Paper 5116-2020

Interpretation Methods for Black-Box Machine Learning Models in Insurance Rating-Type Applications

Gabe Taylor, State Farm Insurance Companies; Sunish Menon, State Farm Insurance Companies; Huimin Ru, State Farm Insurance Companies; Ray Wright, SAS Institute Inc.; Xin Hunt, SAS Institute Inc.; Ralph Abbey, SAS Institute Inc.

ABSTRACT

Traditional Generalized Linear Models (GLMs) are often favored in the insurance realm for rating due to their interpretation simplicity and intuitive distributional assumptions. This is partly driven by insurance regulation needs and partly based off customers' demand of explanation for their rates. In insurance rating, there is often a need to provide credit reason codes and occasionally customer feedback in determining the factors adversely impacting policyholders' individual premium. Machine learning models (sometimes referred as black-box methods), on the other hand, often lack transparency and interpretability but have powerful predicting potential. Actuarially, modelers have to balance between accurate confident pricing and model interpretability, for both the collective customer population and individual customers. In this paper, we demonstrate the value of interpretation methods at the global and local level for a Gradient Boosting Decision Tree model using simulated auto insurance data. We review Partial Dependence (PD), Individual Conditional Expectation (ICE) and Accumulated Local Effects (ALE) plots for global variable level interpretation as a substitute for parameter estimate and variable significance type analysis. We also demonstrate use of Localized Interpretable Model-Agnostic Explanations (LIME) and Shapley values for local prediction explainability. LIME and Shapley values can be used independently or together to provide feedback at an individual customer level. Although these methods can be easily explored on other platforms, such as R and Python, our research was conducted within the SAS® Viya environment, which utilizes pre-packaged action sets for black-box machine learning.

INTRODUCTION

Despite the intuition to revere model performance as the most sought after quality in a predictive model, model transparency takes precedence in the insurance rating setting. In other words, knowing *how* a model makes a prediction is more important than *what* the prediction itself is. Insurance companies are often required to give clear and explicit explanations to regulators, and occasionally due to customer demand, provide detailed explanations to customers about individual rates. Regulators often look for affordability and unfair discrimination, while insurance companies are looking for competitive advantage by better matching price to risk. An insurance company should be equipped to investigate an individual customer's policy using a predictive model to present a report to the customer containing the most influential factors affecting their rates. When companies are using credit for rating, Fair Credit Reporting Act (FCRA) stipulates that insurance companies provide reason codes to the customer (2018). Sometimes the rating program requires that the company provides customers feedback on ways to improve their rate or what they can do to lower their premiums.

The most widely used model in rating an insurance policy is the Generalized Linear Model (GLM) because of its lucidity and interpretability. The intuitive distributional assumptions and innate parametric nature of the model make it a popular modeling technique. However, GLM needs significant one-way visual analysis, such as Exploratory Data Analysis at the univariate level, to capture non-linear behavior of variables because of

underlying linear structure. The accuracy of the predictions may fall short of other models, such as machine learning "black-box" models. However, as the name "black-box" model implies, the models lack transparency in the manner in which they reach predictions. The complex nature of the models presents a problem for potential use in the insurance industry due to aforementioned emphasis on interpretability.

Global and local interpretation methods have been explored that are capable of overcoming the lack of transparency in black-box models. Global methods reflect how the target variable changes in response to changing the inputs of the feature variables. Revealing the relationship of the variables to the target through the use of global methods leads to meaningful diagnostics for variable selection. Local methods indicate which feature variables are most influential on a unique prediction of interest given by the model. With the ability to isolate one prediction, local methods provide a system for insurance companies to investigate individual policies and diagnose the most influential factors affecting the rate in that policy. Black-box models coupled with sufficient interpretation could potentially be utilized by insurance companies to give more accurate rates and useful feedback to their customers when needed.

DATA AND MODELS

To explore the global and local interpretation methods, a black-box Gradient Boosting Machines (GBMs) was built on two million observations of simulated auto insurance data with 15 variables using the action set "decisionTree". For comparison, a Generalized Linear Model (GLM) was also built on the same data using PROC HPGENSELECT. Both models were evaluated using Mean Absolute Percentage Error (MAPE) and Lift, which measure, respectively, the prediction accuracy in the model and how effectively the model segments the target. Lower MAPE suggests better prediction accuracy, and higher Lift indicates better segmentation.

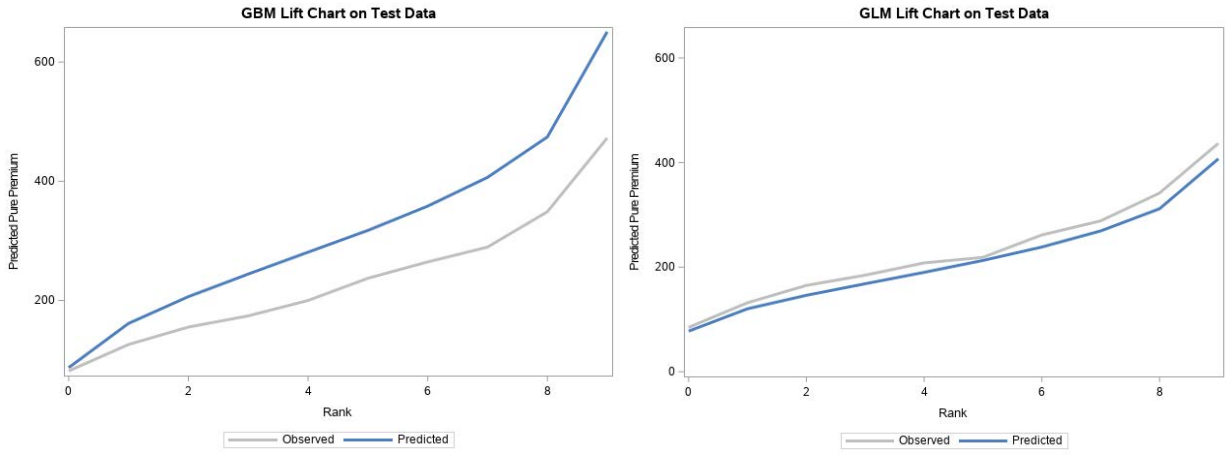


Figure 1. Lift Charts for Gradient Boosting Machines and Generalized Linear Model

As illustrated in Figure 1, the GBM reported a higher lift (14.24), and the GLM had a lower MAPE (1.8%). The lines shown in each Lift chart represent both the average observed and predicted pure premium values across 10 quantiles represented on the x-axes. Steeper slopes for the predicted values translate to higher Lift values, which indicate better model segmentation. The difference between the observed and predicted values is reflected by the amount of space between each of the lines on the chart, therefore, less space expresses lower MAPE. Despite the comparable fit statistics, the GBM captured non-linear relationships between the predictors and the target, which will be discussed in the following sections.

PARTIAL DEPENDENCE

Partial Dependence (PD) plots show how the model predictions change as the feature variable inputs change. One-way PD plots are concerned with displaying the relationship between the model predictions and a single feature variable (Wright 2018). For each level within the chosen feature variable, the entire dataset is replicated holding that level constant. The new replicated data sets are then scored using the model. The levels within the chosen feature variable form the x-axis values, and the average scored predictions for each level form the y-values that are plotted at the respective x-axis levels.

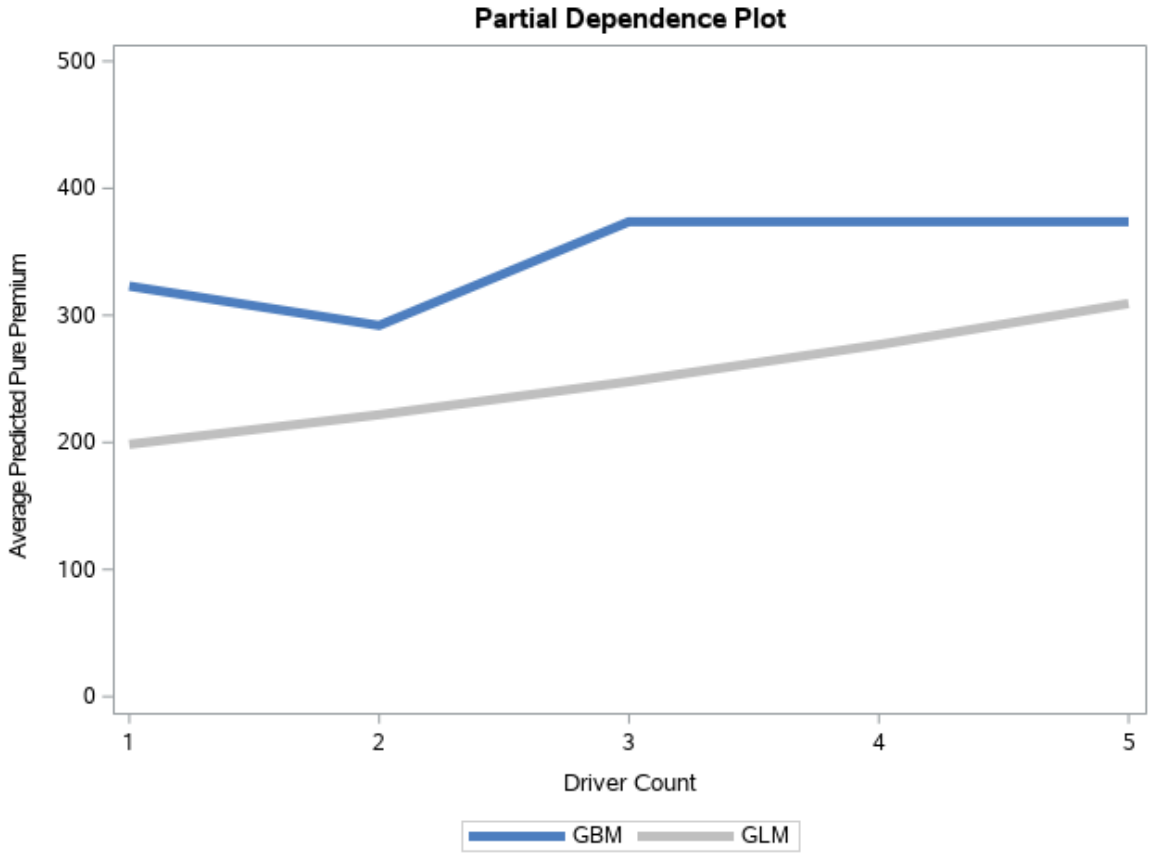


Figure 2. Partial Dependence Plot for Driver Count

The PD plot in Figure 2 depicts the relationship between the feature variable Driver Count, which denotes the number of drivers on a single policy, and the target variable pure premium given by a GBM and a GLM. The grey line representing the GLM reflects a strictly linear relationship with the target. However, the blue line representing the GBM indicates a different, non-linear relationship. According to the PD plot of the GBM, on average, pure premium will decrease from 1 to 2 drivers on a policy and then the premium will actually increase from 2 drivers to 3. More than 3 drivers will not affect premium on average. Because the PD plot of the GBM suggests that the relationship between the Driver Count and pure premium is not perfectly linear, the non-linear behavior of the blue line exhibits the ability of the GBM to capture a relationship of the predictors to the target in a multivariate case that is perhaps more in line with industry knowledge and modeler intuition.

Instead of assuming that all levels within a variable have the same linear effect on pure premium, a PD plot can reveal which levels have more or less influence on the target determined by a black-box model. In insurance rating, knowing which levels within a

variable have the most influential relationship to the target provides meaningful guidance over the assumption that all levels are scaled with the target linearly.

The code below utilizes an action set unique to SAS Viya that automates the calculations for generating a table of values that will create a PD plot:

```
proc cas;
loadactionset "astore";
action explainModel.partialDependence
    result           = pd_res
    table            = 'mod_data_3'
    modelTable       = 'GB_astore'
    inputs           = {{name='X_01_02'}, {name='X_01_05'}, {name='X_01_09'},
                        {name='X_01_20'}, {name='X_01_24'}, {name='X_01_26'},
                        {name='X_01_28'}, {name='X_01_30'}, {name='X_02_03'},
                        {name='X_02_07'}, {name='X_02_17'}, {name='X_02_22'},
                        {name='X_02_28'}, {name='X_03_16'}, {name='X_03_20'}}
    predictedTarget  = "P_Target"
    analysisVariable = {name="X_01_30", nBins= 10}
    seed                = 1234;
    saveresult pd_res dataset=pd;
   run;
quit;

proc sgplot data = merged ;
   series x = X_01_30  y = AvgYHat /
   lineattrs = (color = BIGB thickness = 5 legendlabel = "GBM" name = "GBM");

   series x = X_01_30  y = AvgYHat0 /
   lineattrs =    (color = LIGR thickness = 5 legendlabel = "GLM" name =
   "GLM");

   yaxis label = "Average Predicted Pure Premium" min = 0 max = 500;
   keylegend "GBM "GLM";

run;
```

This action call uses the following parameters:

- table: the original dataset from which the model was trained
- modelTable: the table containing the scoring code for the black-box model
- inputs: specifies the model input variables to use in the analysis
- predictedTarget: specifies the variable that contains the model's predictions
- analysisVariable: specifies the analysis variable and its related attributes
- seed: seed number to be used for random sampling of data

In spite of its simplicity and the ability to depict non-linear relationships that black-box models are able to capture, PD plots can be misleading if there are correlations between the predictors in the model. For example, if variable x1 = b*x2 + c, simple replication of the data is not going to capture this relationship. In fact, it will under-represent the correlation when x1 and x2 values are replicated. Also, unrealistic observations could be simulated during the process of replicating the data sets, which could lead to misguided interpretation. In the insurance application, the rating variable pool is often a collection of correlated variables, so interpretation methods less sensitive to correlation could be used together with PD plots to get a more accurate representation of the relationships between the predictors and the target.

INDIVIDUAL CONDITIONAL EXPECTATION

Individual Conditional Expectation (ICE) plots, similar to PD plots, illustrate how the model predictions change as the feature variable inputs change. For a chosen feature variable, the complementary variables for a single chosen observation are replicated for each level within the chosen feature variable. The replicates are scored and then the predictions are plotted for each respective level. This process is repeated for a given number of sample observations. An intuitive explanation of an ICE plot is that a PD plot simply represents the average of all the lines on an ICE plot (Wright 2018).

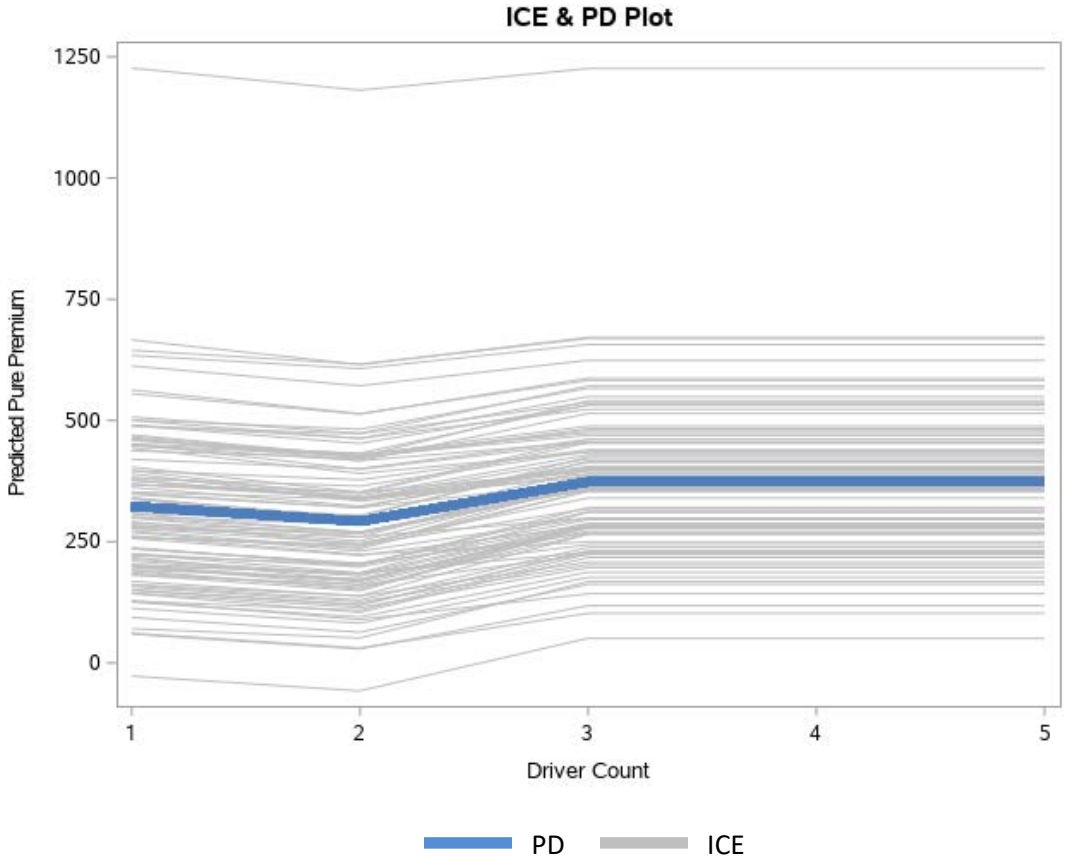


Figure 3. Individual Conditional Expectation Plot for Driver Count

The ICE plot above was generated using a macro developed by Ray Wright (Wright 2018). The grey lines in Figure 3 represent single sampled observations and how the predicted pure premium changes for each number of drivers on a policy. The single blue line is the average of all of the grey lines, which is the same GBM line shown in the PD plot in Figure 2, only drawn at a slightly different scale.

Since many different observations are plotted, ICE plots can reveal interactions among the variables in the model. If the lines on a plot are excessively intersecting, that behavior would be indicative of an interaction present in the model. This interaction detectability is an advantage of ICE plots. However, because of the similarity in permutation sampling akin to PD plots, ICE is susceptible to the same issue of correlation among the variables as PD.

If there is little correlation present among model inputs, both PD and ICE plots are valid model interpretation methods that have potential insurance application due to the ease of calculation and implementation. Capturing non-linear behavior that more accurately

describes the relationship of the predictors to the target provides a distinct advantage in the rating process. However, the issue of correlation may be insurmountable in some cases, which leads into the next global interpretation method that has the potential to overcome the limitations of PD and ICE.

ACCUMULATED LOCAL EFFECTS

Similar to PD and ICE plots, ALE plots perform the same function of reflecting the relationship of the predictions to a feature variable as the inputs of that given feature variable change. However, instead of averaging across all predictions as the PD and ICE do, ALE averages the *difference* in predictions for specified intervals within the domain of the given feature variable. To calculate the difference in prediction for a single instance, replace the value corresponding to the given feature variable to be plotted with the upper and lower limit of the given interval, score those predictions, and take the difference of the two. Repeat the same process for the other instances in the interval. Averaging the differences of the instances results in the ALE for the given interval. This process is repeated for all specified intervals within the chosen feature variable. By averaging the difference in the predictions in each interval rather than averaging across the entire domain of the variable nullifies the effect of other correlated variables (Molnar 2019).

COMPUTING THE ALE FUNCTION

The following steps calculate the values for an ALE Plot with the variable Car Age and the target as pure premium using hypothetical training data. An additional predictor, Horsepower, is introduced for illustration purposes.

Car Age	Horsepower
3	150
6	275
6	300
8	300
9	125

1. Three intervals have been established for the Car Age variable – 1:4, 4:7, and 7:10.
2. Each instance is replicated and scored by replacing the upper and lower limits of the interval.
3. Take the difference of the predictions.

Car Age Interval	Horsepower	Predicted Premium	Difference
1	150	200	0
4	150	200	

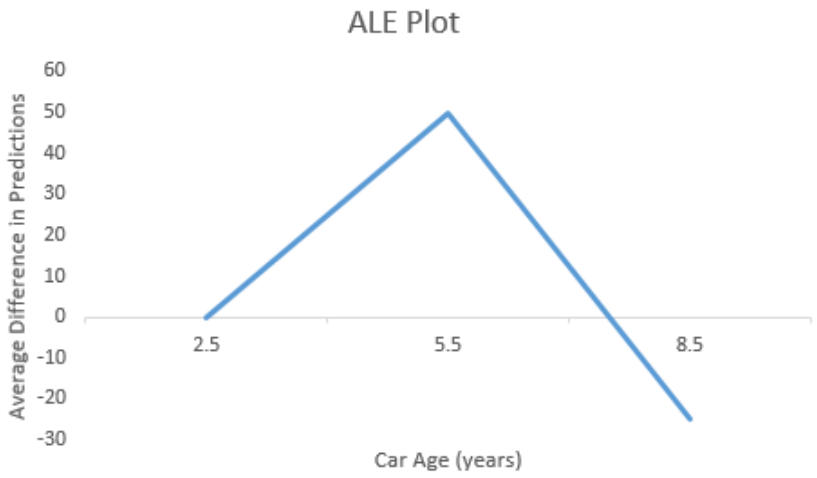
Car Age Interval	Horsepower	Predicted Premium	Difference
4	275	300	75
7	275	375	
4	300	375	25
7	300	400	

Car Age Interval	Horsepower	Predicted Premium	Difference
7	300	400	-25
10	300	375	
7	125	200	-25
10	125	175	

4. Once all differences have been calculated, they are averaged within the interval.

Car Age Intervals	Average Difference in Predictions
1:4	0
4:7	50
7:10	-25

5. The final step is to plot the average differences within each interval. The y-axis values represent the average differences between the predictions, and the x-axis values take the same values as the predictor. However, because the effects are isolated to intervals, the slope of the average differences will change at the midpoint of each interval, rather than the changing in relation to original values of the predictor.



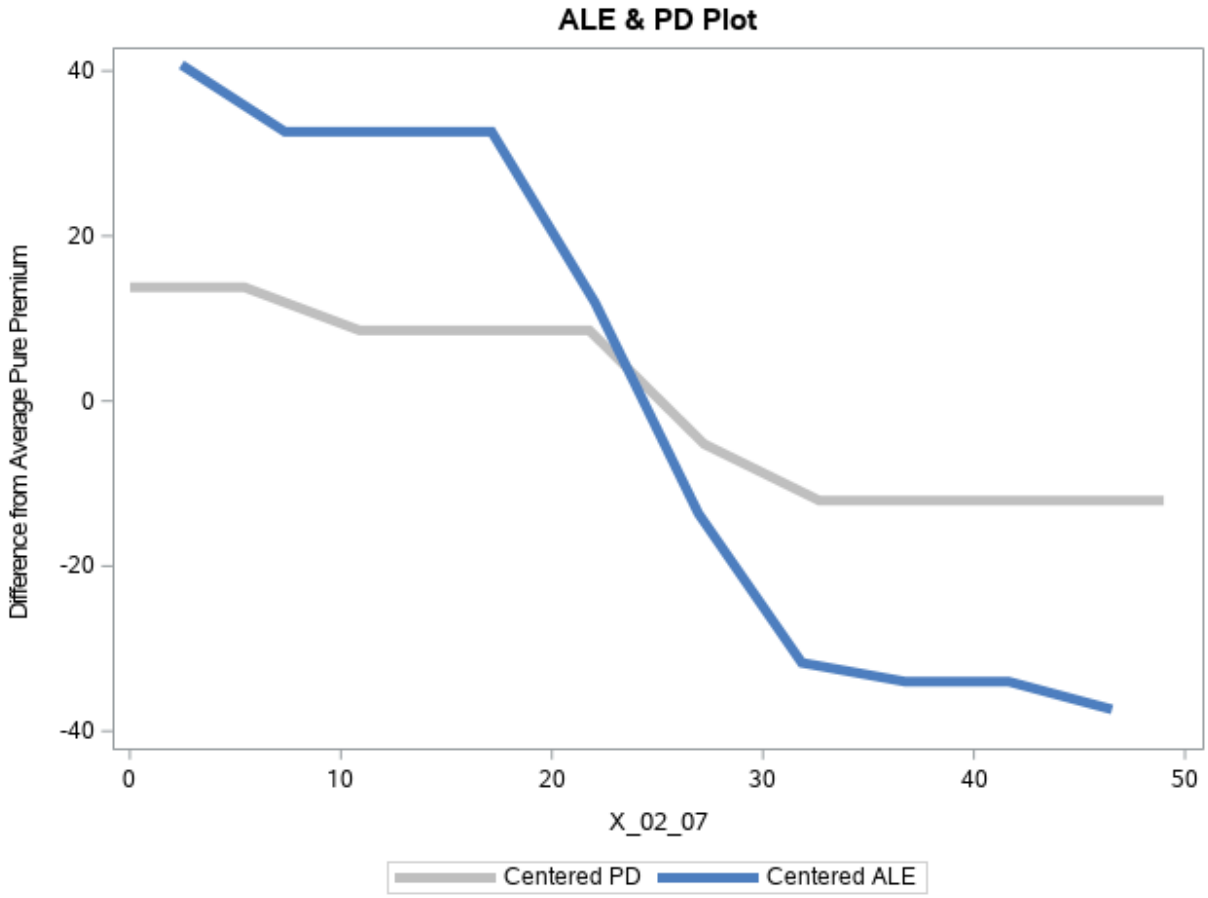


Figure 4. Centered Accumulated Local Effects and Partial Dependence Plot for X_02_07[1]

ALE plots can be interpreted in the same manner as PD plots, however, ALE plots are centered at the mean due to calculating the average difference in predictions. To interpret the ALE plot in Figure 4, when the variable X_02_07 takes a value of 10, the average predicted pure premium will be greater than the overall average predicted pure premium by about 30. For comparison, the centered PD for the same variable is represented by the grey line. The disparity in behavior between the ALE and the PD can be explained by the fact that the variable X_02_07 is strongly correlated with other variables in the model. The relatively flat appearance of the PD line is commonly observed when model inputs are correlated. Therefore, the PD plot does not represent the relationship between X_02_07 and the target as accurately as the ALE plot.

In the presence of correlations, ALE plots offer clear advantages over PD plots. While both methods provide critical insight on the relationship of the target and the feature variables, rating models may include correlated features, rendering PD plots less reliable. Therefore, ALE plots are a strong alternative when predictors are correlated and PD plots may be an incomplete representation.

[1] Due to confidentiality, we could not reveal the true name of this variable, so the X_XX_XX naming convention represents a masked variable

LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS

As given in the name, Local Interpretable Model-Agnostic Explanations (LIME) attempt interpreting a black-box model locally, as opposed to globally as all the previous methods have done. Rather than examining the relationship between the predictions and a single variable, LIME investigates the relationship of a single prediction and the variables that most influenced that prediction. The overall concept of LIME is to choose a prediction and build a local surrogate model that is interpretable based on the characteristics of that prediction. The process is as follows:

- Once a specific prediction is chosen, generate a new data set by randomly drawing samples from a Gaussian distribution that is centered at the selected prediction for each variable. The variance of each of the Gaussian distribution is calculated from the marginal distribution of each variable in the original data set.
- Score the new data set using the original black-box model.
- Next, weight the new scored samples in relation to their proximity of the chosen prediction.
- A new weighted, interpretable model is trained on the generated data set that can then be used for subsequent local interpretation for the prediction of interest.

Variable	Estimate	QueryValue
Intercept	331.9389481	.
X_01_02	-0.150527448	50
X_01_05	-2.789804348	4
X_01_09	-0.365817893	5
X_01_20	0	0
X_01_24	25.832148632	3
X_01_26	-9.376502819	-1
X_01_28	-0.087431652	347
X_01_30	233.69072833	1.0604
X_02_03	-0.133557695	721
X_02_07	-0.631645733	26
X_02_17	-0.03159703	718
X_02_22	-0.060356433	718
X_02_28	-0.079236681	719
X_03_16	0	0
X_03_20	0	0

ModelPred	ExplainerPred	ExplainerRMSE
396.22516531	380.05243474	64.847616829

Figure 5. Output for LIME Action Set

The results shown above represent the LIME values of a chosen prediction in the data set. The output is able to provide insight on which variables will most alter the prediction. The leftmost variable column denotes which variables were included in the original model. For each variable in the original model, an estimate is given by the local model. Note that an estimate of 0 suggests that the variable was excluded from the local model through LASSO regularization. The values displayed in the rightmost column of the table are the original values of the prediction of interest. Each estimate in the table provides a *local* explanation for the chosen prediction, meaning that attempting to draw global

interpretations about the model outside of the local area will lead to misguided and false explanations. For example, the coefficient for a variable in the local model for one prediction may be significantly different than the coefficient for the **same variable** for another prediction, meaning that the estimates given by a local model should not be interpreted globally. In this example, the variable X_01_30 has the largest valued estimate, which suggests that changing the input of the X_01_30 will change this specific prediction more significantly than the other variables.

The code below generates the local surrogate model by employing the action set "explainModel" and LIME:

```
data query;
   set mycaslib.mod_data_3(obs=1);
run;

proc casutil;
   load data = query outcaslib = 'mycaslib'
   casout = "query";
run;
proc cas;
  loadactionset "explainModel";
  explainModel.linearExplainer /
    table            = "mod_data_3"
    query            = "query"
    modelTable       = "GB_ASTORE"
    modelTableType   = "ASTORE"
    predictedTarget  = "P_TargetO"
    seed             = 1234
    preset           = "LIME"
    inputs           = {{name='X_01_02'}, {name='X_01_05'}, {name='X_01_09'},
                        {name='X_01_20'}, {name='X_01_24'}, {name='X_01_26'},
                        {name='X_01_28'}, {name='X_01_30'}, {name='X_02_03'},
                        {name='X_02_07'}, {name='X_02_17'}, {name='X_02_22'},
                        {name='X_02_28'}, {name='X_03_16'}, {name='X_03_20'}}
    nThreads         = 1;
   run;
quit;
```

This action call uses the following parameters:

- table: the original dataset from which the model was trained
- query: a single row from the data to be investigated
- modelTable: the table containing the scoring code for the black-box model
- modelTableType: the type of table containing score code
- predictedTarget: specifies the variable that contains the model's predictions
- seed: seed number to be used for random sampling of data
- preset: specifies the type of model explanation
- inputs: variables to be included in the local surrogate model
- nthreads: number of threads used

With the ability to investigate a single prediction, LIME has the potential to equip insurance companies with a powerful tool. Insurers can explore a single policy and extract the most influential factors that could impact an individual's rate, and supply meaningful feedback to their customers.

SHAPLEY VALUES

In place of building a local surrogate model for a single prediction, Shapley values evaluate the degree to which each variable contributed to making that specific prediction. The method is rooted in cooperative game theory, where a certain overall gain is achieved among the players, but the distribution of payout will be unique because certain players may have contributed more than others achieving the gain (Cohen 2005). This concept can translate directly to machine learning interpretability. For each prediction, there are some variables that contribute to the prediction more than others. Shapley values reveal which features have the most significant influence for a specific prediction. The sum of all the Shapley values will sum to the difference in the average prediction and the prediction of interest.

COMPUTING THE SHAPLEY VALUES

Using the same hypothetical data from the ALE example and an additional variable, Length of Ownership, we will calculate the Shapley value for **Horsepower** for the **third** prediction.

Car Age	Horsepower	Length of Ownership	Predicted Premium
3	150	2	200
6	275	2	350
6	**300**	**3**	**375**
8	300	4	400
9	125	9	175

If there are *p* predictors in a model, the amount of possible coalitions needed to calculate a single Shapley value will be 2^{p-1}. In our example, we will have 4 coalitions:

a. None
b. Car Age
c. Length of Ownership
d. Car Age, Length of Ownership

For all possible coalitions, calculate a prediction **with** the variable of interest and **without** the variable of interest, and then take the difference of the two. For coalitions that exclude certain variables, input those values by randomly selecting an observation from the original dataset. In this example, the real values are highlighted in **blue** and the random values in **grey**. In practice, for each coalition this process is **repeated many times** and the differences are averaged. For this example, only a **single** contribution for each coalition is calculated.

1. The first coalition includes **no predictors**, so we generate those values by randomly selecting those values from the original data set.

Car Age	Horsepower	Length of Ownership	Predicted Premium
9	**125**	**2**	**225**
3	**300**	**3**	**350**

The difference with and without Horsepower in this coalition is 350 − 225 = 125.

2. The second coalition includes **Car Age**.

Car Age	Horsepower	Length of Ownership	Predicted Premium
6	275	2	350
6	300	9	325

The difference with and without Horsepower in this coalition is 325 − 350 = -25.

3. The third coalition includes **Length of Ownership.**

Car Age	Horsepower	Length of Ownership	Predicted Premium
6	300	3	375
8	300	3	400

The difference with and without Horsepower in this coalition is 400 − 375 = 25.

4. The fourth coalition includes **Car Age** and **Length of Ownership**

Car Age	Horsepower	Length of Ownership	Predicted Premium
6	150	3	200
6	300	3	375

The difference with and without Horsepower in this coalition is 375 − 200 = 175

5. Averaging all the differences would give
    a. (125 + -25 + 25 + 175) / 4 = 75

Hence, the Shapley value for Horsepower in this example is **75**.

6. Since all Shapley values sum to the difference in the average prediction and the prediction of interest, the interpretation of the value given above relies on the Shapley values of the remaining features.
    a. The Shapley values should sum to:

        375 (actual prediction) − 300 (average prediction) = 75

    b. An example for the remaining Shapley values would be

        Horsepower: 75 (from above)

        Car Age: 50

        Length of Ownership: -50

c. Horsepower contributed the most to the prediction of interest in comparison to Car Age and Length of Ownership.

However, a significant drawback to Shapley values is that the calculations are extremely computationally expensive. The possible coalitions needed for a Shapley values increases exponentially as the amount of features in the model increase. The next section discusses a method that overcomes the problem of intensive calculation and limitations of the interpreting Shapley values.

SHAPLEY ADDITIVE EXPLANATIONS

Computing classical Shapley values can give insight into which variables have the most influence on the certain prediction, but it becomes computationally cumbersome as the dimension of the data set increases. However, Lundberg and Lee (2017) developed a method called KernelSHAP, which uses special weighted regression to compute **approximated** Shapley values that are also local regression coefficients. KernelSHAP has been shown to decrease computation times significantly compared to computing classical Shapley values.

Variable	BinaryVariable	Estimate	QueryValue	LowerBound	UpperBound
Intercept		293.37000809	.	.	.
X_01_02	_bCode_X_01_02_	56.099316824	50	33.442153172	66.557846828
X_01_05	_bCode_X_01_05_	17.823189837	4	3.4121432724	4.5878567276
X_01_09	_bCode_X_01_09_	9.0464622609	5	4.3823245572	5.6176754428
X_01_20	_bCode_X_01_20_	5.9031931928	0	-0.044472077	0.0444720768
X_01_24	_bCode_X_01_24_	1.3531534799	3	2.9034053129	3.0965946871
X_01_26	_bCode_X_01_26_	63.22492428	-1	-1.386191369	-0.613808631
X_01_28	_bCode_X_01_28_	0.5285984128	347	327.25247714	366.74752286
X_01_30	_bCode_X_01_30_	10.647232274	1.0604	1.0416521615	1.0791478385
X_02_03	_bCode_X_02_03_	5.0369320978	721	704.45281919	737.54718081
X_02_07	_bCode_X_02_07_	-4.592245358	26	24.468025305	27.531974695
X_02_17	_bCode_X_02_17_	-21.87224486	718	677.99810176	758.00189824
X_02_22	_bCode_X_02_22_	-32.93646117	718	679.14532114	756.85467886
X_02_28	_bCode_X_02_28_	-20.03943703	719	693.38618557	744.61381443
X_03_16	_bCode_X_03_16_	17.728350883	0	-0.076864353	0.0768643528
X_03_20	_bCode_X_03_20_	-5.095857694	0	-0.109831942	0.1098319417

ModelPred	ExplainerPred	ExplainerRMSE
396.22516531	396.22511551	80.852900619

Figure 6. Output for Local Model Using KernelSHAP

The table above displays the Shapley values for the same prediction of interest used in calculating the LIME estimates in Figure 5. While the estimates shown in Figure 6 are regression coefficients for a local surrogate model, they are also the Shapley values for the prediction of interest. A common misinterpretation of these estimates is the idea that when removing a variable from the model, the prediction will change by the same amount of the Shapley value of the removed variable. This interpretation is incorrect because a Shapley value is the contribution to the **difference** between the actual prediction and the mean prediction. By removing a variable from the model, the actual prediction and the mean prediction on the new model, created by removing the variable, will change.

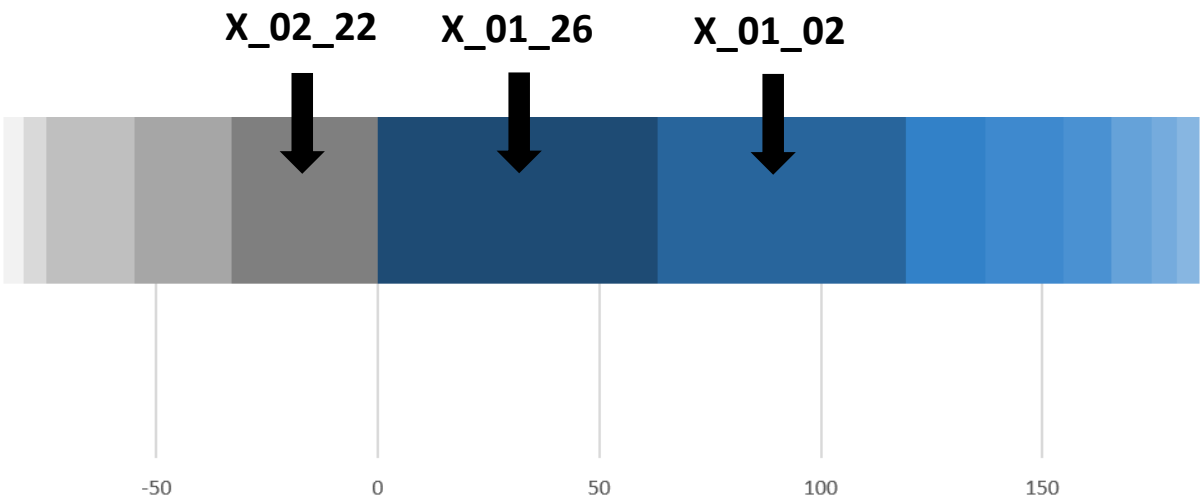
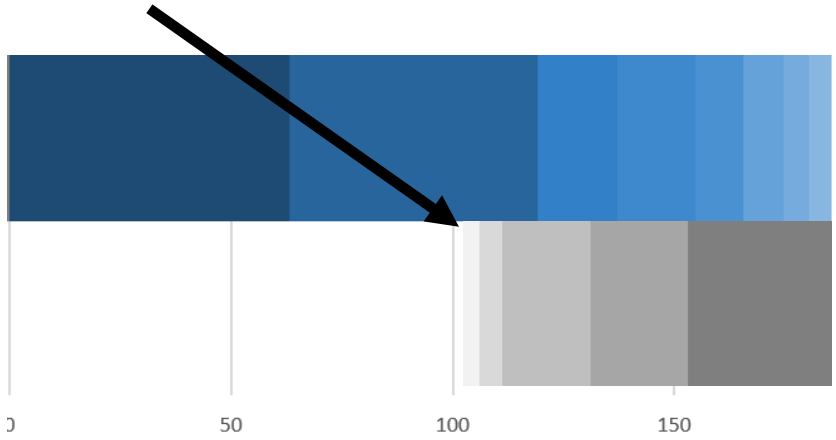


Figure 7. Shapley Values Plot

A better visual representation of the Shapley values given in Figure 7 can be observed in the graph above, where the blue shaded boxes represent the positive Shapley values and the grey boxes correspond to the negative Shapley values. The visual aid clearly highlights that the variables X_01_26 and X_01_02 contribute most to the prediction, and the variable X_02_22 influences the prediction the most negatively. In essence, X_01_26 is the factor that most influences the price of this customer's premium.

The aid also illustrates the fact that the Shapley values sum to the difference between the actual prediction and the mean prediction. By overlapping the grey area (negative Shapley values) onto the blue area (positive Shapley values) the point at which the two areas cancel out each other would be the value of the **difference** between the actual prediction and the mean prediction. In this case that value is 396 (actual) − 293 (mean) = **103**.



The code below generates the Shapley values by employing the action set "explainModel" and KernelSHAP:

```
proc cas;
  loadactionset "explainModel";
  explainModel.linearExplainer /
    table            = "mod_data_3"
    query            = "query"
    modelTable       = "GB_ASTORE"
    modelTableType   = "ASTORE"
    predictedTarget  = "P_TargetO"
    seed             = 1234
    preset           = "KERNELSHAP"
    inputs           = {{name='X_01_02'}, {name='X_01_05'}, {name='X_01_09'},
                        {name='X_01_20'}, {name='X_01_24'}, {name='X_01_26'},
                        {name='X_01_28'}, {name='X_01_30'}, {name='X_02_03'},
                        {name='X_02_07'}, {name='X_02_17'}, {name='X_02_22'},
                        {name='X_02_28'}, {name='X_03_16'}, {name='X_03_20'}}
    nThreads         = 1;
   run;
quit;
```

This action call uses the following parameters:

- table: the original dataset from which the model was trained
- query: a single row from the data to be investigated
- modelTable: the table containing the scoring code for the black-box model
- modelTableType: the type of table containing score code
- predictedTarget: specifies the variable that contains the model's predictions
- seed: seed number to be used for random sampling of data
- preset: specifies the type of model explanation
- inputs: variables to be included in the local surrogate model
- nThreads: number of threads used

In addition to the rapid computation time compared to regular sampling Shapley values, the most significant advantage of KernelSHAP is that it combines the methods of Shapley values and LIME. Since Shapley values are included in the calculations of KernelSHAP, the same desirable properties affixed with Shapley values translate to KernelSHAP.

REASON CODES AND CUSTOMER FEEDBACK

Through the use of the local methods discussed in the previous section, insurers would be equipped to provide reason codes and more complete customer feedback. For example, suppose a customer named John recently inquired about his car premium rate. The insurer could input the customer's information through a black box model in order to reach a prediction regarding the price of John's rate. By applying the local interpretation method, KernelSHAP, to the prediction given by the black box model, the company concludes that the length of time that John has owned the car is the most influential factor affecting his premium. LIME could provide additional insight on which direction John could toggle the value of the most significant variables to improve his rate. If the LIME coefficient corresponding to length of ownership is negative, LIME suggests that an increase in the length of time that John owns his car will lower his premium on average. While the above example only examines the most influential variable, in practice, insurers could provide a complete report of all of the factors affecting a customer's rate in a ranked order of impact for a fully detailed explanation of the customer's premium.

Credit reason codes related to credit scores are well known to most savvy credit card users. Similarly, in insurance industry it is common to use credit-based scores to rate and underwrite customers. The correlation between an individual's financial behavior and insurance risk is very well known. Therefore, credit-based scores are often included in the pricing models in addition to traditional insurance rating variables. As required by FCRA law, insurers are required to disclose up to four adversely impacting credit variables. In the event that one or more of the credit variables in the model result in a significantly large Shapley value for a specific prediction, the company would be able to identify variables strongly impacting the rates. Through the implementation of Shapely in conjunction with LIME, one can identify top four adversely impacting credit variables.

For example, Figure 6 includes a comprehensive list of the Shapley values for the predicted rate of a single customer. In order to determine the features most significantly contributing to the predicted rate, the insurer will examine the four predictors with the largest positive Shapley values, which are X_01_26, X_01_02, X_01_05, and X_03_16. Next, referring to Figure 5, which includes the LIME estimates for the same prediction, the coefficients for X_01_26, X_01_02, and X_01_05 are negative, indicating that an increase in the values for those variables would most likely result in a decrease in the customer's predicted rate. Hence, the insurer could advise the customer, if applicable, to increase the values associated with the variables X_01_26, X_01_02, and X_01_05 in effort to lower the customer's current rate.

CONCLUSION

Machine learning black-box models have shown promise over traditional Generalized Linear Models with respect to model performance, but have yet to be fully implemented in the insurance industry due to the lack of intrinsic interpretability. However, as a result of the development of black-box model interpretation methods, black-box models could be applied to insurance rating for more accurate pricing and better customer feedback. The global methods presented in this paper, such as ALE plots, illuminate the relationship of rating variables to the predicted pure premium that would otherwise be less understood. With knowledge of these relationships, modelers can take measures to improve subsequent model performance leading to more accurate pricing. The local methods outlined in this paper, Shapley values and LIME, enable insurers to communicate to their customers which factors are most adversely affecting customer rates, and potential courses of action in order to lower their current rates.

REFERENCES

Cohen, S. B., Ruppin, E., & Dror, G. (2005, July). Feature Selection Based on the Shapley Value. In *IJCAI* (Vol. 5, pp. 665-670).

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765-4774).

Molnar, C. (2019). *Interpretable machine learning: a guide for making Black Box Models interpretable*.

Wright, R. (2018). "Interpreting Black-Box Machine Learning Models Using Partial Dependence and Individual Conditional Expectation Plots." *Proceedings of the SAS Global 2018 Conference.* Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1950-2018.pdf

Fair Credit Reporting Act. (2018, November 16). Retrieved from https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/fair-credit-reporting-act

ACKNOWLEDGMENTS

I want to thank Zixuan Jiang for providing the data and the software resources necessary for this project, David Podwojski for his managerial guidance during the early stages of this project, and Heather Pierce for her logistical support.

CONTACT INFORMATION

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

Gabe Taylor
State Farm Insurance Companies
309-361-5940
gabe.z.taylor@gmail.com