Innovation and Technology

Interpretable Machine Learning for Insurance

April 2021

Interpretable Machine Learning for Insurance

An Introduction with Examples

AUTHORS

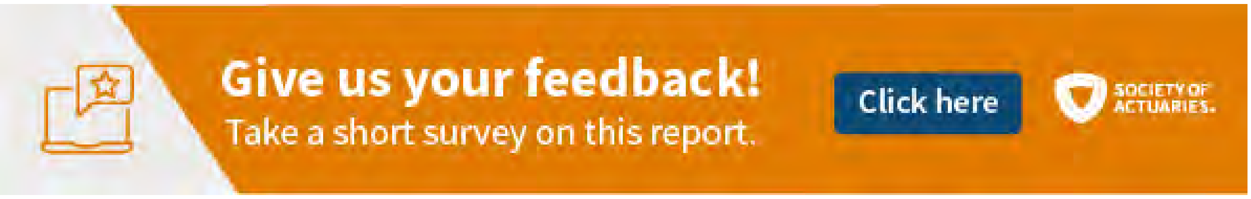
Larry Baeder
Data Scientist
Milliman, Inc.

Peggy Brinkmann, FCAS, MAAA, CSPA
Principal and Consulting Actuary
Milliman, Inc.

Eric Xu, FCAS, MAAA
Actuary
Milliman, Inc.

SPONSOR

Actuarial Innovation and Technology
Steering Committee



Caveat and Disclaimer

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries or its members. The Society of Actuaries makes no representation or warranty to the accuracy of the information.

CONTENTS

Interpretable Machine Learning for Insurance

An Introduction with Examples

Executive Summary

This project was awarded funding by the Society of Actuaries to provide an introduction to the methods and best practices for reviewing machine learning models developed for insurance applications.

Machine learning algorithms fit models based on patterns identified in data and can be very complex. Modelers and users of machine learning models must carefully test models to avoid overfitting to the data used to train the model. Many, but not all, of the best practices used to review and validate linear models can also be applied to machine learning models, but a significant challenge for machine learning models is interpretability. Understanding the relationships between the model inputs and outputs builds additional confidence in any model but is particularly challenging for many types of machine learning models. And in many applications, it is necessary for users and reviewers of models to understand whether the relationships between the model inputs and outputs make business sense, are compliant with laws and regulations, and/or can be explained to individuals impacted by the use of the model.

In this report, we describe and illustrate a range of methods for interpreting machine learning models from the growing field of Interpretable Machine Learning (IML). These methods provide a way to understand and analyze a model that is not inherently transparent. Each method is not tied to a specific modeling algorithm, so they can be applied to any modeling problem. The methods fall into the following categories:

- Feature importance
- Methods to understand relationships between model inputs and outputs (main effects)
- Methods to identify and visualize interaction effects

Some methods are limited to only one category while others apply to multiple categories. Feature importance can be used to understand variable significance and justify a feature's inclusion or exclusion from the model. Metrics that are directly related to the impact on prediction error or decompositions of model contributions to the predicted value are preferable. Methods based on permutation can be distorted when features are correlated.

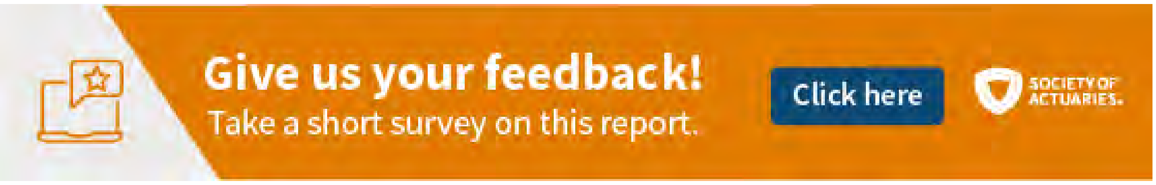
Methods to understand the main effects help to visualize the relationship between predictor variables and model output. Methods based on the marginal distribution of an individual variable, such as partial dependence plot (PDP) or individual conditional expectation (ICE), can be distorted by variable correlations. Accumulated local effects (ALE) is an alternative method that attempts to address the issue of correlation in the features but does not work as well for categorical variables. SHAP values avoid these problems but are computationally intensive. That said, there are efficient algorithms available to calculate approximate SHAP values for some types of models.

Methods to identify the most important interactions in a model include the H-statistic, ICE plots, and SHAP values. The H-statistic is computationally intensive and can be misleading when dependent relationships or correlations are present. ICE plots indicate the presence of interactions, but do not identify the specific

variables involved. The SHAP interaction calculation is a more reliable measure of interactions. Methods used to visualize main effects, such as PDP and SHAP plots, can be adapted to visualize interaction effects.

Each method is illustrated with examples. This report is accompanied by code prepared in an R Markdown file that creates these examples. The data used for these illustrations is the 2014-15 Individual Life Experience Report that was published by the SOA[1].

Additional IML methods outside the scope of this paper include surrogate decision trees and local interpretable model-agnostic explanations (LIME). Fairness and disparate impact are important tests for machine learning models; best practices for testing for fairness and remedies for disparate impact are not covered in this paper.



[1] https://www.soa.org/resources/research-reports/2019/2009-2015-individual-life-mortality/

Section 1: Introduction

Insurers vie for market share and profitability by appropriately setting premiums commensurate with predicted losses and expenses for each policyholder, as well as carefully selecting risks. Insurers that utilize accurate predictive models can avoid overcharging for low-risk policies and undercharging for high-risk policies.

Certain types of insurance that are legally required for consumers, such as auto and home insurance, typically face a high amount of regulatory scrutiny. Insurers in the United States often must obtain state regulatory approval of their pricing models before being able to sell new insurance products or change pricing of existing products. Beyond that, model users must feel assured that the variables and the relationships between variables used in the models are logical and intuitive, both to themselves, as well as to stakeholders affected by the model results. In the case of pricing models, price stability and moral hazard are among other important considerations.

The combination of predictive power and interpretability have historically made generalized linear regression models (GLMs) a popular choice for insurers. However, advances in computing power and the vibrant open-source communities of Python and R have exposed insurers to the prospect of using new types of models with the possibility of increased predictive power. Gradient boosting machines, random forests, and neural networks have become more commonplace in modern business applications as they offer alternative methods of prediction over simpler linear methods. However, these complex models, often referred to as "black boxes," can be difficult to understand or interpret, creating a difficult hurdle to their adoption.

In this paper, we aim to survey techniques that can enable users to interpret more complex models. We first outline the current state of predictive modeling standards and best practices within the insurance industry. We then explore concepts from the emerging field of interpretable machine learning which we believe can assist actuaries better understand and review complex models. We utilize a Gradient Boosted Machine (GBM) model fit on life insurance mortality data to illustrate the use of interpretable machine learning techniques.

While these methods can help provide insights into complex models, this is not an exhaustive list of possible procedures. However, we believe they could be a useful part of a larger undertaking where new best practices are established.

The authors of this paper are familiar with the US environment and will make reference to the US regulatory framework, NAIC, state regulators and other elements that are specific to the US environment. However, we assume the techniques articulated here are not dependent to a particular jurisdiction.

1.1 EXISTING STANDARDS AND BEST PRACTICES FOR MODEL REVIEW

Actuarial Standard of Practice 56 - Modeling (ASOP 56) was adopted October 1, 2020 to provide some high-level guidance applicable to actuaries designing, developing, selecting, modifying, using, reviewing, or evaluating models. ASOP 56 defines a model as "a simplified representation of relationships among real world variables, entities, or events using statistical, financial, economic, mathematical, non-quantitative, or scientific concepts or equations."[2] Recommended practices include consideration of whether there is a

[2] http://www.actuarialstandardsboard.org/asops/modeling-3/#28-model, Accessed 2/12/2020.

material risk of the model overfitting the data and the reasonableness of the model output. That said, ASOP 56 does not prescribe any specific metrics or diagnostics.

Regulatory requirements for predictive modeling review lack standardization across jurisdictions, and state insurance departments in the US vary in their desire and capacity to be able to review predictive models in depth. Moreover, state insurance laws rarely contain specific predictive modeling documentation requirements. To facilitate model review, insurance regulators in several states have begun to request additional detail related to predictive models used by insurers[3], in some cases requiring predictive modeling checklists for insurers to complete as part of the rate filing process.

In the US, the National Association of Insurance Commissioners (NAIC) Property and Casualty Insurance Committee has adopted a white paper entitled *Regulatory Review of Predictive Models*, intended to provide state insurance departments with best practices related to the review of GLMs used in auto and homeowners insurance ratemaking.[4] The paper was written, in part, due to the lack of guidance from the ASOPs that specifically addresses the use of models such as GLMs, and may help encourage consistency in regulatory requirements from state to state. However, the paper focuses on the review of GLMs and does not explicitly detail guidance for non-linear models.

While many of the information elements for reviewing GLMs described in the white paper can also be provided for other types of models, some items that are not easily obtainable for complex models include:

- Tests of individual variable significance, such as confidence intervals, chi-square tests, p-values, or F-tests
- Illustrations of the relationship between each characteristic/rating variable and model output/risk of loss, and a rational explanation for why an increase in a predictor variable should increase or decrease the outcome being predicted (e.g., frequency, severity, loss costs, expenses)
- Explanation of rationale and impact of variable interactions in the model

According to the NAIC best practices, overall lift charts may not provide enough insight into how the model is operating. The items above are useful to a regulator and/or other model stakeholders because they can help to establish confidence that the relationships in the model are not spurious, temporary, or limited to the specific datasets, and that there is a reasonable and stable relationship to the predicted outcome.

These efforts highlight insurers' desire to use more predictive models and insurance regulators' desire to ensure compliance with relevant state laws and regulations. Nevertheless, best practices for review of more complex models remain a challenge.

1.2 INTERPRETABLE MACHINE LEARNING

Lack of interpretability has been a barrier to the widespread adoption of complex, non-linear models, especially in highly regulated industries such as insurance. While there is no method to make a black box model fully transparent, the growing field of interpretable machine learning (IML) is developing techniques to make the black boxes less opaque.

[3] https://www.insurancejournal.com/blogs/right-street/2019/10/27/546717.htm, Accessed 2/7/2020.

[4] https://content.naic.org/sites/default/files/inline-files/9-15%20CASTF%20-%20Predictive%20Model%20White%20Paper%209-09-2020.pdf

Open-source IML packages make building complex models and applying model interpretation methods very easy with only a few lines of code. As a result, it may be tempting to blindly trust the results of those default methods without considering other methods or identifying potential pitfalls.

In the sections that follow, we will review multiple interpretation methods and provide examples to illustrate their advantages and disadvantages. We remind the reader that none of these methods will be sufficient on their own to fully interpret a model.

The various tests and technical validations should give the technical team (model user, model builder) insight into how the model is behaving and what variables and features are critical to the model results. There is no single scoring variable with a passing/failing grade and each technique provides insight within its level of validity. These technical elements could be selected and contextualized when communicating with a less technical audience to increase their confidence level with the model results. Understanding these tests and technical validations are also useful beyond the technical team to understand and interpret the model results provided.

Section 2: Overview of Mortality Case Study

2.1 CASE STUDY DATA

The SOA's Individual Life Experience Committee (ILEC) published the 2014-15 Individual Life Experience Report. This dataset includes composite data from 2009-2015. For the case study, the data was restricted to term life insurance policies that were within the initial policy term, issued after 1980, and the issue age was at least 18 years old. Term life mortality is well understood in the life insurance industry, which makes it a good case study for this paper as it is easier to identify poorly fit models and problematic areas in the interpretation methods.

The features used in the mortality model are:

- **Attained age** – the sum of the policyholder's age at policy issue and the number of years they have held the policy.
- **Duration** – the number of years (starting with a value of one) the policyholder has had the policy.
- **Smoking status** – if the policyholder is considered a smoker or not.
- **Preferred class** – an underwriting structure used by insurers to classify and price policyholders. Different companies have different structures with the number of classes ranging from two to four. The lower the class designation, the healthier the policyholders who are put into that class. Thus, someone in class 1 of 3 (displayed as 1_3 in this paper) is considered healthier at time of issue than someone in class 3 of 3.
- **Gender** – A categorical feature in the model with two levels, male and female.
- **Guaranteed term period** – the length of the policy at issue during which the premium will remain constant regardless of policyholder behavior or health status. The shortest term period in the data is five years with increasing lengths by five years up to 30 years. Term period is used as a categorical feature with six levels.

Smoking status and Preferred class are gathered at policy issue. As the policy ages, the effect of these variables is expected to decline as the underwriting effects "wear off" and mortality rates regress toward the mean. The conventional approach for designing mortality tables is to use issue age for the initial years of the policy ("select period") and attained age for the subsequent years ("ultimate"). To streamline our examples, we focused on a model using attained age only.

The response variable used in this case study is mortality rate, defined as the number of deaths divided by policies exposed. Policies exposed was used as a weight in the model.

2.2 CASE STUDY MODEL

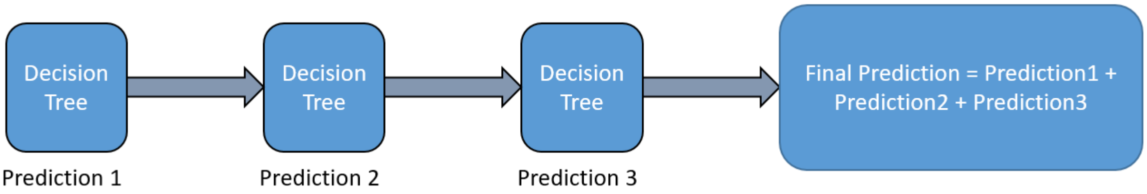
For the case study in this paper, we used the eXtreme Gradient Boosting (XGBoost) implementation of the GBM as an example of a complex "black box" model. Gradient Boosting Machines (GBMs) are a popular machine learning framework, widely used because of their predictive power, flexibility in implementation, and adaptability to many different problem types.

XGBoost models[5] are constructed using an iterative process, loosely depicted in Figure 1. The boosting process begins with a single model, usually a decision tree or simple average model. The model predictions

[5] For more details on gradient boosted trees: https://xgboost.readthedocs.io/en/latest/tutorials/model.html

are evaluated and another model is fit on the residual errors of the first model. This subsequent model improves the predictions incrementally based on the learning rate specified in the model. This process is repeated either for a set number of iterations or until another stopping criteria is met. This process creates a single predictive model stemming from multiple decision trees.

Figure 1
XGBOOST ITERATIVE PROCESS.



In machine learning algorithms, hyperparameters are specified that control the model-fitting process. For GBMs, hyperparameters define the degree of complexity of each decision tree, such as the tree depth and number of observations in a terminal node, and control the model-iteration process in a variety of ways, such as how strongly the model responds to poor predictions (known as the "learning rate") and criteria for stopping the iterations.

In any modeling problem, overfitting is one of the key dangers and GBMs are easily overfit due to their iterative nature. Overfitting occurs when a model performs well on a specific dataset but is not generalizable to the population that data represents. When a model is overfit, it is fitting to the noise in the data.

While properly fit GBMs can create accurate predictive models, the predictive power comes at the cost of interpretability. In a GLM, one can determine the amount that each variable contributes to the prediction for any given observation by examining model coefficients. GBMs have no coefficients like GLMs that can be interpreted to understand predictions. GBM predictions are often the result of hundreds of decision trees that have been aggregated to produce a single prediction. As a result, we cannot easily explain the relationship between the model inputs and outputs.

To reduce run times, we fit XGBoost models that predicted mortality rates based on the policyholder characteristics on a 10% random sample, or 529,708 observations. This data was split into a 60% training set and 40% test set using simple random sampling. Larger sample sizes were tested, but there were no significant differences in the results. A Poisson count objective function was used to model the target variable of mortality rate, number of deaths divided by policies exposed, with the case weight of policies exposed. The target variable is the value that the model predicts, and the weight is the amount of emphasis the model gives to each particular observation. Grid search with cross validation was performed to tune the hyperparameters for the final model. Specific details of the modeling process can be found in the accompanying code provided with the report.

Section 3: Feature Importance

Confidence intervals, chi-square tests, p-values, or F-tests quantifying the statistical significance of individual predictor variables are not available for methods such as GBMs. Instead, many algorithms provide various measures of feature importance, which rank the variables from most important to least important. None of these measures provide insights into the direction of the relationship between the predictor variables and the model output. There are several evaluation metrics to choose from, each with their own advantages and disadvantages.

3.1 GAIN, COVER, AND FREQUENCY

3.1.1 DESCRIPTION

XGBoost's built-in feature importance method has three evaluation metrics: gain, cover, and frequency. Gain represents the overall improvement in prediction accuracy by using that feature in the decision trees of the model. Cover represents the proportion of observations that are influenced by splitting on a feature in the decision trees. Finally, frequency is how often that feature is used to split the data in the overall model (Chen & Guestrin, 2016).

The feature importance rankings by gain, cover, and frequency can vary. Gain is considered the most useful because it directly measures improvement in the prediction, while cover and frequency do not directly measure predictive power. While these measures provide a simple global view of which features are most important to the model, they have the following disadvantages:

- Correlated features may have their importance reduced when they contain similar information. Consider two highly correlated features, X1 and X2. A single decision tree model may select a split on X1 or X2; when the variables are correlated, the choice of split can be influenced by relatively small differences in the dataset, such as a different sort order or random sampling. When a GBM trains its multiple decision trees, some of the model's individual decision trees may choose to split on X1, others may choose to split on X2. Consequently, the relationship the model is finding may be highly important between its response variable and the correlated features X1 and X2, but the associated features would individually have lower-ranked importance relative to the actual importance.
- These measures do not come with tests of statistical significance of the variable contribution.

3.1.2 CASE STUDY EXAMPLES

Table 1 shows the three measures for the mortality model with the features ranked by gain by default. The values are automatically converted into proportions by XGBoost, with each column totaling to one to aid comparison between features.

Table 1
XGBOOST FEATURE IMPORTANCE

	Gain	Cover	Frequency
Attained Age	0.772	0.822	0.397
Preferred Class	0.131	0.125	0.286
Duration	0.054	0.016	0.171
Smoker	0.025	0.035	0.062
Gender	0.013	0.001	0.049
Guaranteed Period	0.004	0.000	0.034

Overall, the feature importance results are consistent across the different metrics with only one reversal in ordering. All the feature importance methods (gain, cover, frequency) indicate attained age as the most important feature in this model. This result is consistent with the general understanding that mortality rate is strongly related to age. After attained age, preferred class is the next most important variable by all metrics.

All the metrics except cover have duration as the third most important feature followed by smoker. Ranking by cover flips these two with smoker being third most important and duration being fourth. This is an example of how the rankings can differ depending on the metric. Gender and guaranteed period are fifth and sixth most important for all metrics.

When analyzing feature importance, sometimes questions arise as to why a feature is higher or lower in the rankings than expected. To better understand the effects of the individual features, we need to dive into the main effects and interactions to understand how each feature is influencing the model predictions.

It is important to keep in mind the previous discussion regarding correlated variables when analyzing the above table. To the extent, for example, that attained age and duration are related, this will impact the gain shown by one variable versus the other and can give a false display that one variable is more predictive than the other.

3.2 PERMUTATION FEATURE IMPORTANCE

3.2.1 DESCRIPTION

An alternative to the default XGBoost feature importance is permutation feature importance. Permutation feature importance evaluates the change in overall model error rate when a feature’s values are replaced with random values, breaking the relationship between the feature and the target. If a feature is important, then the shuffling of the values would meaningfully increase the model error. The features are then ranked based on the change in model error rate. The calculation is described below (Molnar, 2019):

1. Calculate the model error rate with original data, X: $e_0 = L(y,\ f(X))$ where $f(X)$ is the model prediction; y is the actual value of the target variable; $L(\,)$ is the loss function, which for the mortality study is the Poisson negative log-likelihood function.
2. For each feature in model, $j\ =\ 1, \dots, p$:
    a. For a set number of permutations, $i\ =\ 1, \dots, m$
        i. Create new model matrix, X_{perm} , with values of feature j permuted.
        ii. Calculate new model error rate: $e_{j,i} = L\left(y, f\left(X_{perm}\right)\right)$
        iii. Calculate the feature importance as ratio, $FI_{j,i}\ =\ e_{j,i}\ /\ e_0$, or difference, $FI_{j,i}\ =\ e_{j,i}\ -\ e_0$.

Multiple permutations are needed to average the differences or find confidence intervals for the error rates. For most models, it would take an unreasonable amount of time to compute the error rates for all possible permutations. Therefore, a small number of permutations is often used, with a common default being five. The final value may be stated as a ratio or a difference, but the feature importance ranking will be the same. The ratio calculation has the advantage of being comparable across models and is easier to interpret.

Advantages of permutation feature importance include:

- The use of multiple permutations allows for the creation of confidence intervals for the feature importance metrics. The range of error rates may be quite large, so quantifying the uncertainty can be important.
- Permuted feature importance captures both main and interaction effects.

Disadvantages include:

- Permuting correlated features may result in the creation of unrealistic observations. This may result in bias in the feature importance metrics.
- Correlated features may have their importance reduced when they contain similar information.

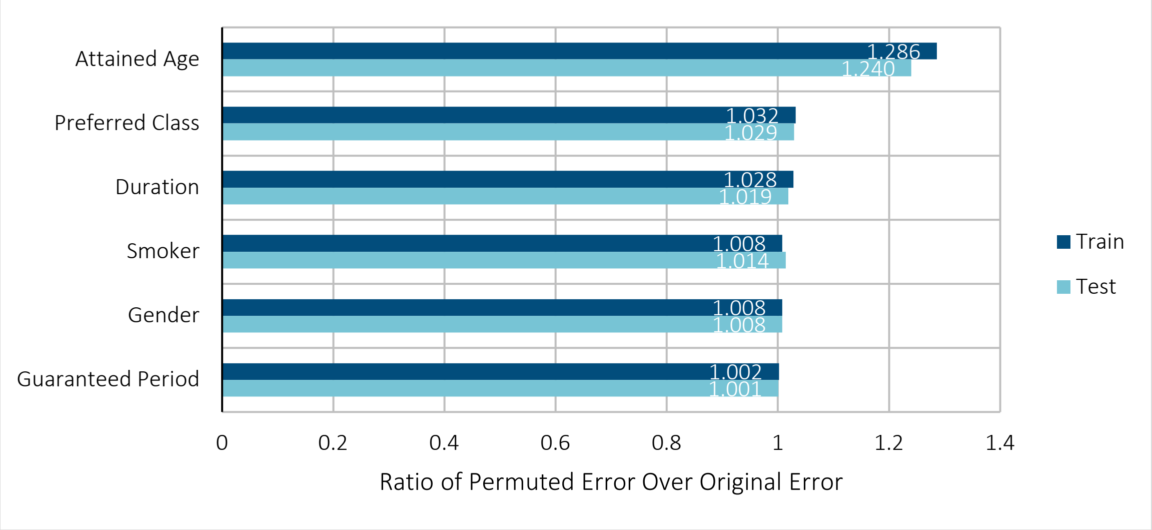
In GLM modeling, statistical significance is the standard method with which to remove features with spurious relationships. While multiple permutations can be used to create confidence intervals on feature importance metrics, it is possible for a variable to have a relatively high feature importance on the training data but not on test data. Permutation feature importance metrics calculated on test data can provide additional insight into potential overfit and/or spurious correlations in the model. Another technique is to introduce random variables into the model and examine their importance relative to other features. Using feature importance in these ways can provide a practical method to identify overfit and remove potentially non-significant features.

3.2.2 CASE STUDY EXAMPLES

Figure 2 compares permutation feature importance for both the train and test data. The variable rank for the permutation feature importance is the same as the unpermuted gain metric from Table 1 and the rank is consistent between train and test data. We do need to interpret the metrics with caution, however, because of unrealistic combinations of features that are created by the permutations. For instance, policyholders with an attained age of 18 can only have a duration value of one. When the values of duration are permuted, an observation with an attained age of 18 may have their duration set to a larger number that is not possible.

Figure 2

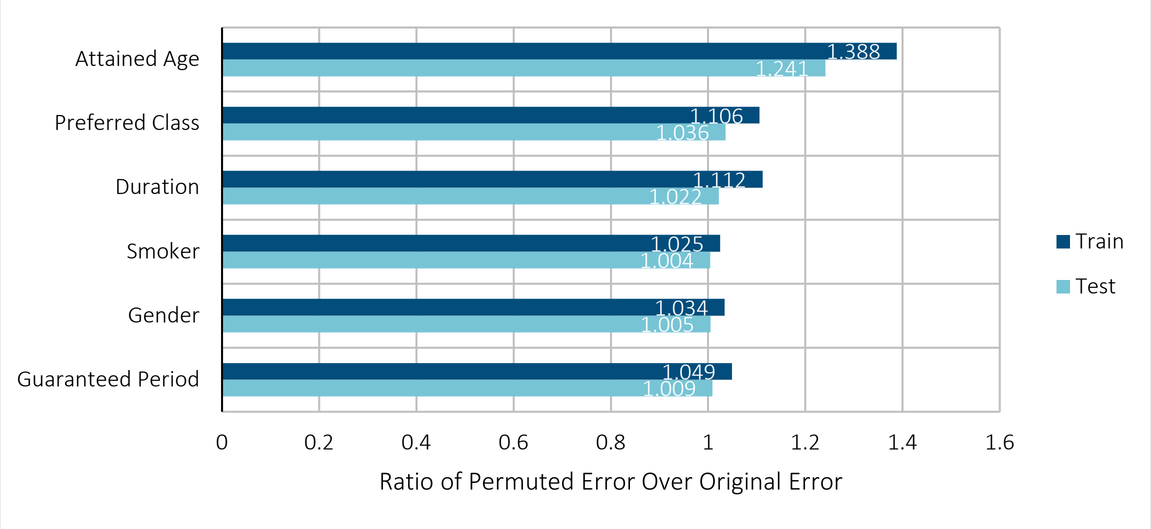
PERMUTATION FEATURE IMPORTANCE FOR MORTALITY MODEL



In Figure 3, the permutation feature importance is shown for a model that was intentionally overfit. All features have a large difference between the train and test feature importance values. For instance, with attained age, the model error rate increases by 39% on the training data, but only 24% on the test data. When the difference between the train and test becomes large, it suggests that the model could be overfit on the training data and the features have an inflated importance.

Figure 3

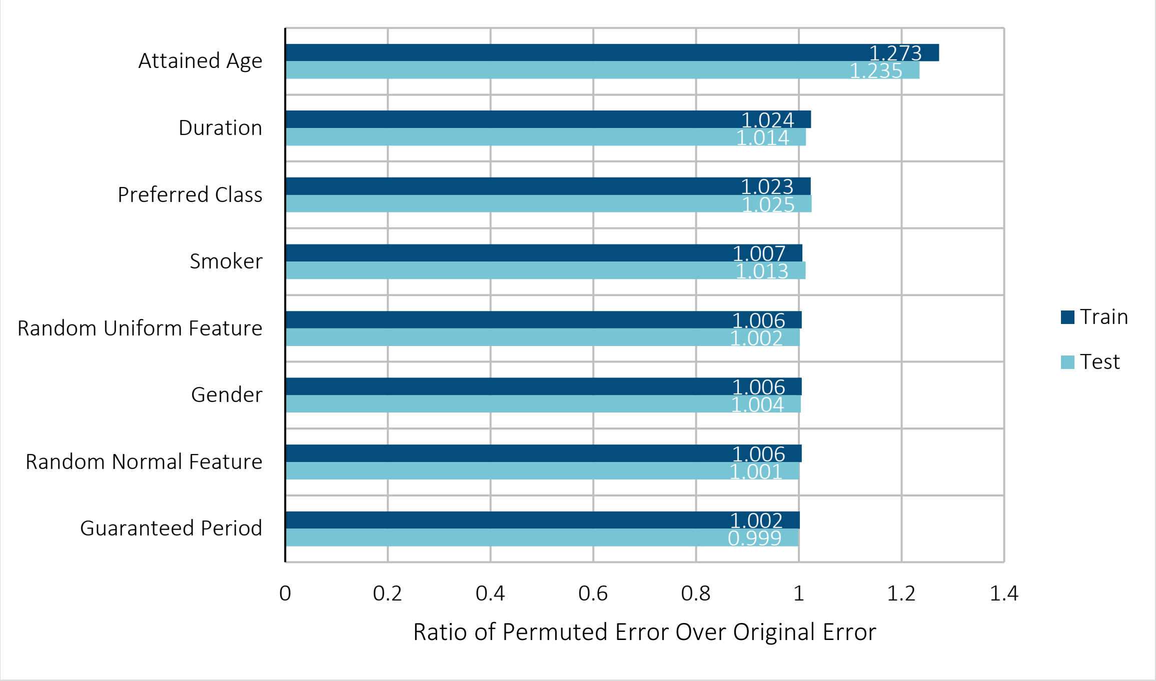
PERMUTATION FEATURE IMPORTANCE FOR OVERFIT MORTALITY MODEL



As another diagnostic, we fit a new model with two random features, one from a uniform distribution and one from a standard normal distribution. Figure 4 shows that the random features were more important than gender and guaranteed period on the train data, suggesting that these variables may not be significant and reliable predictors in this model.

Figure 4

PERMUTATION FEATURE IMPORTANCE FOR MORTALITY MODEL WITH RANDOM FEATURES



Section 4: Main Effects

While feature importance provides a summarized ranking of which features highly influence the model, it does not provide enough information for users to understand the individual effects for each feature in the model.

In a GLM context, the coefficient value of each feature can be used to understand the direction, magnitude, and shape of a feature's effect on the predicted value. This is not the case in more complex models such as XGBoost. As a result, alternative methods of analysis will be required. The following interpretable machine learning techniques aim to provide insights into the relationship between a change in a predictor variable and the model output for a complex model.

4.1 PARTIAL DEPENDENCE PLOTS

4.1.1 DESCRIPTION

The partial dependence plot (PDP) computes the marginal effect of a given feature on the prediction within a predictive model. This is accomplished by fixing the value of the predictor variable of interest, calculating the model prediction for each observation holding the variable of interest constant, and calculating the average model prediction across all the observations. This is repeated for all the values of the predictor variable and the results are plotted. The process is described below (Friedman, 2001):

1. For each level, i, of the selected feature (continuous variables are binned):
   a. For all observations, modify the value of the selected feature to i.
   b. Using the modified observations and the existing model, predict the response variable value for every observation.
   c. Calculate the average predicted values for all observations.
2. Plot the average predicted values for each level (y-axis) against the feature levels (x-axis).

For a basic GLM without interactions or other non-linear regressors, the PDP algorithm would yield straight-line relationships with the slope equal to the feature's coefficient. In any non-linear model such as the XGBoost model, however, PDP can display more complex relationships and enable users to estimate the general direction, magnitude, and shape of a feature's influence.

When comparing multiple PDPs for different features, it is important to note the magnitude of the y-axis between plots. By default, PDPs are usually plotted on an appropriate scale for only that feature and not on a common scale that allows for comparison between features.

Advantages of PDP include:

- PDP are calculated using a short algorithm that is easy to implement and understand.
- When the feature is uncorrelated with other features, PD plots directly show how the prediction is affected by changes in the feature value.
- The plot represents the model's interpretation of the data by stating the change in prediction if a feature is set to an alternative value.

Disadvantages:

- PDP can be computationally expensive as it recalculates predictions on the entire dataset per point on a single feature plot.
- PDP assumes independence between features, so correlation between features can cause potentially misleading interpretations. A review of feature correlation values is recommended prior to running a PDP.
- The calculation in the PDP is an average and is, therefore, susceptible to any skewed data or outliers.

The level of confidence we have in the estimate of the effect for a given level of a feature is determined by the distribution of observations for that feature and the comparison of its PDP to other interpretation methods. With more data, the model is more likely to find correct relationships and the PDP calculation is more likely to be accurate.

The partial dependence algorithm can be applied to more than one feature at a time. The algorithm cycles through all the possible combinations of the features and finds the predicted values for all observations. This is a way to potentially find interactions between features. If two features are used, the traditional visualization is to put the values into a heatmap to show how the prediction changes across the levels of the two features. Line charts can also be used but can become cluttered, so a subset of levels can be more useful. There are currently no practical methods for visualizing relationships among three or more features.

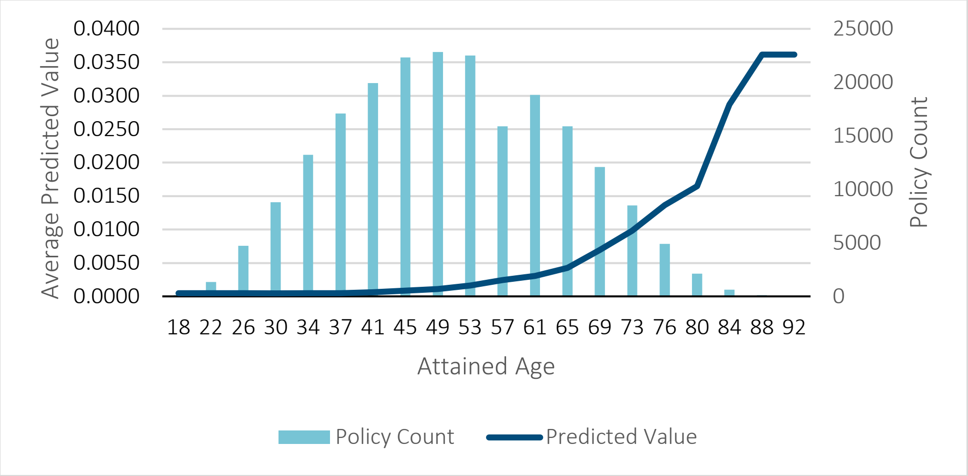
For the purpose of PDP interpretation, it is critically important to compare the PDP predictions against the distribution of values for the feature. First, the model itself is unlikely to be able to accurately generate predictions in areas of the distribution where data is sparse. Second, the PDP calculation in those relatively sparse areas will be computed largely using data that does not actually contain those values, resulting in less precision. The same model may generate more precise predictions if those areas were more heavily populated due to more credible data. Therefore, caution is recommended when interpreting PDPs in areas along the feature distribution where there are relatively few observations.

4.1.2 CASE STUDY EXAMPLES

Figure 5 displays the PDP by attained age and the relative sparse data for the youngest and oldest ages.

Figure 5

PARTIAL DEPENDENCE PLOT AND POLICY COUNT HISTOGRAM FOR ATTAINED AGE



Like permutation feature importance, the independence assumption of the PDP is a primary concern when interpreting and applying PD plots. If features are correlated or have a dependent relationship, then forcing the individual features to independently take all possible values creates unrealistic observations in the process. These observations may cause a distortion in the perceived marginal effect shown in the resulting PDP.

Figures 6 and 7 below show the PDPs for duration and preferred class, respectively.

Figure 6

PARTIAL DEPENDENCE PLOT AND POLICY COUNT HISTOGRAM FOR DURATION

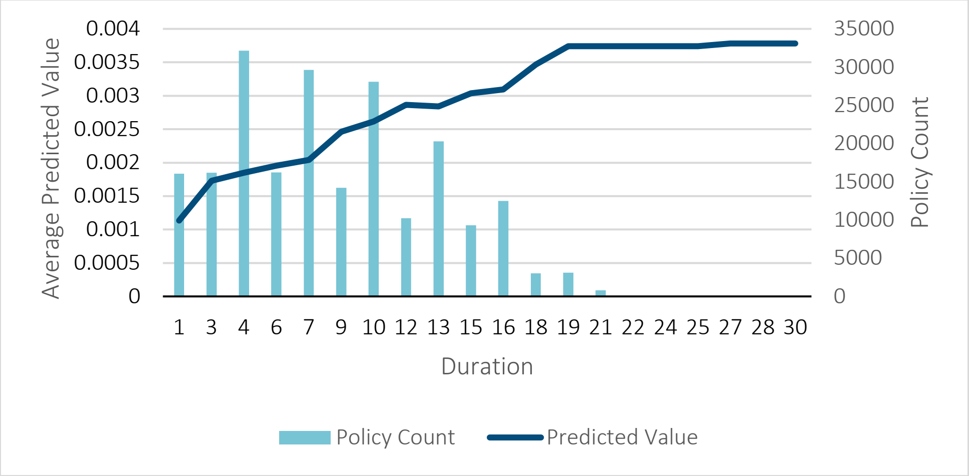
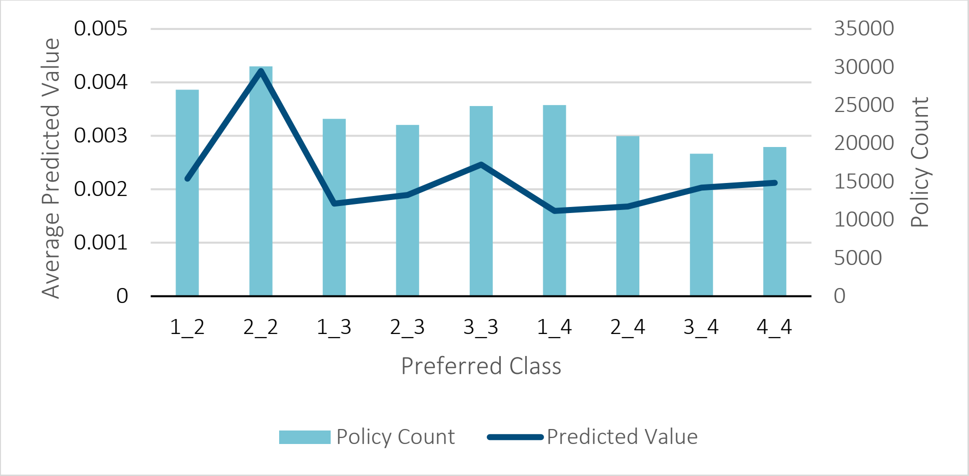


Figure 7

PARTIAL DEPENDENCE PLOT AND POLICY COUNT HISTOGRAM FOR PREFERRED CLASS



In Figure 7, the PDP for preferred class shows an increasing mortality risk as the ranking within a preferred class increases, 2_2 has higher mortality than 1_2. This is consistent with the structure of preferred classes where the lowest number in the preferred class is composed of the healthier subsets of the population. This plot shows that the model identifies the underlying structure of the preferred classes where the healthiest risks are placed in class 1 (e.g., 1_2, 1_3, or 1_4).

4.2 INDIVIDUAL CONDITIONAL EXPECTATION

4.2.1 DESCRIPTION

The partial dependence algorithm finds the average predicted value if each observation was given every possible value of a feature. The Individual Conditional Expectation (ICE) uses the same process, but does not average predicted values, and all the hypothetical observations are plotted (Hall & Gill, 2019). This results in predicted values for each observation across the range of feature values and moves the analysis from the aggregate level to the individual level, seeking a more nuanced view of the feature's effect. The ICE plot may find relationships that are lost due to the aggregation in the PDP algorithm, which is the average of the ICE plot.

Advantages of ICE plots are:

- More intuitive than the PDP because each line represents one observation or group of similar observations.
- Ability to show different effects for different levels of features that are lost in the averaging of the PDP. These differing effects may be indications of interactions and indicate new avenues of investigation.

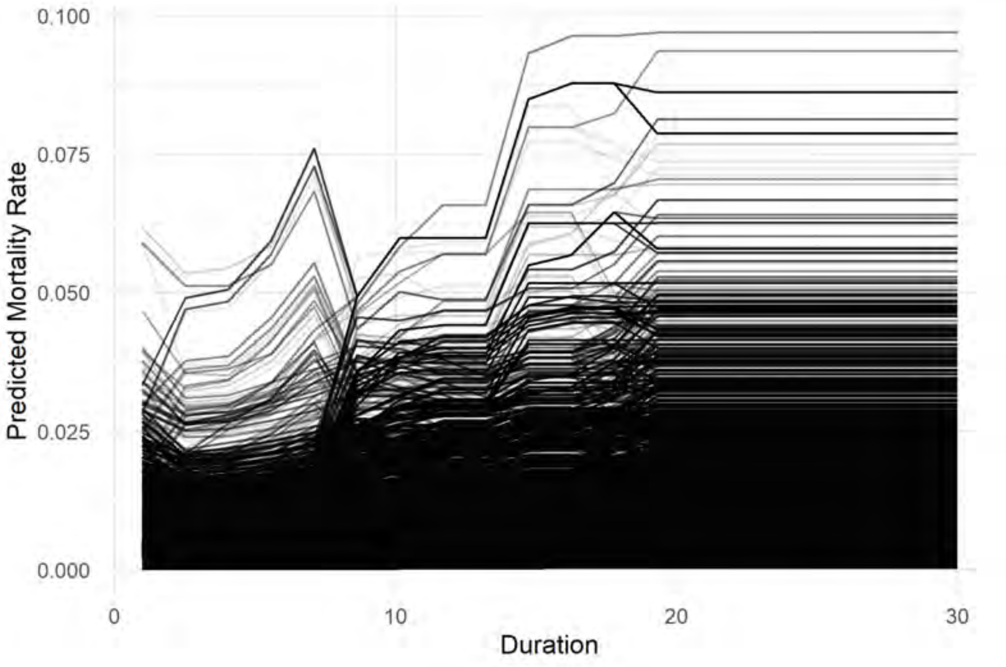
Disadvantages include:

- The ICE plot can only show one feature at a time with no direct measurement of interactions.
- Correlation in the features is problematic for the same reasons as PDPs.
- The plot can be overwhelming with too many lines to decipher specific trends. Sampling observations can help, but at the loss of detail.

4.2.2 CASE STUDY EXAMPLES

As seen in Figure 8, the ICE plot can be difficult to read when there is a large amount of data. Formatting the lines to be partially transparent can help in readability, but the plots may still appear dense and difficult to read. The particular lines can be noisy with counter-intuitive results due to the granularity of the ICE plot. This noise is one of the main arguments for the use of PDPs over ICE plots.

Figure 8
ICE PLOT FOR DURATION

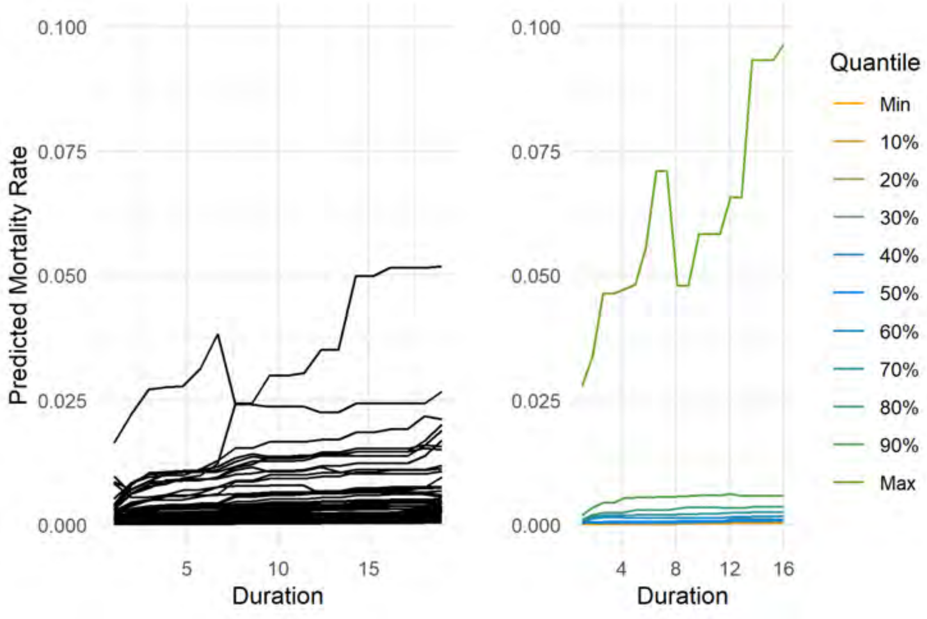


One potential solution for the denseness of the ICE plot is to plot a sample of all observations, which can be seen in the left graph of Figure 9. This makes the ICE plot more readable but comes at a cost: the output can be highly variable and highly dependent on the sample of observations. Sampled ICE plots may potentially miss relationships that don't arise in the sample. The variance can be reduced by increasing the sample size, but this reintroduces clutter to the ICE plot.

An alternative method is to sample values around the quantiles of the predicted value, illustrated on the right in Figure 9. This method involves finding a set of observations around quantiles of interest (e.g., 10%, 20%, etc.) and usually the minimum and maximum predicted values.

Figure 9

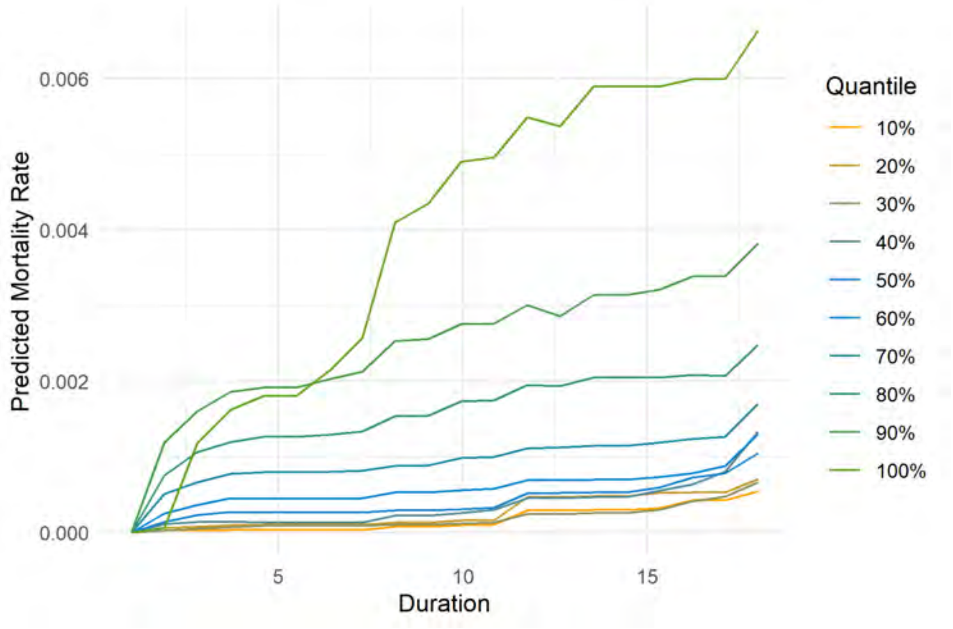
SAMPLED ICE PLOT, RANDOM SAMPLE (LEFT) AND QUANTILE SAMPLE (RIGHT)



A final alternative is to randomly sample multiple observations from each predicted value quantile and average the ICE within each quantile, rather than selecting a single record to represent the quantile. This is shown below in Figure 10. Since we are interested in comparing the relative relationship between quantiles, Figure 10 was centered so each line starts at zero for easier comparison.

Figure 10

CENTERED ICE PLOT FOR RANDOM SAMPLE OF 10 OBSERVATIONS FOR EACH DECILE OF PREDICTED VALUES



When we see the shape of the relationship vary by quantile, this is an indication of potential interaction effects. In other words, the subgroups identified by the various quantiles have a different relationship between model input and model output. In Figure 10, we see that, for the higher-risk groups, duration has a bigger impact on the predicted mortality than for the lower-risk groups. This is an indication (but not necessarily proof) of an interaction and should be investigated.

4.3 ACCUMULATED LOCAL EFFECTS

4.3.1 DESCRIPTION

Neither PDP nor ICE can be fully trusted in the case of correlated features. Accumulated Local Effects (ALE) avoids the independence assumption by calculating and accumulating smaller, incremental changes in the feature effects. ALE shows the expected and centered effects of a feature, like a coefficient in a GLM.

The ALE statistic divides a continuous variable into smaller intervals to create the conditional distribution where the change in predicted value effects is accumulated over the preceding intervals to find the main effect for the feature. The procedure to calculate ALE is as follows (Apley & Zhu, 2016):

1. Divide the feature into a set number of intervals (usually based on quantiles).
2. For all observations within each interval:
    a. Find the model predicted value setting the feature value equal to the upper limit of the interval.

   b. Find the model predicted value setting the feature value equal to the lower limit of the interval.
   c. Calculate the difference in predicted values between the upper limit and lower limit calculations (by subtracting step b. from step a., i.e. [a.] – [b.]).
   d. Calculate the average of the differences in predicted values (from step c.) across all observations in the interval.
   e. Sum all preceding intervals including the current interval.
3. Center the interval values by subtracting a constant so that the mean value is zero.

The ALE calculation requires an ordering of the feature values. For continuous features, the intervals are based on the quantiles of the feature values and, therefore, have a natural ordering to them. Categorical features may not have a natural order; therefore, an artificial ordering must be created for categorical features prior to running the ALE calculation.

The first important characteristic of the ALE algorithm is the use of the conditional distribution. As discussed earlier, correlation in the features can cause the PDP to create unrealistic observations that are used in its calculation. By using the conditional distribution for any given value, the ALE will only estimate the main effect using similar and realistic observations. As an example, in the mortality model only similar attained ages with appropriate durations would be used to estimate the feature main effect at any level.

Advantages of ALE include:

- Unbiased estimation of feature main effect that is not susceptible to correlation between features.
- Faster computation time based on the number of intervals in a continuous feature or levels in a categorical feature.
- Clear interpretation that is analogous to the interpretation of a GLM coefficient.

Disadvantages include:

- An order is imposed on categorical features for the purposes of the calculation even though the feature has no ordering.
- The number of intervals for a continuous feature affects the accuracy of the curve. If there are too many intervals, then insignificant noise will enter into the plot, while too few intervals will smooth the curve and hide results.
- Interaction or second-order effects require the data to be binned by two variables, which will usually create varying count sizes. The different count sizes in the bins will result in varying degrees of accuracy. These effects also require the main effect to interpret the entire relationship to the prediction, so the interaction plot is not interpretable by itself.

4.3.2 CASE STUDY EXAMPLES

Figure 11 shows the distribution and ALE plots generated for attained age. The distribution for attained age is different than the distribution for PDP because the bucketing for the histogram is different, but the overall shape of ALE is similar to PDP for attained age. The ALE value in Figure 11 for attained age 32 is -0.0023, which means that when an observation has an attained age of 32, its predicted value will be 0.0023 less than the average prediction.

Figure 11

DISTRIBUTION AND ALE PLOT FOR ATTAINED AGE

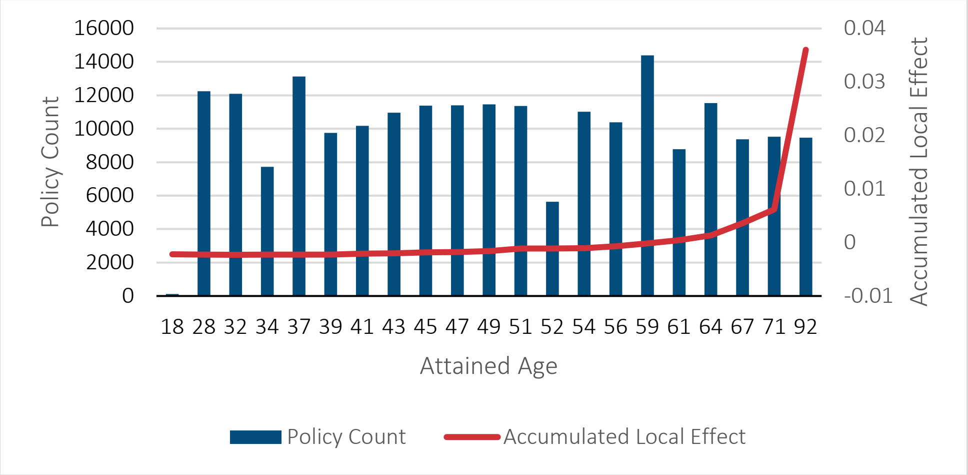
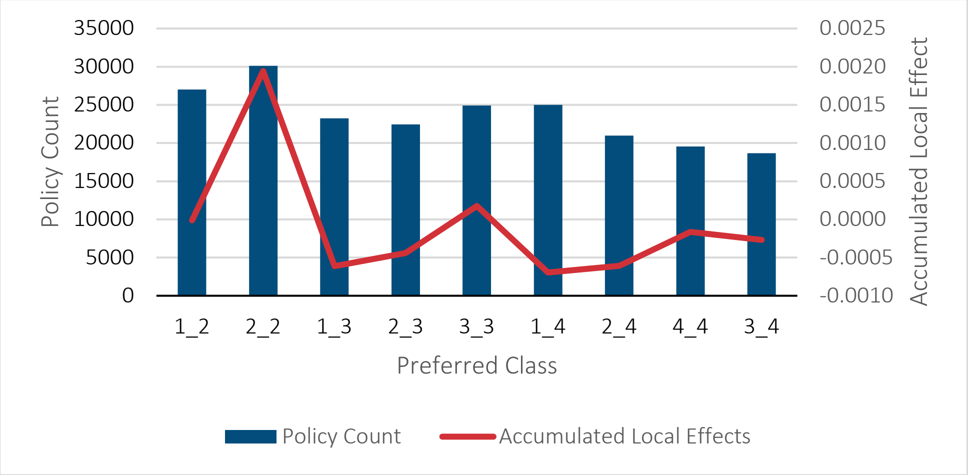


Figure 12 shows the distribution and ALE plot generated for preferred class from our mortality study. The plot is consistent with the PDP results where increased ranking number is related to higher mortality rates. However, in the ordering process, 4_4 was found to be closer to 2_4 than 3_4. The other classes were ordered in a way similar to a human ordering. Despite the out-of-order levels, the ALE algorithm still finds that class 4_4 has higher mortality rates than 3_4, which is what is expected due to the structure of preferred class.

Figure 12

DISTRIBUTION AND ALE PLOT FOR PREFERRED CLASS



(*The ordering for preferred class 4_4 is out of order due to the distance calculation from the ALE algorithm. The algorithm calculates distances between categorical features and found that 4_4 was mathematically* *closer to 2_4 than 3_4. We left the ordering from the algorithm to show that the ALE algorithm may reorder features differently from the order that a human would intuitively order them.*)

4.4 SHAPLEY ADDITIVE EXPLANATIONS

4.4.1 DESCRIPTION

PDP, ICE, and ALE each provide a way to measure a feature's main effect in a model. As these methods come with their own respective caveats, researchers continue to strive for newer and better ways to interpret machine learning models. SHapley Additive exPlanations (SHAP) are a newer and promising development in interpretable machine learning (Lundberg & Lee, 2017).

SHAP values are based on the game theory concept of Shapley values. Shapley values investigate the effect of including a feature in the model by the order in which it is added. The contribution of the feature might change depending on when the information from that feature is added to the model. The following description of the Shapley value calculation process provides an intuition for SHAP values:

1. Fit a null model with no features.
2. For the possible number of features, $j = 1, \dots, p$:
   a. Fit models for all the permutations of j features.
   b. Calculate the expected predicted value for a model with each permutation.
3. Calculate the marginal contribution between models with the feature of interest and those preceding without the feature of interest.
4. Each marginal contribution is weighted equally, but some marginal contributions will be the same and combined in the final calculation.
5. Shapley value is the sum of the weights multiplied by the marginal contributions for all the permutations of the feature of interest.

Table 2 illustrates the process for calculating a Shapley value for a model with three features. Each row shows one of the possible orders in which the feature can be added. The marginal contribution is calculated by subtracting the prediction from the first model in which the feature of interest appears from the last model in which it does not appear. This value represents the amount the feature of interest contributes to the prediction.

Table 2
SHAPLEY CALCULATION EXAMPLE FOR AGE WITH THREE MODEL FEATURES (AGE, GENDER, DURATION).

One Feature Model	Two Feature Model	Three Feature Model	Weight	Marginal Contribution: Age
{**A**}	{**A**, G}	{**A**, G, D}	1/6	{**A**} – {}
{**A**}	{**A**, D}	{**A**, G, D}	1/6	{**A**} – {}
{G}	{**A**, G}	{**A**, G, D}	1/6	{**A**, G} – {G}
{G}	{G, D}	{**A**, G, D}	1/6	{**A**, G, D} – {G, D}
{D}	{**A**, D}	{**A**, G, D}	1/6	{**A**, D} – {D}
{D}	{G, D}	{**A**, G, D}	1/6	{**A**, G, D} – {G, D}

The final Shapley calculation for age is calculated as:

$$SHAP_{Age} = \frac{1}{3} * \{A\} + \frac{1}{6} * (\{A,G\} - \{G\}) + \frac{1}{6} * (\{A,D\} - \{D\}) + \frac{1}{3} * (\{A,G,D\} - \{G,D\})$$

In general, the formula is:

$$SHAP_i = \sum w * [f(S \cup \{i\}) - f(S)]$$

where $f(S \cup \{i\})$ is the model where feature *I* is first introduced, and $f(S)$ is the model prior to the additional of feature *i*.

This process is computationally expensive, as it must be repeated for each feature in the model and for every observation in the dataset. Calculating Shapley values would also require refitting all possible models with every possible combination of features. SHAP algorithms such as TreeSHAP (Lundberg, Erion, & Lee, 2018) and KernelSHAP find approximations for these models using the existing model that is being interpreted.

The SHAP algorithm is run for each observation and feature, producing a table with one row per observation and one column per feature; Table 3 below shows SHAP values for a sample observation. Each value states how that feature contributed to that observation's deviation from the average prediction. For instance, the value of -0.024 for smoking status contributed to moving its prediction slightly lower than the average prediction. This effect size can be compared to the values for gender, duration, or preferred class, which also contributed to a lower prediction but with larger magnitudes. Like GLM parameters, the SHAP values can be added together, with the inverse of the link function applied, to calculate a predicted value.

Table 3
INPUT AND SHAP VALUES FOR ONE OBSERVATION IN MORTALITY STUDY

	Smoker	Gender	Duration	Attained Age	Guaranteed Period	Preferred Class
Input values	Nonsmoker	Female	5	73	10 year	1_4
SHAP values	-0.024	-0.229	-0.246	1.735	0.021	-0.376

Advantages:

- Strong theoretical basis from Shapley values in game theory.
- The TreeSHAP algorithm reduces the complexity of the computation, resulting in significantly faster algorithm speeds compared to the true Shapley value calculation.
- Allows for a unified interpretation of a model, including feature importance, feature main effects, and feature interactions.

Disadvantages:

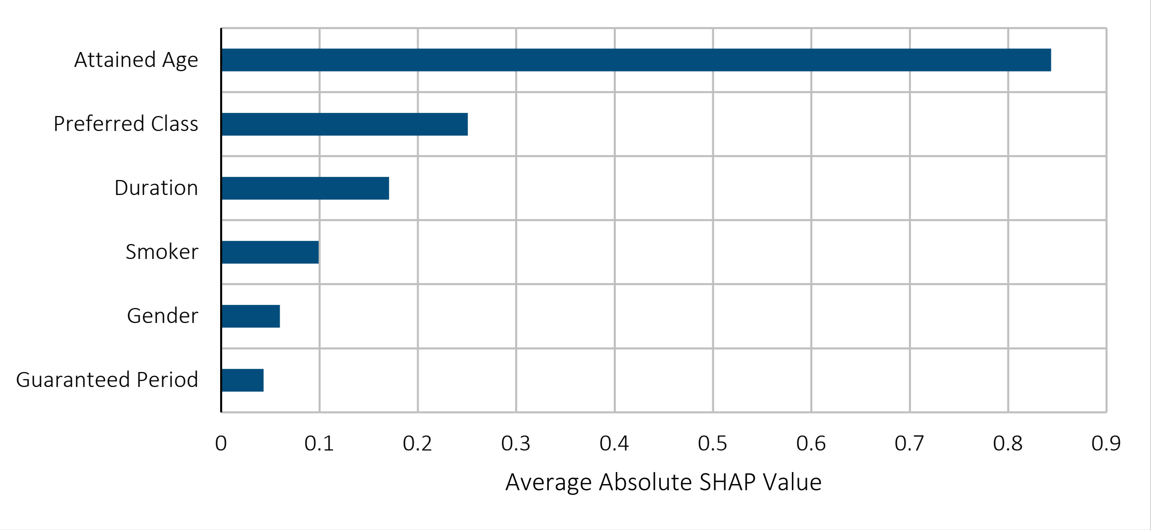
- Approximation methods introduce disadvantages not present in theoretical Shapley values:
    - KernelSHAP is slow and ignores feature dependence.
    - TreeSHAP can attribute a contribution to features that truly have no influence on the prediction.

4.4.2 CASE STUDY EXAMPLES

The nature of SHAP values enables it to be used as a feature importance metric, as well as understanding main effects and interactions. Features with the largest average absolute SHAP values are the most important since they have the largest contribution to the prediction. This value is calculated and averaged across all observations and the features are ranked and plotted from highest to lowest. Figure 13 shows such a SHAP-based feature importance ranking for the mortality study, which is similar to the results of the other feature importance methods presented in Section 3.

Figure 13

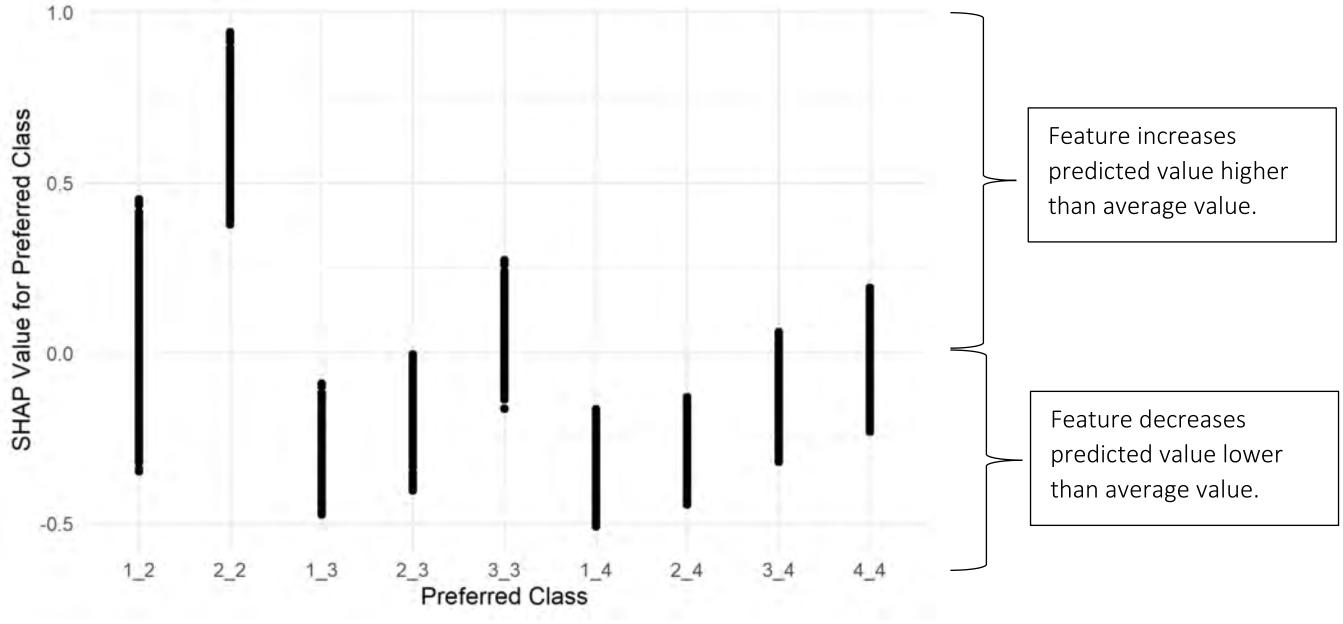
SHAP FEATURE IMPORTANCE FOR MORTALITY STUDY



SHAP values are typically plotted as scatterplots with the feature of interest on the x-axis and the SHAP value on the y-axis, such as in Figure 14. The scatterplot shows the shape and magnitude of the feature's effect on the prediction and can be interpreted similarly to a PDP or ALE plot. The SHAP values for preferred class show a similar relationship to the PDP in Figure 7.

When interpreting SHAP plots, the key area of focus is the vertical dispersion at any given value of the feature. Without interaction effects present, contribution to the prediction would be consistent across all observations, meaning that all the points would fall in a narrow range. Vertical dispersion, as demonstrated by preferred class 1_2 in Figure 14, indicates that, at a particular value of the feature, there is a large amount of variance in the feature's contribution to the prediction, which suggests the presence of interaction effects.

Figure 14
SHAP SCATTERPLOT FOR PREFERRED CLASS



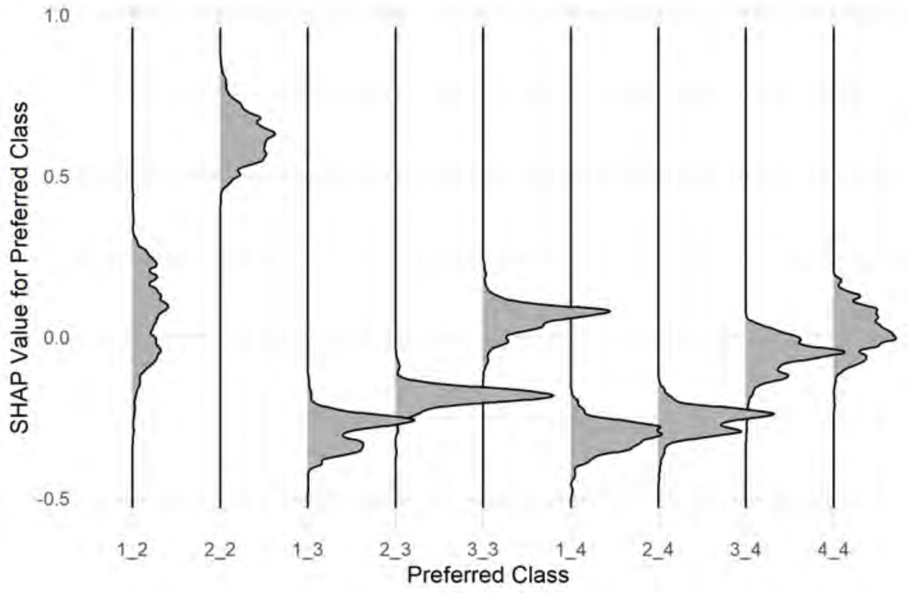
Feature increases predicted value higher than average value.

Feature decreases predicted value lower than average value.

As can be seen in Figure 14, large datasets will cause scatterplots to have clusters of points that are likely to overlap, making the distribution of values unclear. Boxplots, violin plots, and ridge plots are alternative options that display the distribution of values clearer than scatterplots do as outliers in scatterplots may misrepresent the amount of vertical dispersion. In Figure 15, we display a ridge plot version of SHAP to show a fuller picture of the distribution of SHAP values (and degree of interaction effects) at each level of preferred class.

Figure 15

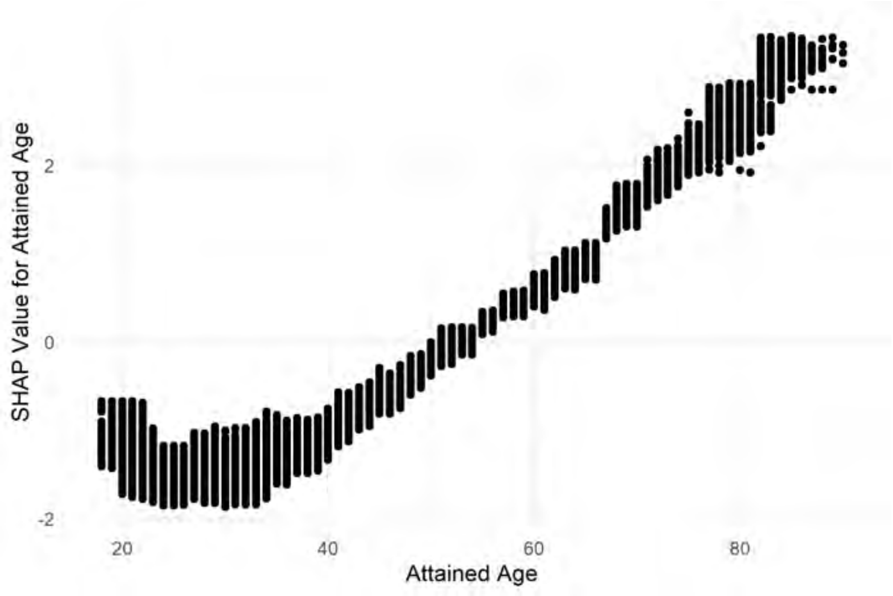
SHAP RIDGE PLOT FOR PREFERRED CLASS



For continuous features, however, the ridge plot is usually not a viable option due to the high number of feature levels which makes interpretation more difficult. A scatterplot provides a better visualization starting point for continuous features.

Interpreting the continuous SHAP plot is no different from interpreting a categorical plot; we also focus on the overall shape and the amount of vertical dispersion of the SHAP values. In Figure 16, we see that the expected mortality increases with attained age. We also see a downward slope at younger ages, which is not apparent in the PDP, ICE, and ALE plots.

Figure 16
SHAP SCATTERPLOT FOR ATTAINED AGE



4.5 COMPARISON OF PARTIAL DEPENDENCE AND SHAPLEY ADDITIVE EXPLANATIONS

An initial comparison between the PDP and SHAP for attained age shows similar directions, but different shapes, to their plots. Since both methods claim to show the effect attained age has on the prediction, the critical question is: which relationship is correct? To answer this question, it is important to note that the plots are showing different values. PDP displays the average expected prediction for each value of attained age, while SHAP displays the attained age's contribution to the deviation from the average prediction.

To make an accurate comparison, we must connect the average prediction with the SHAP values. Since the SHAP value for each feature is its contribution to that observation's deviance from the overall average prediction, we can reproduce the actual predicted value from the SHAP values. The calculation can either be related to the log scale of the SHAP values (1) or the response scale of the prediction (2):

$$\log(f(x)) = \log(\bar{y}) + SHAP_{AA} + SHAP_D + SHAP_{PC} + SHAP_{GP} + SHAP_S + SHAP_G + \beta_0 \qquad (1)$$

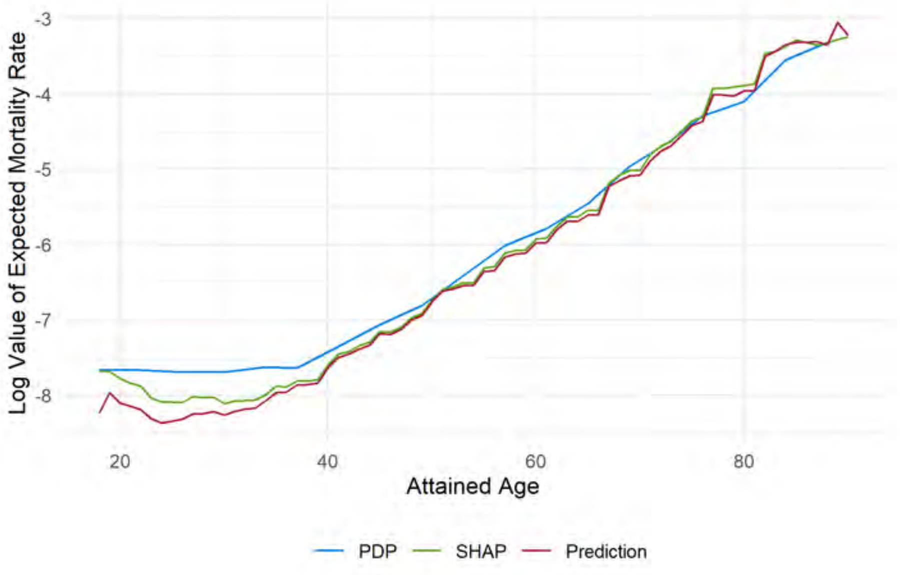
$$f(x) = \bar{y} * e^{SHAP_{AA}} * e^{SHAP_D} * e^{SHAP_{PC}} * e^{SHAP_{GP}} * e^{SHAP_S} * e^{SHAP_G} * e^{\beta_0} \qquad (2)$$

- $\bar{y}$ is the average predicted value for the whole dataset.
- β_0 is an intercept that represents the SHAP value for the average prediction.
- $SHAP_{AA}$, $SHAP_D$, $SHAP_{PC}$, etc. represent the SHAP values for attained age, duration, preferred class, etc.

The values calculated in the PDP algorithm are on the predicted scale by default and are log-transformed to make our comparison in Figure 17.

Figure 17

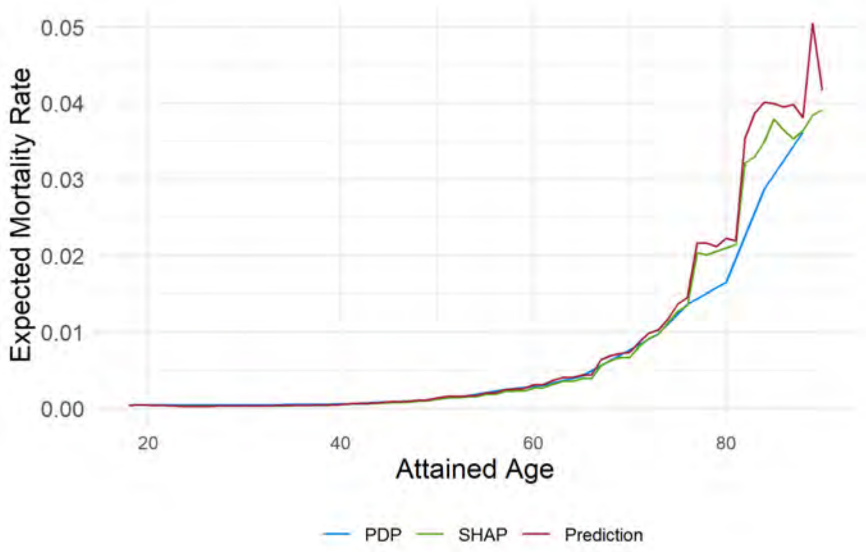
COMPARISON OF PDP (BLUE), SHAP (GREEN), AND PREDICTION (RED) VALUES ON THE LOG SCALE



The log scale in Figure 17 may not be ideal in some interpretation settings. Viewing this analysis on the response scale may be more intuitive, as shown in Figure 18, but because of the change in scale, the separation in the lower attained ages is hard to see on the response scale as opposed to the log scale, at least for the mortality case study in this paper. Experimentation is encouraged to determine what works best for each individual case.

Figure 18

COMPARISON OF PDP (BLUE), SHAP (GREEN), AND PREDICTION (RED) VALUES ON THE RESPONSE SCALE



The transformations described allow us to make a true comparison of the PDP and SHAP plots for attained age. The most disagreement between the different values is at lower and higher values of attained age. These are the portions of the distribution with the most correlation to duration and lower policy counts, so it is not surprising that the different values disagree somewhat in those areas. Otherwise, the relationships have the same shape for both PDP and SHAP. This consistency of results, after accounting for transformation, ultimately lends confidence that the model has a reasonable mortality relationship across attained age values.

Our guidance for the use of PDP, ICE, ALE, and SHAP:

- Examine the correlation matrix of features used in a model. When reviewing a model, if no correlation matrix is provided, ask for one to be provided.
- For uncorrelated variables, PDP and ALE can be used to examine model effects, especially if results are similar for both methods.
- For variables with correlations, considering removing one of the two correlated features and re-fitting and/or using SHAP to examine the marginal effects.
- ICE and SHAP can help identify variables involved in interaction effects.

SHAP values can also be used to explain the contribution of each feature for individual observations, providing a framework for “reason codes” that explains what is driving the model output.

Finally, we note that if the relationships uncovered by these methods are unacceptable for business or regulatory reasons, it is possible in the XGBoost package to place monotonic constraints that force selected features to only increase or decrease over the range of values.

Section 5: Interaction Effects

One of the key elements in understanding a predictive model is examining its interaction effects. Interaction effects occur when the impact of a change in a variable depends on the values of other features. In this section, we examine methods to identify and examine interactions in complex models.

5.1 FRIEDMAN'S H-STATISTIC

5.1.1 DESCRIPTION

Interaction effects in complex models are difficult to identify. The H-statistic quantifies the magnitude of interaction effects for each feature and can also be used to quantify the magnitude of interaction effects between one feature and each of the other features. The H-statistic calculation can be interpreted as the amount of variance explained by the interaction effect.

While the H-statistic has a clever algorithm, it is relatively computationally expensive and may not be practical in a lot of situations. Building predictive models is an iterative process, so having a multi-day process for an iterative process could be prohibitive to its usefulness.

Friedman's H-statistic is rooted in the idea that the effect of an interaction between x_j and x_k can be approximated using partial dependence as follows (Friedman & Popescu, 2008):

1. Calculate $PD_j(x_j)$, the average predicted value achieved by setting x_j equal to a given level, while holding all other features (x_{-j}) unchanged.
2. Calculate $PD_k(x_k)$, the average predicted value achieved by setting x_k equal to a given level, while holding all other features (x_{-k}) unchanged
3. Calculate $PD_{jk}(x_j^{(i)}, x_k^{(i)})$, defined as a two-feature partial dependence function in which all combinations of features j and k are considered while holding all other features constant.
4. The two-way H-statistic for variables j and k is:

$$H_{jk}^2 = \sum_{i=1}^{n} [PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j{}^{(i)}) - PD_k(x_k{}^{(i)})]^2 \; / \; \sum_{i=1}^{n} PD^2{}_{jk}(x_j^{(i)}, x_k^{(i)})$$

If x_j and x_k have no interaction, then it follows that:

$$PD_{jk}(x_j^{(i)}, x_k^{(i)}) = PD_j(x_j) + PD_k(x_k)$$

and the H-statistic would be zero. With an interaction present, the two-way H-statistic for variables j and k can be interpreted as the proportion of variance in prediction that is explained by the interaction.

Due to the volume of partial dependence computations necessary to calculate H-statistics, this procedure is highly computationally intense. We also note that the exact details of implementing the H statistic in practice are out of scope for this paper and direct readers to (Friedman & Popescu, "Predictive learning via rule ensembles", 2008).

Also, the H-statistic relies heavily on the partial dependence function, which makes it susceptible to all the issues associated with partial dependence plots. In particular, the creation of unrealistic observations in the partial dependence function can inflate the H-statistic to indicate a large magnitude of interaction that is not actually present.

5.1.2 CASE STUDY EXAMPLES

Table 4 shows the H-statistic for the mortality case study. The feature with the highest interaction value is attained age, but there is concern that this value might be inflated due to the correlation between attained age and duration.

Table 4

OVERALL H-STATISTIC FOR MORTALITY STUDY

	H-Statistic
Attained Age	0.505
Duration	0.499
Preferred Class	0.262
Gender	0.134
Guaranteed Period	0.078
Smoker	0.061

Based on the ranking from Table 4, we calculated the H-statistic for every other variable interacting with attained age, shown in Table 5. Attained age had the highest interaction with preferred class.

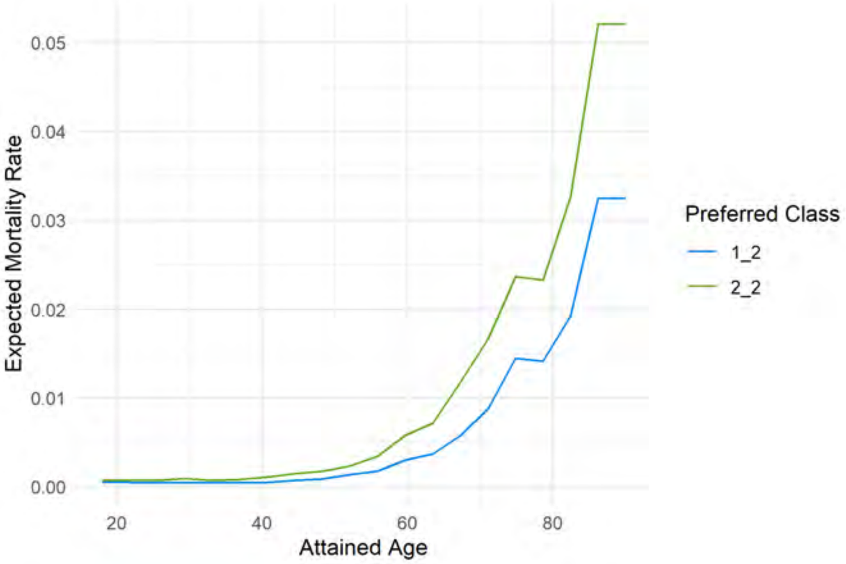
Table 5

H-STATISTIC FOR ATTAINED AGE WITH OTHER FEATURES

	H-Statistic
Preferred Class:Attained Age	0.248
Duration:Attained Age	0.187
Gender:Attained Age	0.168
Smoker:Attained Age	0.112
Guaranteed Period:Attained Age	0.032

To examine the nature of the interaction effects, we used PDPs. Figure 19 shows the two-feature PDP of attained age by preferred class 1_2 and 2_2, which were identified in Table 5 to have the highest degree of interaction effects. This plot shows the curve for attained age is different at the different levels of preferred class.

Figure 19
TWO-FEATURE PDP FOR ATTAINED AGE AND PREFERRED CLASS, FILTERED TO PREFERRED CLASSES 1_2 AND 2_2



5.2 SHAPLEY ADDITIVE EXPLANATIONS

5.2.1 DESCRIPTION

Research in Shapley values identified methods to deconstruct the SHAP values into separate interaction values (Fujimoto, Kojadinovic, & Marichal, 2006). The calculation is similar to the normal SHAP calculation but finds the marginal contribution of adding two features instead of just one. Using the notation from section 4.4.1, the SHAP interaction calculation for two features is:

$$SHAP_{i,j} = \sum w * [f(S \cup \{i,j\}) - f(S \cup \{j\}) - (f(S \cup \{i\}) - f(S))]$$

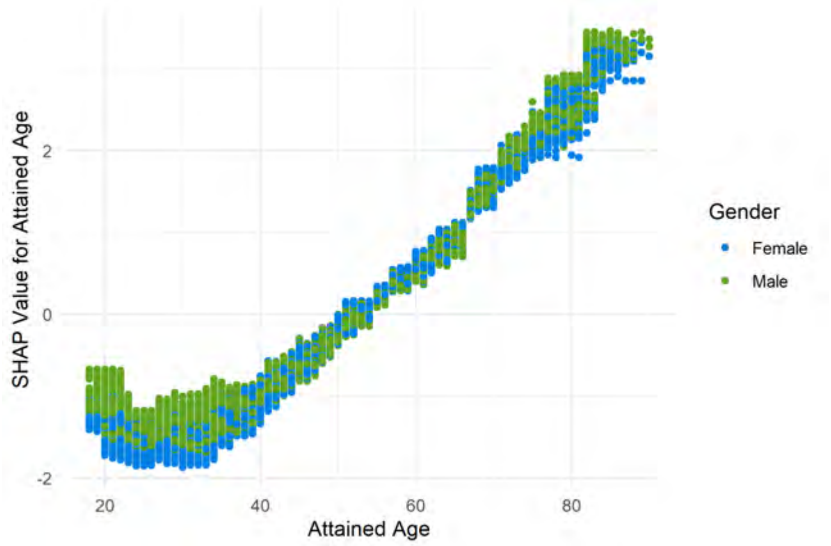
The result of this calculation is a matrix of size $p\ x\ p\ x\ n$ where p is the number of features in the model and n is the number of observations. The interaction value between the two features is split evenly, giving each feature equal weight of the interaction, $SHAP_{i,j} = SHAP_{j,i}$.

5.2.2 CASE STUDY EXAMPLES

The common method for finding interactions is to plot the SHAP values for one feature and color the plot by another. This plot would show how the contribution for a feature may change depending on the value of another feature. However, these plots often do not show a clear pattern to the interactions, either because the features have too many levels to distinguish between them, or the colors overlap too much to display clear views of any interaction effects. Figure 20 shows the SHAP values for attained age colored by gender. From the plot, it appears that the difference in mortality between males and females is greater at younger and older ages.

Figure 20

SHAP PLOT FOR ATTAINED AGE BY GENDER

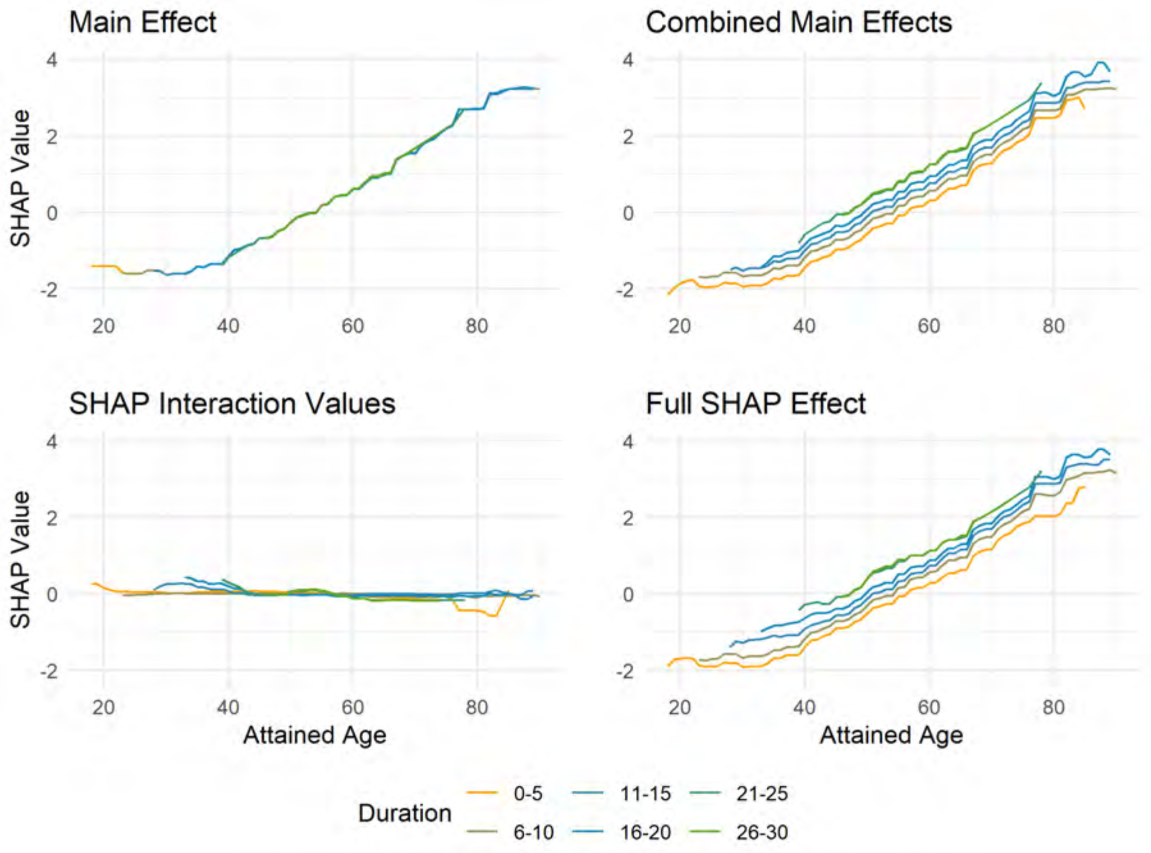


The H-statistic signaled a relatively strong interaction between attained age and duration, but this is potentially biased due to the correlation between these variables. To investigate this interaction further, we looked at the SHAP values.

Figure 21 shows a decomposition of the SHAP values of attained age and duration. The top left plot is the main effect ($\beta_{AA}\ x\ AttainedAge$) for attained age colored by duration. The top right plot combines the main effects for the duration and attained age ($\beta_{AA}\ x\ AttainedAge +\ \beta_D\ x\ Duration$). Duration separates the attained ages into parallel lines. The bottom left plot shows the interaction effect ($\beta_{AAxD}\ x\ AttainedAge\ x\ Duration$). There appears to be some effect at lower and higher attained ages, but the overall size of the effect is small compared to the magnitude of the main effects. The bottom right plot combines the interaction with the main effects ($\beta_{AA}\ x\ AttainedAge +\ \beta_D\ x\ Duration +$ $\beta_{AAxD}\ x\ AttainedAge\ x\ Duration$). The telltale sign of an interaction would be lines that are no longer parallel, which would indicate that the effect of attained age changes at different levels of duration. This analysis indicates some interaction at the extremes of the two features but, overall, there appears to be no strong interaction effect, contrary to the H-statistic.

Figure 21

SHAP INTERACTION ANALYSIS FOR ATTAINED AGE AND DURATION



5.3 PREDICTION FUNCTION

5.3.1 DESCRIPTION

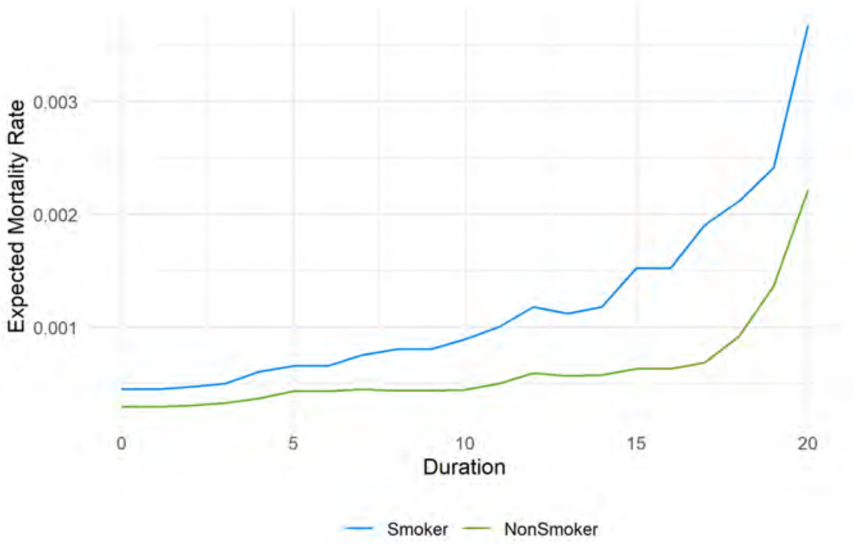
Another method of visualizing interactions is a function that creates hypothetical observations and finds their predicted values. The function allows us to isolate features to investigate how the model uses those features. We can also investigate different combinations of characteristics to see how the model would treat those combinations.

The prediction function takes hypothetical observations as inputs, performs the necessary data prep to put those observations into the model, and makes a prediction. That prediction is then plotted across a relevant feature to see the change in prediction across that feature.

5.3.2 CASE STUDY EXAMPLES

The prediction function relies heavily on the assumptions of the hypothetical observation. In Figure 22, we assumed a 20-year guaranteed period, preferred class 1_2, and male. Attained age begins at 40 years old but increases as duration increases due to their dependent relationship. Given these characteristics, we can see that a smoker’s mortality rises at a different rate than nonsmokers as duration increases.

Figure 22
PREDICTION FUNCTION COMPARING SMOKERS AND NONSMOKERS



5.4 COMPARISON OF INTERACTION METHODS

On the use of H-statistics, SHAP, and prediction functions:

- Use H-statistics and PDPs with caution in situations with a large number of observations, variables, and/or correlated variables.
- SHAP values can be used to identify and visualize the most influential interaction effects.
- Use caution with the prediction function as the results may be highly influenced by the choice of assumptions for the hypothetical observation.

As with main effects, there is also an interaction constraint option in XGBoost where the modeler can define the features that are allowed to interact with each other.

Section 6: Correlated Features

For GLMs and other variations of linear models, correlation, multicollinearity, and aliasing (perfect correlation) among predictor variables can cause standard deviations of coefficients to be large and coefficients to behave erratically, causing issues with interpretability. With machine learning models, there has historically been less focus on correlations or dependent relationships among predictor variables, with the exception of high-dimensional problems where the number of potential predictor variables is much bigger than the number of observations. However, as we have outlined in several sections, correlation can cause the interpretation methods to provide faulty results for machine learning methods.

Fortunately, we can use many of the same best practices from linear models to identify and address correlations impacting interpretability for machine learning models. The first step is to understand which features are highly correlated; the next step is to determine if the correlations are causing instability in the model; and the last step is to determine how to modify the model to address the instability.

With linear models, an effective practice to determine if correlation is problematic is to build multiple models, one with all features and others with one feature removed and examining the effects on the model coefficients and standard errors. Alternatively, regularization (such as Lasso, Ridge, and Elastic Net) can be applied to shrink or eliminate the coefficients of correlated features when training a linear model. For machine learning models, we can use a similar approach, fitting multiple models and examining the feature importance, main effects, and interaction effects to see if the results are stable when correlated features are in the model.

Correlated features are a common occurrence in modeling projects and can cause major issues. The simplest method for handling correlated features is to pick one to stay in the model and drop the other. Sometimes the two features contribute different information to the model, and it would be preferable to keep them. Using the principle components as the model features is one way to solve this problem. Another is to create a two-stage model. In a two-stage model, a first model is fit with one of the correlated features. The prediction from that model is used as an offset in a second model that contains the other correlated feature. In this way, the effects of the two features are isolated from each other. None of these solutions is perfect, but they do provide options for dealing with correlated features.

6.1 CORRELATION METRICS

6.1.1 DESCRIPTION

The first step is to identify which features have high correlations. The simplest way to do this is with a correlation matrix, chi-square tests, or Cramer’s V statistics. These statistics put a numeric value on the strength of the correlation between pairs of features.

While variance inflation factors are a popular method for identifying multicollinearity in linear models, they are calculated using ratios of variances in model coefficients, and therefore do not transfer over to tree-based methods where such statistics are not available.

6.1.2 CASE STUDY EXAMPLES

For continuous features, the Pearson Correlation is the simplest method to determine the size of the correlation. From Table 6, we see that issue age and attained age are highly correlated.

Table 6
CORRELATION MATRIX FOR CONTINUOUS FEATURES

	Attained Age	Duration	Issue Age
Attained Age	1.00	0.29	0.94
Duration	0.29	1.00	-0.06
Issue Age	0.94	-0.06	1.00

For categorical features, Cramer's V statistic provides a method for correlated quantification. Preferred class and smoking status have a dependent relationship, so it is logical that they have a high correlation value.

Table 7
CRAMER'S V MATRIX FOR CATEGORICAL FEATURES

	Preferred Class	Smoker	Gender	Guaranteed Period
Preferred Class	1.00	0.63	0.03	0.09
Smoker	0.63	1.00	0.02	0.04
Gender	0.03	0.02	1.00	0.03
Guaranteed Period	0.09	0.04	0.03	1.00

When comparing continuous and categorical features and the categorical feature has only two levels, then Point biserial correlation is a good option, as shown in Table 8. If the categorical features have more than two levels, then logistic regression, Kruskal-Wallis H Test, or ANOVA would be appropriate. These methods have more assumptions and require more in-depth analysis to verify the results[6].

Table 8
CORRELATION MATRIX BETWEEN CONTINUOUS FEATURES AND CATEGORICAL FEATURES WITH TWO LEVELS

	Smoker	Gender
Attained Age	-0.06	-0.03
Duration	-0.02	0.00
Issue Age	-0.05	-0.03

[6] For more information: https://rcompanion.org/rcompanion/d_06.html

6.2 IMPACT OF CORRELATION ON FEATURE IMPORTANCE

6.2.1 DESCRIPTION

Correlated features may cause the model to split the importance among multiple features. By removing features one at a time, the feature importance of the other variables may change, providing insights into the impact of the correlation on the model.

6.2.2 CASE STUDY EXAMPLES

In the raw mortality study data, issue age is highly correlated with attained age with a Pearson correlation coefficient of 0.94. A secondary model was built, which included both attained age and issue age. In Figure 23, it is clear that the model picked attained age to be the more informative of the two features and rarely used issue age. This is an example of the model handling the correlated features fairly well in terms of the feature importance. However, the permutation importance for attained age increased from train to test because the correlations were broken through the permuting of values, illustrating the sensitivity of the permutation feature importance to correlated features in the model.

Figure 23

PERMUTATION FEATURE IMPORTANCE FOR MORTALITY MODEL INCLUDING ISSUE AGE

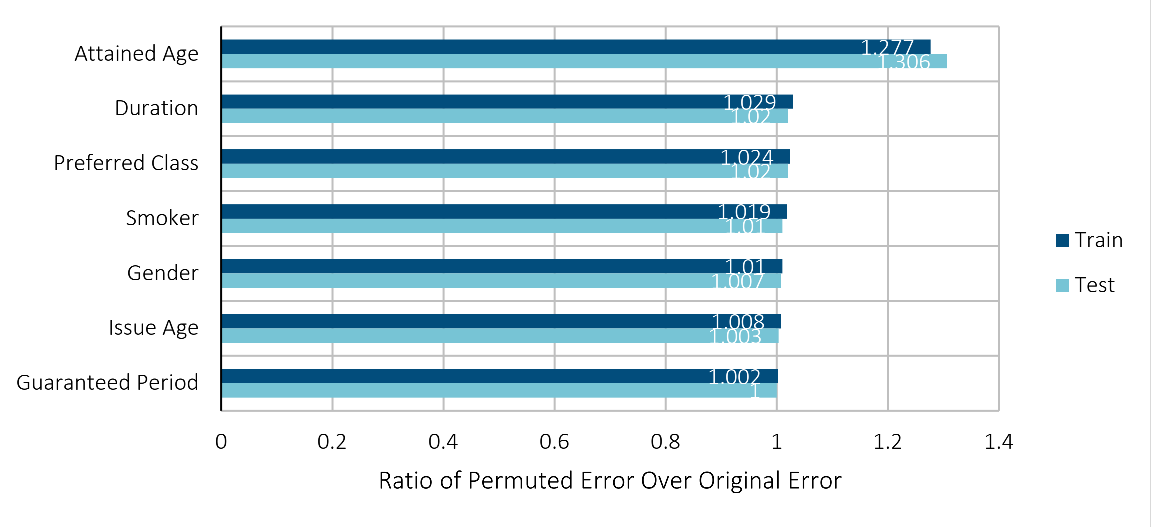


Table 9 compares the feature importance for the models with and without issue age. In this example, the variables most impacted by adding issue age to the model were attained age and smoker.

Table 9

PERMUTATION FEATURE IMPORTANCE – TRAINING DATA

	Primary Model Excluding Issue Age	Secondary Model Including Issue Age	Absolute Change	Relative Change
Attained Age	1.286	1.277	-0.009	-0.700%
Duration	1.032	1.029	-0.003	-0.291%
Preferred Class	1.028	1.024	-0.004	-0.389%
Smoker	1.008	1.019	+0.011	1.091%
Gender	1.008	1.010	+0.002	0.198%
Issue Age	NA	1.008	NA	NA
Guaranteed Period	1.002	1.002	+0.000	0.000%

6.3 IMPACT OF CORRELATION ON MAIN EFFECTS

6.3.1 DESCRIPTION

PDP and ICE plots make an assumption that features are independent; simulations have shown them to produce inaccurate results when this assumption is violated[7]. The concern with correlated features, or a dependent relationship, is that it will cause the algorithm to create unrealistic observations and distort the estimated impacts of the predictor variables on the model output.

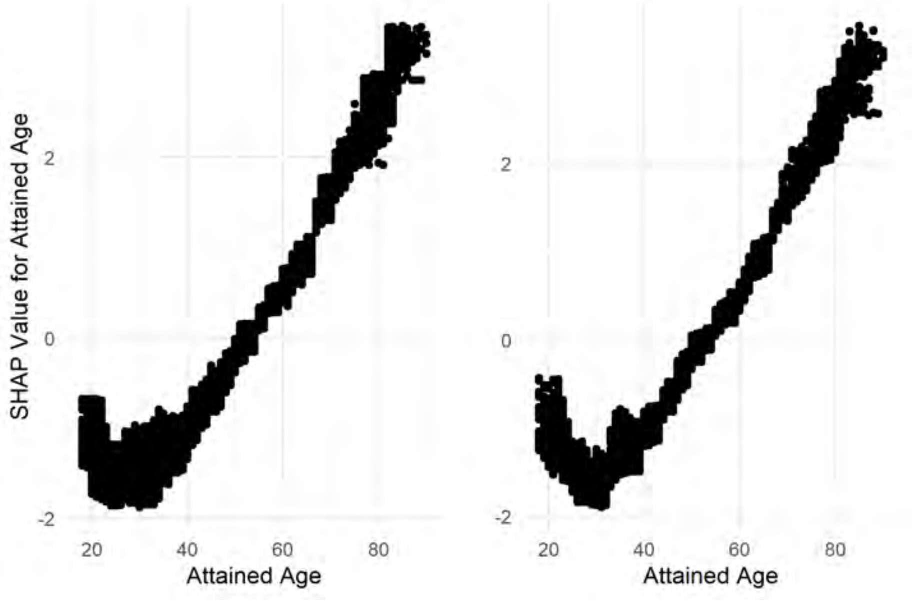
Unlike PDP and ICE, SHAP values are a representation of the model and estimate how much a feature contributes to an observation's deviation from the average prediction. SHAP values may be influenced by the model allocating effects among correlated features. We can compare the SHAP plots for various models to understand the degree to which the presence of correlated variables in the model are impacting the relationship between the model inputs and outputs.

6.3.2 CASE STUDY EXAMPLES

In Section 6.1.2, we noted the high degree of correlation between attained age and issue age; in Section 6.2.2, we examined the permutation feature importance for a secondary model including issue age. Figure 24 compares the SHAP plots for attained age with and without issue age in the model.

Figure 24

SHAP PLOT FOR ATTAINED AGE (PRIMARY MODEL LEFT, SECONDARY MODEL RIGHT)



[7] https://compstat-lmu.github.io/iml_methods_limitations/ale-pdp.html

Duration is the difference between attained age and issue age. Figure 25 shows a similar comparison of SHAP effects for duration between the two models.

Figure 25

SHAP PLOT FOR DURATION (PRIMARY MODEL LEFT, SECONDARY MODEL RIGHT)

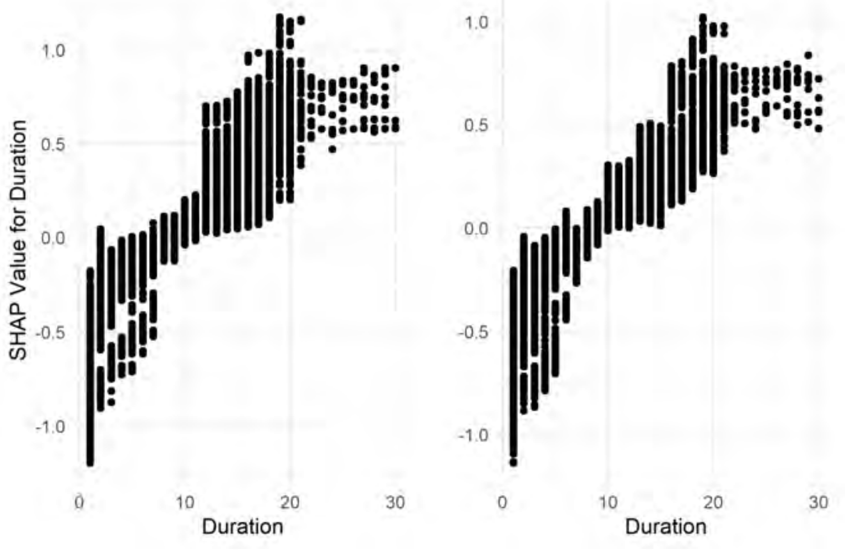


Figure 26 shows the SHAP plot for issue age from our secondary model. Comparing the y-axis of Figure 26 to those of Figures 24 and 25, we see that the impact of issue age on model predictions is smaller in magnitude than the impact of attained age or duration. We also can see that the model is indicating mortality rates are higher at younger issue ages and mortality rates are lower for issue ages over 75. To the extent that the younger issue ages have a higher average duration than the older issue ages, this could be reflecting the "wearing off" of underwriting effects. However, this effect is more directly measured by duration, which is also in the model, so we do not have a clear explanation for this relationship between issue age and mortality after accounting for duration and attained age.

Figure 26
SHAP PLOT FOR ISSUE AGE

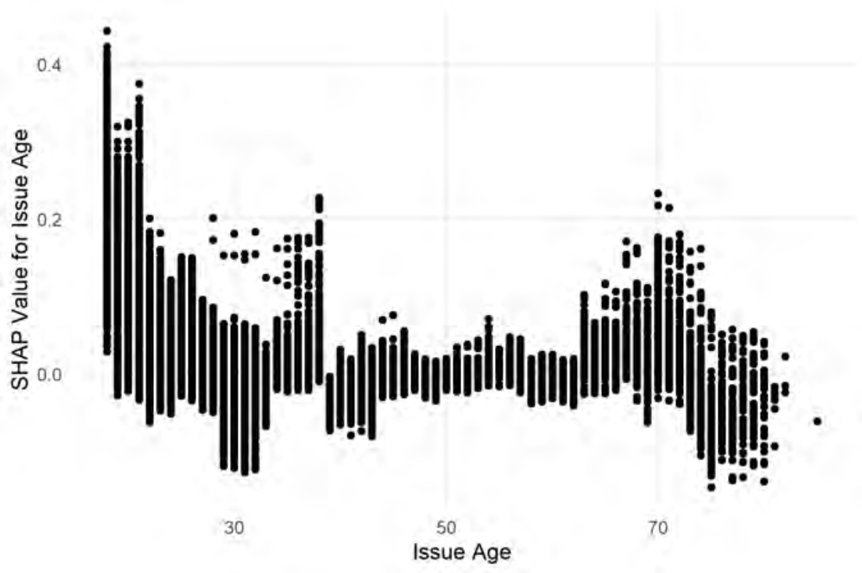


Figure 26 raises some questions about the inclusion of issue age in the model due to the lack of interpretability of the effect, as evidenced by the lack of slope going across the x-axis. Another way to evaluate the impact of including issue age is to look at how different the predictions are between these models and how accurate the predictions are. Figure 27 displays a histogram of the difference between the two models. Given that the model effects for attained age and duration are fairly similar, and the average effect of issue age is near zero for a large number of records (between ages 30 and 70, as seen in Figure 26), the differences in the model estimates are relatively small for a large proportion of the observations.

Figure 27

HISTOGRAM OF DIFFERENCE IN PREDICTION BETWEEN PRIMARY AND SECONDARY MODEL (FILTERED TO DIFFERENCES BETWEEN -0.0025 TO 0.0025)

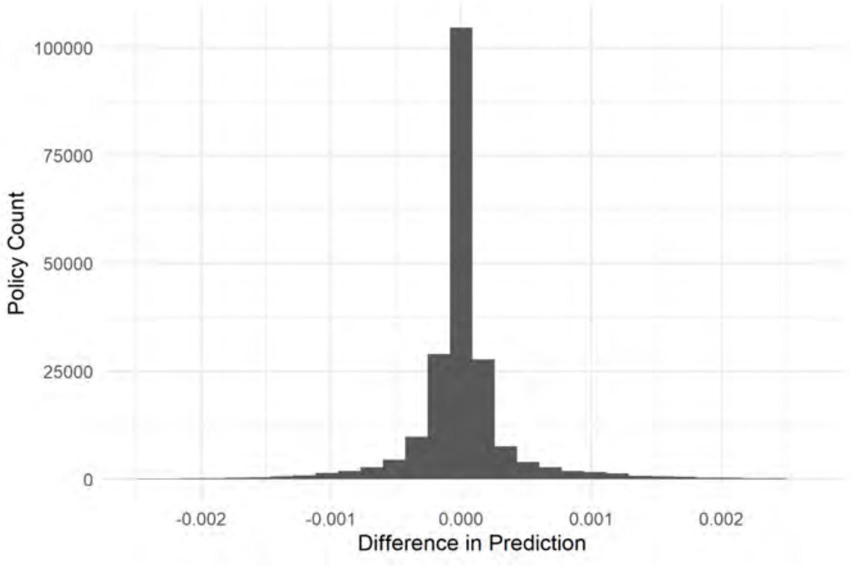
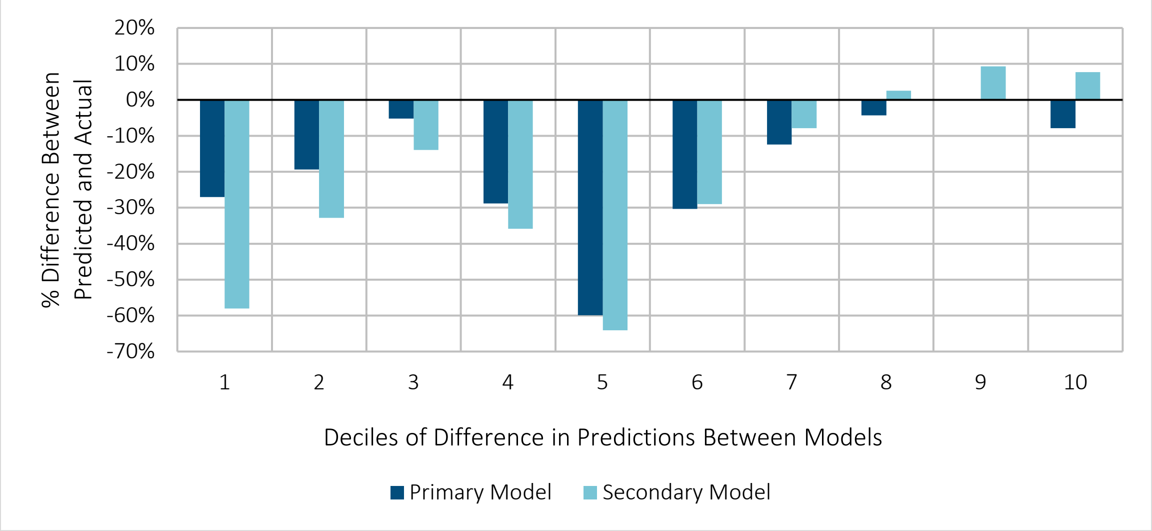


Figure 28 displays a two-way lift chart comparing the models. This chart creates ten deciles based on the difference in prediction between the two models. Decile 1 contains observations where the primary model prediction was lower than the secondary model; in Decile 10, the primary model predictions are higher than the secondary model predictions. In each quantile, we calculate the difference between the actual mortality rate and the predicted mortality rate (e.g. actual divided by predicted mortality minus one) for each model. For most of the deciles where the differences in model predictions are the greatest, the differences between actual and predicted were smaller for the primary model, which suggests that the model with issue age does not predict as well as the model without it.

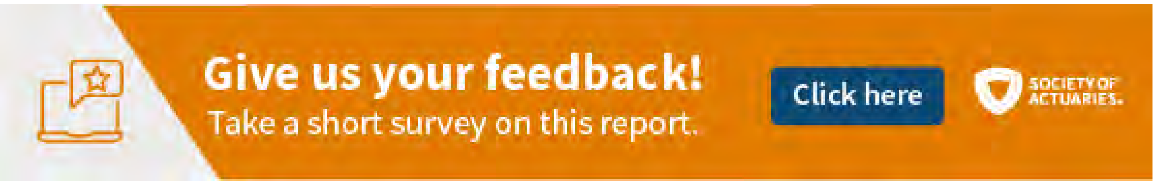
Figure 28
TWO-WAY LIFT CHART BETWEEN PRIMARY MODEL AND SECONDARY MODEL



Section 7: Conclusion

Machine learning models are powerful tools that are growing in use by the insurance industry[8]. The interpretation of data to identify fraudulent claims, assumption setting in life insurance, and the creation of underwriting guidelines are some of the areas in which machine learning can provide predictive ability superior to GLMs. However, in many applications there are other considerations besides raw predictive ability in selecting and using a model, such as identifying potential bias and discrimination, determining compliance with applicable laws and regulations, and explaining the model to individuals impacted by its results.

We have reviewed several methods for interpreting machine learning models. Our investigation into these methods produced promising results in our life insurance case study, as well as insights into the strengths and weaknesses of different approaches. The field of interpretable machine learning is developing rapidly, and it is our hope that studies such as these will advance the development of best practices for the review of machine learning models and increase their acceptability.



[8] https://emerj.com/ai-sector-overviews/machine-learning-at-insurance-companies/

Acknowledgments

The authors’ deepest gratitude goes to those without whose efforts this project could not have come to fruition: the volunteers who generously shared their wisdom, insights, advice, guidance, and arm’s-length review of this study prior to publication. Any opinions expressed may not reflect their opinions nor those of their employers. Any errors belong to the authors alone.

Talex Diede, ASA, MAAA

David Evans, FCAS, MAAA

Matthias Kullowatz, ASA, MAAA

Nicholas Hanoian

Kshitij Srivastava

Gabriele Usan

David Wang, FSA, FIA, MAAA

Cody Webb, FCAS, MAAA, CPCU

The authors also thank the Project Oversight Group for their diligent work overseeing project development and reviewing and editing this report for accuracy and relevance.

Project Oversight Group members:

Tom Callahan, FSA, MAAA

Michael Destito, FSA, CERA

Annie Girard, FSA, FCIA

Yuanjin Liu, ASA

Vikas Sharan, FSA, FIA, MAAA

Yi Yue Zhang, ASA, ACIA

At the Society of Actuaries:

Korrel Crawford, Senior Research Administrator

R. Dale Hall, FSA, MAAA, CERA, Managing Director of Research

Mervyn Kopinsky, FSA, EA, MAAA, Senior Experience Studies Actuary

David Schraub, FSA, MAAA, CERA, Senior Practice Research Actuary

References

Apley, D. W., & Zhu, J. (2016). "Visualizing the effects of predictor variables in black box supervised learning models.". *arXiv preprint arXiv:1612.08468*.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. Retrieved from https://xgboost.readthedocs.io/en/latest/

Friedman, J. H. (2001). "Greedy function approximation: A gradient boosting machine". *Annals of Statistics*, 1189-1232.

Friedman, J. H., & Popescu, B. E. (2008). "Predictive learning via rule ensembles". *The Annals of Applied Statistics*, 916-54.

Fujimoto, K., Kojadinovic, I., & Marichal, J.-L. (2006). Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 72-99.

Hall, P., & Gill, N. (2019). *An Introduction to Machine Learning Interpretability.* O'Reilly.

Lundberg, S. M., & Lee, S.-I. (2017). "A unified approach to interpreting model predictions.". *Advances in Neural Information Processing Systems.*

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). "Consistent individualized feature attribution for tree ensembles.". *arXiv preprint arXiv:1802.03888*.

Molnar, C. (2019). *"Interpretable Machine Learning, A Guide for Making Black Boxes Explainable".* https://christophm.github.io/interpretable-ml-book/.

These third-party links are being provided for informational purposes only. The content in these third-party links do not necessarily reflect the opinions of Society of Actuaries or the Education & Research Section. Neither the Society of Actuaries nor the Education & Research Section are responsible for the reliability, accuracy or content of the third-party site(s). If you have questions about the content on such sites, please contact the site directly.

About the Society of Actuaries

With roots dating back to 1889, the Society of Actuaries (SOA) is the world's largest actuarial professional organization with more than 31,000 members. Through research and education, the SOA's mission is to advance actuarial knowledge and to enhance the ability of actuaries to provide expert advice and relevant solutions for financial, business and societal challenges. The SOA's vision is for actuaries to be the leading professionals in the measurement and management of risk.

The SOA supports actuaries and advances knowledge through research and education. As part of its work, the SOA seeks to inform public policy development and public understanding through research. The SOA aspires to be a trusted source of objective, data-driven research and analysis with an actuarial perspective for its members, industry, policymakers and the public. This distinct perspective comes from the SOA as an association of actuaries, who have a rigorous formal education and direct experience as practitioners as they perform applied research. The SOA also welcomes the opportunity to partner with other organizations in our work where appropriate.

The SOA has a history of working with public policymakers and regulators in developing historical experience studies and projection techniques as well as individual reports on health care, retirement and other topics. The SOA's research is intended to aid the work of policymakers and regulators and follow certain core principles:

Objectivity: The SOA's research informs and provides analysis that can be relied upon by other individuals or organizations involved in public policy discussions. The SOA does not take advocacy positions or lobby specific policy proposals.

Quality: The SOA aspires to the highest ethical and quality standards in all of its research and analysis. Our research process is overseen by experienced actuaries and nonactuaries from a range of industry sectors and organizations. A rigorous peer-review process ensures the quality and integrity of our work.

Relevance: The SOA provides timely research on public policy issues. Our research advances actuarial knowledge while providing critical insights on key policy issues, and thereby provides value to stakeholders and decision makers.

Quantification: The SOA leverages the diverse skill sets of actuaries to provide research and findings that are driven by the best available data and methods. Actuaries use detailed modeling to analyze financial risk and provide distinct insight and quantification. Further, actuarial standards require transparency and the disclosure of the assumptions and analytic approach underlying the work.

Society of Actuaries
475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
www.SOA.org