SESUG Paper 242-2019

Advanced Statistical Modeling within Machine Learning and Big Data Analytics

Niloofar Ramezani, George Mason University

ABSTRACT

Machine learning, big data, and high dimensional data are very important topics of recent years which some consider as the wave of the future. Therefore, it is crucial to learn about appropriate statistical techniques to be applied within these relatively new topics. Since statistical approaches have been established for many years and their efficiency have already been evaluated, they can benefit newer data-related fields.

Machine learning is an important topic these days as it involves a set of many different methods and algorithms that are suited to answer diverse research questions. Thus, choosing a proper algorithm is a critical step in the machine learning process to ensure it truly fits the solution proposed in answering a problem at hand (Segal, 2004). To better understand and select machine learning algorithms when dealing with real data, it is helpful to understand them within the framework of statistics and divide them into two main groups based on the nature and dimension of data and type of outcome variables. Depending on the nature of variables, classification and regression machine learning methods are discussed here and appropriate statistical techniques for modeling them are presented in this study for low and high dimensional data. Within this paper, these two types are differentiated and related algorithms and statistical techniques, as well as the SAS® procedures, are discussed in order to answer real world problems including high dimensional data scenarios. These procedures include, but are not limited to, PROC HPFOREST, PROC FACTOR, PROC CLUSTER, PROC LOGISTIC, PROC GENMOD, PROC REG, PROC GLM, PROC PRINCOMP, and GLMSELECT.

INTRODUCTION

To better understand machine learning algorithms, data analytics, and when each algorithm and method needs to be applied to low or high dimensions of data, one needs to understand them within the framework of statistics and choose the appropriate modeling techniques based on the number of variables compared to the number of subjects and the format of the response variables. In many scenarios, we deal with low-dimensional data. However, if the number of variables or features is higher than the number of samples, the data are considered high-dimensional and require different analytical methods and dimension reduction approaches. Regardless of the dimension of our data, depending on the type of the outcome, there are two types of machine learning methods; classification and regression for categorical and continuous response variables, respectively. Within this paper, we will differentiate these two types first within the low-dimensional data scenarios and mention related algorithms and statistical techniques that can be used to answer real world problems. Then the concept of high-dimensional data and some of the methods that can appropriately handle such data will be discussed. For each of the aforementioned methods, the SAS procedures and options are discussed with the goal to gather in one paper different SAS programming tools for classification and regression algorithms of low and high-dimensional data.

This paper adopts an educational approach for different levels of learners, to first familiarize the readers with different data scenarios and the proper statistical and machine learning algorithms to appropriately model the data and then discuss the SAS options to implement such methods. Author believes the first step in ensuring the correctness and reliability of any quantitative analysis within machine learning, statistics, data analytics engineering, or any other field is to explore and learn about the characteristics of the data, and then choose the proper statistical and machine learning approach for data modeling. Based on years of experience in teaching and statistical consulting, author strongly believes in the importance of strengthening the foundations of statistical knowledge before rushing into programming and producing output to ensure the accuracy of the results. This paper aims to educate the readers with proper modeling options while exploring different SAS procedures to perform such models on the real-world data.

LOW-DIMENTIONAL DATA

When dealing with lower dimensions of data, it is important to understand what algorithms are appropriate to use for modeling different types of outcome variables. Classification models such as logistic regression approaches are used for categorical responses and regression models are used for continuous outcome variables. Here, a few of these models are introduced and the SAS procedures to apply them to real cross-sectional data are explained.

CLASSIFICATION

When modeling categorical outcomes, the assumption of normality is not met and therefore, the linear regression approaches cannot be fitted to such responses anymore. Moreover, the relationship between the outcome and input variables, or predictors, is no longer linear. For such circumstances, binary, multinomial, and ordinal logistic regression approaches are a few of the robust methods which enable us to model the relationship between non-normal categorical response and the predictor variables. Ramezani and Ramezani (2016) discussed multiple methods and algorithms as well as examples to fit predictive models to categorical non-normal target variables. Consult Agresti (2007) to learn more details regarding such techniques.

Binary classification in machine learning will model binary outcomes, which are outcomes with only two categories, while multi-label classification covers everything else such as customer segmentation, audio and image categorization, and text analysis for mining customer sentiment (James, Witten, Hastie & Tibshirani, 2013). Generalized linear models (GLM) and classification techniques assist researchers to model discrete binary and categorical outcome or target variables. Binary logistic regression as well as multinomial and ordinal logistic regression models are introduced and briefly discussed here. These models can help applied researchers and practitioners in modeling binary and categorical response variables. While discussing ordinal logistic models, three types of logit functions are briefly discussed. These logit functions are cumulative logit, adjacent-categories logit, and continuation-ratio logit.

Binary Logistic Regression

Logistic regression allows building a predictive model between a categorical target variable and multiple input variables. Logistic regression, which is a GLM, helps predicting the presence or absence of a characteristic or outcome based on values of one or a set of input variables. The power of logistic regression over a liner model is that, through the addition of an appropriate link function to an ordinary linear model, the target or response variable does not need to follow a normal distribution while the input variables may be any combination of discrete and continuous variables. When the dependent variable is binary following a binomial distribution, the logit link function is widely used within the GLM, making the predictive model a binary logistic regression (Atkinson & Massari, 1998). Logistic regression coefficients can be used to estimate ratios for each of the independent variables in the model (Lee, 2005).

In Logistic Regression, instead of measuring the average change of the response, which is the case in linear regression models, probability of the outcome is measured by the odds of occurrence of an event. Change in probability of the occurrence of one category of the response (Y) is not constant (linear) with constant changes in each predictor (X). This means the probability of success ($Y = 1$), given X, is a non-linear function, commonly a logistic function. As mentioned above, the most common form of logistic regression uses the logit link function so it is easily understandable to show the logistic regression equation as

$$logit\ (p) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k,$$

where p is the probability of success and $logit(p)$ is the natural logarithm of probability of success over probability of failure. PROC LOGISTIC and PROC GENMOD are two SAS procedures that can be used to fit a binary logistic regression model. The call to PROC LOGISTIC is displayed as:

```
PROC LOGISTIC DATA=(mention the dataset name here);
      CLASS (list the categorical variables here)/PARAM=REF;
      MODEL (specify the response variable here) (EVENT='1')=(list all the
      predictor variables used here)/LACKFIT CORRB;
RUN;
```

The CLASS statement is to specify all categorical variables used in the model. PARAM=REF option will result in creating dummy variables for a categorical variable as oppose to the default within the LOGISTIC procedure which is effect coding. MODEL statement is where the response variable is specified followed by the predictors after an equal sign. Using the EVENT='1' option in the model statement, defines the event of interest to be modeled as 1, which generally is used to show the occurrence of the event. 1 can be replaced with 0 if modeling the nonoccurrence is of interest. Modeling category 1 as the event of interest instead of the nonoccurrence category, 0, is also possible by using the DESCENDING option in the PROC LOGISTIC statement. LACKFIT is another option that can be added within the model statement to test the goodness of fit of the model using Hosmer-Lemeshow test. This test will check to see if there is any difference between the observed and predicted values of the response variable.
PROC GENMOD is another SAS procedure that can be used to perform a binary logistic regression as below:

```
PROC GENMOD DATA=(mention the dataset name here) DESCENDING;
      CLASS (list the categorical variables here);
      MODEL (specify the response variable here)=(list all the predictor
      variables here)/DIST=BIN CORRB;
RUN;
```

Within this procedure specifying DIST=BIN is needed to impose performing a binary logistic regression model. This is because the GENMOD procedure can be used for performing a handful of other models as well; therefore, the distribution of the response needs to be specified to define the type of the performed model. Within GENMOD procedure specifying that we want to model 1 as the event of interest instead of 0 for the dependent variable can be done by simply adding the DESCENDING option to the procedure.

Multinomial and Ordinal Logistic Regression

The multinomial logistic regression model is an extension of the binomial logistic regression model and should be used when the dependent variable has more than two nominal (unordered) categories. When the response categories are not ordered, procedures such as PROC LOGISTIC, with the specification of LINK=GLOGIT option in the MODEL statement, can be used to fit a multinomial logistic regression. A sample code is shown below:

```
PROC LOGISTIC DATA=(mention the dataset name here);
      CLASS (specify the response variable here)(REF="1") (list the
      categorical predictor variables here)/PARAM=REF;
      MODEL (specify the response variable here)=(list the predictor
      variables here)/LINK=GLOGIT;
RUN;
```

PROC SURVEYLOGISTIC with the specification of LINK=GLOGIT option can also be used to perform the same analysis. The GLIMMIX and HPGENSELECT procedures can also be used to fit this model by specifying the DIST=MULT and LINK=GLOGIT options in the MODEL statement. All of the aforementioned procedures fit the model using maximum likelihood estimation. PROC CATMOD can also be used to fit the multinomial logistic model using maximum likelihood by default or using weighted least squares after specifying the WLS option.

When the response categories are ordered, a multinomial regression model still can be used but in that case, some information about the way the response categories are ordered will be lost (Agresti, 2007). Instead, when modeling data with ranked multiple response outcomes, an ordinal logistic regression model is recommended which preserves the ordering related information, but the model is slightly more complex (Ramezani, 2015). Due to this complexity, assumptions validation, and limitations of modeling options offered by statistical packages, the use of such models for ordered information is still infrequent.
Therefore, it is important to highlight the importance of such models that will result in robust estimates when dealing with ordinal target variables.
In ordinal logistic regression models, there are different logit functions that should be used within the GLM framework. Cumulative logit, adjacent–categories logit, and continuation ratio logit are briefly explained below, based on the description of Ramezani (2015), and notations used in Agresti (2007). Table 1 differentiates these logit functions based on their respective equations.

The cumulative logit function used in the ordinal logistic models basically models categories $\leq j$ versus categories $> j$, where j is the cut-off point category decided by the researcher based on the research question (Hosmer & Lemeshow, 2013). To fit such model with a cumulative logit function, PROC LOGISTIC and PROC GENMOD may be used as below:

```
PROC LOGISTIC DATA=(mention the dataset name here);
      CLASS (specify the response variable here)(REF="1") (list the
      categorical predictor variables here)/PARAM=REF;
      MODEL (specify the response variable here)=(list the predictor
      variables here)/LINK=CLOGIT SCALE=NONE AGGREGATE RSQ LACKFIT;
RUN;
```

Within the model statement of the PROC LOGISTIC, LINK=CLOGIT is added to specify the cumulative logit link function.

PROC GENMOD may also be used to fit the same model as below:

```
PROC GENMOD DATA= (mention the dataset name here) RORDER=data DESCENDING;
      CLASS (specify the response variables here)(REF="1") (list the
      categorical predictor variables here);
      MODEL (specify the response variables here)=(list the predictor
      variables here)/DIST=MULTINOMIAL LINK=CUMLOGIT;
RUN;
```

Within the MODEL statement of the PROC GENMOD, DIST=MULTINOMIAL is added to enforce the use of the multinomial distribution for the categorical outcome variable and the LINK=CUMLOGIT specifies the use of the cumulative logit link function in the ordinal logistic regression model.

The adjacent-categories logit function used in ordinal logistic models is to model two adjacent categories**.** Within this model, only adjacent categories will be used within odds of the model. This will result in using local odds ratios for interpretations, whereas within the cumulative logit models, the entire response scale is used for the model and therefore, cumulative odds ratio is used for interpreting the results.
Fitting an ordinal logistic regression with adjacent categories logit function in SAS is not as straight forward as when cumulative logit link is used. There still is not a SAS built-in procedure for this type of analysis. Requiring more effort, PROC NLMIXED can be used to perform the adjacent categories logit model. The likelihood functions need to be defined within the NLMIXED procedure, which can be time consuming specially in the presence of several independent variables in the model. Using PROC CATMOD to perform this type of analysis is also recommended in some books including Allison (2012) but it causes some issues in the output reported by this procedure (Ramezani, 2015).

The continuation-ratio logit function can also be used in ordinal multinomial logistic models**.** This model is useful when a sequential mechanism determines the response outcome (Agresti, 2007). Mechanisms like survival through various age periods of subjects would be suitable for such models. For more details and examples about the application of these models to real data, see Ramezani (2016).
Fitting an ordinal logistic regression with continuation-ratio logit function is not easy either due to not having a built in procedure in SAS to perform this type of analysis. There exists some issues when running models using a continuation-ratio logit function using PROC CATMOD. Agresti (2013) suggests using PROC GENMOD for the continuation ratio logit models that performs better than PROC CATMOD, yet is not easy to use. Another option when running the continuation ratio model is using PROC LOGISTIC and then restructure the original dataset to create binary response (Allison, 2012). Having this new binary outcome, PROC LOGISTIC produces can provide the same results as NLMIXED. Within this procedure the PARAM=GLM coding in the CLASS statement should be used (High, 2013).

Table 1, which its extended version can be found in Ramezani and Ramezani (2016), summarizes multinomial and ordinal logistic models and the SAS procedures that can be used to fit such models to real data.

Logistic Models	Logit Function	SAS Procedure
Multinomial Logistic Regression	$logit(P) = log\left(\frac{P(Y=j)}{P(Y=J)}\right)$	PROC LOGISTIC & PROC GENMOD (link=glogit)
Ordinal Logistic Regression (Cumulative Logit)	$logit\ (P) = log\left(\frac{P(Y\leq j)}{P(Y>j)}\right)$	PROC LOGISTIC & PROC GENMOD (link=clogit)
Ordinal Logistic Regression (Adjacent Categories Logit)	$logit\ (P) = log\left(\frac{P(Y=j)}{P(Y=j+1)}\right)$	PROC NLMIXED & PROC CATMOD
Ordinal Logistic Regression (Continuation Ratio Logit)	$logit\ (P) = log\left(\frac{P(Y=j)}{P(Y\geq j+1)}\right)$	PROC GENMOD & PROC CATMOD

Table 1. Logistic Regression Models: Different Logit Functions and Respective SAS Procedures

REGRESSION

When modeling continuous data and outcome measures, regression approaches and algorithms can be fitted. Linear regression is by far the simplest and most popular example of a regression algorithm used in different fields. Linear regression is widely used in supervised machine learning problems and focuses on regression problems. Every machine learning problem can be considered as an optimization problem where we want to find either a maximum or a minimum of a specific function of input and target variables. This function, which is usually referred to as a loss function, is defined for every machine learning algorithm. A regression function can be optimized using the same system in linear programming; therefore, a linear regression is generally considered as an optimization problem. Regression approaches are helpful with investigating the vast amounts of data and establishing the relationships that exist among different variables in a model. Through linear regression models, we can use one or multiple input variables to predict the expected value of a target response variable. This is done by fitting a line to the data, which is called regression line if it is the best and most optimal fit, then by using the estimates of the coefficients of the regression model and considering the characteristics of this line, the average response values can be predicted. This is achievable by executing an iterative process that updates the coefficients or parameters of the model at every step by reducing the loss function as much as possible. Once the procedure reaches the minimum of the loss function, the iterative process is considered completed and estimates of the parameters can be extracted for prediction. Least square estimation is the most common algorithm, and one of the most efficient techniques, commonly used to minimize the loss function and estimate the best coefficients for the regression model (Montgomery, Peck & Vining, 2012). To predict the response, within simple linear regression model, we use only one predictor or input variable while within multiple linear regression, more input variables can be used. In general and conditioned on using predictors that have a strong correlation with the response variable, multiple predictors together can explain a higher amount of the variation of the response. A multiple linear regression model can be written as

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \epsilon,$$

where Y is the response variable, $X_1, \ldots, X_k$ are k predictor or input variables ($k > 0$), $\beta_0, \ldots, \beta_k$ are the model parameters, and ϵ is a random error term. Different procedures such as PROC REG and PROC GLM in SAS can help users to fit a simple or multiple linear regression model to continuous responses. PROC REG may be used as below:

```
PROC REG DATA=(mention the dataset name here);
      MODEL (specify the response variable here)=(list the continuous
      predictor variables here);
RUN ;
```

The REG procedure does not support categorical predictors directly and so does not have a CLASS statement to specify classification or categorical predictor variables. Requiring some effort, users have to recode them into dummy variables, a series of 0 and 1 values, and use them in the model. Therefore, the number of dummy variables created to be used in the MODEL statement will be one less than the number of levels for each categorical input variable. For instance, a binary variable such as gender will be recoded to a simple 0 and 1 and then listed among the continuous predictor variables in the MODEL statement. A three-level categorical variable becomes two variables, and so on.

In the presence of several predictors, to avoid overfitting the regression model by including all of the predictors in the model, performing variable selection procedure is recommended. Through variable selection, variables that are playing a more important role in explaining the variation of the response variable are selected to be used in the regression model. There are multiple variable selection methods that can be performed within a linear regression such as all subset selection, forward selection, backward selection, and stepwise selection. If one wants to apply a stepwise variable selection procedure while fitting the regression model to the data, SELECTION=stepwise can be added within the MODEL statement after specifying the model, while separating them with a forward slash. Other variable selection methods such as backward and forward can be selected as well.

The manual recoding of the categorical variables within a REG procedure is analogous to the reference cell recoding that can be used in PROC GLM. The only issue with this dummy coding followed by the REG procedure is when variable selection methods are imposed. This is because sometimes part of a categorical variable can be selected through the variable selection procedure and part of it is left out of the list of the effective variables. PROC GLM makes it easier to include categorical predictor variables in a linear regression model, and avoid the previously mentioned issues if variable selection needs to be enforced. It can be fitted as below:

```
PROC GLM DATA=(mention the dataset name here);
      CLASS (list the categorical predictor variables here);
      MODEL (specify the response variable here)=(list the predictor
      variables here)/SOLUTION CLPARM;
RUN ;
```

The specification of the SOLUTION within the MODEL statement will help producing a solution to the normal equations, meaning the parameter estimates of the regression model will be printed in the output. PROC GLM displays a solution by default when the regression model involves only continuous predictor variables. Therefore, this option needs to be added only when there are categorical predictor variables in the regression model and seeing the solution for models with classification effects are of interest.

The inclusion of the CLPARM is recommended when the SOLUTION option is also specified. This option will produce confidence limits for the parameter estimates the results of all ESTIMATE statements.

One problem that may arise when fitting multiple linear regression to a number of predictors is when there exists correlation among the predictors. This causes a problem that is referred to as multicollinearity, which can make it more difficult to trust the results of the regression model and the accuracy of the estimations. Checking the correlation matrix of the variables is one option to see if there exist strong correlations among predictors. The other option is to check for the variance inflation factor (VIF). Based on the magnitude of the VIF per predictor, one can decide if the predictor is triggering a minor, moderate or extreme multicollinearity issue. The VIF values can easily be printed by adding a VIF option to the MODEL statement within a REG procedure as below:

```
PROC REG DATA=(mention the dataset name here);
      MODEL (specify the response variable here)=(list the continuous
      predictor variables here)/VIF;
RUN;
```

Another important aspect to consider when fitting linear regression models is checking for the assumptions of the model, which can be found in any regression reference (Montgomery et al., 2012). The GLM procedure automatically produces the residual plots that can help with diagnosing the model and can expose some assumptions violation problems. To see a list of all plots, after activating the ODC GRAPHICS, PLOTS=all can be added to the PROC GLM statement. How to read these plots is not the

main concern of this paper but certain patterns and trends in the residual diagnostics can show some assumption violations. As explained in detail in Montgomery et al. (2012), transformation of the response variable is sometimes needed to satisfy certain assumptions. Transformations are typically used to help inducing the homogeneous or constant variance assumption or to transform a nonlinear model into an approximately linear model. The method can also be applied to change multiplicative effects into additive effects using transformations such as natural log.

To check for the model assumptions first, the ODS GRAPHICS ON is added before running the GLM procedure to print the created graphics. When this option is used, it should be closed at the end of the procedure by typing the ODS GRAPHICS OFF, after the RUN statement. Furthermore, the OUTPUT statement is added to the GLM procedure to create a new SAS data set that saves diagnostic measures calculated after fitting the model. At least one specification within this statement is required. Within the OUTPUT statement OUT= (name) can be added to gives the name of the new data set. If OUT and a name is not specified, SAS uses its default to give the data set a name. An example SAS code is provided below with multiple options added to the OUTPUT statement.

```
ODS GRAPHICS ON;
      PROC GLM DATA=(mention the dataset name here) PLOTS=all;
            CLASS (list the categorical predictor variables here);
            MODEL (specify the response variable here)=(list the predictor
            variables here);
            OUTPUT OUT=stat COOKD=cook_d DFFITS=df_fits H=hat_value
            LCL=pred_lci UCL=pred_uci PRESS=res_del R=Residual RSTUDENT= st_r;
      RUN ;
ODS GRAPHICS OFF;
```

If it is of interest of the users, a P can also be added at the end of the MODEL statement, following a forward slash, to print the predicted and residual values. COOKD is added to the above SAS code to display the Cook's D influence statistic. DFFITS is added to produce the standard influence of each observation on the respective predicted value. Adding the H option will display the leverage values. LCL and UCL will display the lower and upper bound of a confidence interval for an individual prediction, respectively. PRESS will produce the residual for each observation that results from dropping that observation and then predicting it on the basis of all other observations. R will produce and display the residuals, and RSTUDENT displays the studentized residual with the current observation deleted. There are many other options that can be added here to display more information.

After investigating the residual plots and other parts of the output, if the users needed to opt for a transformation to resolve the assumption violation issues, the Box-Cox method can be adopted to help them decide about the proper type of transformation. As shown how to perform this method in the following SAS code, this approach is helpful in identifying an appropriate transformation for the response variable based on a set of predictor variables.

```
ODS GRAPHICS ON;
      PROC TRANSREG DATA=(mention the dataset name here) TEST;
            MODEL BOXCOX((specify the response variable here))=
            IDENTITY((list the continuous predictor variables here));
      RUN;
ODS GRAPHICS OFF;
```

After figuring out the best transformation, it can be applied to the response and the residual plots should be checked again to see if the assumption violation issues have been alleviated.

HIGH-DIMENTIONAL DATA

When dealing with high dimensional data, methods such as principal component analysis, random forest(s), shrinkage-based models, and cluster analysis are highly recommended (Wickham & Grolemund, 2016). Principal component analysis can be used to reduce the dimension of the data by creating linear functions of the existing variables. Radom forest models can be applied to both categorical and continuous response variables in two forms of classification random forest and regression random

forest, respectively. Breiman (2001) proposed random forest (or random forests), which uses a group of decision trees. Within random forests, in addition to constructing each tree using a different bootstrap sample of the data, how the classification or regression trees are constructed are changed in a way to build the most optimal group of trees for data sets. This strategy turns out to perform fairly well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overfitting of the data (Liaw & Wiener, 2002). Shrinkage-based methods such as ridge regression and the least absolute shrinkage and selection operator (LASSO) can also deal with a very large number of predictors and do not suffer from overfitting, which would happen if a multiple linear regression model is applied to the same data set. Therefore, they can be used for high dimensions of data and LASSO can even assist researchers in variable selection of several variables. Cluster analysis can be looked at as a case reduction technique, which reduces the dimension of the data by grouping together data samples that are similar in some way according to certain criteria of interest. Below each of these models are introduced and the SAS procedures to apply them to real data are explained.

PRINCIPAL COMPONENT ANALYSIS

Pearson (1901) and Hotelling (1933) introduced principal component analysis (PCA) to describe the variation in a set of correlated and uncorrelated variables within a data set in terms of a set of uncorrelated variables. PCA is a variable dimension-reduction tool that is capable of reducing a large set of variables to a small set of new uncorrelated variables, or components, that still contains most of the information in the original large set of variables. This mathematical procedure transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components. This will assist researchers to not to be negatively affected by the multicollinearity issue, that was explained within regular multiple linear regression models, through creating a new set of uncorrelated variables (Jolliffe, 2011) . Among the newly created components of PCA, the first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Therefore, the first few principal components will account for most of the variation and can be used instead of all the variables to reduce the dimension of the data (Dunteman, 1989).

PROC PRINCOMP is one of the procedures used to perform a PCA on a dataset and outputs the scores to a newly created data set. This data set can be called anything within the OUT option (for instance, OUT= score to call the data set ‘score’). OUT=SAS-data-set creates an output SAS data set that contains the original data in addition to the principal component scores. To create a permanent SAS data set to store the scores and save them to be used later, a two-level name should be specified. The following code shows a general PCA using the default for the number of principal components to be equal to the number of variables.

```
PROC PRINCOMP DATA=(mention the dataset name here) OUT=score;
      VAR (list the variables here);
RUN;
```

Once an initial PCA is performed on the data set, more options can be added to the code. The following code includes more modeling options.

```
PROC PRINCOMP N=6 PLOTS(NCOMP=3)=PATTERN;
      VAR (list the variables here);
RUN;
```

In the PROC PRINCOMP statement, N=number specifies the number of principal components to be computed; in the example above six principal components are specified to only calculate the first six principal components, instead of it being equal to the number of variables, which is the default.

Additionally, the statement PLOTS=PATTERN will provide the graphical plots of the PCA. The option (NCOMP=number; here number=3) requests for the graphical output comparing only the first three principal components in this example. If this option is not specified, the graphical output will compare the first six requested principal components.

PROC FACTOR is another procedure in SAS that can be used to perform a PCA. A simple code can be written as below:

```
PROC FACTOR DATA=(mention the SAS dataset name here) METHOD=principal
ROTATE=v SCREE SCORE OUTSTAT=save.factors_all;
      VAR (list the variables here);
RUN;
```

METHOD=principal specifies the use of the PCA as the FACTOR procedure is capable of performing other models, such as factor analysis, as well. ROTATE=v is to specify the rotation type as varimax; other rotation types can also be specified here. The SCREE option produces a scree plot of the eigenvalues that is visually helpful in deciding how many components to use. The SCORE option requests the scoring coefficients to be computed and reported. The OUTSTAT= option saves the results in a specially structured SAS data set. Make sure that the SAS data library has the libref save. The name of the data set, in this case factors_all, is arbitrary. Other options can also be added after the PROC FACTOR to specify more modeling details such as MINEIGEN=0, which enforces all components with variance greater than zero to be retained.

Similar to the PRINCOMP procedure, within the FACTOR procedure, the number of principal components to be computed can be specified, which in the following example, it is specified as the first six principal components.

```
PROC FACTOR DATA=raw METHOD=principal NFACTORS=6 OUT=save.scores;
      VAR (list the variables here);
RUN;
```

Creating plots of the scores of the principal components is not as straightforward as it was within the PRINCOMP procedure through the PLOTS option. A separate PLOT procedure needs to be created after running the PROC FACTOR to produce these plots and the pairwise comparisons need to be defined within the PLOT option as below:

```
PROC PLOT;
      PLOT factor1*factor2 factor1*factor3 factor2*factor3;
RUN;
```

As it can be seen, three pairwise graphical comparisons of the first three principal components are requested to be created in the output. More principal components and their comparisons can be added to this code, if one is interested.

DECISION TREES AND RANDOM FOREST

Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the output of this class by individual trees. The term “random forests” came from random decision forests that was first proposed by Tin Kam Ho in 1995. A tree is called a classification tree when the target variable is categorical and it is called a regression tree when the target variable is continuous. When classification decision trees are used within the forest building, the respective random forest is referred to as classification random forest. On the other hand, when regression decision trees are used while building the random forest for modeling continuous target variables, the random forest is called a regression random forest (Hartshorn, 2016).

In order to run a random forest in SAS, PROC HPFOREST can be used. Using this procedure, the target variable, outlining variables, and their categorical or continuous nature need to be specified. TARGET is to list the target or response variable and INPUT is to specify the input or predictor variables. LEVEL after each of these commands should be added to specify the type of the variable. The LEVEL option for the INPUT statement defines the predictors as binary, nominal, ordinal, or interval. LEVEL=binary is to specify a binary variable where there are two categories of the variable, LEVEL=nominal is for noting that the variable is categorical mainly for more than two categories of the variable, LEVEL=ordinal to specify the categorical variable where there exists an order among the categories, and LEVEL=interval is to let SAS know that the variable is numerical or continuous.

```
PROC HPFOREST DATA=(mention the dataset name here);
      TARGET (response variable name should be mentioned here)/LEVEL=(specify
      type of the variable here);
      INPUT (list the categorical variables here)/LEVEL=(specify type of the
      variables here; nominal should be specified for categorical variables);
      INPUT (list the numerical variables here) /LEVEL=(specify type of the
      variables here; interval should be specified for numerical variables);
RUN;
```

The INPUT lines can be condensed into one line to include the list of all the variables together as below:

```
input <predictor1, predictor2, predictor3,predictor4>/
level= <binary, nominal, ordinal, interval> ;
```

After running the code, a series of tables will appear in the SAS output that include the model information such as, number of randomly selected variables to test each node or possible split in each tree and the maximum number of trees. By default, the “Inbag Fraction” is set at 60% and the out of bag (OBB) is at a rate of 40%. The “Prune Fraction” is defaulted at “0″ and the closer it is set to “1″, the lower the level of growth the tree will have.

HPFOREST automatically uses only the valid variables that have no missing records under any of the observations. Table of “Baseline Fit Statistics” will include information such as the percentage of mis-classification the model identified. 100 percent minus this percentage, or the remaining percentage, will be the rate of accurate classification the random forest identifies. The higher percentage of the accurate classification shows that the majority of the sample has been classified correctly in each of the randomly selected samples. The “Fit Statistics” table of the decision trees shows the result of the analysis of the fitness of the trees within the forest. The column “Miscalculation Rate (Train)” shows the miscalculation rate for each tree and the rate tends to decrease by the increase in the tree number. The closer to the bottom of the table is where the minimum miscalculation rate is generally observed. Another table the model generates is the “Loss Reduction Variable Importance” table, which outlines the rank of importance of variables in terms of how each variable contributes to the predictability of the model.

By performing classification or regression random forest as a data-mining algorithm, we can select important explanatory variables in the process of predicting the outcome, response, or target variable whether categorical or continuous. In addition, this exercise allows us to use a combination of categorical and quantitative variables. In sum, this forest lets us know which variables are important, but not the relationship to one another.

SHRINKAGE METHODS

Two extensions of the linear regression are ridge regression and LASSO, which are used for regularization. When applying multiple linear regression models, more features or predictors could be added to the model compared to the simple linear regression. Having more features may seem like a perfect way for improving the accuracy of the trained model by reducing the loss within the loss function. This is because the trained model will be more flexible and will take into account more parameters that can potentially explain more of the response variation. On the other hand, researchers and practitioners need to be cautious while adding more features to the model as this may increase the likelihood of overfitting the data. Every data set within a research study can include noisy samples as the samples are taken at random. The noise in the sample may lead researchers to an inaccurate fitted model and if not trained carefully, it will lead to a model with lower quality. The model might also memorize the noise that exists within the noisy dataset instead of learning the trend of the data. Moreover, overfitting can happen in linear models when dealing with multiple features. As a result, if not filtered and explored up front, some features can be more destructive to the accuracy of the model than helpful, repeat information that are already expressed by other features, and add high noise to the dataset.

Therefore, statisticians have always tried assisting the applied researchers in preventing the overfitting issue. One of the most common mechanisms for avoiding overfitting is through regularization. A regularized machine learning model is a model that its loss function contains another element that should be minimized as well. The loss function includes two elements. The first one is what is used within regular

linear regression models, which is the sum of the distances between each prediction and its ground truth or the observed value. The second element added to the loss function, which is used in ridge regression and LASSO models, is the regularization term. Both models are explained below and the SAS procedures for performing each are explained.

Ridge Regression

Ridge regression allows a gentle trade-off between fitting the model and at the same time making sure that we are not overfitting it. Ridge regression is an extension, and at the same time an alternative technique, to linear regression. It is basically a regularized linear regression model, which was explained above. The tuning parameter is a scalar that should be learned as well, using a method called cross validation. Ridge regression also helps alleviating multicollinearity and so avoids the bias of least square estimates and the increased standard error of the coefficients. Increased standard errors that can happen due to multicollinearity means some variables are shown statistically insignificant when in fact they might be significant, if multicollinearity was not present. By helping to reduce the standard errors of the coefficients, ridge regression makes estimates of the parameters of the model more reliable.

An important fact we need to notice about ridge regression is that it enforces the coefficients to shrink and be lower in magnitude, but it does not force them to be zero. Therefore, it shrinks them to zero but does not set them equal to zero. That is, it will not get rid of irrelevant features and variables but rather minimize their impact on the trained model to reduce the likelihood of the overfitting of the model (Friedman, Hastie & Tibshirani, 2001).

Beyond the overfitting issue, having many input variables at play in a multiple linear regression sometimes poses a problem of choosing the inappropriate variables for the linear model, which gives undesirable and unreliable outputs as a result. Ridge regression can help overcoming this issue as well through regularization in which an extra variable (tuning parameter) is added and optimized to offset the effect of multiple variables in linear regression that ultimately can reduce the noise.

When trying to perform ridge regression, the first step is standardizing the target and input variables by subtracting their means and dividing the result by their standard deviations to harmonize the scales of measurement. The next step would be changing the diagonals of the correlation matrix, which would normally be 1, by adding a small bias or a K-value. This is why this model is called ridge regression, since a “ridge” is created in the correlation matrix by adding a bit to the diagonal values. Increasing this k-value will eventually drive the regression coefficients to zero, which is why large K-values are avoided within the ridge regression. There are different ways to decide about the K-value. Common methods include using a “Ridge Trace” plot to have a better visual understanding of where the regression coefficients stabilize, choosing the smallest value of K after the regression coefficients stabilize, choosing the value of K where R-Square is not changed significantly, and choosing the value of K where VIF is close to 2.

PROC REG can be used to fit a ridge regression as below:

```
PROC REG DATA=(mention the dataset name here) OUTVIF OUTEST=estimates RIDGE=0
to 0.05 by 0.001 plots(only)=ridge(unpack VIFaxis=log);
      MODEL (specify the response variable here)=(list the continuous
      predictor variables here);
      PLOT / RIDGEPLOT;
RUN;
```

OUTEST = estimates is added to the REG procedure to create a dataset called ‘estimates’ with the model estimates. OUTVIF is added to the procedure to tell SAS to write the VIF to the OUTEST = estimates. The RIDGE= option requests the ridge regression technique in the REG procedure. RIDGE = 0 to 0.05 by 0.001 tells the procedure to perform the ridge regression where the value of K starts at 0 and goes to 0.05 by increments of 0.001. If the K-value is known, it can be specified here instead of going for a range of K-values, which helps to see which one performs better within the model. After choosing the K-value, the REG procedure can be run with the preferred K-value.

Within this SAS code, each of the individual plots for ridge traces and VIF traces are generally of interest, so the unpack sub-option of the plots(only)=ridge option is added. The PLOT statement is designated to

display scatter plots of the target and input variables. RIDGEPLOT is added to the PLOT statement to request the ridge trace for the ridge regression.

LASSO Method

LASSO is a linear regression technique, which also performs regularization on variables in consideration. Similar to the ridge regression, LASSO is another extension built on regularized linear regression, but with a small difference, which is that the regularization term is in absolute value. Setting the coefficient equal to zero, when the tuning parameter allows the model to do so, has a large impact on the model results. LASSO method overcomes the disadvantage of ridge regression by not only punishing high values of the coefficients, but actually setting them to zero if they are not relevant, which means getting rid of the irrelevant variables. Therefore, fewer features may be included in the final model than originally entered into the model, which is a huge advantage. This is why LASSO is considered a variable selection technique (Hastie, Tibshirani & Wainwright, 2015).

LASSO shares a very similar statistical analysis evident in ridge regression, except it differs in the regularization values. This means, LASSO considers the absolute values of the sum of the regression coefficients, hence the 'shrinkage' feature, and sets the coefficients to zero. This helps reducing the errors and noise completely. PROC GLMSELECT can be used to fit LASSO in SAS. The SAS code to perform this model can be written as below:

```
PROC GLMSELECT DATA=(mention the dataset name here) PLOTS=all;
      MODEL (specify the response variable here)=(list the predictor
      variables here)/SELECTION=lasso(STOP=none CHOOSE=CP);
RUN;
```

Within this procedure the PLOTS=all option is added to request all the plots that come with the analysis. SELECTION=lasso option is added to the MODEL statement to impose the use of the LASSO method.

The CHOOSE= option enables specifying a criterion to use for picking a model from the sequence of models obtained by the selection process. If a CHOOSE= criterion is not specified, then the model based on the STOP= option is the selected model. SELECTION=lasso (CHOOSE=CP) will result in a sequence of models to be obtained by the LASSO algorithm using the Mallow's C(p) criterion for picking the best model. From the models in this sequence, the one yielding the smallest value of the Mallow's C(p) statistic is chosen as the final model. Other criteria for picking the best model can also be specified here. Additionally, a stopping criterion with the STOP= option can be added to the procedure.

To cross validate the model, the data need to be split randomly into the test and training sets within the LASSO procedure. To do so, first, a SURVEYSELECT procedure should be defined to split the data and then the LASSO procedure should be performed. The following PROC SURVEYSELECT randomly splits the observations in the data set into training (60%) and test (40%) roles. These percentages can vary.

```
PROC SURVEYSELECT DATA=(mention the dataset name here) OUT=traintestdata
SEED = 123 SAMPRATE=0.6 METHOD=SRS OUTALL;
RUN;
```

Next, the GLMSELECT needs to be performed to apply LASSO regression to select a subset of the input variables that minimizes the prediction error for the response variable.

```
PROC GLMSELECT DATA=traintestdata PLOTS=all SEED=123;
      PARTITION ROLE=selected(TRAIN='1' TEST='0');
      MODEL (specify the response variable here)=(list the predictor
      variables here)/SELECTION=lar(CHOOSE=cv STOP=none) CVMETHOD=random(10);
RUN;
```

In the above GLMSELECT procedure, least-angle regression (LAR) is the selection method and cross-validation is used as a criterion for choosing the best model. As it can be seen from the option CVMETHOD=random(10) added to the MODEL statement, k=10 fold validation is used here. Using the PARTITION statement, PROC GLMSELECT assign 60% of the observations for model validation and 40% of the observations for model training.

Cluster Analysis

Clustering is a case reduction technique, which operats based on grouping together data samples that are similar in some way, according to some criteria that researchers pick, and separating dissimilar cases from each other based on some dissimilarity measures. Both similarities and dissimilarities can be used while defining different clusters and grouping cases together to reduce the dimension of the cases of the data set.
Cluster analysis is also considered a form of unsupervised learning. This means, there generally are no examples or theory demonstrating how the data should be grouped together and instead it is left up to the data to guide researchers and practitioners through this grouping based on the data characteristics. Therefore, clustering can be seen as a method of data exploration and a method of looking for patterns or certain structure in the data. The goal is to group together "similar" data and the important question is how to find the similarities and dissimilarities and use them to group the data (Anderberg, 2014).

There exists no single answer in how to define the similarities among the sample cases. It depends on what researchers want to find or emphasize in the data; this is one reason why clustering can be a flexible tool based on what researchers need. The similarity measure is often more important than the clustering algorithm used in the data exploration as if the similarity is not measured properly, the clustering algorithm cannot do anything to fix that issue. Measures such as Pearson correlation coefficient and Euclidean distance are some examples of measures that quantify the similarities among the data points. Different clustering algorithms include, but are not limited to, hierarchical agglomerative clustering, K-means clustering and quality measures, and self-organizing maps.

There exist different SAS procedure to perform each type of the cluster analysis. PROC CLUSTER is one of the procedures that enable us to perform cluster analysis in SAS. A simple SAS code can be written as below:

```
PROC CLUSTER SIMPLE NOEIGEN METHOD=EML RMSSTD RSQUARE OUTTREE=tree;
      ID subject;
      VAR (list variables here);
RUN;
```

Within this procedure, the SIMPLE option is added to display descriptive statistics. The NOEIGEN option suppresses computation of eigenvalues to save processing time if the number of variables is large, but it should be used only if the variables are nearly uncorrelated or if there is no interest in the cubic clustering criterion. The METHOD= specification determines the clustering method used by the procedure. Here the EML is specified to impose the performing of the maximum-likelihood hierarchical clustering approach. There are 11 options that can be used here. These options are AVERAGE, CENTROID, COMPLETE, DENSITY, EML, FLEXIBLE, MCQUITTY, MEDIAN, SINGLE, TWOSTAGE, and WARD.

The RMSSTD option displays the root mean square standard deviation of each of the clusters. The RSQUARE option displays the R^2 and semi-partial R^2 to evaluate the cluster solution. The NONORM option could be added to prevent the distances from being normalized to unit mean or unit root mean square with most methods. The OUTTREE=SAS-data-set option creates an output data set that can be used by the TREE procedure to draw a tree diagram. The name given to this data set should be a two-level name if one intends to save it. If the OUTTREE= option is omitted, the data set is named by using the DATA convention and is not permanently saved. If creating an output data set is not of interest, use OUTTREE=_NULL_. Within the ID statement, the values of the ID variable is added to identify observations in the displayed cluster history and in the OUTTREE= data set. If the ID statement is omitted, each observation is denoted by OB*n*, where *n* is the observation number. The VAR statement lists numeric variables to be used in the cluster analysis. If the VAR statement is omitted, all numeric variables not listed in other statements are used.

PROC VARCLUS and PROC MODECLUS are among other procedures for performing cluster analysis. Another procedure that can be used to apply cluster analysis is PROC FASTCLUS. The FASTCLUS procedure performs a non-hierarchical or disjoint cluster analysis on the basis of distances computed from one or more quantitative variables. PROC HPCLUS and PROC FASTCLUS can also be used to perform k-means clustering. The aforementioned procedures and many more SAS procedures can assist SAS users to perform different types of clustering to achieve their case reduction goals.

CONCLUSION

To summarize, in this paper, author tried to gather together a summary of different algorithms and methods for modeling classification and continuous response or target variables within both low dimensional and high dimensional data. Author's goal was to assist readers in the understanding of different statistical methods, using their algorithms within machine learning and data analytics, and provide SAS programming options and examples to make it easier for users to adopt these methods in their data analyses. Various statistical strategies and algorithms depending on the type of response variables used in each model were explained based on the dimension of the data. Models used for modeling of categorical and continuous target variables were first explained for low-dimensional data. Next different approaches for analyzing high-dimensional data were explained and used for dimension reduction. These techniques can assist researchers in modeling of the bigger dimensions of data, describing them and making inferences about them, and predicting the behavior of the data.

Different approaches, based on the type of the variables involved in a study, were presented to help working with data sets in different dimensions and formats, reduce the dimensions of big data sets, and get efficient results. These methods are not the only methods and algorithms that can be adopted in each of the explained data scenarios, but this can be a start to learn about and apply appropriate modeling techniques to each data set based on the characteristics of the data and the nature of the variables used in building quantitative models. Once these methods are understood and the differences become clear, the data analysis stage is much easier to perform and many resources are available for programming and modeling in different software packages.

REFERENCES

Agresti, A. (2007), An Introduction To Categorical Data Analysis. Willey Series in Probability and Statistics, second edition.

Agresti, A. (2013). Categorical data analysis (3rd ed.). New York: Willey.

Allison, P. D. (2012). *Logistic regression using SAS: Theory and application*. SAS Institute.

Anderberg, M. R. (2014). *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks* (Vol. 19). Academic press.

Atkinson, P. M., & Massari, R. (1998). Generalised linear modelling of susceptibility to landsliding in the central Apennines, Italy. *Computers & Geosciences*, *24*(4), 373-385.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Dunteman, G. H. (1989). *Principal components analysis* (No. 69). Sage.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.

Hartshorn, S. (2016). Machine Learning With Random Forests And Decision Trees: A Visual Guide For Beginners. *Kindle Edition*.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

High, R., Models for Ordinal Response Data (2013). SAS Global Forum, Paper 445-2013

Hosmer, D., Lemeshow, S., Applied Logistic Regression (2013). Willey Series in Probability and Statistics, third edition.

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science* (pp. 1094-1096). Springer, Berlin, Heidelberg.

Lee, S. (2005). Application of logistic regression model and its validation for landslide susceptibility mapping using GIS and remote sensing data. *International Journal of Remote Sensing*, *26*(7), 1477-1491.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis* (Vol. 821). John Wiley & Sons.

Ramezani, N. (2015). Approaches for missing data in ordinal multinomial models*. In JSM Proceedings,* Biometrics section, New Methods for Studies with Missing Data Session. Alexandria, VA: American Statistical Association Journal.

Ramezani, N. (2016). Analyzing non-normal binomial and categorical response variables under varying data conditions. *In proceedings of the SAS Global Forum Conference.* Cary, NC: SAS Institute Inc.

Ramezani, N., Ramezani, A. (2016). Analyzing non-normal data with categorical response variables. In proceedings of the Southeast SAS Users Group Conference. Cary, NC: SAS Institute Inc.

Segal, M. R. (2004). Machine learning benchmarks and random forest regression.

Wickham, H., & Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data*. " O'Reilly Media, Inc.".

RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Niloofar Ramezani
Department of Statistics, George Mason University
nramezan@gmu.edu