

“An EXCEL-Based Decision-Making Tool”

AUTHORS

Ajay K. Aggarwal

ARTICLE INFO

Ajay K. Aggarwal (2004). An EXCEL-Based Decision-Making Tool. *Problems and Perspectives in Management*, 2(2)

RELEASED ON

Thursday, 02 September 2004

JOURNAL

"Problems and Perspectives in Management"

FOUNDER

LLC “Consulting Publishing Company “Business Perspectives”



NUMBER OF REFERENCES

0



NUMBER OF FIGURES

0



NUMBER OF TABLES

0

© The author(s) 2025. This publication is an open access article.

An EXCEL-Based Decision-Making Tool

Ajay K. Aggarwal¹

Abstract

Managers in manufacturing and service organizations often face decision-making situations involving prediction or classification problems. The superiority of artificial intelligence (AI) modeling techniques over traditional statistical methods is generally well-established. This paper demonstrates the creation of a decision-maker friendly forecasting model that uses the spreadsheet program EXCEL as the front end with a trained hybrid neural network model imbedded as a Visual Basic module. A key part of the Microsoft Office Suite, EXCEL, is widely popular as a decision-making tool because of its versatility. It is useful for number crunching, creating graphics, automating tasks with macros, and permitting the import of data in several different formats. The model proposed in the paper works in runtime mode, and enables decision-makers to play what-if games via the familiar EXCEL interface. The methodology is completely transferable across industries for similar decision-making situations.

Introduction

Managers are noted for their decision-making abilities. Whether an organization is engaged in manufacturing products or providing services, the importance of making the right decision cannot be overstated. Decision-making has been complicated by the globalization and Internet revolution of the 1990s. In particular, these phenomena have significantly enhanced the amount of information available to decision-makers. Managers often find themselves inundated with data while being forced to respond to decision situations in a timely manner. While technologically sound commercial solutions for many decision problems are available, managers are often uncomfortable with the technical sophistication required to implement them.

In the literature, the superiority of artificial intelligence techniques over traditional statistical models has been well-established. Most notably, neural networks have been used for providing solutions to problems ranging from credit-card fraud detection, bank failures prediction, and sales forecasting to hiring productive employees, projecting hospital discharge dates, and predicting customer churn in telecommunication firms. Despite their prowess as a decision-support tool, neural networks adoption has been lack-luster relegating their status to a "black box", even by technically savvy managers. One of the potential reasons for this is the lack of versatile and easy-to-use model building programs, and complex user-interfaces.

There is a need in the marketplace for decision-making tools that can combine the simplicity of friendly user-interfaces with the technically savvy of sophisticated algorithms. This paper presents such a tool. The tool presented uses EXCEL spreadsheet as a front-end. Decision-makers generally like EXCEL for its versatility. It is useful for number crunching, creating graphics, automating tasks with macros, and permitting the import of data in several different formats. For instance, it can be used for creating budget reports, performing intricate financial analysis, or analyzing marketing data. Graphics such as pie charts, line and bar graphs can be deployed to enhance data presentation. Actions that need to be performed repeatedly can be captured in a macro and executed when needed. Formats such as Lotus, QuatroPro, dBase, XML, etc. may be imported into EXCEL with ease.

The spreadsheet imbeds trained neural network models in a Visual Basic module. Prediction and classification problems can be handled through this approach. Prediction problems involve prediction of a variable value based upon several inputs. For instance, demand forecast for an item may be a function of time, historical demand, information about competition, and advertis-

¹ Else School of Management, Millsaps College, 1701 North State St, Jackson, MS 39210, USA. Phone: (601) 974-1270; Fax: (601) 974-1260; E-Mail: aggarak@millsaps.edu

ing budget. Classification problems involve placing the data into one of two or more categories. For instance, customers may be classified into high-discount-worthy, moderate-discount-worthy and non-discount-worthy based upon income, sales history, longevity of account, and credit worthiness.

This paper consists of three sections. The first section introduces the neural network concept. The second one demonstrates neural network modeling for a decision-making problem and imbeds the resulting model inside EXCEL. Finally, the third section offers some concluding remarks on the technique presented.

The Neural Networks Concept

Neural networks may be viewed as a highly interconnected network of nodes spread out over several layers. Generally speaking they comprise of an input layer, one or more hidden layers, and an output layer. Figure 1 depicts a simple neural network having n input nodes, m nodes in one hidden layer, and r output nodes. In the example shown, the n neurons in the input layer neurons receive their input data or "signals" for processing. The hidden layer neurons weigh the input feeds, sum them up, and then feed the result to r output layer neurons following transformation via a logistic sigmoid or other function. The procedure eventually leads to production of output signals at the r output layer neurons.

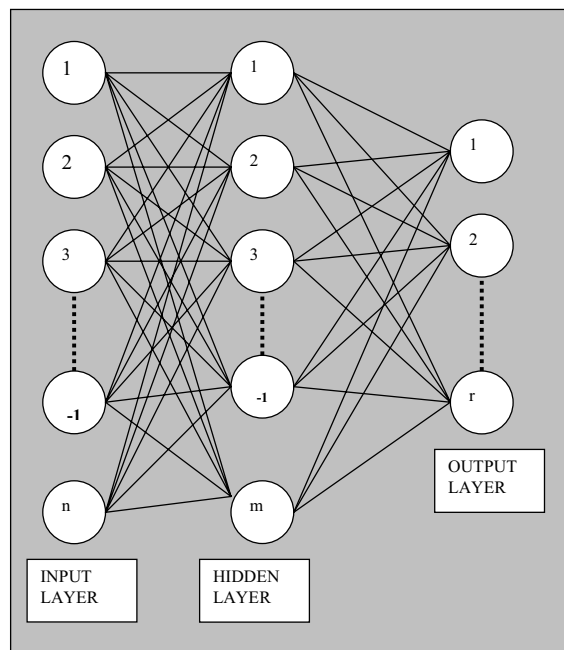


Fig.1. A Typical Neural Network

Back propagation is the most popular type of neural network used in forecasting. It typically starts with a presumed network architecture that employs random weights along the different network branches. Training data comprising of input-output sets is provided to initiate its development for a particular application. The network begins with the examining the first set of input values X_i , where $i=1, n$. These are forward propagated from the input layer through the network until an output is produced Y_k , $k=1, \dots, r$ is produced. This is compared to the desired output. If the results differ by more than a specified tolerance, weights along the different network branches are adjusted by propagating the differences backwards from units higher up in the network based upon a back-propagation formula. This concept is used repeatedly until a trained network that produces

output within the desired tolerance results. In case the training fails to produce the desired result, two remedies may be attempted. The first one involves changing the data presentation. The second one involves changing the network parameters. Bishop [2] offers a comprehensive technical discussion of the related concepts.

Data presentation may be altered by data transformation or variable selection. Data transformation deals with changing the data to make it better conform to the output variable pattern. For instance, taking logarithm of large input values may facilitate the training process. In addition, categorical data (e.g. male/female, yes/no, true/false, etc.) may be converted into numerical variables. Skewed data may be controlled by describing the variable as belonging to a combination of open and closed-ended intervals. Variable selection typically involves selection of variable subsets that are likely to produce better output. This typically involves using a technique like stepwise regression. It explores the contributions of each variable in isolation as well as in the presence of others. It can help remove non-contributing variables, and boost network performance.

Network parameters modification could involve a combination of several possible changes such as the number of neurons per layer, the number of layers, the learning rate, the learning rule (e.g. Kalman filter rule, Adaptive Gradient rule, etc.), the starting and ending tolerances, the nature of the transfer function, etc.

Recently several researchers have reported some success with production of hybrid models. These models are basically neural networks that tend to exploit the usefulness of conventional statistical techniques within the framework of neural network processing. To facilitate this, these models embellish the usual neural network input variables with appropriate statistical measures. Figure 2 shows a hybrid neural network model with $n+2$ input nodes, which includes the original n nodes (see Figure 1) and 2 statistical measures S_i and S_j . The hidden and output layers are configured as before. The final result shows the promise of producing output that outpaces those produced by either the neural network or the conventional statistical technique by itself. Once a trained network is produced, it is tested on a portion of the data not viewed by it in the course of the training phase. Production of desirable output at this junction validates the network.

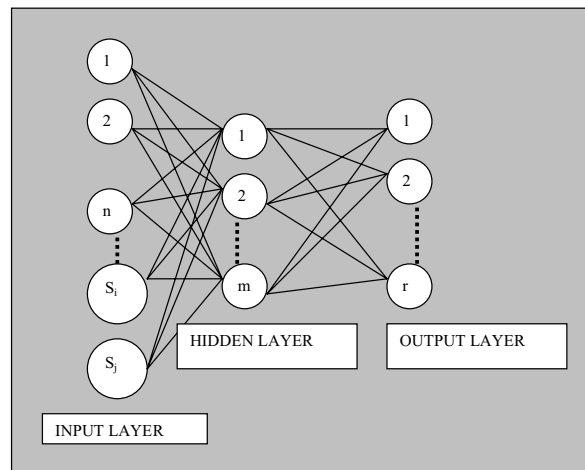


Fig. 2. A Typical Hybrid Neural Network

An Example

Selecting interns is an annual undertaking at surgical residency programs throughout the United States. While the criteria for admissions vary widely, the program director has the responsibility for establishing and implementing one in place to select the best candidates. A detailed discussion of related literature and related issues is presented in [1]. The 36 variables considered include performance measures, interests, personal characteristics, and job-motivation data.

Using the Predict software [12], a neural network having 36 input variables is created. The software is a cinch to use. The user is prompted initially for location of data file, and then for information about the problem type and data properties. Spreadsheet, database, and text files are all included in the long list of acceptable data formats. For information pertaining to the problem and the data, the software displays default responses for all questions. In case the user disagrees with the default response, the option of choosing from a displayed list exists. For new users, the software includes demonstrations, as well as a "Quick Start Guide" which is quite helpful.

In the example presented there is only one predictor or output variable – the likelihood of admission. The output variable ranges from zero for poor candidates to one for excellent prospects. The data available for the training, testing, and validation consists of 338 observations collected over a decade. The Predict software uses the backward propagation neural network technique for forecasting. The software is loaded as an add-in for EXCEL. The user specifies the location of the data file, and the different input and output variables. The program uses the EXCEL data for training, testing, and validation. The program asks the user several qualifying questions including the problem type, extent of noise level, data analysis and transformation level, input variable selection level, and the neural network search level.

The problem type can be prediction or classification. The noise levels choices are clean, moderately clean, noisy, and very noisy. Clean data implies that identical inputs result in identical outputs in the training data. The extent to which this doesn't happen determines the noise level. The data transformation, variables selection, and network choices are all five-fold – namely none, superficial, moderate, comprehensive, and exhaustive. Data transformation deals with the modifications performed on the data to make it more amenable to training. For instance, extremely large and small values may be suppressed, and some functional transformations may be performed. Variable selection deals with the flexibility in choosing which variables to include in the model. Network choices deals with the manner in which the final neural network models evolve from initial networks. The network building commences immediately upon selection of the data and the desired features.

The network training, testing and validation results are instantly available upon completion of training. A key output provided by the software is the flash code that stores the trained network parameters in Visual Basic. Network training is conducted using the Predict software. Since data ordering has a bearing on the data selected for training or testing, 30 sets of data were randomly created by repeatedly using the RANDOM and SORT functions in EXCEL. The Predict software partitioned each data set into two parts – 236 observations were kept for training, while 102 ones were kept for testing and validation. The procedure employed for creation of a typical dataset was as follows. A column that stored a random number was made as the first column of the dataset. The entire dataset was then sorted in ascending order on the basis of the first column value. Table 1 presents the overall details of the neural network training and testing results. As can be seen the best neural network produced 84.6% matches, with a correlation coefficient of 0.71, and a root mean square error of 0.337.

Table 1

Basic Neural Network Performance

Model	Neural Network Model		
	R	RMS	%matches
1	0.64	0.365	80.8%
2	0.69	0.345	83.4%
3	0.63	0.371	79.3%
4	0.64	0.367	80.5%
5	0.66	0.361	80.5%
6	0.71	0.334	82.5%
7	0.66	0.360	80.5%
8	0.62	0.373	79.8%
9	0.65	0.362	81.1%
10	0.66	0.360	81.1%
11	0.67	0.355	82.2%
12	0.64	0.367	79.9%
13	0.66	0.361	79.9%
14	0.71	0.338	83.1%
15	0.66	0.360	82.5%
16	0.60	0.382	76.3%
17	0.61	0.379	79.9%
18	0.66	0.358	82.2%
19	0.66	0.358	82.2%
20	0.71	0.337	84.6%
21	0.65	0.364	81.4%
22	0.66	0.357	82.2%
23	0.66	0.360	82.0%
24	0.71	0.336	82.5%
25	0.63	0.370	77.8%
26	0.64	0.365	79.3%
27	0.63	0.369	80.5%
28	0.62	0.374	77.8%
29	0.67	0.353	82.8%
30	0.64	0.368	78.4%

An earlier study [1] using the Brainmaker professional software along with the genetic training and optimizer packages employing the parent dataset confirmed that the neural network output performed statistically better than discriminant analysis. The hybrid model design was explored with the intention to produce a model with better predictive powers using the discriminant analysis results along with the basic neural network model. To that end, the Mahalanabis distances were input along with the raw variable data in the neural network model. Table 2 presents the results of the hybrid model and compares them with the simple neural network model. The best hybrid model results in 91.4% match, a correlation coefficient of 0.84, and a root mean squared error of 0.259. Table 3 presents the results of the matched t-test that confirms that the hybrid model per-

formed statistically superior to the neural network model. The t value of -4.09 is significant at the 0.0003 level, supporting the superiority of hybrid models.

Table 2

Comparing Model Performances

Model	Neural Network Model			Hybrid Model		
	R	RMS	%matches	R	RMS	%matches
1	0.64	0.365	80.8%	0.70	0.339	83.7%
2	0.69	0.345	83.4%	0.84	0.259	91.4%
3	0.63	0.371	79.3%	0.66	0.360	82.8%
4	0.64	0.367	80.5%	0.65	0.365	80.5%
5	0.66	0.361	80.5%	0.77	0.303	87.9%
6	0.71	0.334	82.5%	0.79	0.292	87.9%
7	0.66	0.360	80.5%	0.65	0.364	80.2%
8	0.62	0.373	79.8%	0.72	0.330	84.9%
9	0.65	0.362	81.1%	0.64	0.368	80.5%
10	0.66	0.360	81.1%	0.66	0.361	82.0%
11	0.67	0.355	82.2%	0.63	0.370	81.7%
12	0.64	0.367	79.9%	0.62	0.374	79.9%
13	0.66	0.361	79.9%	0.64	0.367	81.7%
14	0.71	0.338	83.1%	0.65	0.363	81.4%
15	0.66	0.360	82.5%	0.74	0.321	85.8%
16	0.60	0.382	76.3%	0.64	0.366	79.9%
17	0.61	0.379	79.9%	0.63	0.372	80.8%
18	0.66	0.358	82.2%	0.71	0.337	82.8%
19	0.66	0.358	82.2%	0.80	0.287	89.9%
20	0.71	0.337	84.6%	0.69	0.345	84.3%
21	0.65	0.364	81.4%	0.65	0.363	80.5%
22	0.66	0.357	82.2%	0.65	0.365	80.8%
23	0.66	0.360	82.0%	0.71	0.334	83.7%
24	0.71	0.336	82.5%	0.75	0.317	86.4%
25	0.63	0.370	77.8%	0.82	0.275	90.2%
26	0.64	0.365	79.3%	0.65	0.364	80.2%
27	0.63	0.369	80.5%	0.66	0.358	81.1%
28	0.62	0.374	77.8%	0.71	0.337	84.3%
29	0.67	0.353	82.8%	0.70	0.339	84.9%
30	0.64	0.368	78.4%	0.65	0.363	80.2%

Table 3

Paired Samples Test to Compare Network Performances

Paired Samples T-Test	Paired Differences			t	df	Significance (2-tailed)
	Mean	Std. Dev.	Std. Error Mean			
BASIC - HYBRID	-8.43	11.31	2.06	-4.09	29	0.0003

The Predict software enables retrieval of the “Flash Code” that created the trained neural network. To exploit this for the benefit of the decision maker, it was decided to imbed the flash code inside a Visual Basic module within EXCEL. This automatically generated a function called XLFC that produced network output when presented with the range of input values. To utilize the function, only the range of input values needs to be included as parameter of the XLFC function. The system responds by inserting the output value generated by the neural network model for the input values provided.

In order to facilitate an interactive working hybrid model predictor in EXCEL two additional tasks have to be performed. First, the EXCEL spreadsheet has to be equipped with the intelligence to generate the Mahalanabis distances for any decision-maker input for the 36 variables. Second, a friendly way to elicit decision maker input has to be planned.

The first task is accomplished by using a leading statistical package – SPSS version 11 for Windows. Discriminant analysis is performed on the raw data. Mahalanabis distances for any data values can be computed by utilizing the values of the unstandardized canonical discriminant function coefficients and the values of the predictor variable (MATCH) at the group centroids. Table 4 shows the details of the computations that are conducted in EXCEL. The canonical discriminant function coefficients (CDFC) and the grouping variable function values were obtained directly from the discriminant analysis reports produced by the statistical package SPSS.

The second task is facilitated by dividing up the 36 variables into two sets – one that required a YES/NO input and the other that requires numeric input. EXCEL formulas are written to convert all categorical data to numeric. In addition, provisions are made to regroup the 36 variables into their original training order along with the two Mahalanabis distance values at the end. All of the processing occurs in a part of the spreadsheet hidden from the decision maker view. In figure 3, the admissibility rating of 0.04 is computed by using the XLFC function on the 36 input values displayed above it. The display permits the decision-maker to change any or all values of the basic 36 variables, and instantly displays the changes in the predictor variable. To a decision-maker familiar with EXCEL, the interface presents a friendly way to use a state-of-the-art technology. The decision-maker also has the ability to employ all features of EXCEL when using the model. For instance, the format of all cells, colors, zoom rate, etc. may be altered to suit the decision-maker’s taste. Ad hoc reports may be generated by copying and pasting, or printing needed information from any part of the system.

Computation of Mahalanabis Distances

MAHALANABIS DISTANCES COMPUTATION FOR A TYPICAL CASE							
Variable	CDFC	Value	CDFC*Value	Variable	CDFC	Value	CDFC*Value
AGE	0.113	26	2.933	HANDS	-0.319	0	0.000
US	-1.893	1	-1.893	HOBBIES	0.401	0	0.000
MILITA	-0.444	0	0.000	LEADER	0.068	0	0.000
UMC	0.779	0	0.000	JOCK	-0.403	1	-0.403
GAP	-0.076	0	0.000	LCS	0.305	1	0.305
WORKED	-0.027	0	0.000	HOP	0.440	1	0.440
BOARD	-0.189	2	-0.377	LFS	0.268	4	1.071
SCORE	0.000	219	0.054	LNS	0.188	0	0.000
ORGANIZ	0.095	0	0.000	RANKYN	-0.735	1	-0.735
RES	0.035	0	0.000	RANK	0.015	69	1.003
PUBL	0.048	3	0.143	AOA	-0.090	0	0.000
PMC	0.059	2	0.118	SURGGR	0.365	4	1.462
SICK	-0.372	0	0.000	ANATGR	0.225	4	0.899
SICKNOW	-0.073	0	0.000	REPEAT	-0.070	0	0.000
SEX	0.354	0	0.000	INT1	0.030	5	0.148
ALWAYS	0.436	0	0.000	INT2	0.549	5	2.743
PARENT	0.342	0	0.000	INT3	0.672	5	3.362
PSURG	-0.594	0	0.000	Constant	-10.723	1	-10.723
SICKKID	-0.442	0	0.000				
						DISCRIM SCORE=	0.549

(Sum of all CDFC*Value terms)

CDFC= Canonical Discriminant Function Coefficients

Grouping Variable	Function at Group Centroid	Computation Needed to Compute the Mahalanabis Distance
Match-0	-0.596	$(-0.596 - 0.549)^2 = 1.310$
Match-1	1.097	$(1.097 - 0.549)^2 = 0.301$

(Note: All Shaded values were obtained directly from SPSS)

APPLICANT CLASSIFICATION SYSTEM			
Is the applicant a US Citizen?	Yes	What is the applicant's age (in years)?	26
Has the applicant served in the military?	No	What is the gap before medical school (in years)?	1
Is the applicant your own graduate?	No	How many boards has the applicant taken? (0-2)	2
Has the applicant worked before?	No	What is the average board score? (Default is 0)	219
Is there a history of sickness in the applicant's family?	No	Rate the applicant's research organization ability (0-6)	3
Is there major sickness in the applicant's family currently?	No	Rate the currency of published works (0-2)	0
Is the applicant a male?	Yes	Number of publications (0-30)	1
Did the applicant always want to be a surgeon?	No	Rate applicant's membership in professional orgs. (0-2)	2
Are any of the applicant's parents a Physician?	Yes	Rate the relevance of applicant's hobbies (0 to 1)	0
Are any of the applicant's parents a surgeon?	No	Recommendations from faculty surgeons (0-7)	1
Was the applicant sick as a kid?	No	Recommendations from non-surgical faculty (0-7)	1
Is the applicant good with hands?	No	Enter the applicant's class rank if available (Default is 99)	68
Does the applicant possess leadership skills?	Yes	Grade on Surgical Internship (0-5)	5
Is the applicant active in several sports?	Yes	Grade on Anatomical Internship (0-5)	3
Did the chief of surgery write a recommendation letter?	Yes	Grade by first interviewer (0-6)	5
Will the applicant rank high for our own program?	Yes	Grade by second interviewer (0-6)	5
Is the applicant's class rank available?	Yes	Grade by third interviewer (0-6)	4
Is the applicant a member of Alpha Phi Omega?	No		
Did the applicant repeat any courses?	No		

Admissibility Rating (0=Deny, 1=Admit)
--

0.04

Fig. 3. Applicant Classification System in EXCEL

Conclusion

Managers can add a tool to their decision-making arsenal. The modeling technique presented in this research is fully transferable across industries for similar decision-making situations. Once the decision maker knows the input and output variables, neural network models, hybrid or otherwise, can be tested and imbedded in the EXCEL spreadsheet. The familiarity of spreadsheets makes the imbedded neural network a much easier sell with decision-makers. The imbedded network and related functionalities attach themselves to the EXCEL application on any host com-

puter, making this architecture feasible for group decision making. The high speed processors in today's personal computers and lap-tops make it possible to have all computational results of "what-if" scenarios available in real-time. Its simple construct is quite appealing and may pave the way for widespread usage of this technology in other arenas.

References

1. Aggarwal, A.K., Travers, S., Scott-Conner, C., "Selecting Residents for a Surgical Residency: A Neural Network Approach," *Cybernetics & Systems: An International Journal*, Volume 31, Number 4, pages 417-430, 2000.
2. Bishop, C.M., "Neural Networks for Pattern Recognition," Oxford, 1995.
3. Chen, H., Grant-Muller, S., Mussone, L. and Montgomery, F., "A Study of Hybrid Neural Network Approaches and the Effects of Missing Data on Traffic Forecasting," Vol. 10, 2001.
4. Goonatilake, S. and Khebbal, S., "Intelligent Hybrid Systems," Wiley, 1995.
5. Hiar, J.F., Anderson, R.E., and Tatham, R.L., "Multivariate Data Analysis," Second Edition, Macmillan, 1987.
6. Johnson, D.E. "Applied Multivariate Methods for Data Analysis," Duxbury Press, (ITP), 1998.
7. Kumar, A., "A Study of Composite or Hybrid Classifiers for Knowledge Discovery," *INFORMS Journal of Computing*, Vol. 11, No. 3, Summer 1999.
8. Lattin, J., Carroll, J.D., and Green, P.E., "Analyzing Multivariate Data," Thompson, 2003.
9. Lawrence, J., "Introduction to Neural Networks: Design, Theory, and Applications," California Scientific Software Press, 1993.
10. Markham, I.S. and Ragsdale, C.T., "Combining Neural Networks and Statistical Predictions to Solve the Classification Problem in Discriminant Analysis," Vol. 26, Number 2, March/April 1995.
11. McKeown P.G. and Piercy, C.A., "Learning to Program with Visual Basic," Second Edition, Wiley, 2002.
12. NeuralWorks Predict: Complete Solution for Neural Data Modeling, NeuralWare, Inc., 1995.