Data Mining Models and Enterprise Risk Management

9

Datamining applications to business cover a variety of fields.[1] Risk-related applications are especially strong in insurance, specifically fraud detection.[2] Fraud detection modeling includes text mining.[3] There are many financial risk management applications, with heavy interest in developing tools to support investment. Automated trading has been widely applied in practice for decades. More recent efforts have gone into sentiment analysis, mining text of investment comments to detect patterns, especially related to investment risk.[4]

There are a number of data mining tools. This includes a variety of software, some commercial (powerful and expensive) as well as open-source. Open-source classification software tools have been published.[5] There are other modeling forms as well, to include application of clustering analysis in fraud detection.[6] We will use an example dataset involving data mining of bankruptcy, a severe form of financial risk.

Bankruptcy Data Demonstration

This data concerns 100 US firms that underwent bankruptcy.[7] All of the sample data are from the USA companies. About 400 bankrupt company names were obtained from the Compustat database, focusing on the companies that went bankrupt over the period January 2006 through December 2009. This yielded 99 firms. Using the company Ticker code list, financial data ratios over the period January 2005–December 2009 were obtained and used in prediction models of company bankruptcy. The factor collected contain total asset, book value per share, inventories, liabilities, receivables, cost of goods sold, total dividends, earnings before interest and taxes, gross profit (loss), net income (loss), operating income after depreciation, total revenue, sales, dividends per share, and total market value. To obtain non-bankrupt cases for comparison, the same financial ratios for 200 non-failed companies were gathered for the same time period. The LexisNexis database provided SEC fillings after June 2010, to identify firm survival with CIK code.

D.L. Olson, D.D. Wu, *Enterprise Risk Management Models*, Springer Texts in Business and Economics, DOI 10.1007/978-3-662-53785-5_9

Table 9.1 Attributes in bankruptcy data

No	Short name	Long name
1	fyear	Data year—Fiscal
2	cik	CIK number
3	at	Assets—Total
4	bkvlps	Book value per share
5	invt	Inventories—Total
6	Lt	Liabilities—Total
7	rectr	Receivables—Trade
8	cogs	Cost of goods sold
9	dvt	Dividends—Total
10	ebit	Earnings before interest and taxes
11	gp	Gross profit (Loss)
12	ni	Net income (Loss)
13	oiadp	Operating income after depreciation
14	revt	Revenue—Total
15	sale	Sales-turnover (Net)
16	dvpsx_f	Dividends per share—Ex-date—Fiscal
17	mkvalt	Market value—Total—Fiscal
18	prch_f	Price high—Annual—Fiscal
19	bankruptcy	Bankruptcy (output variable)

The CIK code list was input to the Compustat database to obtain financial data and ratios for the period January 2005–December 2009 to match that of failed companies.

The data set consists of 1321 records with full data over 19 attributes as shown in Table 9.1. The outcome attribute is bankruptcy, which has a value of 1 if the firm went bankrupt by 2011 (697 cases), and a value of 0 if it did not (624 cases).

This is real data concerning firm bankruptcy, which could be updated by going to web sources.

Software

R is a widely used open source software. Rattle is a GUI system for R (also open source) that makes it easy to implement R for data mining.

To install R, visit https://cran.rstudio.com/

Open a folder for R.

Select Download R for windows.

To install Rattle:

Open the R Desktop icon (32 bit or 64 bit) and enter the following command at the R prompt. R will ask for a CRAN mirror. Choose a nearby location.

- install.packages("rattle")

Enter the following two commands at the R prompt. This loads the Rattle package into the library and then starts up Rattle.

- library(rattle)
- rattle()

If the RGtk2 package has yet to be installed, there will be an error popup indicating that libatk-1.0-0.dll is missing from your computer. Click on the OK and then you will be asked if you would like to install GTK+. Click OK to do so. This then downloads and installs the appropriate GTK+ libraries for your computer. After this has finished, do exit from R and restart it so that it can find the newly installed libraries.

When running Rattle a number of other packages will be downloaded and installed as needed, with Rattle asking for the user's permission before doing so. They only need to be downloaded once.The installation has been tested to work on Microsoft Windows, 32bit and 64bit, XP, Vista and 7 with R 3.1.1, Rattle 3.1.0 and RGtk2 2.20.31. If you are missing something, you will get a message from R asking you to install a package. I read nominal data (string), and was prompted that I needed "stringr". On the R console (see Fig. 9.1), click on the "Packages" word on the top line:Give the command "Install packages" which will direct you to HTTPS CRAN mirror. Select one of the sites (like "USA(TX) [https]") and find "stringr" and click on it. Then upload that package. You may have to restart R.

Data mining practice usually utilizes a training set to build a model, which can be applied to a test set. In this case, 1178 observations (those through 2008) were used for the training set and 143 observations (2009 and 2010) held out for testing. To run a model, on the **Filename** line, click on the icon and browse for the file "bankruptcyTrain.csv". Click on the **Execute** icon on the upper left of the Rattle window. This yields Fig. 9.2:Bankrupt is a categoric variable, and R assumes that is the Target (as we want). We could delete other variables if we choose to, and redo the Execute step for the Data tab. We can **Explore**—the default is **Summary**. **Execute** yields macrodata, identify data types as well as descriptive statistics (minima, maxima, medians, means, and quartiles). R by default holds out 30 % of the training data as an intermediate test set, and thus builds models on the remaining 70 % (here 824 observations). The summary identifies the outcome of the training set (369 not bankrupt, 455 bankrupt).

We can further explore the data through correlation analysis. Figure 9.3 shows the R screen with the correlation radio button selected.**Execute** on this screen yields output over the numerical variables as shown in Fig. 9.4:

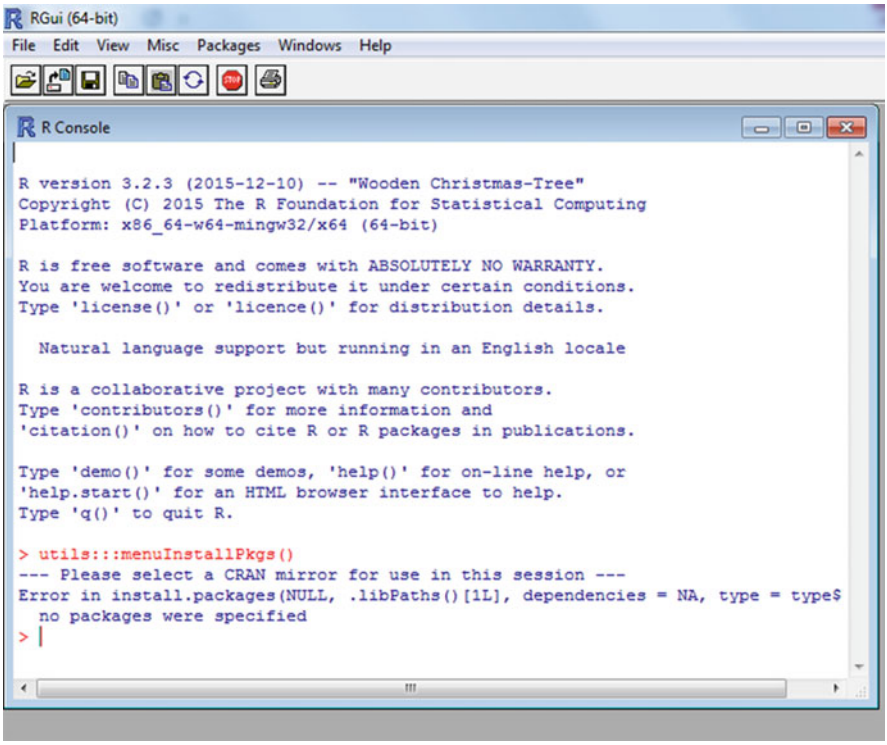

Fig. 9.1 R console

Figure 9.4 indicates high degrees of correlation across potential independent variables, and further analysis might select some for elimination. Numerical correlation values are also provided by R. The dependent variable was alphabetical, so R didn't include it, but outside analysis indicates low correlation between bankruptcy and all independent variables—the highest in magnitude being 0.180 with cost of goods sold (cogs) and with total revenue (revt).

Decision Tree Model

We can click on the **Model** tab and run models. Data mining for classification models have three basic tools—decision trees, logistic regression, and neural network models. To run a decision tree, select the radio button as indicated in Fig. 9.5:Note that the defaults are to require a minimum of 20 cases per rule, with a maximum number of 30 branches. These can be changed by entering desired values in the appropriate window. **Execute** yields Fig. 9.6:Rattle also provides a graphical display of this decision tree, as shown in Fig. 9.7:This model begins with the variable revt, stating that if revt is less than 78, the conclusion is that bankruptcy

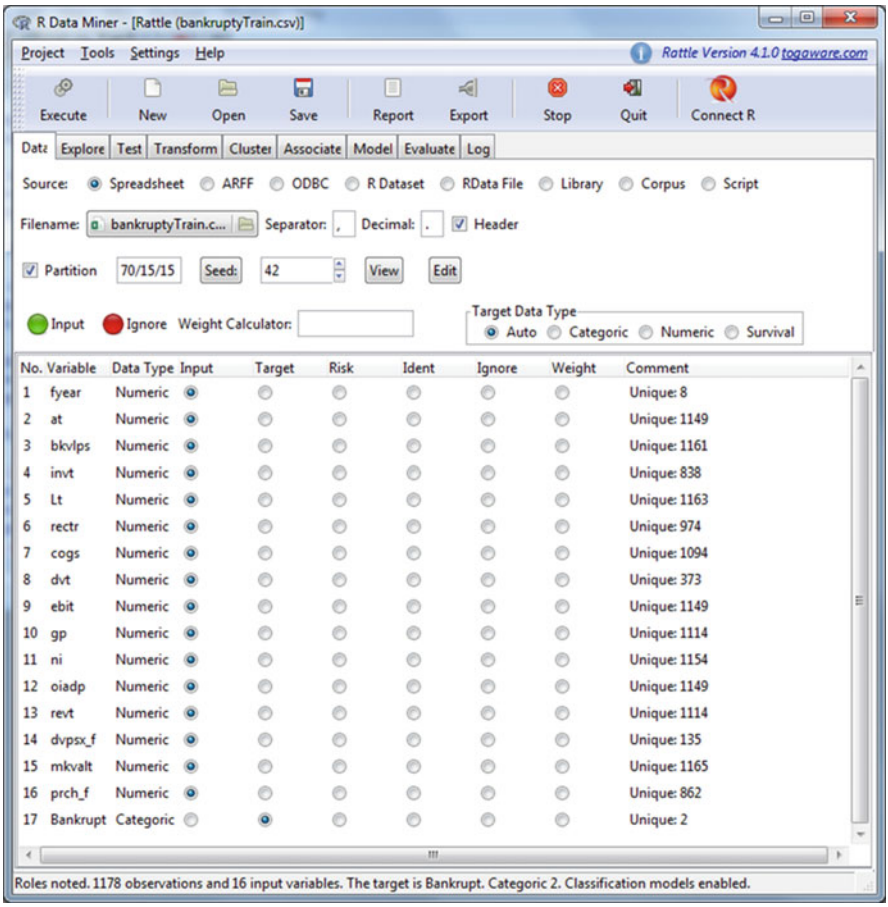

Fig. 9.2 LoanRaw.csv data read

would not occur. This rule was based on 44 % of the training data (360 out of 824), over which 84 % of these cases were not bankrupt (count of 304 no and 56 yes).

On the other branch, the next variable to consider is dvpsx_f. If dvpsx_f was less than 0.215 (364 cases of 464, or 44 % of the total), the conclusion is bankruptcy (340 yes and 24 no, for 93 %).

If revt $\geq$ 78 and dvpsx_f $\geq$ 0.215 (100 cases), the tree branches on variable at. If at $\geq$4169.341, the conclusion is bankruptcy (based on 31 of 31 cases). If at<4169.341, the model branches on variable invt.

For these 69 cases, if invt $<$ 16.179 (23 cases), there is a further branch on variable at. For these 23 cases if at <818.4345, the conclusion is bankruptcy (based on 13 of 13 cases). If at $\geq$818.4345, the conclusion is no bankruptcy (based on 7 of 10 cases).

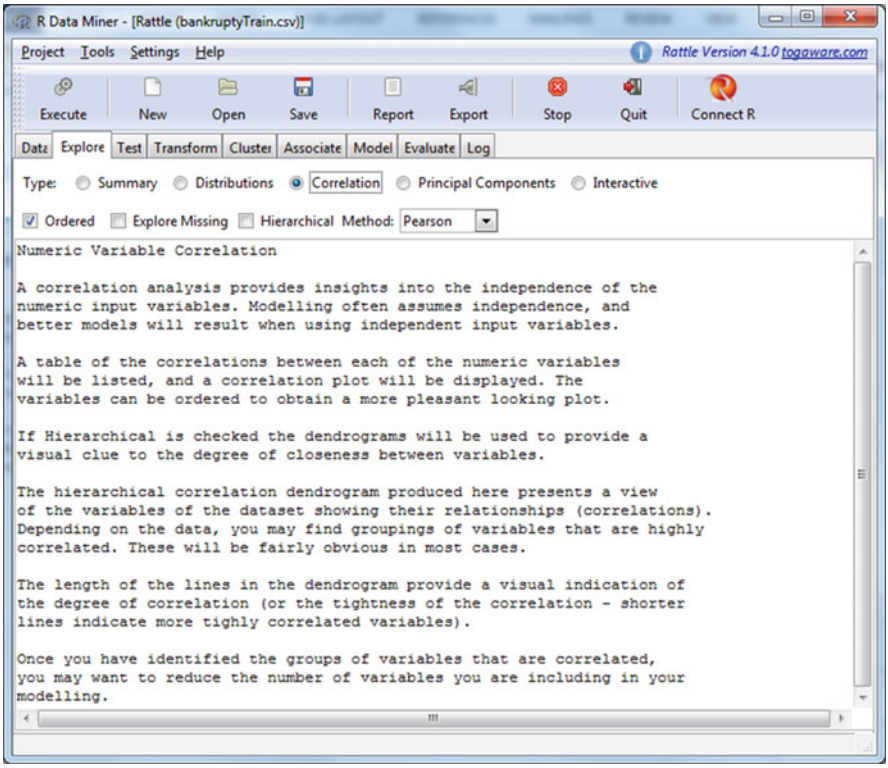

Fig. 9.3 Selecting correlation

If invt $\geq$ 16.179 (46 cases), the model splits further on invt. If invt $<$ 74.9215, the conclusion is no bankruptcy (based on 18 of 18 cases). If invt $\geq$ 74.9215, there is further branching on variable mkvalt. For mkvalt $<$ 586.9472, the conclusion is bankruptcy based on 11 of 14 cases. If mkvalt $\geq$ 586.9472, the conclusion is no bankruptcy (based on 13 of 14 cases).

This demonstrates well how a decision tree works. It simply splits the data into bins, and uses outcome counts to determine rules. Variables are selected by various algorithms, often using entropy as a basis to select the next variable to split on (Table 9.2).This model shows overall accuracy of 164/176, or 0.932. These validation were over the same period over which the model was built, up to 2008. We now test on a more independent testing set (2009–2010) as shown in Table 9.3:Here the overall correct classification rate is 126/143, or 0.881. The model was correct in 80 of 90 cases where firms actually went bankrupt (0.889 correct). For test cases where firms survived, the model was correct 46 of 53 times (0.868 correct).

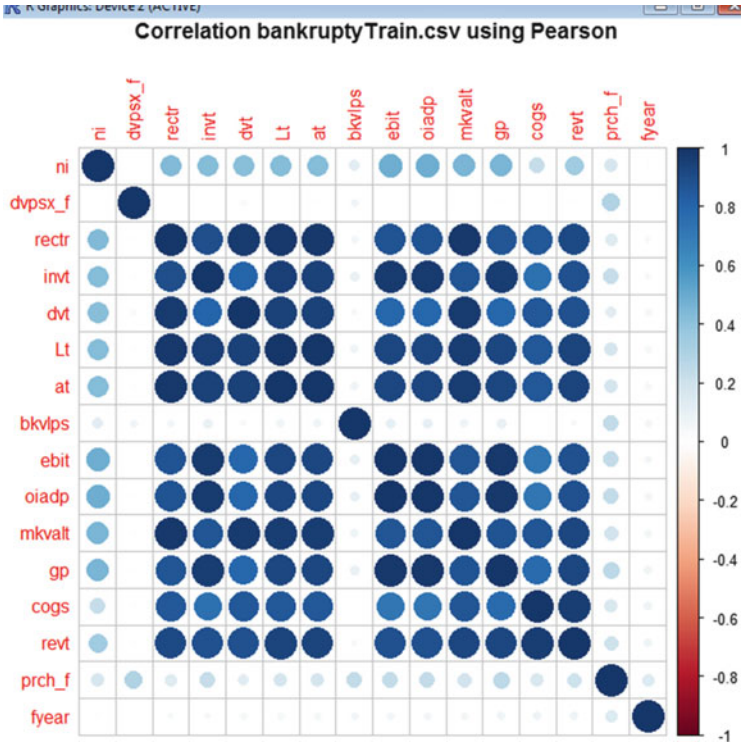

Fig. 9.4 Correlation plot

Logistic Regression Model

We can obtain a logistic regression model from Rattle by clicking the Linear button in Fig. 9.8, followed by the Logistic button.**Execute** yields Fig. 9.9 output:

Note that R threw out two variables (oiadp and revt), due to detected singularity. This output indicates that variables rectr and gp are highly significant. Further refinement of logistic regression might consider deleting some variables in light of correlation output. Here we are simply demonstrating running models, so we will evaluate the above model on both the validation set (Table 9.4) and the test set.This model shows overall accuracy of 158/176, or 0.898. This is slightly inferior to the decision tree model. We now test on a more independent testing set (2009–2010) as shown in Table 9.5:Here the overall correct classification rate is 111/143, or 0.776. The model was correct in 78 of 90 cases where firms actually went bankrupt (0.867 correct). For test cases where firms survived, the model was correct 33 of 53 times (0.623 correct).

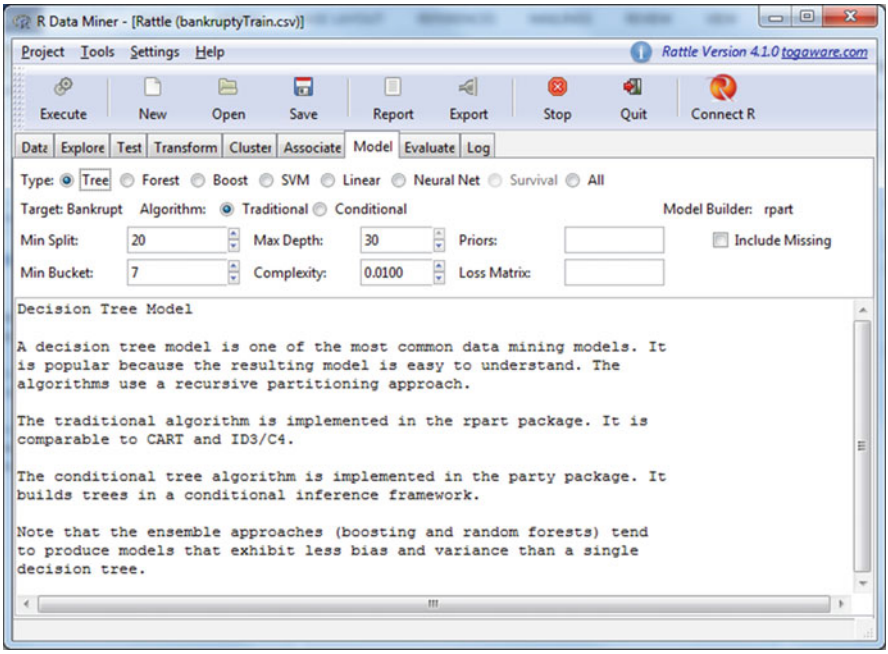

Fig. 9.5 Selecting decision tree

Neural Network Model

To run a neural network, on the Model tab, select the neural net button (see Fig. 9.10):**Execute** yields a lot of values, which usually are not delved into. The model can be validated and tested as with the decision tree and logistic regression models. Table 9.6 shows validation results:This model shows overall accuracy of 156/176, or 0.886. This is slightly inferior to the decision tree model. We now test on a more independent testing set (2009–2010) as shown in Table 9.7:Here the overall correct classification rate is 121/143, or 0.846. The model was correct in 75 of 90 cases where firms actually went bankrupt (0.833 correct). For test cases where firms survived, the model was correct 46 of 53 times (0.868 correct).

Here the decision tree model fit best, as shown in Table 9.8, comparing all three model test results.All three models had similar accuracies, on all three dimensions (although the decision tree was better at predicting high expenditure, and correspondingly lower at predicting low expenditure). The neural network didn't predict any high expenditure cases, but it was the least accurate at doing that in the test case. The decision tree model predicted more high cases. These results are typical and to be expected—different models will yield different results, and these relative advantages are liable to change with new data. That is why automated systems

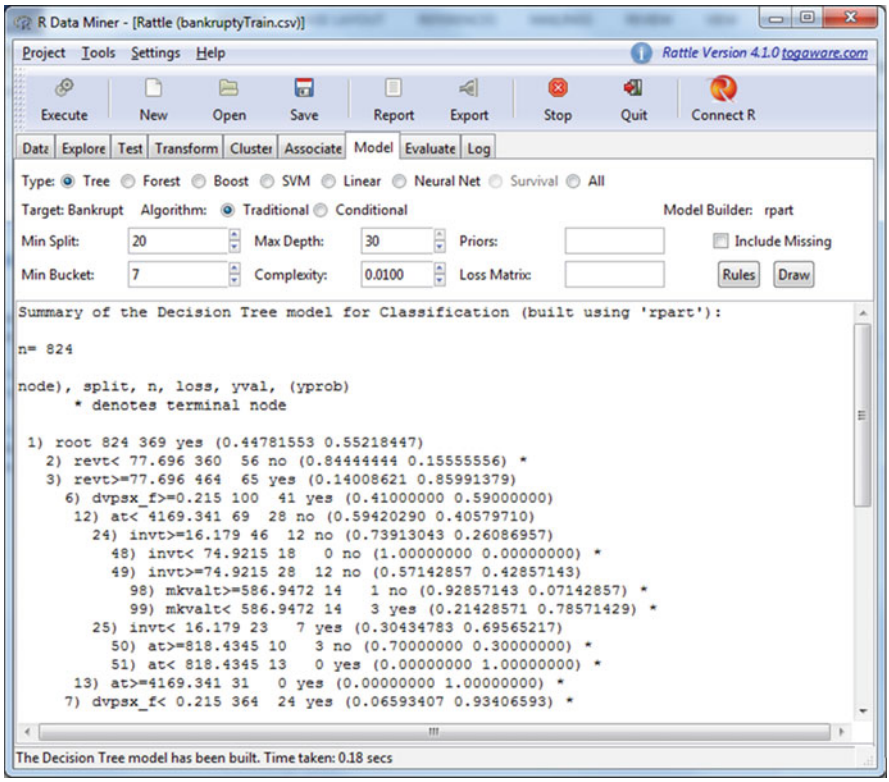

Fig. 9.6 Default decision tree model

applied to big data should probably utilize all three types of model. Data scientists need to focus attention on refining parameters in each model type, seeking better fits for specific applications.

Of course, each model could be improved with work. Further, with time, new data may diverge from the patterns in the current training set. Data mining practice is usually to run all three models (once the data is entered, software tools such as Rattle make it easy to run additional models, and to change parameters) and compare results. Note that another consideration not demonstrated here is to apply these models to new cases. For decision trees, this is easy—just follow the tree with the values for the new case. For logistic regression, the formula in Fig. 9.9 could be used, but it requires a bit more work and interpretation. Neural networks require entering new case data into the software. This is easy to do in Rattle for all three models, using the Evaluate tab and linking your new case data file.

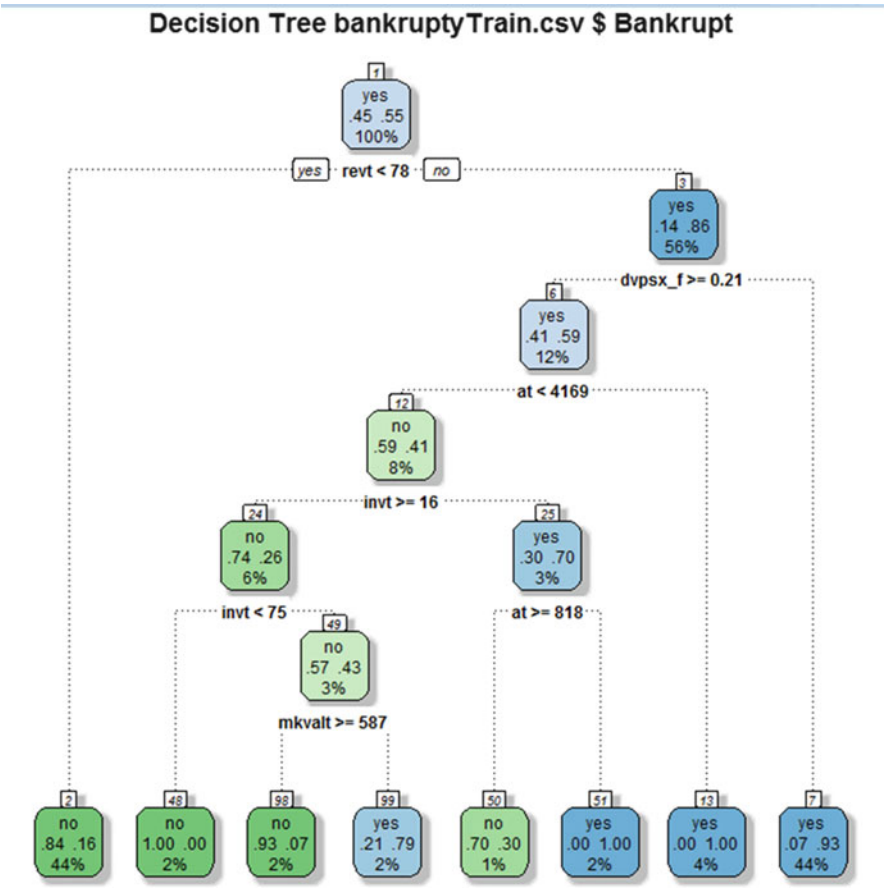

Fig. 9.7 Rattle graphical decision tree

Table 9.2 Coincidence matrix for validation set of decision tree model

	Model not bankrupt	Model bankrupt	
Actual not bankrupt	70	6	76
Actual bankrupt	6	94	100
	76	100	176

Table 9.3 Coincidence matrix for test set of decision tree model

	Model not bankrupt	Model bankrupt	
Actual not bankrupt	80	10	90
Actual bankrupt	7	46	53
	87	56	143

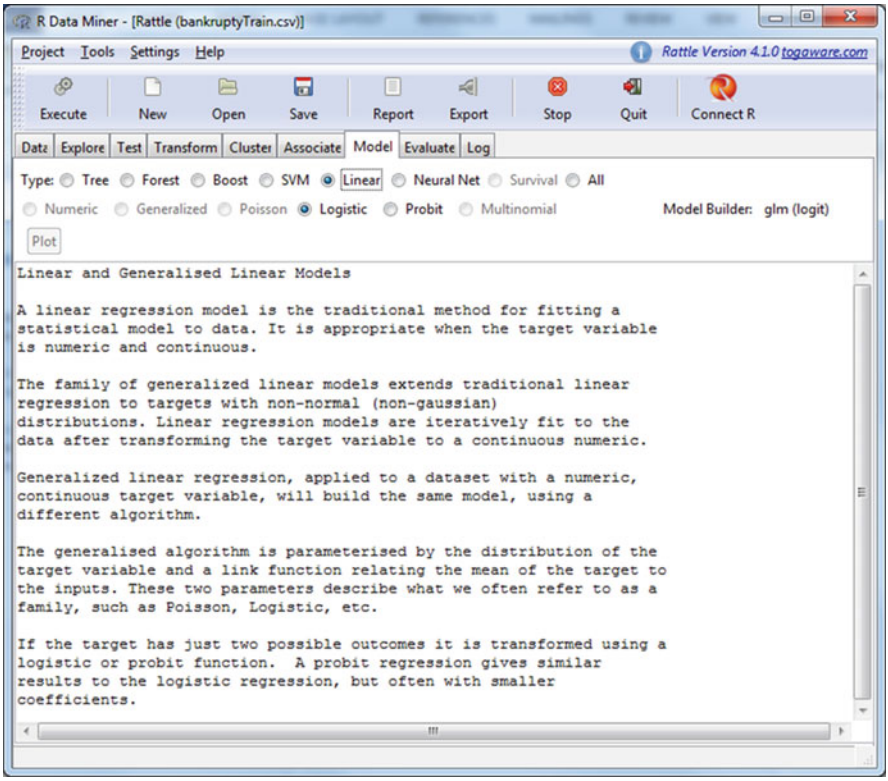

Fig. 9.8 Selecting logistic regression

Summary

We have demonstrated data mining on a financial risk set of data using R (Rattle) computations for the basic classification algorithms in data mining. The advent of big data has led to an environment where billions of records are possible. We have not demonstrated that scope by any means, but it has demonstrated the small-scale version of the basic algorithms. The intent is to make data mining less of a black-box exercise, thus hopefully enabling users to be more intelligent in their application of data mining.

We have demonstrated an open source software product. R is a very useful software, widely used in industry and has all of the benefits of open source software (many eyes are monitoring it, leading to fewer bugs; it is free; it is scalable). Further, the R system enables widespread data manipulation and management.

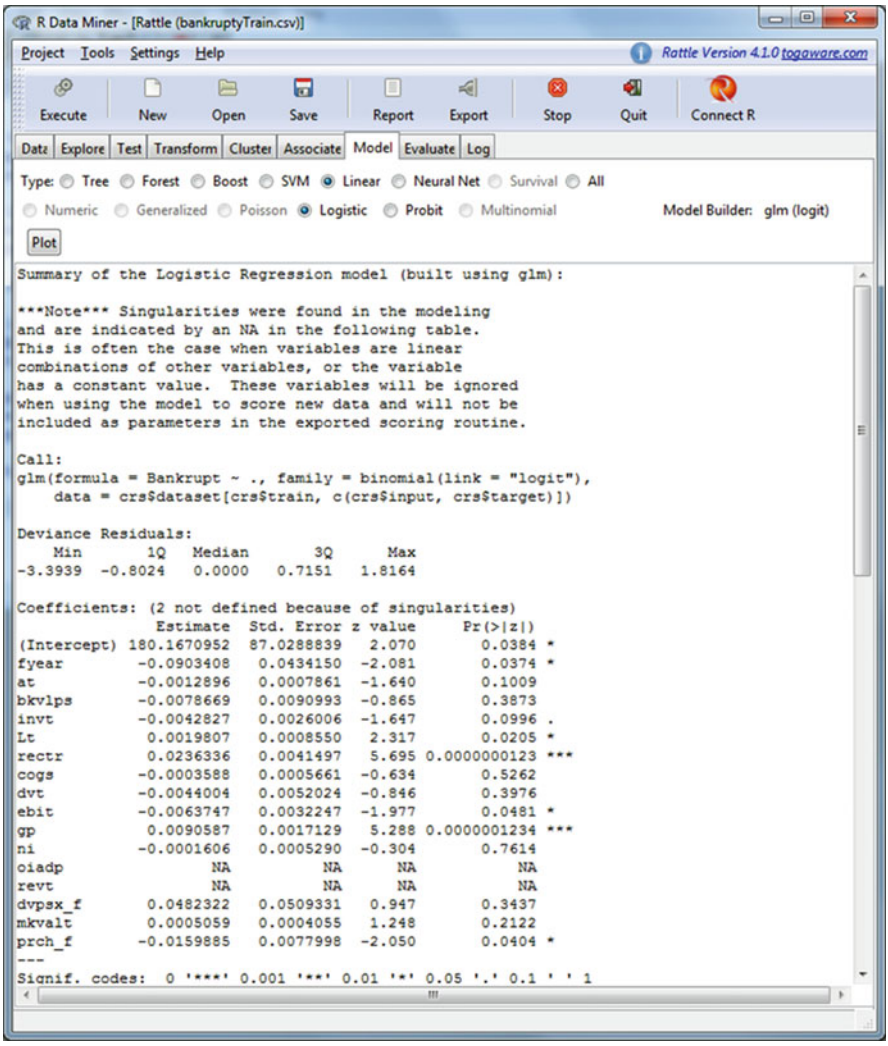

Fig. 9.9 Logistic regression output

Table 9.4 Coincidence matrix for validation set of logistic regression model

	Model not bankrupt	Model bankrupt	
Actual not bankrupt	72	4	76
Actual bankrupt	14	86	100
	86	90	176

Table 9.5 Coincidence matrix for test set of logistic regression model

	Model not bankrupt	Model bankrupt	
Actual not bankrupt	78	12	90
Actual bankrupt	20	33	53
	98	45	143

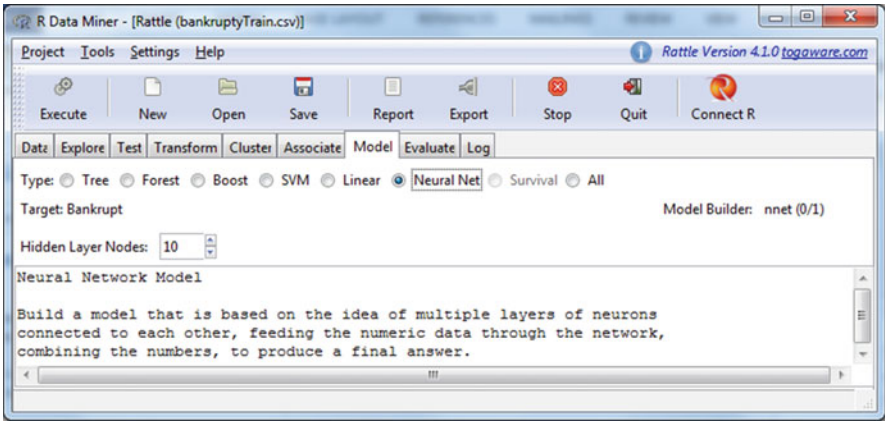

Fig. 9.10 Selecting neural network model

Table 9.6 Coincidence matrix for validation set of neural network model

	Model not bankrupt	Model bankrupt	
Actual not bankrupt	67	9	76
Actual bankrupt	11	89	100
	78	98	176

Table 9.7 Coincidence matrix for test set of neural network model

	Model not bankrupt	Model bankrupt	
Actual not bankrupt	75	15	90
Actual bankrupt	7	46	53
	82	61	143

Table 9.8 Comparative test results

Model	Correct not bankrupt	Correct bankrupt	Overall
Decision tree	0.889	0.868	0.889
Logistic regression	0.867	0.623	0.776
Neural network	0.833	0.867	0.846

Notes

1. Olson, D.L. and Shi, Y. (2006). *Introduction to Business Data Mining*. Irwin/McGraw-Hill.
2. Debreceny, R.S. and Gray, G.L. (2010). Data mining journal entries for fraud detection: An exploratory study. *International Journal of Accounting Information Systems* 11(3), 157–181; Jan., M., van der Werf, J.M., Lybaert, N. and Vanhoof, K. (2011). A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications* 38(10), 13,351–13,359.
3. Holton, C. (2009). Identifying disgruntled employee systems fraud risk through text mining: A simple solution for a multi-billion dollar problem. *Decision Support Systems* 46(4), 853–864.
4. Groth, S.S. and Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. *Decision Support Systems* 50(4), 680–691; Chan, S.W.K. and Franklin, J. (2011). A text-based decision support system for financial sequence prediction. *Decision Support Systems* 52(1), 189–198; Schumaker, R.P., Zhang, Y., Huang, C.-N. and Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems* 53(3), 458–464; Hagenau, M., Liebmann, M. and Neumann, D. (2013). Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems* 55(3), 685–697; Wu, D.D., Zheng, L. and Olson, D.L. (2014). A decision support approach for online stock forum sentiment analysis. *IEEE Transactions on Systems Man and Cybernetics: Systems* 44(8), 1077–1087.
5. Olson, D.L. (2016). *Data Mining Models*. Business Expert Press.
6. Jans, M., Lybaert, N. and Vanhoof, K. (2010). Internal fraud risk reduction: Results of a data mining case study. *International Journal of Accounting Information Systems* 11, 17–41.
7. Olson, D.L., Delen, D., and Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction, *Decision Support Systems,* volume 52 (2), 464–473.