

The community behind RapidMiner is smaller than expected. Even if the FOSS solution attracted lots of attention, Rapid-I is still running the major part of the software development. The forum is a good indicator of this phenomenon: the number of topics in the "Development" section is a minor contribute to the forum, which is dominated by the "Problems and Support" section.

Rapid-I, however, is taking good care of the community. Some Rapid-I developers manage the community wiki, write the official RapidMiner blog and, aware of the importance of the forum, monitor its activities daily. In our experience, we have received an answer from an official member of the Rapid-I staff, to any question posted, in reasonable time (less than 24 hours). These online resources are available for any user and no registration is needed.

RapidMiner is supported by SourceForge.net, one of the main portals for FOSS projects. The website hosts the system files and offers extra functions to support the project and its users. The "Feature Request Tracker", for example, is made to gather new ideas and suggest them to the developing team, while the "Bug Report Tracker", shown in SoruceForge.net and host by Rapid-I webpage, is supposed to facilitate the bug fixture and the release of new patches.

- **Support.**

The simpler support is offered by a list of webinars which can be purchased through the Rapid-I shop page at a net cost which goes from 150€ to 300€. If the

user would rather have a training course, she can book a personal or group session in Dortmund, Germany, for two days. The cost is between 1000€ and 1500€, accordingly to the group size.

The RapidMiner enterprise edition can be bought contacting the Rapid-I. As we said, this edition is technically identical to the community based edition but, in addition, it offers extra support which can be adapted to suit the customer's purposes. The user can requested the Enterprise Edition under both a FOSS license or a close source license.

Finally, Rapid-I invites the customers to contact them for any further request, such as professional support, consultancy and data analysis.

- **Documentation.**

RapidMiner 4 is well documented with online resources and downloadable manuals. The more interesting resources are the "GUI Manual" [Rapid-I 09a] which introduces the software interface, the video tutorials [Rapid-I 10c] and the thick "User Guide, Operator Reference and Developer Tutorial" [Rapid-I 09b] which illustrates all the features and operators available for the user. In addition, the "RapidMiner 4 Class Documentation" [Rapid-I 09c] is the list of official APIs that can be handy for users who want to extend the software or integrate it into their own code.

These valuable papers has not been created for the new release yet. Indeed, RapidMiner 5 is in big lack of documentation. At the time of writing, the available resources for the new version are just some short guidelines (e.g. the "Installation and Starting Guide" [Rapid-I 10d]), which can be found in the Rapid-I web pages. The daring customers who want to start using the new version, have to learn from experience and from the built in tutorial. This interesting resource introduces the program in twenty six steps, but it is not as clear as expected: the descriptions are short and it is often hard to understand the purposes of the flow used in the example.

5.4 KNIME

Konstanz Information Miner

KNIME (Konstanz Information Miner) is the data mining FOSS solution developed by Konstanz University. The original project was created in 2004 by the chair for "Bioinformatics and Information Mining" at the University of Konstanz, Germany. KNIME, which is currently used for teaching and researching at the University, became soon popular in the pharmaceutical world. In addition, in the last years the system has increased its user share spreading in more industry areas.

Image 5.5 (KNIME Logo)

The FOSS solution is growing in popularity. It is also easy to find good reviews of the software in the literature about data mining FOSS: [Chen 07] survey is a good example.

KNIME, which is still younger and less popular then RapidMiner, has some singular interesting functions to analyse. The software engineers from Konstanz University focused their efforts on special data handling (e.g. pharmaceutical data) and on the "hiliting" function. Hiliting is a original selection tool that let the user analyse the data with a straightforward method: the data selected by the user in one view are immediately hilited (selected) in all other views too. The user, for example, could hilite some unexpected records on a graph and analyse their characteristics on the data table.

The user interface is intuitive, therefore the learning curve for this tool is not as steep as the one for RapidMiner. The nodes are subdivided into sensible groups and per each node or group a description is provided: the user,

therefore, can easily navigate through the node selection. The GUI makes use of a modular workflow approach which opens a new window for each data view. The data views are easy to handle using the “hiliting” function and some extra features to modify colour, shape and size of the plotted values.

5.4.1 Software Related Characteristics

FUNCTIONALITIES: which technical features are supported?

- **Integration with different techniques.**

Clementine 12 offers 25 nodes for the modelling algorithms, RapidMiner has around 75 nodes (excluding the ones from Weka), while the number of nodes implemented in KNIME is somewhere in the middle, around half the amount of nodes in RapidMiner. However, the algorithms used in KNIME’s nodes still cover the more important modelling issues.

In addition to those nodes, KNIME can expand the library importing the nodes from some original extensions. The more interesting to mention are the Weka and the R-project extension. These extra nodes are grouped into a separate folder, which also contains some features to integrate the modelling algorithms into the usual flows.

Finally, the “Java Snipper” node and the “Python Scripting” extension allow the user to execute arbitrary Java code and Python scripts. The user can easily integrate a well known coding language in the flow to manipulate the data.

We will mention some more extensions and their extra nodes. They are all well integrated with the basic functions of KNIME.

- **Integration with data sources and other products.**

The list of importing nodes offered by KNIME is not as exhaustive as the one for RapidMiner. By contrast, the few basic nodes can be optimised to many issues, working on their extensive settings. The approach used in KNIME is similar to the one used in Clementine.

The “File Reader” node is a good example: the node implements a generic function to read any kind of flat file. The data is read following the user settings and displayed in a table preview, which facilitates the user in discovering eventual importing errors.

In addition, KNIME supports any database which can be linked through a JDBC- compliant bridge. A whole group of nodes is dedicated to database connections.

In addition to nodes for flat files and databases, KNIME uses some specific nodes to handle the ARFF and PMML standards. The ARFF file format is a popular standard for data mining systems, developed at the Department of Computer Science of The University of Waikato for use with the Weka machine learning system. The second standard is the already mentioned Predictive Model Markup Language (PMML). As we said, PMML is the standard used to import and export data handling models, developed by the Data Mining Group.

Finally some extensions can augment the data types used by KNIME. A good example is the data type used to represent molecules, particularly interesting for chemical and pharmaceutical industries.

- **Integration with user’s platform.**

The KNIME developing team declares that KNIME Desktop version 2.1 has been tested on Windows XP, and Vista, both 32 bit and 64 bit versions and Linux, both 32 bit and 64 bit [KNIME 10]; we run our tests on Windows Vista 64 bit and Windows 7 64 bit.

An experimental KNIME build for MacOS X was released with version 2.1. This requires a 64 bit Intel-based architecture with Java 1.6.

The KNIME Software Development Kit (SDK) is a package containing Eclipse Ganymede Sr2 with the Graphical Editing Framework, Java Runtime Environment 1.6 and KNIME 2.1. KNIME SDK is available for Microsoft Windows and Linux systems, both 32bit and 64bit versions. This package is meant to facilitate the development of new extensions.

- **Extensibility.**

KNIME is developed in Java and based on the Eclipse project: these two platforms facilitate the possibilities to extend the software. Moreover, the creation of the KNIME Software Development Kit (SDK) is supposed to maximise the extensibility of the program. The KNIME group claims that thanks to the KNIME SDK and through its modular API, it is easy to extend the data mining tool.

Some original extensions, offered by the KNIME developing group, can be installed to provide additional functionalities. The statistic nodes from the R project, for instance, can be linked to the system through the "R Integration" module. The "Weka Integration" is another extension which offers around 100 nodes from the popular data mining solution.

In addition KNIME can be extended with (*a*) some chemistry functionalities to use extra types and nodes; (*b*) the Eclipse project BIRT (Business Intelligence and Reporting Tools); (c) the "Math Formula" node, which can evaluate free-form mathematical expressions; (*d*) a node made to export data as XLS files; (*e*) the "External Tool" feature made to run an external program on the data; (*f*) the "Python Scripting" plug-in; (*g*) the "Distance Matrix" package; (*h*) an integration of the library for support vector machines (LIBSVM).

All the extensions can be installed automatically using the "Software Updates and Add-ons" feature. This tool is helpful to manage the numerous plugins created for KNIME, which are divided between "Installed Software" and "Available Software". The user can use the extensions tool to access to the

information about the extensions, such as version number, developer team or extra details about the plugin functions.

PERFORMANCE

- **Data pre-processing capability.**

The pre-processing nodes, which are collected into the “Data Manipulation” folder, do not have enough functionalities to respond to every kind of need.

Initially, the KNIME developing team decided to focus on the data mining functionalities, while some data handling nodes have been added only in a second time. The developing team is still working on this part of the project: for example, KNIME 2.1 has introduced a whole group of functions which can handle the time series.

Finally, as we already mentioned in the previous sections, KNIME is highly tuneable through the nodes’ settings, which can facilitate the user in the data manipulation. Some extra nodes, however, would be of great help.

- **Algorithms’ performance.**

The general results obtain with KNIME are at the same level of RapidMiner and Clementine. The system spends some extra time on analysing and loading the data, but performs better on the algorithms.

The intermediate results, which are stored during the flow execution, can be accessed at any time, without spending extra computational time. This characteristic is a significant improvement for the system performance, since each node needs to be processed just once. The user could even change the following nodes, without losing the information already processed.

- **Stability-reliability.**

We have tested KNIME for a couple of months without having stability problems. The tests were conducted in order to cover the largest number of different techniques and to exploit all the characteristics of the software.

A webpage in KNIME official site is dedicated on the known issues that the developing group is working on. The list is particularly helpful for those users who experience a problem and need to know if it is a machine or user mistake.

Once the issue is solved, a patch is released on the website. This can be installed manually or automatically, using the “Software Updates and Add-ons” window.

Finally, the analysis read in [Chen 07] shows great results for KNIME, which is considered the more interesting system between data mining FOSS solutions.

USER FRIENDLY

- **Usability.**

KNIME has a positive learning curve: we managed to create the first flow in few minutes and get interesting results in the first day.

The Graphical User Interface (GUI) is basically similar to the one used in RapidMiner. The window is subdivided into several tabs, each having a different function. Each process is created with colourful nodes and connecting lines, and displayed in a tab in the middle of the window. The flow chart is used in RapidMiner, KNIME and Clementine: it seems like this approach has become the more popular for data mining software. In contrast with RapidMiner, several processes can be open in KNIME at the same time, without reducing the performance.

As we already mentioned, the process flow is progressively saved in order to ensure that any table or chart can be always available for the user. Final and intermediate results are stored on the hard disk, together with the flow files: the

user can, therefore, open them in any moment, even days after the process creation. When the flow is modified, any intermediate unaffected node is still available, together with its table or chart.

Similarly to RapidMiner, KNIME offers the possibility to select and arrange the tabs accordingly to the personal preferences. Many of these tabs have similar functions, but different names are used for the two systems.

The “Outline” tab is an overview of the process. The “Console” tab shows the log file, which is automatically updated. The operators list can be found in the “Node Repository”, while the “Favorite Nodes” collect the personal favourite node, the most frequently used and the last selection. The node selection is much easier than in RapidMiner, simply because there are less nodes and they are organised in more sensible folders.

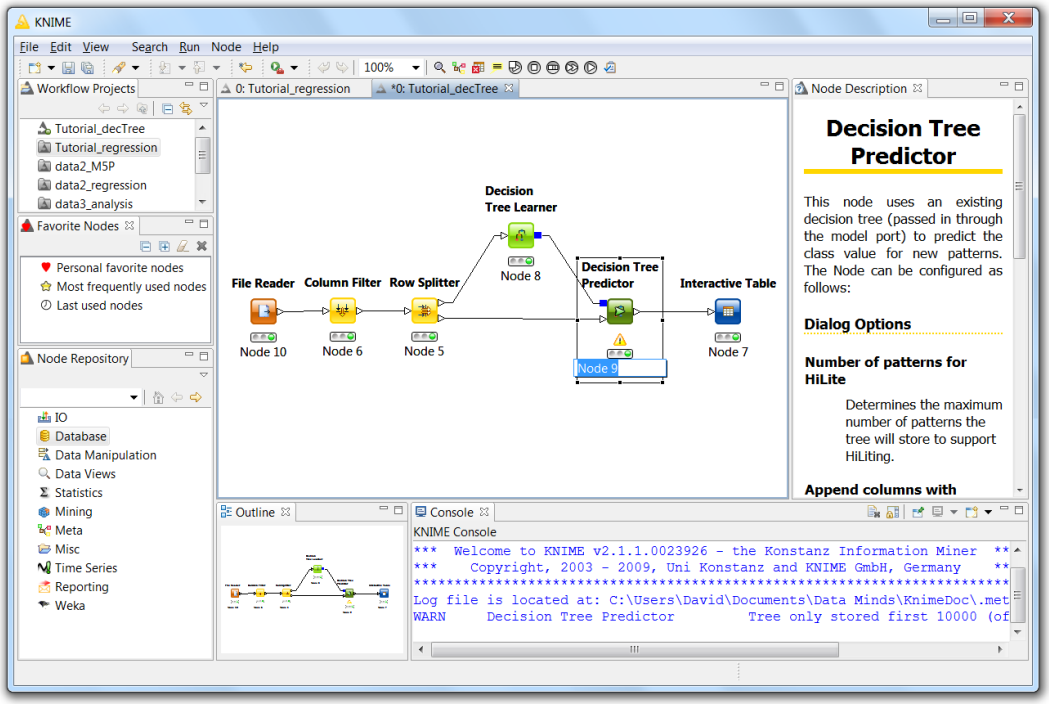

Image 5.6 (KNIME Graphical User Interface)

The “Node Description” tab facilitates the user in the selection of the correct node, showing a synthetic description for any node, or folder, selected in the “Node Repository” list or in the “Outline” tab. The information displayed in the

"Node Description" tab are shown all the time on the screen, resulting more user friendly than the popup message approach used in RapidMiner and Clementine.

In contrast, the nodes' setting are accessible through a separate window, which can be open double clicking on the selected node. This choice is probably forced by the big amount of options per each operator, which would hardly fit any tab. As we mention before, KNIME has a reduced number of node, but each of them is highly tuneable.

Finally KNIME is easy to install and the management of the main program and its extension is facilitated by the user friendly "Software Updates and Add-ons" tool.

- **Data and model visualisation.**

One of the more interesting features of KNIME is the "hiliting" function. This innovative approach is useful to facilitate the data visualisation: the data hilited (i.e. selected) in one table view are hilited in all the others visualisations of the same data set. This is particular useful when, for example, the user wants to locate the standalone values in a plot view and analyse their characteristics in a different table.

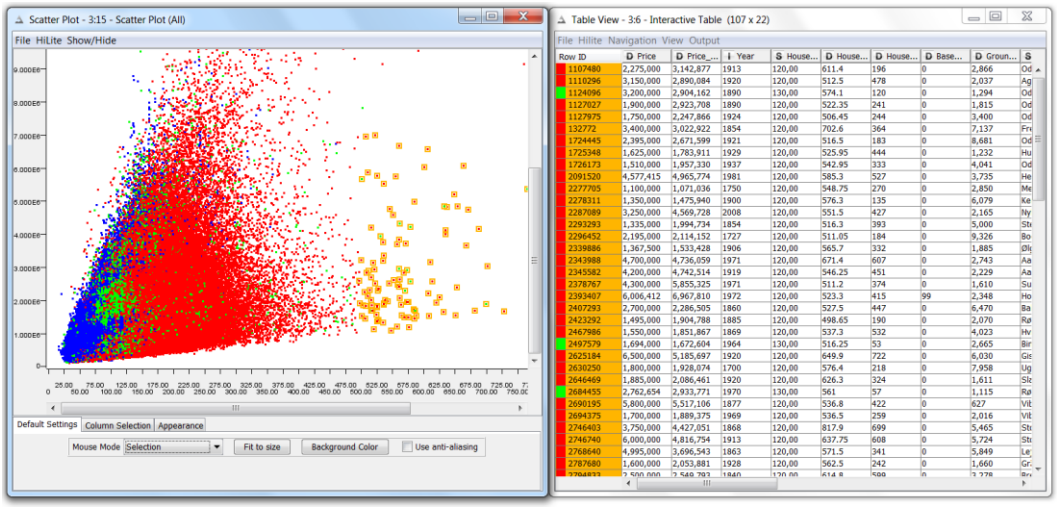

Image 5.7 (KNIME "Scatter Plot" and "Table View" tools: hilited values in orange)

The “Scatter Plot” tool is at the same level of the one used in Clementine, while the RapidMiner visualisation tool is more advanced. KNIME offers a group of nodes to manage colour, size and shape of the plotted data. In contrast, RapidMiner has integrated all these features in the same tool, facilitating the user in the data visualisation.

5.4.2 Community and Supporting Company Related Characteristics

- **License.**

Like RapidMiner and many other FOSS, KNIME uses a dual licensing policy: the users can decide between adopting the FOSS version which goes under a GNU General Public License v.3, and contacting the University of Konstanz to discuss a different kind of license.

The community edition can be downloaded and used by anyone. Its limitations are imposed by the GNU GPLv3, that we discuss in Section 2.1. The proprietary software, whose features do not differ from the community version’s, is free from any limitation.

In addition to the general GNU GPL, the license used by KNIME guarantees some extra permissions: the new users’ nodes, created extending the “classesNodeModel”, “NodeDialog”, and “NodeView”, are not considered derivative work of KNIME and, therefore, do not go under the GPL. This additional agreement is in accordance with Section 7 of the GNU General Public License v.3 [GPL 07].

Finally, the software Eclipse, tied together into KNIME, has its own Eclipse Public License (EPL).

- **Community.**

The KNIME project files are hosted by SourceForge.net. As already mentioned, the host website offers a wide range of supporting features, such as forum, bugs tracker, support and features requests trackers, etc.

In addition, KNIME website has a section dedicated to the community, where we can read information about the plugins developed outside the official developing team. Finally users and developers can access to a forum, which is, however, still quite small.

Although the KNIME community is not big and active, the numerous partner companies and sponsors supporting the project are indeed a good sign of interest for the FOSS system. Some of these supporting companies have also developed extra nodes, which can be directly requested.

- **Support.**

KNIME.org is the official community edition website, while KNIME.com is the commercial portal for support and maintenance, which offer a basic "Desktop" subscription and a superior "Report Designer Support" contract.

The support contracts include regular updates and patches, also in-between regular releases, and direct contact to the KNIME development team. A number of "free support requests" is set accordingly to the type of contract which is stipulated. Finally the customer can use a 30% discount for one day user training at the KNIME offices in Zurich, Switzerland.

The prices depends on the number of customers to support and requests needed. KNIME.com subdivides the groups into four categories: (*a*) single user with around 15 requests per year; (*b*) small groups for up to ten users, with 50 requests per year; (*c*) departments or small and medium enterprises, with maximum 50 users and 100 requests per year; (*d*) enterprises groups, with unlimited users and requests.

A KNIME Desktop Subscription is 2000€ for a single user, 7000€ for a small group and 15'000€ for a department. The Report Designer Subscription is more

expensive: 3000€ for a single user, 10’000€ for a small group and 20’000€ for a department. Any larger support contract needs a meeting to agree a personalised quote.

In addition, KNIME.com provides trainings for users and developers. The courses are divided into basic and advanced level, and the cost is around 750€ per day. The customers are invited at KNIME offices in Zurich, Switzerland, for the duration of the training, which is normally a couple of days.

- **Documentation.**

The online resources are the most valuable documentation for KNIME: apart from the introduction documents, a whole section of KNIME.org is dedicated to the system guides.

KNIME is introduced through a description of its features, some screenshots, and helpful “screencasts”: three videos recorded from a KNIME system about (*a*) its workbench, (*b*) a generic introduction of the program and (*c*) the admired hiliting function.

Four study cases shows how to build the flow for some of the main issues: the flow files are available online in order to let the user try the examples on her own computer. In addition, some real applications of KNIME are described and documented in external articles. Their links are available on KNIME.org.

In the “Documentation” section, the user can find an installation guide, a beginners manual and a advanced user guide. The text of these resources is clear, detailed and enriched with screenshots and pictures. In addition, the advanced user can found information about how to develop new nodes and extend the system.

Extra information can be found in the “FAQ” page, where eighteen useful frequent answered questions are explained in a clear and high level format. Any other question can be forwarded through an online form.

5.5 Overview Tables

The following tables recapitulate the software characteristics described in the previous sections:

A.1 SOFTWARE RELATED CHARACTERISTICS: Functionalities		
	RapidMiner	**KNIME**
Integration with different techniques	Many nodes with few options.	Few nodes with extensive options.
Integration with data sources and other products	DataSources: personal, files and DBMS. Models: personal, PMML (future development).	DataSources: personal, files and JDBC. Models: personal and PMML.
Integration with user's platform	Microsoft Windows (32/64bit) and Java Runtime Environment.	Microsoft Windows (32/64bit), Linux (32/64bit) and MacOS X (experimental).
Extensibility	Official extensions: Weka, Text Plugin, Value Series, Data Stream, Conditional Random Field. Platform language: Java.	Official extensions: Weka Integration, R Integration, Chemistry, BIRT Reporting, Math Formula, XLS Writer, External Tool, Python Scripting, Distance Matrix and LIBSVM. Platform language: Java. Extension module: KNIME Software Development Kit.

Table 5.7 (Software related characteristics: functionalities)

A.2 SOFTWARE RELATED CHARACTERISTICS: Performance		
	RapidMiner	**KNIME**
Data pre-processing capability	Robust set of ETL functions.	Few ETL functions (future development).
Algorithms' performance	Interesting results. Poor performance on Weka nodes.	Interesting results. Intermediate results stored during the flow execution.
Stability-reliability	Robust solution, appreciated by users and researchers. Frequent release of bug fixes.	Robust solution, appreciated especially in the literature. Frequent release of bug fixes.

Table 5.8 (Software related characteristics: performance)

A.3 SOFTWARE RELATED CHARACTERISTICS: User-friendly		
	RapidMiner	**KNIME**
Usability	Graphical User Interface: Flow chart. Positive learning curve: it takes a couple of days to fully appreciate the system.	Graphical User Interface: Flow chart. Positive learning curve: the user can get interesting results in the first day.
Data and model visualisation	Data visualisation tool: "Plot View", high level.	Data visualisation tool: "Scatter Plot", average level. Interesting and innovative "Hilting" function.

Table 5.9 (Software related characteristics: user-friendly)

B. COMMUNITY AND SUPPORTING COMPANY RELATED CHARACTERISTICS		
	RapidMiner	**KNIME**
License	GNU Affero GPL v3 or proprietary.	GNU GPL v3 with extra permissions, or proprietary.
Community	Still quite small. Managed by supporting company "Rapid-I". Hosted by "SourceForge.net".	Official community website: "KNIME.org". Still quite small. Managed by supporting company "KNIME.com". Hosted by "SourceForge.net".
Support	Offered by supporting company "Rapid-I".	Offered by supporting company "KNIME.com".
Documentation	Few resources for RapidMiner 5. GUI manual, user guide and video tutorials for previous version (RapidMiner 4).	Online introductive videos, downloadable guides and FAQ.

Table 5.10 (Community and supporting company related characteristics)

6 CONCLUSIONS

The word “FOSS” (Free Open Source Software) refers to software under a license which grants the right to access the source code and use, study, and change the software. We must not confuse FOSS with “non-commercial software”: antonyms of FOSS are “closed” and “proprietary” software.

Although several communities are involved in the development and recognition of FOSS, we focused our attention on the two main groups: the Free Software Foundation (FSF) and the Open Source Initiative (OSI). The two communities are essentially dealing with similar guidelines and the same group of software. A remarkable resemblance is the adoption of the GNU General Public License (GPL) which was written by the Free Software Community and soon became part of the Open Source Initiative Certified list. The main difference between the two communities is the approach to the matter: the Free Software Community keeps on fighting a philosophical war against the proprietary software, while the Open Source Initiative focuses more on marketing the FOSS world.

Many companies are adopting FOSS solutions in order to solve different issues: a simple overview of the current market shows that a large number of companies is interested in the deploying or, at least, in the testing of Free Open Source Software. FOSS tools are considered secure enough to rely on, especially if a supporting company manages the FOSS project.

One of the most appealing characteristics of FOSS solutions is the direct cost, which is often zero. We have shown, however, that the total cost can be much higher because of the indirect costs arisen from development, technical support, and maintenance efforts. In other words, although the zero direct cost lets any company try the solution, the large indirect cost keeps out all those companies which need large external support and maintenance, and cannot

afford it. It is not possible to define a general rule which states what is cheaper between Free Open Source Software and proprietary software. Indeed, there is not a unique answer to the issue.

A company interested in the adoption of a FOSS solution has to consider its own financial capabilities, the support needed and the software license. These factors are essential to understand if the company can adopt the FOSS tool. Further important characteristics to consider are the supporting company and the community behind the FOSS project.

The analysis of the FOSS solutions can also be considered valid for business intelligence FOSS. The business intelligence world has always been dominated by proprietary software, even thought, in the last year, the FOSS solutions have grown in number and quality, becoming real competitors in the market. Although the first business intelligence FOSS tools were small solutions meeting a specific need, some projects have lately managed to combine some FOSS tools in order to offer a complete FOSS business intelligence suite.

The main reason why FOSS solutions have problems in breaking through is the reluctance of many companies to change the provider of support when adopting business intelligence tools from a different vendor. The resistance is amplified when the adopted software has been purchased recently.

However, although many companies do not want to have extra expenses, they are open to trying new FOSS solutions for free. The business intelligence FOSS tools are growing in popularity: the raising interest is registered in the literature and in several market surveys.

We have suggested a subdivision of the FOSS business intelligence solutions in (*a*) DataBase Management Systems (DBMS), (*b*) Data Integration tools, (*c*) Analytical tools and (*d*) Business Intelligence Suites. For each subcategory we have found several interesting tools which have gathered a large number of admirers.

There is a striking resemblance between many companies supporting business intelligence FOSS projects in the adoption of the “double licensing”

policy: a proprietary license edition can be purchased, while a community edition, which goes under a FOSS license, is released on the Internet for free. Therefore, the supporting company can grow its own market promoting the free edition, and base its revenues on the proprietary licenses and, especially, on support and maintenance contracts.

Very good examples of companies which built their business on “double licensing” and support contracts are Rapid-I and KNIME.com, respectively supporting the FOSS projects RapidMiner and KNIME.

RapidMiner and KNIME are two of the most interesting data mining FOSS tools. The former is the most popular data mining FOSS, while the latter has gathered many positive reviews in the literature.

RapidMiner is a complete solution which offers a wide selection of data mining algorithms. In addition, the user can access extra functions which cover any stage of the data analysis: a good illustration of this is the big amount of pre-processing methods integrated in the system.

The graphical user interface adopts a “flow chart” which proves to be user-friendly and particularly helpful to create complicated processes. The “Plot View” is another feature which facilitates the data analysis: it is a high level data visualisation tool integrated into the data mining system. The user can easily visualise and investigate the data using the several methods offered by the Plot View” tool.

Overall, RapidMiner is a robust solution which can be adapted to many different needs. This data mining FOSS is, however, lacking some proper documentation which would be particularly helpful for beginner users. Finally the supporting company should optimise the integration of the software extensions, whose performance is not yet perfect.

In contrast to RapidMiner, KNIME is smaller and much more focused on the data mining functions: for example, only few data pre-processing methods are integrated into the system. Although KNIME is not as popular as RapidMiner, it has potential: the developing team has created a system with high usability and

strong innovative tools. The beginner user is able to learn how to use the system in short time thanks to the well structured graphical user interface and the build-in help functions.

The “hilting” tool is one of the more interesting innovations introduced by KNIME: the data “hilted” (selected) in a table or graph are “hilted” in all the other windows which visualise the same data source. This function improves the data analysis, especially regarding the examination of unexpected values.

The resource management is another original feature: for each node of the process flow, KNIME creates a folder where the intermediate results are saved. Once calculated, the results are therefore available for any future adoption. In other words, the processed nodes do not need to be executed again if they are not modified. This function is particularly useful when dealing with heavy processes.

Both RapidMiner and KNIME are therefore interesting solutions which can be considered valid alternatives to proprietary data mining systems.

7 REFERENCES

[Albano 09] Albano A., Basi di dati di supporto alle decisioni, Università di Pisa, 2009.

[Chen 07] Chen X., Ye Y., Williams G., and Xu X., A Survey of Open Source Data Mining Systems, *Lecture Notes in Computer Science*, Vol. 4819, 2007, pp. 3-14.

[CRISP 10] *CRISP-DM - Process Model*, <http://www.crisp-dm.org/Process/index.htm>, 18th February 2010.

[Berthold 09] Berthold M.R., Cebron N., Dill F., Gabriel T.R., Kötter T., Meinl T., Ohl P., Thiel K., and Wiswedel B., KNIME – The Konstanz Information Miner, *SIGKDD Explorations*, Vol. 11, Iss. 1, 2009, pp. 26-31.

[Bitterer 09] Bitterer A., Open-Source Business Intelligence Tools Production Deployments Will Grow Five-Fold through 2012, Gartner RAS Core Research Note G00171189, 2009.

[Bitzer 07a] Bitzer J., Schrettl W., and Schröder P.J.H., Intrinsic motivation in open source software development, *Journal of Comparative Economics*, Vol. 35, Iss. 1, 2007, pp. 160-169.

[Bitzer 07b] Bitzer J., and Schröder P.J.H., Open source software, competition and innovation, *Industry and Innovation*, Vol. 14, Iss. 5, 2007, pp. 461-476.

[Bondur 09] Bondur T., and Weathersby J., BIRT: Building Next Generation BI, *The Open Source Business Resource*, Iss. 9, 2009, pp. 32-36.

[Daffara 09] Daffara C., The SME guide to Open Source Software, fourth edition, 2009.

[Damiani 09] Damiani E., Frati F., and Monteverdi C., Open Source BI adoption, OW2 BI Initiative, 2009.

[DMG 10] *Data Mining Group - Member Profiles*, <http://www.dmg.org/about.html>, 18th February 2010.

[Doshi 06] Doshi A., Chambers A., Grantham P., Lalande P., Milham D., and Reberry D., Direction of Open Source for OSS implementation, in: *IEEE Symposium Record on Network Operations and Management Symposium*, Article N. 1687588, 2006, p. 580.

[Friedman 09] Friedman T., Beyer M.A., and Thoo E., Magic Quadrant for Data Integration Tools, Gartner RAS Core Research Note G00171986, 2009.

[Giogia 08] Giogia A., Cazzin G., and Damiani E., SpagoBI: A distinctive approach in open source business intelligence, in: *2008 2nd IEEE International Conference on Digital Ecosystems and Technologies*, IEEE-DEST 2008**,** Article N. 4635227, 2008, pp. 592-595.

[GNU 06] *Open Source Initiative OSI - FAQAdvocacy*, 2006, <http://web.archive.org/web/20060423094746/www.opensource.org/advocacy/faq.php>, 11th November 2009.

[GNU 08a] *Categories of Free and Non-Free Software - GNU Project - Free Software Foundation (FSF)*, 2008, <http://ftp.heanet.ie/disk1/www.gnu.org/philosophy/categories.html>, 17th February 2010.

[GNU 08b] *What is Copyleft? - GNU Project - Free Software Foundation (FSF)*, 2008, <http://ftp.heanet.ie/disk1/www.gnu.org/copyleft/copyleft.html>, 17th February 2010.

[GNU 09a] *The GNU Operating System*, <http://www.gnu.org>, 11th November 2009.

[GNU 09b] *Overview of the GNU System - GNU Project - Free Software Foundation (FSF)*, 2009, <http://www.gnu.org/gnu/gnu-history.html>, 11th November 2009.

[GNU 09c] *About the GNU Project - GNU Project - Free Software Foundation (FSF)*, 2009, <http://www.gnu.org/gnu/initial-announcement.html>, 11th November 2009.

[GNU 09d] *The Free Software Definition - GNU Project - Free Software Foundation*, 2009, <http://www.gnu.org/philosophy/free-sw.html>, 11th November 2009.

[GNU 09e] *Licenses - GNU Project - Free Software Foundation*, 2009, <http://www.gnu.org/licenses/licenses.html>, 11th November 2009.

[GNU 09f] *Why Open Source Misses the Point of Free Software - GNU Project - Free Software Foundation (FSF)*, 2009, <http://www.gnu.org/philosophy/open-source-misses-the-point.html>, 11th November 2009.

[GNU 09g] *Categories of Free and Non-Free Software - GNU Project - Free Software Foundation (FSF)*, 2009, <http://www.gnu.org/philosophy/categories.html>, 11th November 2009.

[GNU 09h] *Why "Free Software" is better than "Open Source" - GNU Project - Free Software Foundation (FSF)*, 2009, <http://www.gnu.org/philosophy/free-software-for-freedom.html>, 11th November 2009.

[Golfarelli 09] Golfarelli M., Open Source BI Platforms: A Functional and Architectural Comparison, *Lecture Notes in Computer Science*, Vol. 5691, 2009, pp. 287-297.

[GPL 07] *GNU General Public License - GNU Project - Free Software Foundation (FSF), Version 3*, 2007, <http://www.gnu.org/licenses/gpl.html>, 12th November 2009.

[Grossman 09] Grossman R.L., What is Analytic Infrastructure and Why Should You Care*?*, *SIGKDD Explorations*, Vol. 11, Iss. 1, 2009, pp. 5-9.

[Hall 09] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H., The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, Vol. 11, Iss. 1, 2009, pp. 10-18.

[Jones 04] Jones A.R., *Open Source Is Fertile Ground for Foul Play*, 2004, <http://www.devx.com/opensource/Article/20111>, 7th January 2010.

[Kemp 09] Kemp J., Shifting Buying and Evaluation Patterns, *The Open Source Business Resource*, Iss. 9, 2009, pp. 13-16.

[KDNuggets 07] *Poll: Data Mining / Analytic Software Tools*, 2007, <http://www.kdnuggets.com/polls/2007/data_mining_software_tools.htm>, 21st October 2009.

[KDNuggets 08] *Poll: Data Mining Software (2008)*, 2008, <http://www.kdnuggets.com/polls/2008/data-mining-software-tools-used.htm>, 21st October 2009.

[KDNuggets 09] *Data Mining Tools Used Poll*, 2009, <http://www.kdnuggets.com/polls/2009/data-mining-tools-used.htm>, 21st October 2009.

[KNIME 10] *Download KNIME | KNIME*, <http://www.knime.org/downloads/overview>, 3rd March 2010.

[Laplante 09] Laplante P.A., IT predictions for 2009, in: *IT Professional*, Article N. 4747657, Vol. 10, Iss. 6, 2008, pp. 56-59.

[LGPL 07] *GNU Lesser General Public License - GNU Project - Free Software Foundation (FSF), Version 3*, 2007, <http://www.gnu.org/licenses/lgpl.html>, 12th November 2009.

[Madsen 09] Madsen M., BeyeNETWORK Research Report, Open Source Solutions: Managing, Analyzing and Delivering Business

Information, BeyeNETWORK and Third Nature, 1790 30th Street, Suite 310, Boulder, CO 80301, 2009.

[Moglen 00] Moglen E., *Free Software Matters: Free Software or Open Source?*, 2000, <http://emoglen.law.columbia.edu/publications/lu-07.html>, 11th November 2009.

[MySQL 07] *MySQL :: YouTube, Flickr, and Wikipedia to Share their Secrets of Success at the 2007 MySQL Conference & Expo*, 2007, <http://dev.mysql.com/tech-resources/articles/mysqluc-2007.html>, 17th February 2010.

[MySQL 10] *MySQL :: Sun Presents Annual MySQL Awards*, <http://www.mysql.com/news-and-events/generate-article.php?id=2009_08>, 17th February 2010.

[OSI 06a] *The Open Source Definition (Annotated) | Open Source Initiative*, <http://opensource.org/docs/definition.php>, 11th November 2009.

[OSI 06b] *Open Source Initiative OSI - Why Free Software is too Ambiguous:Advocacy*, 2006, <http://web.archive.org/web/20060427221947/www.opensource.org/advocacy/free-notfree.php>, 11th November 2009.

[OSI 06c] *Open Source Initiative OSI - Open Source Case for Hackers:Advocacy*, 2006, <http://web.archive.org/web/20060428130741/www.opensource.org/advocacy/case_for_hackers.php>, 11th November 2009.

[OSI 07] *The Licence Proliferation Project - Open Source Initiative*, 2007, <http://www.opensource.org/proliferation>, 6th January 2010.

[OSI 10] *Report of License Proliferation Committee and draft FAQ - Open Source Initiative*, <http://www.opensource.org/proliferation-report>, 6th January 2010.

[OW2 09] *OW2 Consortium - OW2 Consortium (About.OW2Consortium) – XWiki*, 2009, <http://www.ow2.org/view/About/

OW2Consortium>, 17th February 2010.

[Pechter 09] Pechter R., What's PMML and What's New in PMML 4.0?, *SIGKDD Explorations*, Vol. 11, Iss. 1, 2009, pp. 19-25.

[Perens 97] Perens B., *Debian's "Social Contract" with the Free Software Community*, 1997, <http://lists.debian.org/ debian-announce/ 1997/msg00017.html>, 11th November 2009.

[Perens 09a] Perens B., *The Open Source Definition*, <http://perens.com/Articles/OSD.html>, 12th November 2009.

[Perens 09b] Perens B., *Slashdot Comments | How Many Open Source Licenses Do You Need*, 16th February 2009, <http://news.slashdot.org/ comments.pl?sid=1129863&cid=26875815>, 11th November 2009.

[Perens 09c] Perens B., *Innovation Goes Public! - Free Software vs. Open Source,* <http://perens.com/works/speeches/ InnovationGoesPublic/Speech/3.html>, 11th November 2009.

[Prasads 01] Prasads G., *Linux Today – Ganesh Prasad: Open Source-onomics: Examining some pseudo-economic arguments about Open Source*, 12th April 2001, <http://web.archive.org/web/ 20060718105227/www.linuxtoday.com/infrastructure/ 2001041200620OPBZCY-->, 12th January 2010.

[Rapid-I 10a] *Rapid - I - Download RapidMiner Extensions*, <http://rapid-i.com/content/view/55/85/lang,en>, 18th February 2010.

[Rapid-I 10b] *Rapid - I - Rapid-I is now Observing Member of the Data Mining Group (DMG): PMML Support Planned*, <http://rapid-i.com/content/view/173/1/lang,en>, 18th February 2010.

[Rapid-I 10c] *Rapid - I - RapidMiner Video Tutorials*,

<http://rapid-i.com/content/view/189/198>, 18th February 2010.

[Rapid-I 10d] *Rapid - I - Installation Guide*, <http://rapid-i.com/content/view/17/40>, 18th February 2010.

[Rapid-I 09a] *The RapidMiner GUI Manual 4.6*, 1st October 2009, <http://downloads.sourceforge.net/yale/rapidminer-4.6-guimanual.pdf>, 18th February 2010.

[Rapid-I 09b] *RapidMiner 4.6: User Guide, Operator Reference, Developer Tutorial*, 1st October 2009, <http://rapid-i.com/downloads/tutorial/rapidminer-4.6-tutorial.pdf>, 18th February 2010.

[Rapid-I 09c] *Overview (RapidMiner Class Documentation)*, 2009, <http://rapid-i.com/api/rapidminer-4.6/index.html>, 18th February 2010.

[Richardson 09] Richardson J., Business Intelligence Platform Adoption Intentions 2009, Gartner RAS Core Research Note G00170302, 2009.

[Rexer 07] Rexer K., Gearan P., and Allen H.N., Surveying the Field: Current Data Mining Applications, Analytic Tools, and Practical Challenges, Rexer Analytics, 30 Vine Street Winchester MA 01890, 2007.

[Rexer 08] Rexer K., 2nd Annual Data Miner Survey, Rexer Analytics, 30 Vine Street Winchester MA 01890, 2008.

[Rexer 09] Rexer K., Allen H.N., and Gearan P., 2009 Data Mining Priorities, Practices & Business Trends: Results of 3rd Annual Data Miner Survey, Rexer Analytics, 30 Vine Street Winchester MA 01890, 2009.

[Damiani 09] Damiani E., Frati F., and Monteverdi C., Open Source BI adoption, OW2 BI Initiative, 2009.

[Ruffatti 08] Ruffatti G., OW2 BI Initiative Charter, OW2 BI Initiative, 2008.

[Samoladas 08] Samoladas I., Gousios G., Spinellis D., and Stamelos I., The SQO-OSS quality model: Measurement based open source software evaluation, in: *IFIP International Federation for Information Processing*, Vol. 275, 2008, pp. 237-248.

[Selim 09] Selim S.P., Editorial, *The Open Source Business Resource*, Iss. 9, 2009, pp. 4-5.

[Stallman 08] Stallman R., About the GNU Project - GNU Project - Free Software Foundation (FSF), 2008, <http://www.gnu.org/gnu/thegnuproject.html>, 11th November 2009.

[Thomsen 05] Thomsen C., and Pedersen T.B., A survey of open source tools for Business Intelligence, in: *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery*, Tjoa A. M., and Trujillo J. (eds.), Vol. 3589, Berlin Heidelberg: Springer, 2005, pp. 74-84.

[Thomsen 09] Thomsen C., and Pedersen T.B., A survey of open source tools for Business Intelligence, *International Journal of Data Warehousing and Mining*, Vol. 5, Iss. 3, 2009, pp. 56-75.

[Wang 01] Wang H., and Wang C., Open source software adoption: A status report, *IEEE Software*, Vol. 18, Iss. 2, 2001, pp. 90-95.

[Wheeler 09] Wheeler D.A., *Open Source Software / Free Software (OSS/FS) References*, <http://www.dwheeler.com/oss_fs_refs.html>, 11th November 2009.

[Witten 05] Witten I.H., and Frank E., *Data mining: Practical machine learning tools and techniques with Java implementations*, second edition, Morgan Kaufmann, San Francisco, 2005.