Print

Summary: Lesson 6: Model Building and Scoring Prediction

This summary contains topic summaries, syntax, and sample programs.

Topic Summaries

To go to the movie where you learned a task or concept, select a link.

Introduction to Predictive Modeling

Before you can predict values, you must first build a predictive model. Predictive modeling uses historical data to predict future outcomes. These predictions can then be used to make sound strategic decisions for the future. The process of building and scoring a predictive model has two main parts: building the predictive model on existing data, and then deploying the model to make predictions on new data (using a process called scoring). A predictive model consists of either a formula or rules based on a set of input variables that are most likely to predict the values of a target variable. Some common business applications of predictive modeling are: target marketing, credit scoring, and fraud detection.

Whether you are doing predictive modeling or inferential modeling, you want to select a model that generalizes well – that is, the model that best fits the entire population. You assume that a sample that is used to fit the model is representative of the population. However, any given sample typically has idiosyncracies that are not found in the population. The model that best fits a sample and the population is the model that has the right complexity.

An overly complex model might be too flexible. This leads to overfitting – that is, accommodating nuances of the **random noise** (the chance relationships) in the particular sample. Overfitting leads to models that have higher variance when applied to a population. For regression, including more terms in the model increases complexity.

On the other hand, an insufficiently complex model might not be flexible enough. This leads to underfitting – that is, systematically missing the signal (the true relationships). This leads to **biased inferences**, which are inferences that are not true of the population. A model with just enough complexity—which also means just enough flexibility—gives the best generalization. The important thing to realize is that there is no one perfect model; there is always a balance between too much flexibility (overfitting) and not enough flexibility (underfitting).

The first part of the predictive modeling process is building the model. There are two steps to building the model: fitting the model and then assessing model performance in order to select the model that will be deployed. To build a predictive model, a method called honest assessment is commonly used to ensure that the best model is selected. Honest assessment involves partitioning (that is, splitting) the available data—typically into two data sets: a training data set and a validation data set. Both data sets contain the inputs and the target. The training data set is used to fit the model. In the training data set, an observation is called a **training case**. Other synonyms for observation are example, instance, and record. The validation data set is a **holdout sample** that is used to assess model performance and select the model that has the best generalization. Honest assessment means that the assessment is done on a different data set than the one that was used to build the model. Using a holdout sample is an honest way of assessing how well the model generalizes to the population. Sometimes, the data is partitioned into three data sets. The third data set is a test data set that is used to perform a final test on the model before the model is used for scoring.

PROC GLMSELECT can build a model using honest assessment with a holdout data set (that is, a validation data set) in two ways. The method that you use depends on the state of your data before model building begins. If your data is already partitioned into a training data set and a validation data set, you can simply reference both data sets in the procedure. If you start with a single data set, PROC GLMSELECT can partition the data for you. The PARTITION statement specifies how PROC GLMSELECT logically partitions the cases in the input data set into holdout samples for model validation and, if desired, testing. You use the FRACTION option to specify the fraction (that is, the proportion) of cases in the input data set that are randomly assigned a testing role and a validation role.

The PARTITION statement requires a pseudo-random number generator to start the random selection process, and an integer is required to start that process. If you need to be able to reproduce your results in the future, you specify an integer that is greater than zero in the SEED= option. Then, whenever you run the PROC GLMSELECT step using that seed value, the pseudo-random selection process is replicated and you will get the same results. In most situations, it is recommended that you use the SEED= option and specify an integer greater than zero.

Scoring Predictive Models

After you build a predictive model, you are ready to deploy the model To score new data—referred to here as scoring data—you can use PROC GLMSELECT and PROC PLM. Before you start using a newly built model to score data, some preparation of the model, the data, or both, is usually required.

For example, in some applications, such as fraud detection, the model might need to be integrated into an online monitoring system before it can be deployed. It is essential for the scoring data to be comparable to the training data and validation data that were used to build the model. The same modifications that were made to the training data must be made to the validation data before validating the model and to the scoring data before scoring.

When you score, you do not rerun the algorithm that was used to build the model. Instead, you apply the score code—that is, the equations obtained from the final model—to the scoring data. There are three methods for scoring your data:

Method	Advantages and Disadvantages
Method 1 • Use a SCORE statement in PROC GLMSELECT.	The first method is useful because you can build and score a model in one step. However, this method is inefficient if you want to score more than once or use a large data set to build a model. With this method, the model must be built from the training data each time the program is run.
Method 2 • Use a STORE statement in PROC GLMSELECT. • Use a SCORE statement in PROC PLM.	The second method enables you to build the model only once, along with an item store, using PROC GLMSELECT. You can then use PROC PLM to score new data using the item store. Separating the code for model building and model scoring is especially helpful if your model is based on a very large training data set or if you want to score more than once. Potential probles with this method are that others might not be able to use this code with earlier versions of SAS or you might not want to share the entire item store.
Method 3 • Use a STORE statement in PROC GLMSELECT. • Use a CODE statement in PROC PLM to output SAS code. • Use a DATA step for scoring.	The third method uses PROC PLM to write detailed scoring code, based on the item store, that is compatible with earlier versions of SAS. You can provide this code to others without having to share other information that is in the item store. The DATA step is then used for scoring.

Syntax

To go to the movie where you learned a statement or option, select a link.

```
PROC GLMSELECT DATA=training-data-set
                        VALDATA=validation-data-set;
      MODEL target(s)=input(s) </ options>;
RUN;

PROC GLMSELECT DATA=training-data-set
                     <SEED=number>;
      MODEL target(s)=input(s) </ options>;
      PARTITION FRACTION(<TEST=fraction><VALIDATE=fraction>);
RUN;
```

Sample Programs

Building a Predictive Model

```
%let interval=Gr_Liv_Area Basement_Area Garage_Area Deck_Porch_Area
              Lot_Area Age_Sold Bedroom_AbvGr Total_Bathroom;
%let categorical=House_Style2 Overall_Qual2 Overall_Cond2 Fireplaces
                 Season_Sold Garage_Type_2 Foundation_2 Heating_QC
                 Masonry_Veneer Lot_Shape_2 Central_Air;

ods graphics;

proc glmselect data=statdata.ameshousing3
               plots=all
               valdata=statdata.ameshousing4;
   class &categorical / param=glm ref=first;
   model SalePrice=&categorical &interval /
                  selection=backward
                  select=sbc
                  choose=validate;
   store out=work.amesstore;
   title "Selecting the Best Model using Honest Assessment";
run;


title;
```

Scoring Data

Note:

- In your current SAS session, you must first run the code from the previous demonstration (shown above) before running the code below.
- Replace ***my-file-path*** with the path to your course practice files.

```
proc plm restore=work.amesstore;
   score data=statdata.ameshousing4 out=scored;
   code file="my-file-path/scoring.sas";
run;

data scored2;
   set statdata.ameshousing4;
   %include "my-file-path/scoring.sas";
run;

proc compare base=scored compare=scored2 criterion=0.0001;
   var Predicted;
   with P_SalePrice;
run;
```

Statistics I: Introduction to ANOVA, Regression, and Logistic Regression

Close