View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AIS Electronic Library (AISeL)

Designing a Prototype for Analytical Model Selection and Execution to Support Self-Service BI

Full Paper

Greg Schymik
Grand Valley State University
schymikg@gvsu.edu

Karen Corral
Boise State University
karencorral@boisestate.edu

David Schuff
Temple University
schuff@temple.edu

Robert St. Louis
Arizona State University
stlouis@asu.edu

Abstract

This paper presents a prototype of a modeling tool specifically designed for business analysts with little modeling experience. The proposed tool has an interactive user interface for a dimensional data store that contains a library of analytical models that business analysts can evaluate and use to create models they can run on their own data sets. Using a design science approach, we review the relevant literature in self-efficacy and feedforward to provide a kernel theory that informs the design criteria met by our proof of concept prototype. Specifically, we demonstrate the prototype's user interface with a prediction problem faced by the United States Department of Labor.

Keywords

Design Science, Self-Service BI, Model Building, Feedforward.

Introduction

Self-service business analytics and business intelligence (BI) is currently receiving widespread attention. This is defined as "an approach to data analytics that enables business users to access and work with corporate data even though they do not have a background in statistical analysis, business intelligence (BI) or data mining. Allowing end users to make decisions based on their own queries and analyses …"[1] is a key component. Another important aspect of self-service BI is that the business user should be able to work with data without the intervention of the information technology function (Logi Analytics 2014). However, tools that truly support self-service BI remain elusive – only about one-fifth of business analysts surveyed felt they had access to these self-service BI tools (Logi Analytics 2014).

Most analytics software is largely based on the Sample, Explore, Modify, Model, and Assess (SEMMA) process (SAS Institute 1998; Rohanizadeh and Moghadam 2009). An assumption of the SEMMA process is that the analyst has enough knowledge about statistics and the problem domain to know which variables should be considered for the model, the appropriate functional forms for those variables, and the appropriate functional forms for the interactions among those variables. However, nontechnical

[1] http://searchbusinessanalytics.techtarget.com/definition/self-service-business-intelligence-BI retrieved 8/20/2016.

business users typically do not have these skills. Tools that support self-service BI must be able to address this gap in business users' knowledge.

This paper describes the development of such a tool, specifically designed for business analysts with little modeling experience. Our proposed tool is an interactive user interface for a dimensional data store (proposed by Corral et al. 2015) that contains a library of analytical models on which the business analyst can draw. Using a design science approach, we begin by reviewing the relevant literature in self-efficacy and feedforward to provide a kernel theory that informs the design criteria. Next, we describe an implementation of our proposed design through a proof of concept prototype. We specifically demonstrate the user interface using a prediction problem faced by the United States Department of Labor. We conclude with future directions.

Literature Review

Model Formulation and Existing Solutions

Formulating models is a non-deterministic problem. As Box and Draper (1987) point out, there isn't a single, correct model, just ones that perform better than others. During model formulation, an analyst must weigh tradeoffs between complexity and accuracy, often without an ideal set of variables or data. Because of this, it falls under the definition of what Hevner et al. (2004) call a wicked problem, in that it has "unstable requirements based on ill-defined environmental contexts" (p.81).

Current technology-based solutions do not directly address model formulation. Products such as IBM's SPSS Modeler and SAS' Enterprise Miner help the analyst execute a formulated model through a drag-and-drop graphical user interface (see Figure 1). These model-based analytics tools make the tool easier to use by hiding the complexity of the underlying scripting language, but lack decision aids that guide the user regarding which variables, transformations, and techniques to use to build the model. This represents a gap in the current capability of analytics software. Tools that support model building for non-technical analysts must provide the same complexity "hiding" as current tools while incorporating decision aids to guide the user regarding model selection.

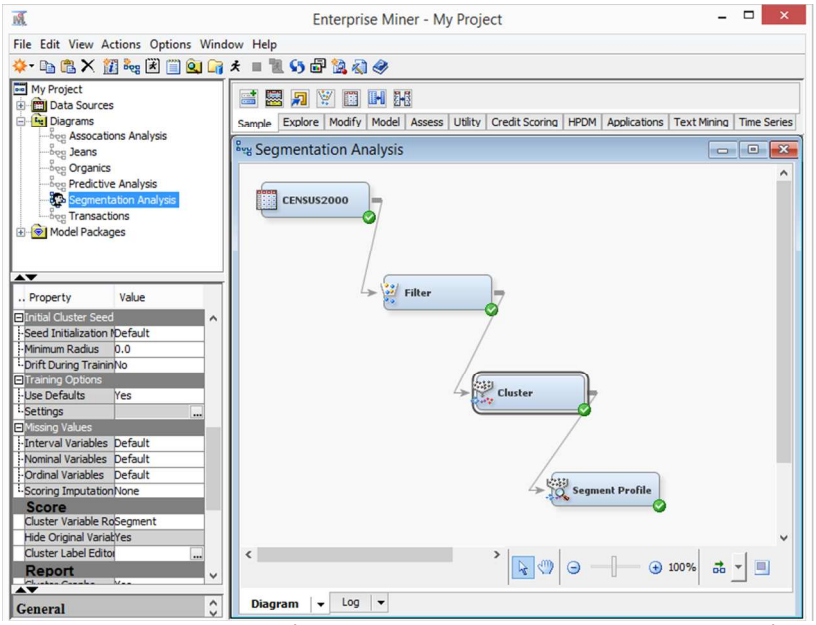

Figure 1. Example of Graphical User Interface for SAS Enterprise Miner

Minimizing the Effects of Low Self-Efficacy

An effective tool to support model building for non-technical analysts must simplify all stages of the model building process. This includes the selection of variables, transformations, and techniques during the model selection process, as well as the construction and execution of the script that implements those

models. This is because non-technical analysts are likely to have less experience and less confidence with model-building software. Confidence with using a software-based decision tool is often characterized as self-efficacy, as it was in the context of spatial decision support systems by Jarupathirun and Zahedi (2007).

A lack of self-efficacy can have serious consequences for a system's evaluation, adoption, and use. Low self-efficacy with statistical software, and anxiety regarding statistics in general, has been shown to negatively impact the perceived usefulness of modeling software, a major antecedent of intention to use (Hsu et al. 2009). Jarupathirun and Zahedi (2007) found self-efficacy positively influences perception of task-technology fit, which can influence the perceived decision quality of a tool. These negative outcomes can have long-term consequences. For example, Johnson et al. (2016) found that unsuccessful experiences with technology negatively influence learning of other technologies. Therefore, a lack of self-efficacy with decision tools is a clear barrier to their adoption and use. This makes building features that minimize the potential negative effects of a user's lack of self-efficacy with model-building software critical to its development.

Feedforward and Outcome Feedback as a Model-Building Decision Aid

Any decision aid must impart information to the analyst regarding their model-building choices. This information can be provided as outcome feedback, showing the consequences of their choice, or feedforward, defined as "generalized information pertaining to the input cues of an analysis that is provided to users prior to the performance of an analysis" (Dhaliwal and Benbasat 1996, p. 348). In general, feedforward has been shown to be more helpful than outcome feedback (Sengupta et al. 1993; Sterman 1989), as users can see the impact of a choice before they commit to the decision.

Cognitive feedback, as opposed to outcome feedback, includes task information about the "relations between the cues and the criterion, information about the criterion or the cues themselves, or both" (Balzer, et. al., 1989, p. 412). Lindell (1976) says feedforward is part of cognitive feedback.

Understanding the impact of a decision in advance through feedforward lowers task uncertainty. Reducing task uncertainty can improve subject performance (Björkman 1972). Intuitive thinking is increased in situations where uncertainty is high, therefore, analytical thinking should be increased under situations when uncertainty is low (Chenoweth et al. 2004). This has direct implications for modeling, as users have been found to use more complex models when uncertainty is reduced (Chenoweth et al. 2004). Feedforward reduces cognitive strain for the decision maker because she/he is given uncertainty-reducing information rather than having to intuit that information from outcome feedback (Björkman 1972).

Feedforward is often operationalized as instruction (Björkman 1972), and this can take many forms. For example, a set of previously used candidate models with the variables selected, showing both transformations for and interactions among the selected variables, is a form of instruction. Analysts can use this instruction to refine their choices.

Application of Design Science Approach

The wicked nature of the problem, the availability of a kernel theory to guide design, and the potential for an IT-based solution, make this an appropriate problem to view through the lens of design science. Design science represents a way of conceptualizing and executing the development of a new IT artifact using a theory-based approach. The use of justificatory (or kernel) theory is not only to provide guidance regarding design, but also to explain why the design works (Gregor and Jones, 2007).

Mandviwalla (2015) provides a helpful framework for conceptualizing theory-informed design of an IT artifact. He states that "kernel theory" can either influence the design properties or provide criteria for evaluating success. Kernel theory and the artifact come together in an iterative prototyping process, where the design is informed by theory, evaluated, and then refined. From this, an artifact emerges that provides new insight into the problem and whose characteristics can inform future designs.

Mandviwalla (2015) also lays out nine types of design science research projects, characterized by the project's relationship with kernel theory (when it is unknown, when it is appropriated, and when new theory is generated) and with the artifact (the identification of an existing artifact, the appropriation of an existing artifact, and the generation of a new artifact).

The use of feedforward as a guide for the design of a decision aid for model building most closely aligns with Mandviwalla's (2015) characterization of a "Type 8" project, where existing theory informs the development of a new artifact. This type of project involves the application of theory to create a new solution to an existing problem. The newly developed solution can be tested through an iterative prototyping process, looking at key metrics tied to the kernel theory – specifically, feedforward and outcome feedback as aids to the model selection process by lowering uncertainty and improving user self-efficacy.

Tool Design

The Underlying Dimensional Model Mart

The model formulation process requires access to a sophisticated body of knowledge that encompasses data, causal relationships, analytical modeling, and domain-specific organizational processes. Specifically, the modeler must have an understanding of:

1. Organizational processes
2. Data available within and outside the organization
3. Hypothesized relationships among the variables to be predicted (the *dependent* variables) and variables that influence the values of the dependent variables (the *independent* variables)
4. Mathematical representations of hypothesized relationships among the variables
5. Measures of model effectiveness.

Nontechnical modelers know as much, or even more, about items 1 and 2 as technical modelers, but tend to struggle with items 3, 4, and 5 in the above list. Corral et al. (2015) proposed a Model Management Warehouse, implemented as a dimensional document mart designed to help with these items (Figure 2 is a modified version of their schema). There are two main reasons why this model is helpful. First, each dimension maps to a major component of an analytical model: the modeling domain, possible dependent and independent variables, possible variable transformations, possible techniques to model the relationships among the variables, and possible measures of model effectiveness. This allows users to intuitively select model components. Second, the structure of the star schema enables the data modeler to pick the precise model for which he/she is searching from the mountain of models that are stored in the model warehouse, and to do so without knowing anything about model structures or statistics.

The data warehouse also contains a set of documents that explain the models with the specified dependent and independent variables, and a date dimension is included to enable modelers to retrieve models developed more recently. Note also that the effectiveness measures allow the modeler to assess the usefulness of the selected model.

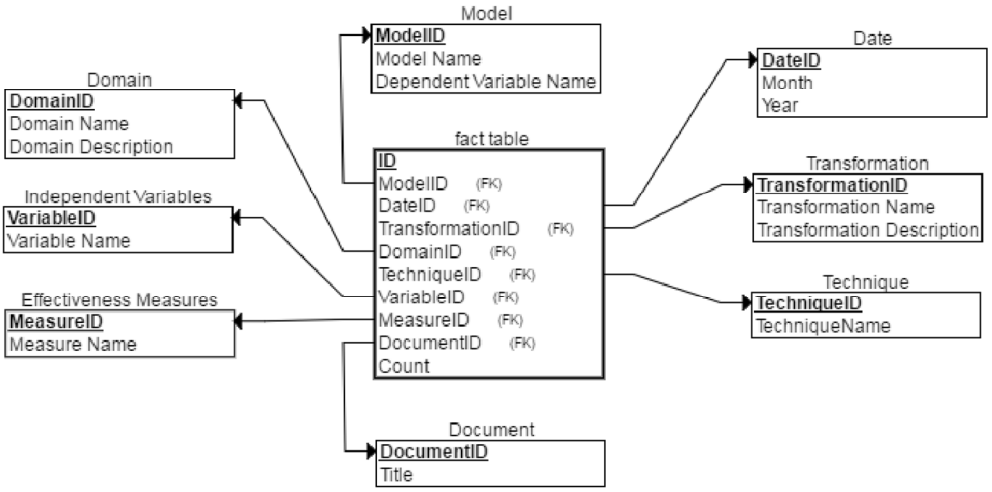

Figure 2. Schema for a Model Management Warehouse (adapted from Corral et al. 2015)

Instantiation of Prototype

Development Platform and Model Performance Metrics

Our solution helps the data modeler by providing a Visual Basic for Applications (VBA)-based interface to the dimensional model store through a Microsoft Excel Pivot Table. We developed a prototype model selection tool built in Excel using VBA and R as the analytics platform. The choice of these tools was driven by their ubiquity (Excel is everywhere and R is a well-supported, open source statistical analysis package) and low cost. The tool requires the following preconditions:

1. All users of the tool must rely on an identical data dictionary
2. The data set that meets the data dictionary requirements must be available in a comma-separated (.csv) file
3. All models must have a single outcome variable
4. The effectiveness of the models must be measureable using well-defined success criteria.
5. Users must have access to Excel and R, and the RScript command-line must be in the user's path.

We developed a working prototype to the dimensional model store developed by Corral et al. (2015) for the US Department of Labor (USDoL). The model warehouse contains benefits-exhaust models for 32 different states. The dependent variable is always a binary variable representing whether or not an unemployment insurance (UI) claimant will exhaust their UI benefits during a specific spell of unemployment. The independent variables represent different characteristics of the claimant, and may vary across state models. As an illustrative example, consider the scenario where a state analyst wants to improve upon the performance of their state's benefits-exhaust model by examining what other states have done.

We use two feedforward cues about model effectiveness to aid the model building process: the receiver operating characteristic (ROC) curve and the decile chart. These cues are not the only measures of model effectiveness, but are appropriate in this scenario because they address the performance of predicting a binary outcome. The ROC curve (see Figure 3 for an example) compares the true positive rate to the false positive rate for a predictive model. This provides a simple metric for model evaluation -- better performing models will have an ROC curve that hugs the vertical axis until it is very close to 1.0 (i.e., a low false positive rate and a high true positive rate). Such a curve would mean that it is possible to identify a high percent of the claimants that exhaust without misclassifying a high percent of non-exhasutees as exhaustees. In Figure 3, one can see that in order to correctly identify 80% of the claimants that exhaust their benefits, one must be willing to misclassify about 38% of non-exhaustees as exhaustees.

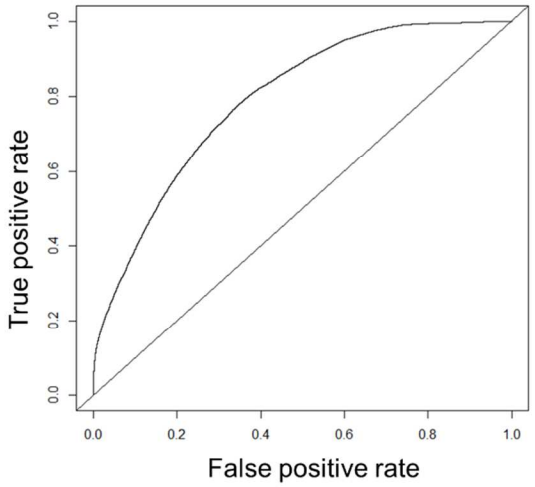

Figure 3. Sample ROC Curve

The ROC curve highlights the tradeoff between identifying an exhaustee and misclassifing a non-exhaustee. The decile chart (see Figure 4) provides more specificity with respect to the number and percent of non-exhauatees that must be mis-classified in order to identify a given number and percent of exhaustees. To construct a decile chart, the population of claimants is divided into deciles based on their predicted probability of exhaustion. For each decile, the chart shows the number and percent of claimants that exhausted and did not exhaust benefits. In Figure 4 one can see that if we classify everyone in the decile that had the highest predicted probability of exhaustion as an exhaustee, then we can identify 3,582 claimants that exhausted their benefits (17.7% of all exhaustees), but we also will misclassify 379 claimants that did not exhaust their benefits as exhaustees (1.96% of all non-exhaustees).

Cell Contents

Count
Chi-square contribution
Row Percent
Column Percent
Total Percent

Total Observations in Table: 39606

UTAH_EXHAUST	DECILE 1	2	3	4	5	6	7	8	9	10	Row Total
0	3830	3307	2544	2305	1893	1595	1348	1157	1020	379	19378
	1847.104	967.068	189.495	69.704	1.022	60.705	179.363	314.735	434.485	1254.114	
	19.765%	17.066%	13.128%	11.895%	9.769%	8.231%	6.956%	5.971%	5.264%	1.956%	48.927%
	96.693%	83.489%	64.226%	58.207%	47.803%	40.268%	34.040%	29.210%	25.758%	9.568%	
	9.670%	8.350%	6.423%	5.820%	4.780%	4.027%	3.404%	2.921%	2.575%	0.957%	
1	131	654	1417	1655	2067	2366	2612	2804	2940	3582	20228
	1769.487	926.430	181.533	66.775	0.979	58.154	171.826	301.509	416.220	1201.415	
	0.648%	3.233%	7.005%	8.182%	10.219%	11.697%	12.913%	13.862%	14.534%	17.708%	51.073%
	3.307%	16.511%	35.774%	41.793%	52.197%	59.732%	65.960%	70.790%	74.242%	90.432%	
	0.331%	1.651%	3.578%	4.179%	5.219%	5.974%	6.595%	7.080%	7.423%	9.044%	
Column Total	3961	3961	3961	3960	3960	3961	3960	3961	3960	3961	39606
	10.001%	10.001%	10.001%	9.998%	9.998%	10.001%	9.998%	10.001%	9.998%	10.001%	

Figure 4. Sample Decile Chart

An important feature of both the ROC curve and the decile chart is that they are readily understandable by non-statisticians and non-technical managers. They provide very intuitive measures of how well a model is performing. Different model performance metrics will be more appropriate for other types of models (models without a binary dependent variable), but those performance measures will not be effective at supporting self-service BI unless they are equally understandable to non-technical managers.

Prototype Walk-through

Upon opening the macro-enabled Excel file (*.xlsm), users see the Model Selection Interface dialog (see Figure 5). The "Select State Model" list box displays the names of all models stored in the warehouse. Selecting a model displays the list of independent variables (and variable transformations) used in that model. In Figure 5, the user selected the "Utah Statistical Model," displaying the ROC curve and the decile chart. By providing this information up-front, the tool reduces the uncertainty in the model-selection process. The user analyzes those model diagnostics, along with the set of independent variables in the model, to determine whether the model is likely to perform satisfactorily with their own data.

If the analyst wants to apply another state's model to her/his state's data, the analyst can click the "Run Model" button. The tool then asks the user to identify the location of the comma-separated (.csv) file containing the dataset, and builds and executes an R script that loads the data set, performs the data transformations, and runs the model with the user's data. The results are displayed in the Model Performance Evaluation interface in a second dialog (see Figure 6). This time, the ROC curve and decile chart serve as outcome feedback, allowing the analyst to validate her/his choice. Since the Utah data was used to populate the warehouse, the ROC curve and decile table in Figure 6 are identical to those in Figure 5 (in general, that will not be the case). When the user is confident that the accuracy of the model is satisfactory, the user may stop the model building process. The application saves the parameters of the

model and the R console output so that it is accessible outside of the session to be run against future data. However, the analyst can also choose to run a new model by returning to the model selection interface.

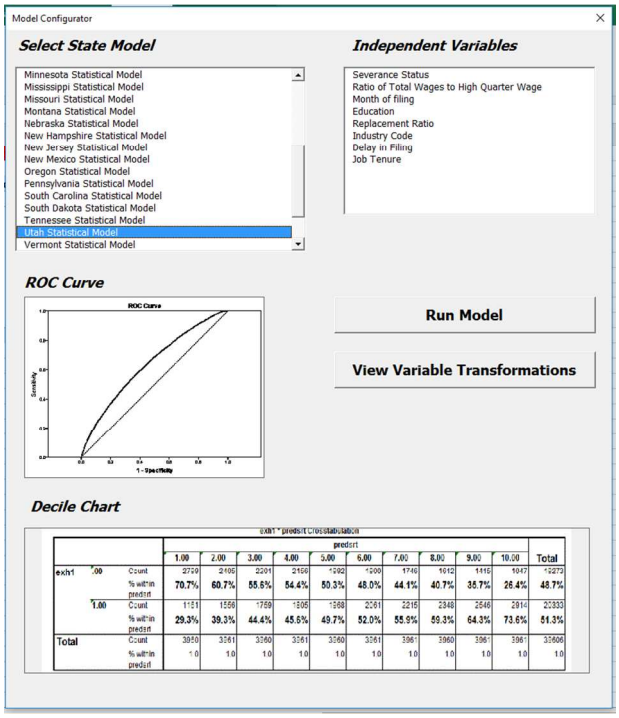

Figure 5. Model Selection Interface

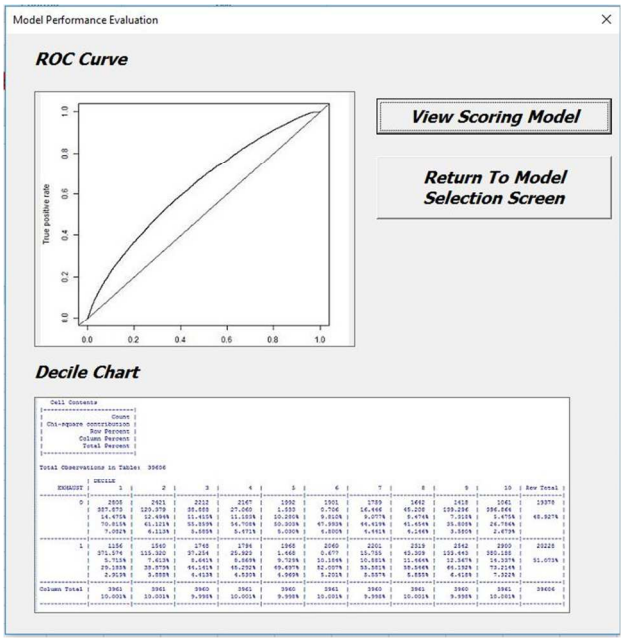

Figure 6. Model Performance Evaluation Interface

Future Work: An Enhanced Model Selection Interface

The working prototype described in the previous section provides a compelling proof-of-concept for self-service BI, demonstrating how our approach can help a non-technical analyst build, maintain, and use models. However, the feedforward cues and outcome feedback provided by that prototype were limited to the ROC curve and decile chart.

We are designing the next iteration of the interface to incorporate additional feedforward cues that will better inform the user's model selection process and provide more flexibility in variable choice. The new interface, like the current one, will allow the user to browse the USDOL's dimensional model warehouse. A "mock-up" of the new Model Selection Interface is shown in Figure 7. We chose C# as the platform because it works well with Microsoft Excel and can easily replace the current version's VBA interface.

In this enhanced design, the user will be able to browse by independent variables used across all models (feedforward), instead of viewing variables one model at a time. This allows us to incorporate several additional feedforward cues. First, the number of states that use a particular independent variable in a model is displayed next to that variable, along with the names of the states. For example, the variable "education" appears in three state models – Alabama, Utah, and Wyoming. Second, below the "Independent Variable List" listbox, the interface provides both an explanation for why the independent variable can be expected to have an impact on the dependent variable; and below the "States Using the Variable" listbox, the interface provides an explanation for why the independent variable is represented as a continuous, transformed continuous, or categorical variable. Taken together, these cues help the user assess whether an independent (predictor) variable is commonly used, if that variable is used in contexts (i.e., states) similar to their own, whether that independent variable is likely to have an impact on the dependent variable, and whether that independent variable should be modeled as a continuous, transformed continuous, or categorical variable. As before, the feedforward cues of the ROC curve and the decile chart for the model are based on the state selected in the "States using…" listbox. These additional feedforward cues (in addition to the ROC curve and the decile chart) provide increased support to the analysts in the model-selection process, giving them more information to reduce their uncertainty and increase their capability in choosing the right model for their state's data.

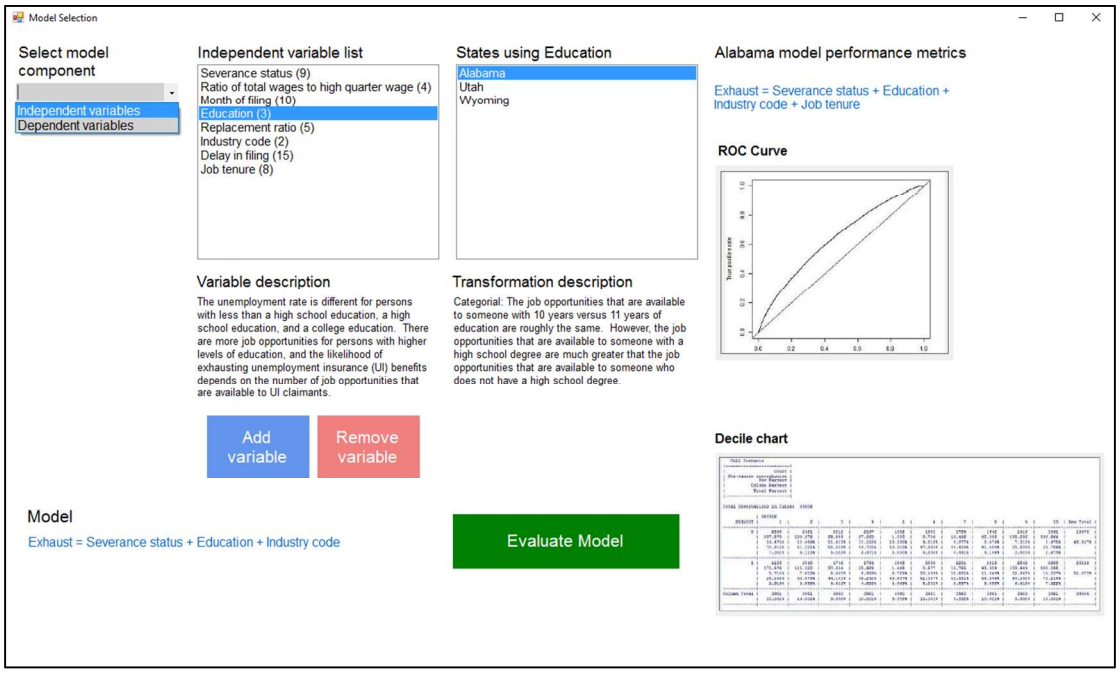

Figure 7. Future Model Selection Interface

In this next version of the tool, the user also will be able to build an original model by adding variables instead of simply applying an existing model to their data. Through the "Add variable" and "Remove variable buttons" the user can create a model that contains any independent variable from any state.

When users believe they have created an appropriate model, they can click the "Evaluate model" button and the Model Performance Evaluation Interface will appear (see Figure 8). The content of this dialog will be similar to the current version, displaying outcome feedback through the ROC curve and decile chart for the user's model applied to her/his state's data. If they click "Accept model," it will save the output and the R code to a file. If they click "Keep working," the application will return to the Model Selection Interface.

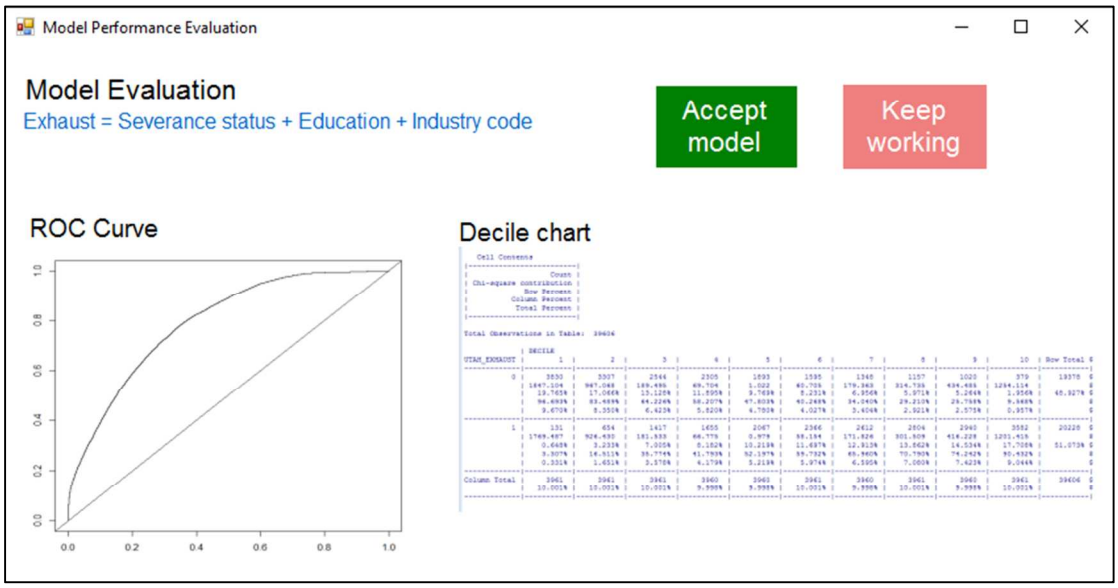

Figure 8. Future Model Performance Evaluation Interface

We will evaluate the efficacy of the proposed interface through a field experiment at the Division of Fiscal and Actuarial Services (DFAS) of the Office of Unemployment Insurance. The DFAS provides technical assistance to the states to help them meet Federal requirements regarding the Unemployment Insurance (UI) program. For example, states must refer UI claimants likely to exhaust their benefits to reemployment services. Each state develops its own model using statistical profiling models to identify claimants. DFAS assists the states by providing support for model building, including training and tracking states' models.

This group is an ideal population for testing our model selection tool because they often do not have extensive backgrounds in statistics and share the same underlying modeling task. We will give the 53 analysts (about one per state) the task of revising their existing benefits exhaust model – many of these models haven't been updated since 2007. Each subject will receive training in basic modeling, but half the subjects will also receive our proposed interface. All subjects will also receive a uniformly constructed data set containing 71 potentially relevant predictors. They will submit their completed models to the research team, who will rate the model based on correctness (i.e., did they choose the best performing model). Each subject will also complete a questionnaire measuring their satisfaction with the model-building process (ie.., Delone and McLean, 1992) and cognitive effort (i.e., Zijlstra and Van Doorn, 1985).

Conclusions

Nontechnical analysts often struggle with building models. Uncertainty with both the model building process and a lack of expertise in the use of statistical software packages acts as a barrier for nontechnical analysts. Corral et al.'s (2015) dimensional model warehouse significantly reduces barriers resulting from uncertainty by providing analysts with examples of models developed by other practitioners for situations similar to their own. These examples not only suggest appropriate independent variables, but also provide suggestions for how to model the functional forms for the relationships between the independent variables and the dependent variable, and how to assess the effectiveness of the models.

We build on Corral et al. (2015) by developing a prototype to assist users in model-building. We integrate feedforward into the prototype in several ways, including (1) information about frequency of use of the independent variables, (2) the context in which practitioners used an independent variable, (3) the reason

why an independent variable can be expected to have an effect on the dependent variable, and (4) the reason why an independent variable should be modeled using a specific functional form. Both inexpensive model building software and feedforward are required to reduce barriers stemming from lack of familiarity with and availability of statistical software packages, the cost of statistical software packages, uncertainty with respect to statistics and the model-building process, and uncertainty with respect to model evaluation. The prototype presented in this paper eliminates these barriers. By seamlessly integrating with the dimensional model management warehouse proposed by Corral et al. (2015), it provides support for the entire modeling life cycle, and has the potential to create a much more widespread use of business intelligence.

References

Balzer, W. K., Doherty, M. E., and O'Connor, R. Jr. 1989. "Effects of cognitive feedback on performance," *Psychological Bulletin* (106:3), pp. 410-433.

Björkman, M. 1972. "Feedforward and feedback as determiners of knowledge and policy: notes on a neglected issue," *Scandinavian Journal of Psychology* (13:3), pp. 152-158.

Box, G. E. P., and Draper, N. R. 1987. *Empirical Model Building and Response Surfaces*, New York, Wiley.

Chenoweth, T., Dowling, K. L., and St. Louis, R. D. 2004. "Convincing DSS users that complex models are worth the effort," *Decision Support Systems* (37:1), pp. 71-82.

Corral, K., Schuff, D., Schymik, G., and St. Louis, R. D. 2015. "Enabling Self-Service BI through a Dimensional Model Warehouse," in *Twenty-first Americas Conference on Information Systems*, Puerto Rico.

Delone, W.H., and McLean, E.R. 1992. "Information Systems Success: The Quest for the Dependent Variable," *Information Systems Research* (3:1), pp. 60-95.

Dhaliwal, J. S., and Benbasat, I. 1996. "The use and effects of knowledge-based system explanations: theoretical foundations and a framework for empirical evaluation," *Information Systems Research* (7:3), pp. 342-362.

Gregor, S., and Jones, D. 2007. "The anatomy of a design theory," *Journal of the Association for Information Systems* (8:5), pp. 312-335.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 77-105.

Hsu, M. K., Wang, S. W., and Chiu, K. K. 2009. "Computer Attitude, Statistical Anxiety and Self-Efficacy on Statistical Software Adoption Behavior: An empirircal study of online MBA learners," *Computers in Human Behavior* (25:2), pp. 412-420.

Jarupathirun, A., and Zahedi, F. M. 2007. "Exploring the Influence of Perceptual Factors in the Success of Web-based Spatial DSS," *Decision Support Systems* (43:3), pp. 933-951.

Johnson, R. D., Li Y., and Dulebohn, J. H. 2016. "Unsuccessful Performance and Future Computer Self-Efficacy Estimations: Attributions and Generalizations to Other Software Applications," *Journal of Organizational and End User Computing* (28:1), pp. 1-13.

Lindell, M. K. 1976. "Cognitive and outcome feedback in multiple-cue probability learning tasks," *Journal of Experimental Psychology: Human Learning and Memory* (2:6), pp. 739-743.

Logi Analytics. *2014 State of Self-Service BI Report*. www.logianalytics.com/bi-trends/infographic-2014-state-of-self-service-bi-report/. Accessed 8 February 2015.

Mandviwalla, M. "Generating and justifying design theory," *Journal of the Association for Information Systems* (16:5), pp. 314-344.

Rohanizadeh, S., and Moghadam, M. 2009. "A Proposed Data Mining Methodology and its Application to Industrial Procedures," *Journal of Industrial Engineering* (4:4), pp. 37-50.

SAS Institute. 1998. Data Mining and the Case for Sampling. *SAS Institute Best Practices Paper*, Carey, NC.

Sengupta, K., and Abdel-Hamid, T. K. 1993. "Alternative conceptions of feedback in dynamic decision environments: an experimental investigation," *Management Science* (39:4) pp. 411-428.

Sterman, J. D. 1989. "Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment," *Management Science* (35:3) pp. 321-339.

Zijlstra, F.R.H., and Van Doorn, L. 1985. "The Construction of a Scale to Measure Perceived Effort." Technical Report. Delft University of Technology.