

# 1 Theme

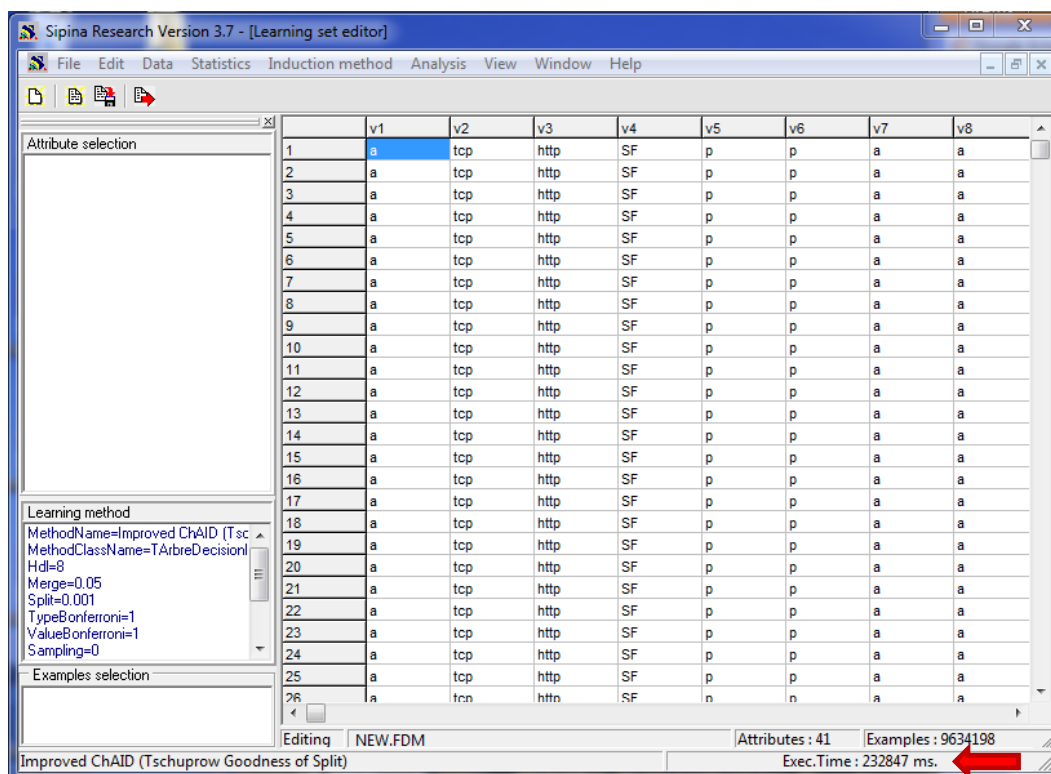
## Studying the behavior of 64-bit version of some tools when they handle a very large dataset.

Because I have recently updated my operating system (OS), I am wondering how the 64-bit versions of **Knime 2.4.2** and **RapidMiner 5.1.011** could handle a very large dataset, which cannot be loaded into main memory on a 32-bit OS. This article completes a [previous study](#) where we deal with a moderate sized dataset with 500,000 instances and 22 variables. Here, we handle a dataset with **9,634,198 instances** and **41 variables**. We have already used this dataset in [another tutorial](#). We showed that we cannot perform a decision tree induction on this kind of database without a swapping system, which is implemented into the SIPINA, on a 32-bit OS. We note that Tanagra can handle the dataset, but this is because it encodes the values of the categorical attributes with a byte. The memory occupation remains moderate.

In this tutorial, we analyze the behavior of the 64-bit Knime and RapidMiner on this database. We use a 64-bit OS, but we have "only" 4 GB of available memory on our personal computer.

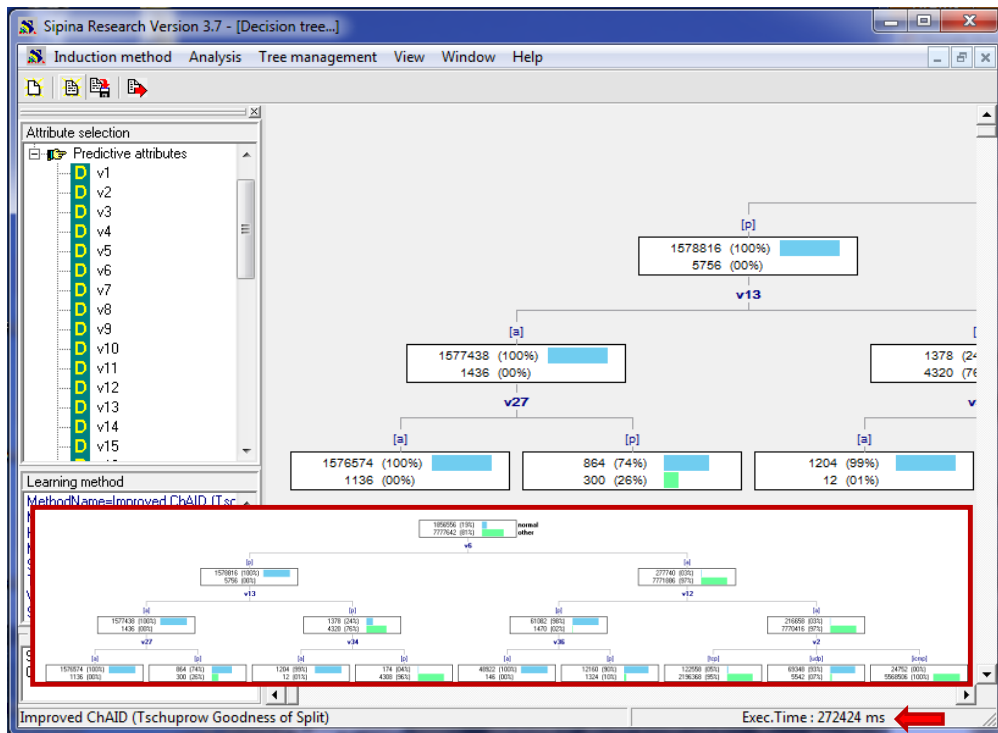
## 2 SIPINA

To ensure that SIPINA can handle this dataset, we must modify the settings into the **SIPINA.INI** initialization file<sup>1</sup>.

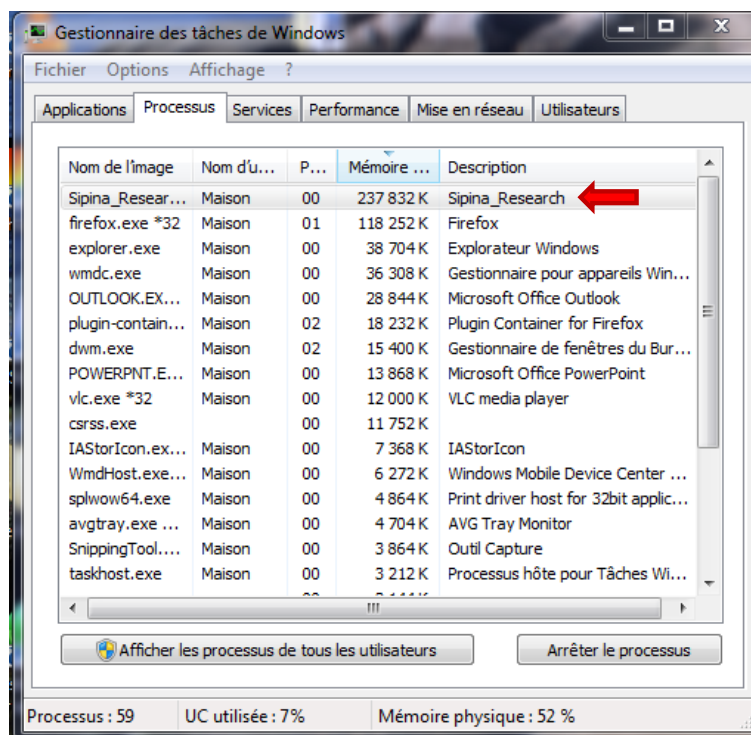


The data file is imported into 232 seconds. The processing time for the decision tree induction is 272 seconds (about 5 minutes).

<sup>1</sup> <http://data-mining-tutorials.blogspot.com/2010/01/dealing-with-very-large-dataset-in.html>

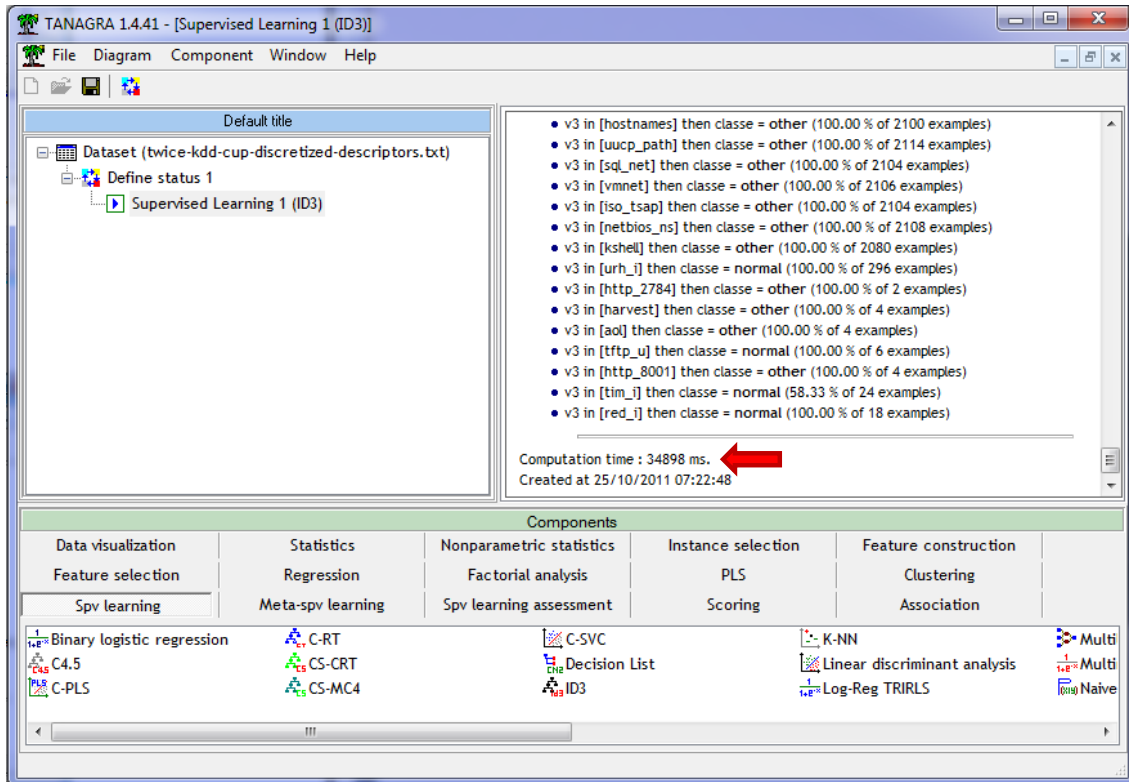


The whole process is already described elsewhere. We note only that the processing time is not influenced by the OS. On the same PC, we have the same performance whatever the OS version (Windows Vista 32-bit vs. Windows 7 64-bit). The memory occupation (about 232 MB) remains moderate in relation to the database size. This is the main interest of a swapping system. We can handle a very large database. The disk size is not a limitation. We note also that the solution is all the more efficient that the disk is performing. If we use a [solid state disk](#) for the temporary files, the computation time will be very likely reduced.

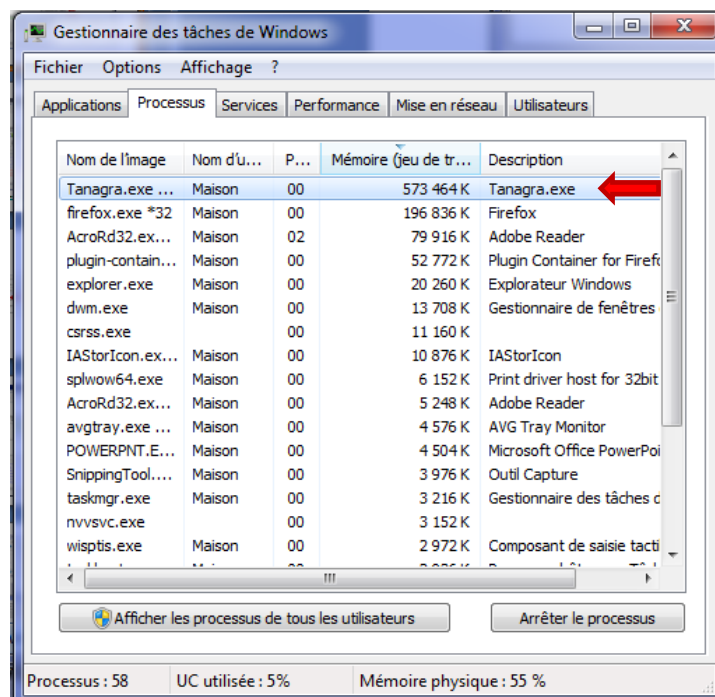


### 3 Tanagra

On the same computer, the performance of Tanagra - which runs under 32-bit mode - under Windows 64-bit version is not modified. The importation time is 87 seconds, and the tree is built in 34 seconds.

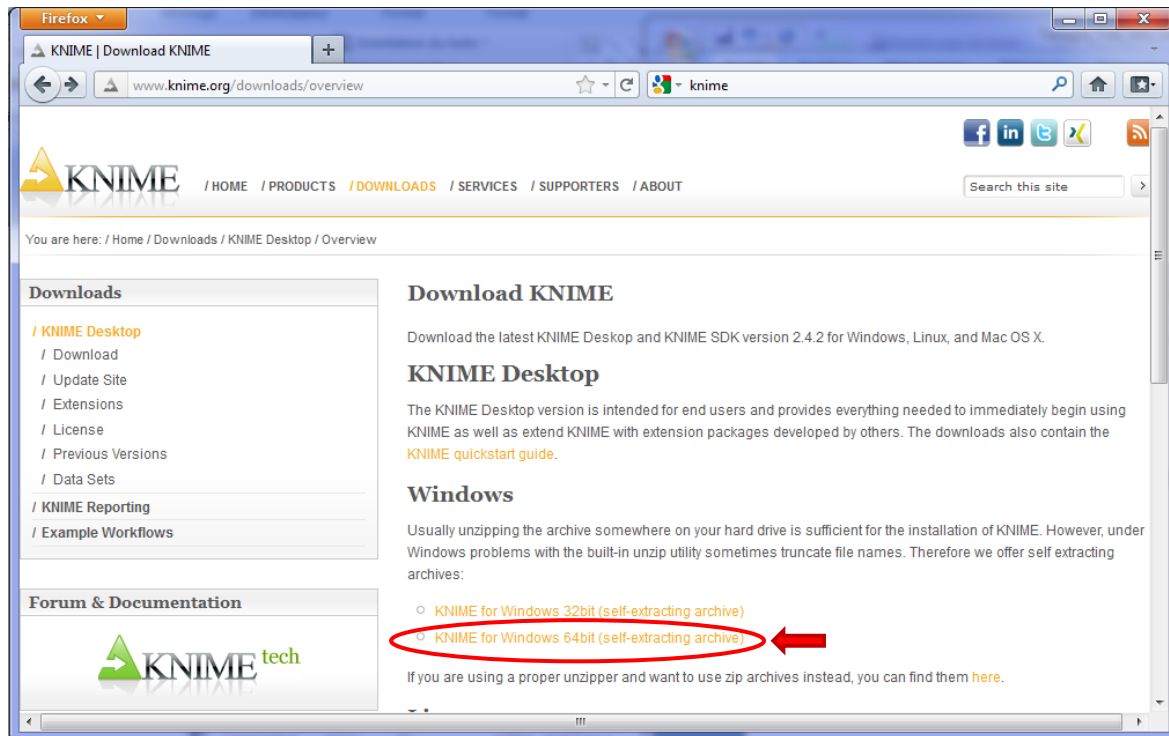


Because, the values of the categorical attributes are encoded with a byte (the number of values for one attribute is limited to 255), the memory occupation ( $\approx 560$  MB) makes the calculations tractable.



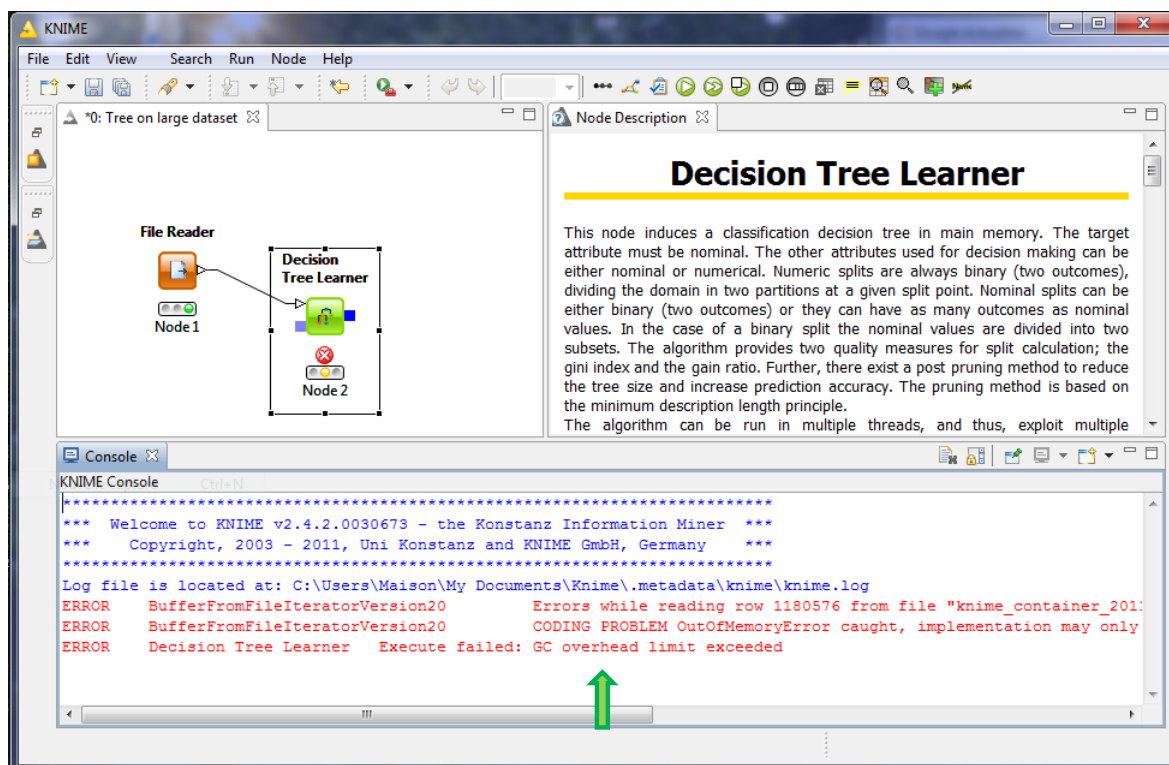
## 4 Knime

On the Knime website, we can choose between the 32-bit and the 64-bit version of the software.

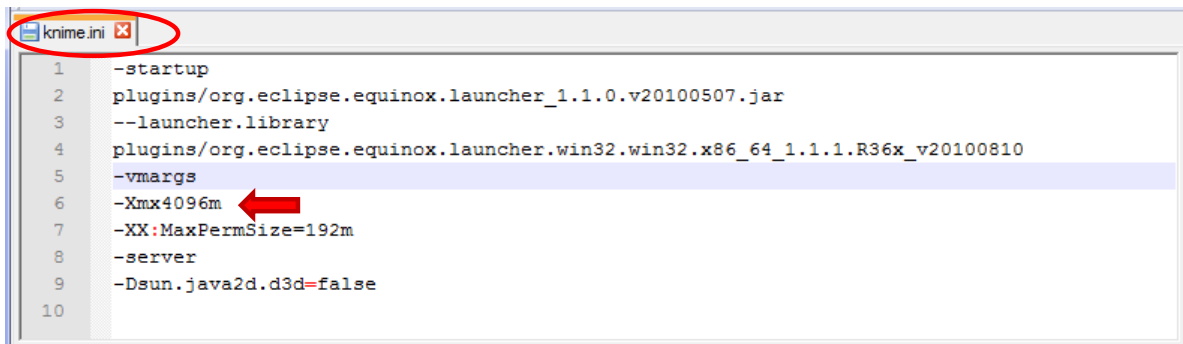


We select this last one for this tutorial. We do not describe the whole definition of the workflow here because we have already done this [previously](#) (section 6.2).

Let us see what happens when we launch the processing. Knime displays an enigmatic error message.



After some search (e.g. <http://www.petefreitag.com/item/746.cfm>), we understand that we can overcome this memory limitation by modifying the heap allocation memory. If we suppose that a double precision coding is used for each value, the memory occupation is (roughly, we do not take into account the eventual meta-information and the memory used for the calculations) about 2.94 GB  $[(9.634.198 \times 41 \times 8) / (1024 \times 1024 \times 1024) \approx 2.94 \text{ Go}]$ . So, we set the following setting into the **Knime.ini** initialization file.

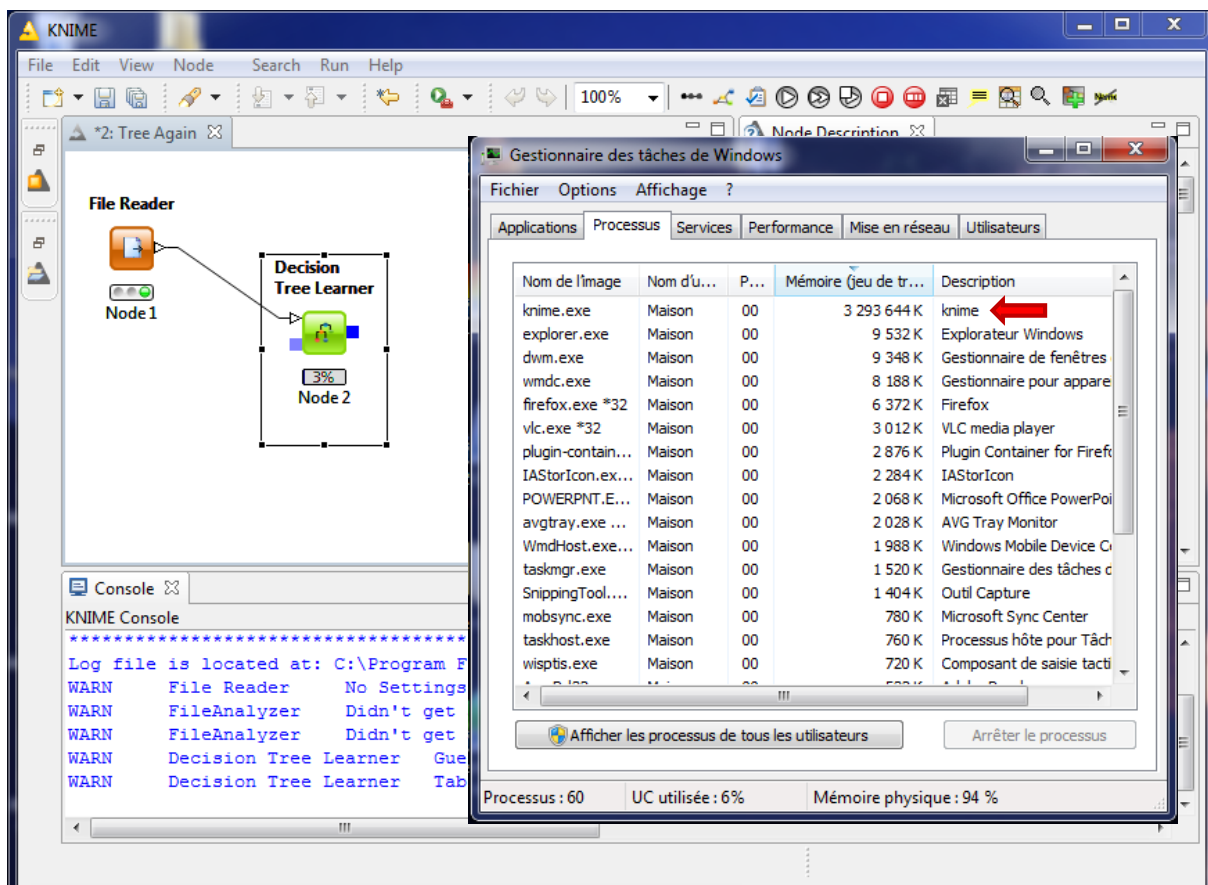


```

1  -startup
2  plugins/org.eclipse.equinox.launcher_1.1.0.v20100507.jar
3  --launcher.library
4  plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.1.R36x_v20100810
5  -vmargs
6  -Xmx4096m
7  -XX:MaxPermSize=192m
8  -server
9  -Dsun.java2d.d3d=false
10

```

Now, Knime can import the dataset. We can launch the decision tree induction process.



The screenshot shows the KNIME software interface with a workflow diagram. The workflow consists of two nodes: 'File Reader' (Node 1) and 'Decision Tree Learner' (Node 2). The 'Decision Tree Learner' node is currently at 3% completion. A Windows Task Manager window is open, displaying the 'Processus' (Processes) tab. The 'knime.exe' process is highlighted with a red arrow, indicating its memory usage. The console window at the bottom shows several 'WARN' messages, including 'File Analyzer Didn't get' and 'Decision Tree Learner'.

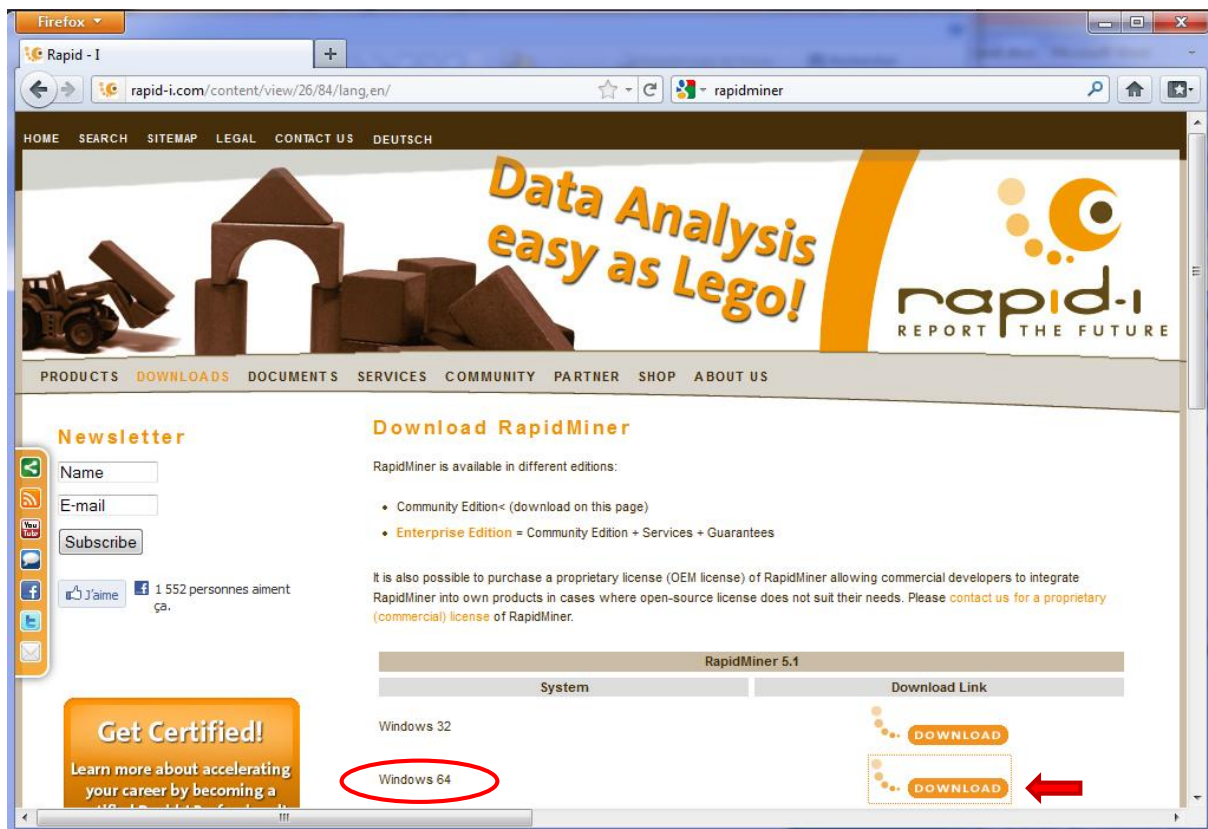
Nom de l'image	Nom d'u...	P...	Mémoire (jeu de tr...	Description
knime.exe	Maison	00	3 293 644 K	knime
explorer.exe	Maison	00	9 532 K	Explorateur Windows
dwm.exe	Maison	00	9 348 K	Gestionnaire de fenêtres
wmdc.exe	Maison	00	8 188 K	Gestionnaire pour appare
firefox.exe *32	Maison	00	6 372 K	Firefox
vlc.exe *32	Maison	00	3 012 K	VLC media player
plugin-contain...	Maison	00	2 876 K	Plugin Container for Firef
IAStorIcon.ex...	Maison	00	2 284 K	IAStorIcon
POWERPNT.E...	Maison	00	2 068 K	Microsoft Office PowerPo
avgtray.exe ...	Maison	00	2 028 K	AVG Tray Monitor
WmdHost.exe...	Maison	00	1 988 K	Windows Mobile Device C
taskmgr.exe	Maison	00	1 520 K	Gestionnaire des tâches c
SnippingTool...	Maison	00	1 404 K	Outil Capture
mobsync.exe	Maison	00	780 K	Microsoft Sync Center
taskhost.exe	Maison	00	760 K	Processus hôte pour Tâch
wisptis.exe	Maison	00	720 K	Composant de saisie tacti

The Knime memory occupation is about 3.14 GB during the calculations. Because the amount of available memory is only 4 GB, Windows begins to use intensively its swap file. The computation time is dramatically increased. So, after about 30 minutes (the tree induction is jammed to 3%, see the screenshot above), I stopped the calculations.

I think if we have more memory, we can increase the Xmx parameter of the KNIME.INI configuration file. And so, we can lead the analysis on the whole dataset. **A user reports to me that he can achieve the process with a PC with 8 GB RAM. The memory occupation is about 7 GB at the end of the decision tree construction.**

From that perspective, because the theoretical limitation is alleviated, the main bottleneck becomes the machine characteristics (the computer mother card capacity and the OS specifications – e.g. Windows 7 - [http://en.wikipedia.org/wiki/Windows\\_7#Physical\\_memory\\_limits](http://en.wikipedia.org/wiki/Windows_7#Physical_memory_limits)).

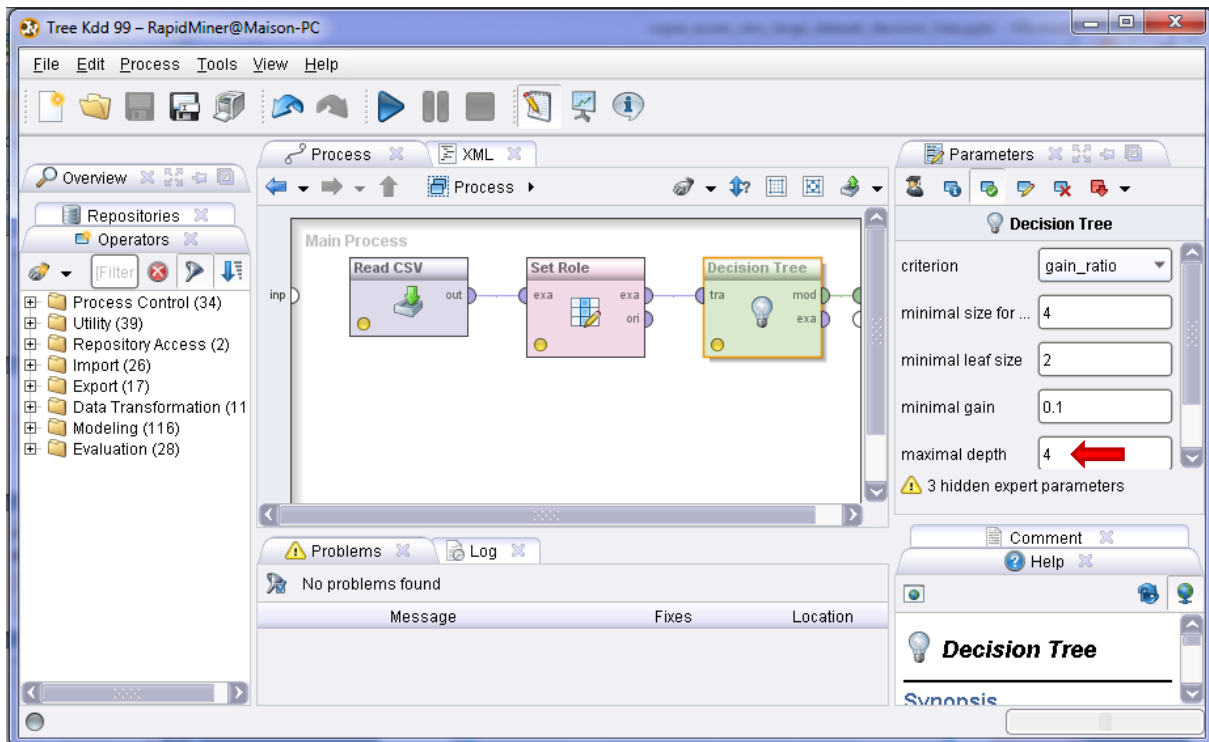
## 5 RapidMiner



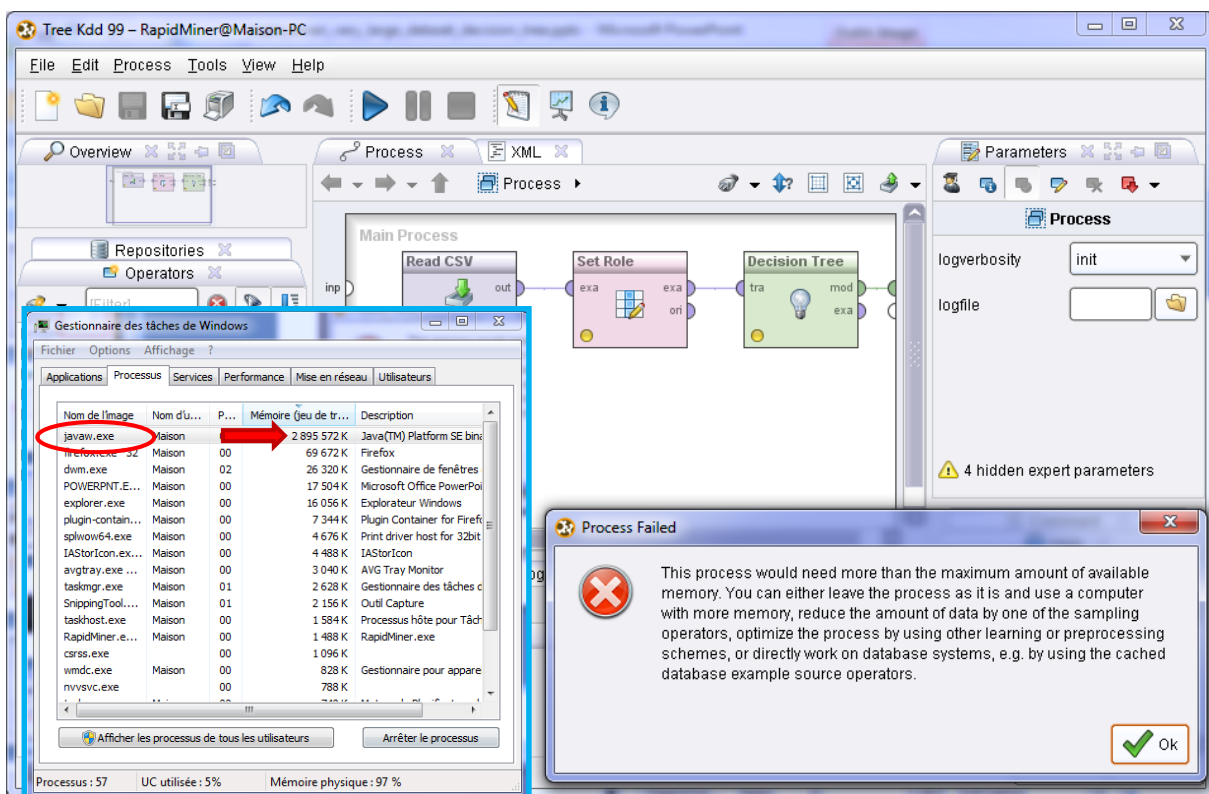
We download the 64 bit version from the RapidMiner website. We use the "Community Edition" which is freely downloadable. It is fully functional. The "Enterprise Edition" incorporates some services and guarantees (see <http://rapid-i.com/content/view/181/190/>).

### 5.1 Processing with the standard settings

We do not detail the construction of the process here. For the reader not familiarized with RapidMiner, they can see the tutorial available on our website: <http://data-mining-tutorials.blogspot.com/2011/09/new-gui-for-rapidminer-50.html>



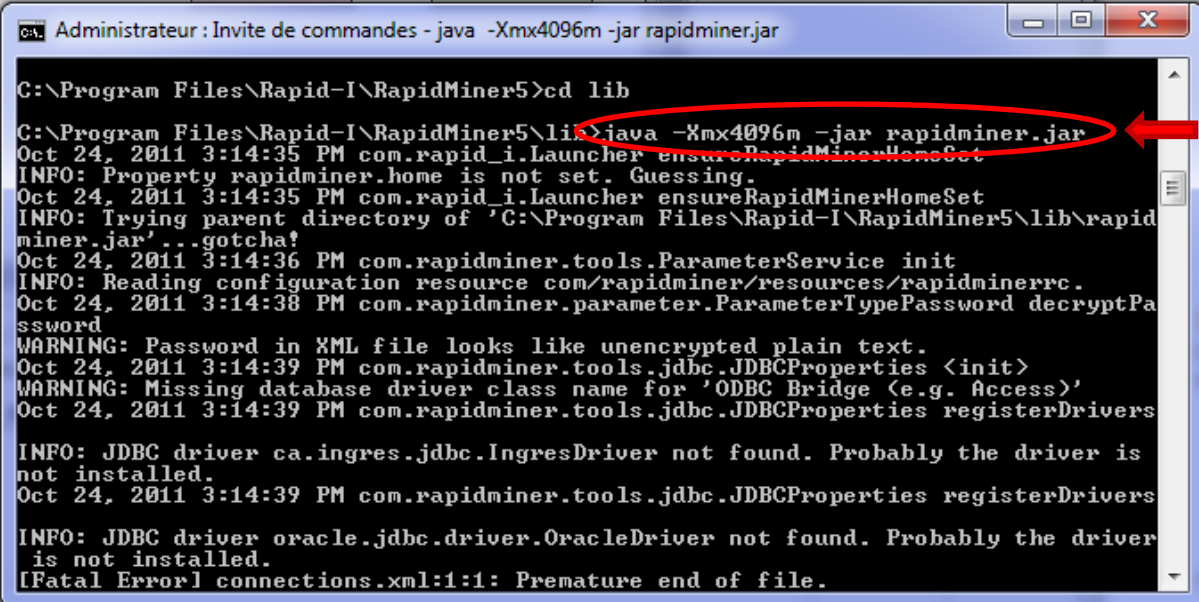
We launch RapidMiner using the shortcut on the desktop. We set the maximum depth of the tree to 4. We start the process, including the importation of the data file. After a few seconds, an error message appears.



The available memory is not enough to achieve the process. We observe that the memory occupation of RapidMiner (through JAVAW.EXE) is 2.67 GB.

## 5.2 Launching from the Windows command line

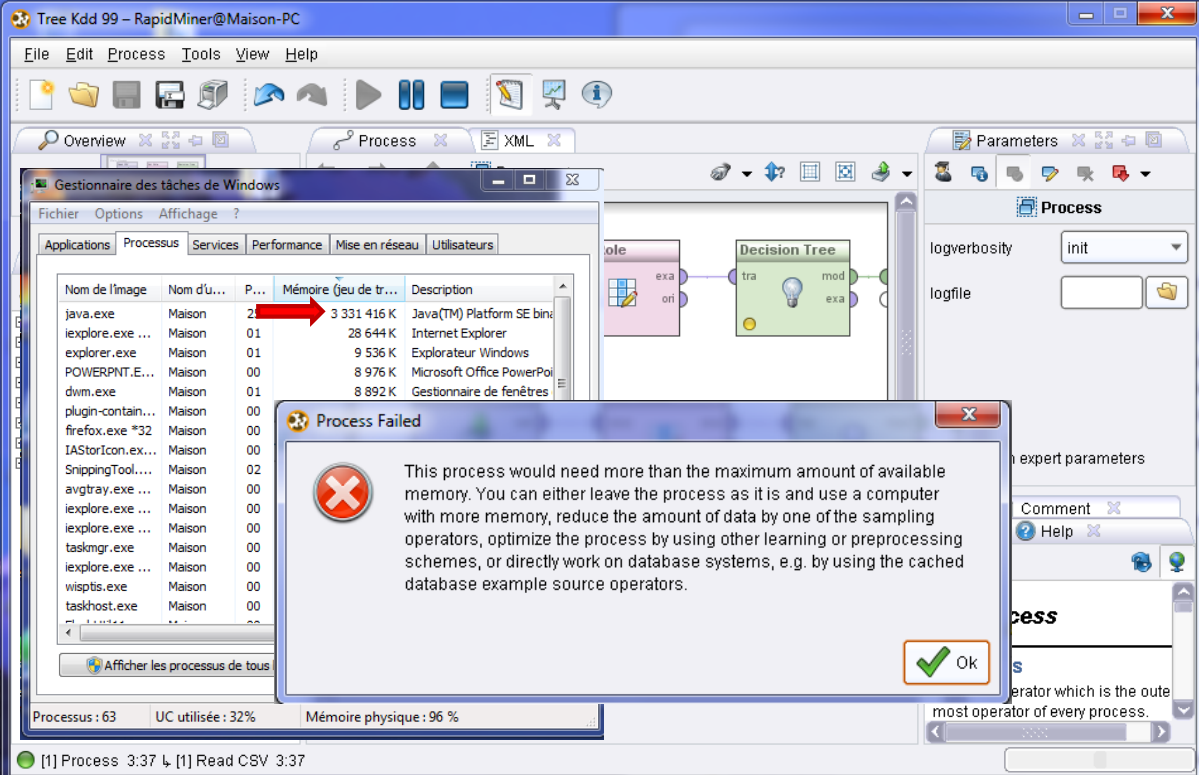
We can modify the settings files into the SCRIPT directory. But, after unsuccessful attempts, I preferred to use the command line in order to control finely the launching settings.



```

C:\Program Files\Rapid-I\RapidMiner5>cd lib
C:\Program Files\Rapid-I\RapidMiner5\lib>java -Xmx4096m -jar rapidminer.jar
Oct 24, 2011 3:14:35 PM com.rapid_i.Launcher ensureRapidMinerHomeSet
INFO: Property rapidminer.home is not set. Guessing.
Oct 24, 2011 3:14:35 PM com.rapid_i.Launcher ensureRapidMinerHomeSet
INFO: Trying parent directory of 'C:\Program Files\Rapid-I\RapidMiner5\lib\rapidminer.jar'...gotcha!
Oct 24, 2011 3:14:36 PM com.rapidminer.tools.ParameterService init
INFO: Reading configuration resource com/rapidminer/resources/rapidminer.rc.
Oct 24, 2011 3:14:38 PM com.rapidminer.parameter.ParameterTypePassword decryptPassword
WARNING: Password in XML file looks like unencrypted plain text.
Oct 24, 2011 3:14:39 PM com.rapidminer.tools.jdbc.JDBCProperties <init>
WARNING: Missing database driver class name for 'ODBC Bridge (e.g. Access)'
Oct 24, 2011 3:14:39 PM com.rapidminer.tools.jdbc.JDBCProperties registerDrivers
INFO: JDBC driver ca.ingres.jdbc.IngresDriver not found. Probably the driver is not installed.
Oct 24, 2011 3:14:39 PM com.rapidminer.tools.jdbc.JDBCProperties registerDrivers
INFO: JDBC driver oracle.jdbc.driver.OracleDriver not found. Probably the driver is not installed.
[Fatal Error] connections.xml:1:1: Premature end of file.
  
```

After few minutes, the process has failed again.



The screenshot shows the RapidMiner interface with a 'Process Failed' dialog box. The dialog box contains the following text:

**Process Failed**

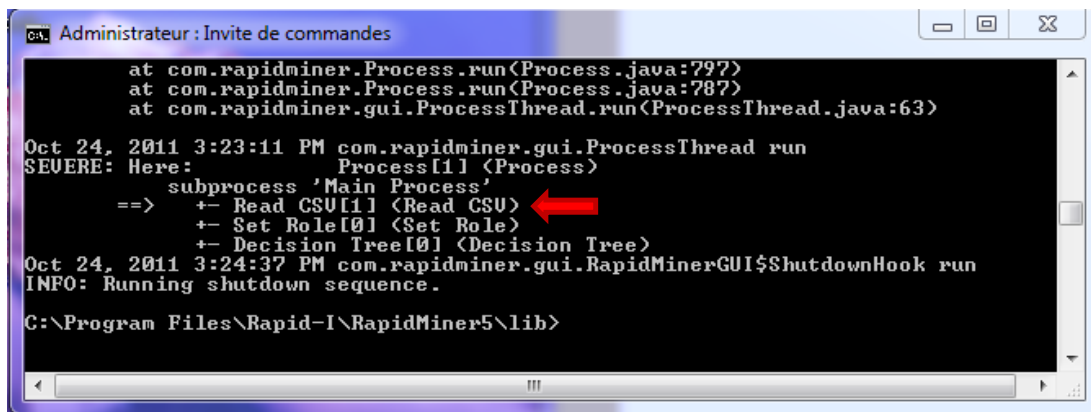
This process would need more than the maximum amount of available memory. You can either leave the process as it is and use a computer with more memory, reduce the amount of data by one of the sampling operators, optimize the process by using other learning or preprocessing schemes, or directly work on database systems, e.g. by using the cached database example source operators.

Ok

In the background, the Windows Task Manager window is open, showing the 'Processus' tab. The 'java.exe' process is highlighted, with a memory usage of 3,331,416 K (3.37 GB).

We note that the memory occupation is 3.17 GB when the process was interrupted. The command window shows that the dataset was not able to be completely imported.





```

at com.rapidminer.Process.run(Process.java:797)
at com.rapidminer.Process.run(Process.java:787)
at com.rapidminer.gui.ProcessThread.run(ProcessThread.java:63)

Oct 24, 2011 3:23:11 PM com.rapidminer.gui.ProcessThread run
SEVERE: Here:
    subprocess 'Main Process'
    ==> +- Read CSU[1] <Read CSU>
        +- Set Role[0] <Set Role>
        +- Decision Tree[0] <Decision Tree>
Oct 24, 2011 3:24:37 PM com.rapidminer.gui.RapidMinerGUI$ShutdownHook run
INFO: Running shutdown sequence.

C:\Program Files\Rapid-I\RapidMiner5\lib>

```

RapidMiner rests on the same technology than Knime (JAVA). No doubt that if we had more memory, he could lead to their term the calculations. Let us note also that RapidMiner proposes us the other strategies to go beyond to the error: the sampling, the using of other data mining method, or working directly inside of a DBMS system. This last suggestion seems really relevant. I think I will study it in a future tutorial. I know that tools such as R propose also this possibility. But, I never tried seriously this kind of solution until now.

### 5.3 Internal coding of the values

The **DATAMANAGEMENT** option of the **READ CSV** operator is very useful for the handling of very large datasets. It enables us to choose the internal coding used for the data representation. **DOUBLE\_ARRAY** is the default setting. Each value is described by a double precision real. The memory occupation is 8 bytes. For our dataset where we have categorical attributes only, each of them having few values, this choice is oversized. We can dramatically reduce the memory occupation by using a suited coding strategy. For instance, if we use the **BYTE\_ARRAY** encoding, that is fully enough for our dataset, we can divide by 8 the memory occupation!

## 6 Conclusion

The shift to the 64-bit version of the data mining tools allows to benefit of the capacities of more powerful computers. We can load a larger dataset and perform the calculations into main memory. About our experiments, if we have a computer with 8 GB RAM, we can complete the analysis.

But, on other hand, this shift from a 32-bit system to a 64-bit one is not really useful if the characteristics of the computer are not improved. Especially, on a low-power machine, the solution such as the one implemented into Sipina, where we write the table into temporary files on the disk, remains a valuable solution.