Expert Views

Predictive disclosure modeling:
A tool for finding misrepresentation in life insurance portfolios

SCOR
The Art & Science of Risk

November 2022

Table of contents

Introduction 3
- What is misrepresentation? 3
- Why should insurers care? 3

Part A: What is a disclosure model? 4
- What is a disclosure? 4
- Visualizing disclosure rates 4
- Building a disclosure model 4
- Summary and discussion 6

Part B: Technical briefing 6
- Data cleaning and standardization 6
- Visualization 6
- Problem definition (X and y) 7
- Training the model 7
- Model evaluation 8
- Model types 9
- Actual / Expected 10
- Confidence intervals 10
- Model interpretation 11

About the SCOR Data Analytics Solutions team 14

Introduction

What is misrepresentation?

With high levels of automation in today's life and health insurance industry, most customers never have to share medical files to apply for insurance. Instead, insurers rely on the self-reported health status of their customers. Naturally, this process introduces additional risk by possibly missing key medical conditions. This could be due to human error, a lack of understanding of the underwriting criteria, or a conscious misleading by the customer or their financial advisor. SCOR UK's 2022 misrepresentation survey finds that misrepresentation in application forms costs the average insurance customer an estimated 5% to 10% in higher premiums.

To reduce risk, SCOR works with insurers to find and correct areas of misrepresentation, consults post-issue sampling (manually reviewing medical evidence such as an applicant's electronic health record after issuing the policy), reviews the automated underwriting processes, and monitors the quality of financial advisory firms. These actions focus on preventing or finding misrepresentation before it is too late, meaning long before claims occur.

Post-issue sampling is a method that allows measuring misrepresentation directly. It is, however, a time-consuming manual process which makes it prohibitively expensive to check all clients. As a solution, SCOR has found an opportunity to harness readily available underwriting data to analyze misrepresentation. This data enables disclosure modeling, which helps SCOR and insurers understand drivers for disclosure rates and areas of misrepresentation. Part A of this whitepaper explains the disclosure modeling process at a high level. Part B provides the technical details of a standard approach to disclosure modeling that SCOR has implemented with various UK life insurers: data cleaning, standardization, visualization, and predictive modeling.

Why should insurers care?

Disclosure modeling brings multiple advantages to various stakeholders in the life insurance journey. For insurers, the benefits of disclosure modeling and distribution quality management in general, include improved risk selection, claims experience, and reinsurance terms. For customers, this leads to increased certainty of coverage and payout of the claim (as the insurer might otherwise deny a claim in cases of misrepresentation). And less misrepresentation means customers will be paying lower premiums on average. At SCOR, offering our expertise to help clients improve their risk management through disclosure modeling is part of our unique value proposition.

Part A: What is a disclosure model?

What is a disclosure?

In insurance, a disclosure is defined by an applicant answering "yes" to a risk-related question on an insurance application form. This could be a medical question such as:

- "Do you have diabetes?"
- "Have you had treatment or advice for raised blood pressure or cholesterol?"
- "Have you suffered a heart attack, angina or any other heart condition?"

It could also include lifestyle questions such as:

- "Have you been banned from driving for driving under the influence of alcohol or drugs?"
- "Have you been skydiving or plan to do so in the future?"

An applicant's disclosure count is the total count of disclosures on their application. Averaging this over a group of applicants, we get the disclosure rate, which changes from one insurer to another insurer (and even for different products of the same insurer) as insurers ask different questions and different numbers of questions. Not all disclosures present the same risk as some may be more severe than others. But for the purpose of this whitepaper, we treat all disclosures equally.

Figure 1: A multi-page online dashboard that analyzes disclosure rates by many different dimensions.

Future extensions could be to weight disclosures according to their severity or focus on modeling disclosures for a single or group of conditions (for example, mental health disclosures).

Visualizing disclosure rates

Visualization of disclosure rates is the first step in understanding potential areas of misrepresentation, and it is greatly helped by a powerful data visualization tool (for example, Power BI, Tableau, or comparable tools available in the market). Figure 1 below shows graphs that split applicants' disclosure rate by customer demographics (e.g., age, gender, BMI, socioeconomic status, smoking status) and product data (e.g., type of benefit, sum assured, policy term), which provide a good understanding of how disclosure rates vary across a portfolio.

A detailed example of such as visualization can be seen in Figure 2, which shows a typical UK insurer's disclosure rate for different BMI rates. In Figure 2, applicants with BMI between 21 and 23 have the lowest disclosure rate at 2.1 disclosures per application, whereas BMIs between 45 and 50 show the highest disclosure rate at 4.7 disclosures per application. The portfolio shows a slight U-curve, with higher disclosure rates for applicants with low BMIs and applicants with high BMIs, which is a typical finding that reflects the correlation between BMI and other medical conditions.

Building a disclosure model

After visualization, building and analyzing a disclosure model is the next step. This helps find where there might be misrepresentation. In short, a disclosure model predicts the expected disclosure count for an applicant based on the applicant's demographic and product data. Each disclosure model is insurer-specific, as disclosure rates vary in definition across insurers and uses machine learning to learn from a dataset of applications. We have included a technical explanation of the modeling techniques in Part B.

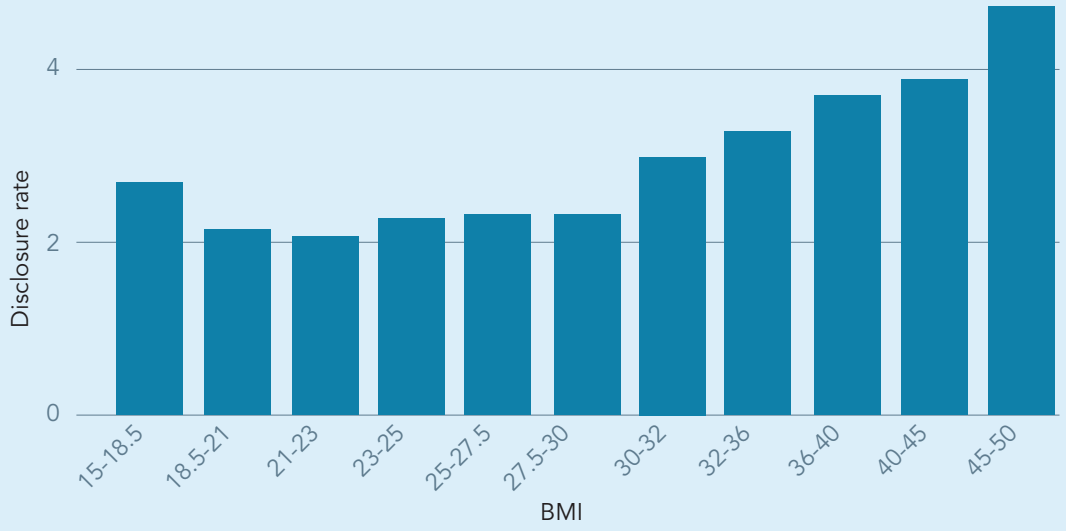
It is highly unlikely that we can predict an applicant's medical disclosures solely based on their age, smoking status, BMI, and some product characteristics. Indeed, in practice, we see a relatively large error when predicting an individual applicant's disclosure count. However, it is important to note that the goal of this disclosure model is not to identify individuals with misrepresentation, but instead, it is to identify opportunities within the portfolio to reduce misrepresentation rates.

In practice, this means grouping applicants together, for example, by distribution firm. By grouping applicants together and looking at their average predicted disclosure count, the error of the average prediction reduces as random errors cancel each other out according to the law of large numbers. Applicants with a distribution firm, for example, may show an actual disclosure rate that is significantly lower than their expected disclosure rate. In that case, this indicates that the firm's applicants have an unexpectedly low disclosure rate based on their demographic and product data. It is a sign of possible misrepresentation that could trigger post-issue sampling for these applicants or further investigation into the distribution firm's practices.

In a second example, we could see that an insurer is picking up more mental health disclosures in online applications than through financial advisors. This could raise the question if there is misrepresentation of mental health questions when an advisor is involved. A disclosure model for mental health disclosures can help answer this question by comparing actual and expected disclosure rates for the two groups. This way, we can account for the different customer mixes and product characteristics between online applicants and applications through advisors. The disclosure model might show no discrepancy between actual and expected disclosure rates for each group. Further analysis of the model could indicate that the difference in actual disclosure rates is explained by age: the online applicants are younger on average, and younger applicants are more likely to disclose mental health.

These examples hint at the power of modeling: as we add more and more variables to the analysis, only machine learning techniques can help account for their influence on disclosure rates.

Figure 2: Disclosure rate by BMI band for an imaginary portfolio of applications reflecting average trends across UK insurers.



Summary and discussion

Datasets of applications with disclosure counts and basic customer and product characteristics are often of non-sensitive kinds and readily available for analysis by insurers. Tapping into the value that these datasets hold, visualization of disclosure rates is the first step in understanding potential areas of misrepresentation. Building and analyzing a disclosure model is the next step and helps find where there may be misrepresentation, even if there is no data about actual events of misrepresentation. However useful, these tools never provide the "smoking gun" that misrepresentation is taking place. In fact, customer segments might genuinely be healthier after controlling for their independent variables such as age, gender, BMI, socioeconomic status, and product data; this would result in a lower actual disclosure rate.

Therefore, a disclosure model should always be used alongside other distribution quality management data and processes and processes. These could entail data about lapse rates, cancellation rates, complaint levels, declined claims, other warning signs of a firm's quality of service delivery, and post-issue sampling. With post-issue sampling, where an applicant's medical records are analyzed to establish the disclosures that should have been made on the application form, we can establish the truth about an applicant's disclosure rate. The power of the disclosure model presented here could be tied to direct post-issue sampling. Groups of applicants with an unrealistic actual-over-expected disclosure rate could be given a higher likelihood for post-issue sampling, leading to higher returns on the expensive practice of analyzing a person's medical files. That said, sampling should always keep a balance between exploration (fully random sampling that helps establish a ground truth of areas of misrepresentation) and exploitation (sampling from higher-risk groups to increase the efficiency of finding misrepresentation).

Part B: Technical briefing

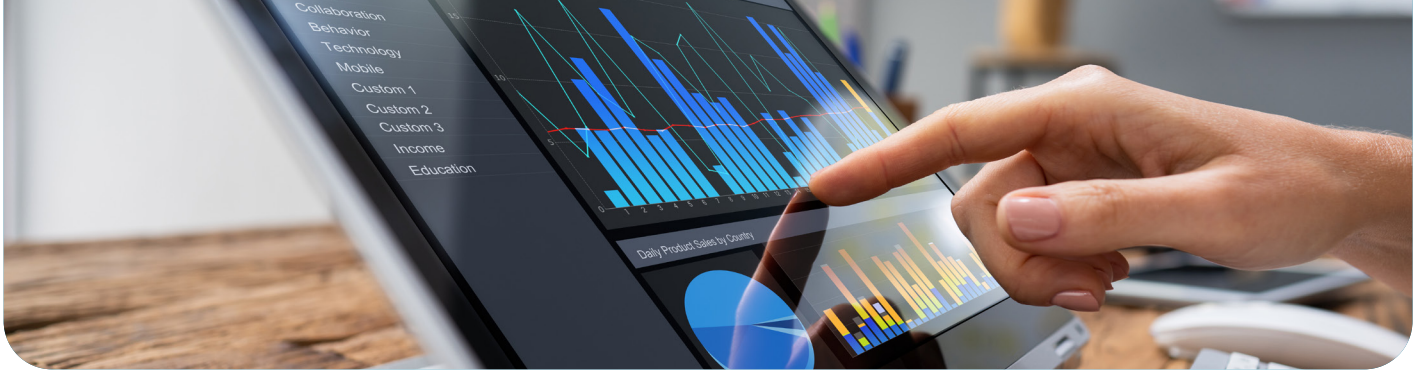
Data cleaning and standardization

The input to a disclosure model is the application data held by the insurer. It contains data on customer demographics (such as age, gender, BMI, socioeconomic status, and smoking status), product data (type of benefit, sum assured, and policy term), and self-reported disclosures (as defined above, e.g., diabetes, hypertension, heart conditions, and family history of cancer). As with any dataset, the data need to go through a process of making it ready for analysis, which is also called data cleaning. Working with many different insurers, SCOR finds it also helpful to bring the datasets into a standardized format to speed up the analysis, as definitions are clear and consistent. But more importantly, standardization helps us benchmark insurers against each other and find common trends.

Python is SCOR's tool of choice for this analysis, as it is a widely-used global standard, offering the convenience of open source, flexibility, and functionality for data manipulation. Provided with datasets that span hundreds of columns and categories that need to be standardized, we use common software engineering techniques such as object-oriented programming and configuration through data. This helps users in creating robust, maintainable, and reusable data pipelines.

Visualization

The first step to understanding what is happening with disclosure rates across an insurance portfolio is to visualize the data. Plots of the disclosure rates across different customer and product dimensions give an initial understanding. For example, we typically see a higher disclosure rate for women, older customers, and customers with a higher BMI. Modern business intelligence tools drastically reduce the effort of creating visuals that are split over many dimensions and need to be updated with each periodic data refresh. At SCOR, we use Power BI for disclosure analysis because it is an



easy tool to learn, collaborate, share, interact with visuals, and update with new data. Examples are shown in Figure 1 in Part A.

Problem definition (X and y)

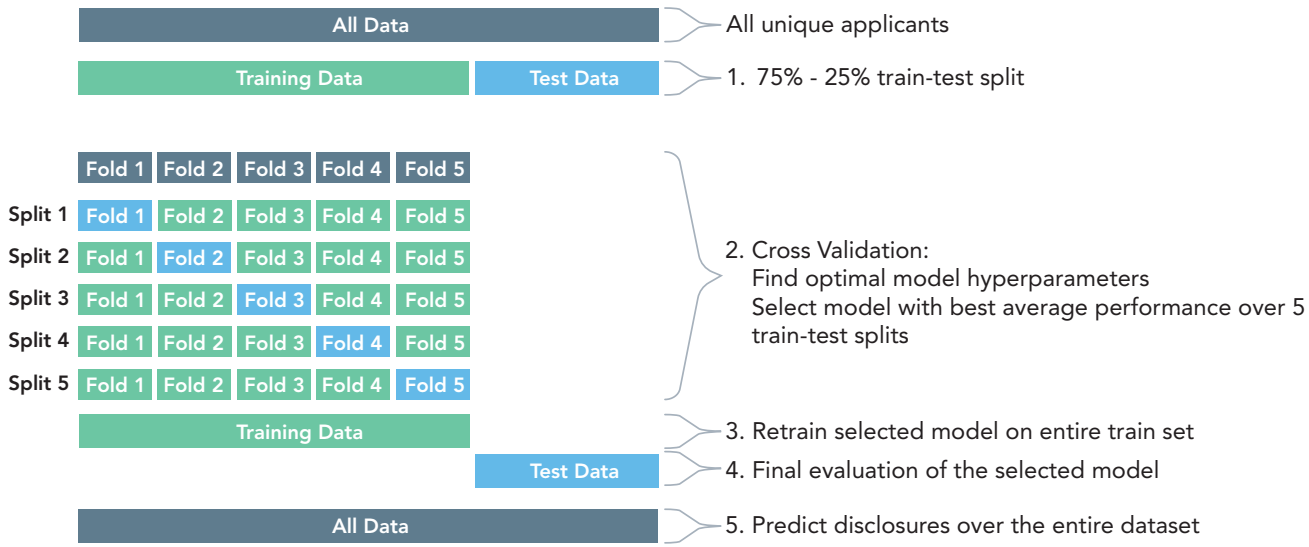
Moving from visualization to modeling the data, we first need to define the inputs and outputs of the model. A key challenge here is that we are looking for (pockets of) misrepresentation at a time when we only know an applicant's self-reported disclosures without verified ground truth data. We first define a machine learning model and, in the next section, apply assumptions that help us extract insights from it. We train the machine learning (ML) model on a target variable y which is disclosure count (see the section What is a disclosure? in Part A). As features or inputs X to the model, we use customer and product data points that are available at the time of underwriting. Examples of such product data are age, gender, socioeconomic status, type of benefit, sum assured, and policy term. These features can be numerical, ordinal, or categorical. As two more inputs to the model, we are following a standard actuarial practice to include smoker status and BMI. These could be seen as disclosures themselves and could also be misrepresented on insurance applications. However, smoker status and BMI are useful variables to predict other medical conditions, which is why we use them as inputs. For a list of recommended features, see Figure 5 in the section Model interpretation.

Training the model

To train a machine learning model, we need to split the available data so that we can keep track of performance, as shown in Figure 3. The splitting is done multiple times with different purposes:

1. We start by randomly sampling all unique applicants into a 75% - 25% train-test split. (These percentages are a good heuristic to follow but can be changed from dataset to dataset depending on the quantity and distribution of the data). The test set is kept separate and never provided to the model to learn from as input. As such, it can be used as an independent random sample on which we can evaluate our final model.
2. Most machine learning models have different configuration settings that can be defined through hyperparameters. This means that we could have many or practically infinite number of different model configurations that we could use. The effectiveness of these models may

Figure 3: Strategy for splitting the available data in train, test, and cross validation splits.



differ widely, so we want to find the optimal (or at least good enough) hyperparameters for our specific problem. We do this by training hundreds of different models, each with different hyperparameter settings. By nature of statistics, some of these models may perform well by coincidence on our specific test set. Some other models may perform well and will generalize well to other unseen data. Cross-validation allows us to find this latter class of models, by splitting the training data up into five different train-test sets, called cross-validation sets. Each model is trained five times, once on each cross-validation set, and the model that performs best on average is selected.

3. The selected model is retrained on the entire train set. This is to leverage the entire training set for training the selected model.
4. We evaluate the selected model on the test set (see section Model evaluation for details).
5. Steps 1 to 4 define a standard machine learning workflow for training and evaluating a predictive model. Normally, a predictive model is then used to make predictions on other (unseen) data where the ground truth is actually not known (that's why you'd want to predict it). However, in our case we use the model to make predictions over the entire applicant dataset The result is a predicted disclosure count that represents an applicant's expected disclosure rate given their age, gender, sum assured, etc. (all input variables).

Model evaluation

We define quantitative and qualitative evaluation criteria for assessing the performance of a disclosure model. In simple terms, we have a predicted disclosure count $\hat{y}$ and the actual disclosure count y. The closer $\hat{y}$ is to y, the better the model performs. There are different ways to define "closeness" and how to average over multiple applicants, see formulas 1 to 3 below.

Formula 1: Mean squared error (MSE) is a common evaluation metric, where a lower MSE is better (meaning predictions are close to their actual values). There is no agreed value for MSE that makes a model 'good'; instead, it is useful to compare the MSE of your model to another MSE on the same data. For example, it can be useful to calculate and compare the MSE of a dummy model that uses the mean disclosure count as the prediction for each applicant's disclosure count.

Formula 2: R-squared (R^2) measures the ratio of the total variance in y that is explained by the predictive model. It is also equivalent to the percentage improvement that a model has over the mentioned dummy mode.

Formula 3: Mean squared logarithmic error (MSLE) is a variation on the MSE that is useful for data with a Poisson distribution. MSLE measures the MSE of the relative error (instead of the absolute error). For example, the MSLE for $\hat{y}$=4 and y=3 (MSLE=0.05) is almost the same as for $\hat{y}$=8 and y=6 (MSLE=0.06; the relative error is the same, but the difference in MSLEs is caused by

1. $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

2. $$R^2 = 1 - \frac{MSE}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_y)^2}$$

3. $$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2$$

y_i	Observed disclosure count for applicant *i*
$\hat{y}_i$	Predicted disclosure count for applicant *i*
n	Total number of applicants
μ_y	Mean observed disclosure count
MSE	Mean squared error
R^2	R-squared
$MSLE$	Mean squared logarithmic error

the +1 in the formula above that helps us deal with predictions with 0).

Qualitatively, it is important to understand if a model overfits the data. An overfitted model uses associations between variables that exist in the training set (called random noise), but do not exist in the population. As a result, an overfitted model does not extrapolate well to data that it has not seen before. We can look at the difference in model performance in the above metrics on the training set with respect to the test set. This difference is also called variance and should be small. Preventing overfitting is important in our disclosure model, not because we are concerned about extrapolation on unseen data (we are not currently using it for that), but because we do not want to model random noise or have the model memorize the data. Our predictions should reflect "the expected disclosure count for an applicant given their inputs X", which breaks down if the model overfits.

Lastly, as another qualitative assessment, it is a good practice to prefer a simple model over a more complex model when both have the same predictive power. The number of coefficients or parameters that a model has and learns can be a good indicator of model complexity.

Model types

Our process for training and evaluating a model means we can treat the machine learning model as a black box. We could ignore how the model arrives at its predictions, as long as the model takes our inputs (X) and produces a disclosure count (y). As we evaluate an independent test set, we know how the black box performs on unseen data. But that is considering the model too simply; there is more to a machine learning model than just its raw performance on the test set.

For example, one class of models, generalized linear models (GLMs), are easy to interpret. By inspecting its coefficients, we can understand how it arrives at its prediction (meaning that to a data scientist, it is actually a white box model: we know what happens under the hood). This allows for simpler and clearer explanations to users of the model as to why a customer has been predicted to have that number of disclosures.

At the same time, this transparency brings an additional layer of evaluation and quality control that helps build confidence in the model. With GLMs, it is easy to prevent overfitting. This is in part because GLMs predict the target variable as a linear combination of the features, which means GLMs can only model linear relationships and cannot model feature interactions (unless those interactions are created as features). This simplicity can be a drawback for more complex datasets and could mean that GLMs do not perform as well as more complex models.

Two examples of GLMs are Ridge regression and Poisson regression. Ridge regression is multiple linear regression with a regularization term that penalizes high coefficient values, thereby helping to prevent overfitting. Poisson regression is similar to Ridge regression, but instead learns to predict the logarithm of the target variable. It then arrives at the final prediction by exponentiating the result of the regression. Poisson regression is a natural choice to model target variables with a Poisson distribution, i.e., where the variable is a (small) count of events or a natural phenomenon like disclosures.

Another class of models is the ensembled or boosted tree-based models such as Random Forest and XGBoost. In contrast to GLMs, these have the advantage of being able to model complex, non-linear relationships and variable interactions. Traditionally, these models are considered black boxes due to their complexity, while training them is considered difficult as they can easily overfit the training data. However, we can open the hood and understand what relationships the model has learned by using modern explainability techniques such as Shapley values. Additionally, the problem of overfitting can be addressed by controlling the complexity and regularization through hyperparameter tuning.

Actual / Expected

Moving from modeling into analysis, one of the key metrics is the actual over expected (A/E) ratio, which is a common focus for analysis in the life insurance sector. The A/E ratio is the actual disclosure count divided by the expected (or predicted) disclosure count. When observing the A/E for a group of applicants, an A/E higher than 1 indicates more disclosures than expected given the group's customer and product characteristics. Likewise, an A/E lower than 1 shows that the group has a lower disclosure rate than expected, which is a sign of possible misrepresentation within the group. As shown in Table 1, a possible grouping of applicants is by the distribution firm that helped them apply for life insurance.

Confidence intervals

In the *Building a disclosure model section of Part A*, we explained that a prediction of a single applicant's disclosure count would not be accurate; but when grouping applicants together, we can find discrepancies among their averaged A/Es. It is important and useful to attach a level of confidence to the A/E, and the ability to do so is a unique selling point of the disclosure model presented in this whitepaper.

Working on the expected average disclosure count for a group of applicants, equations 4 to 7 allow us to build a confidence interval. For example, to construct a 95% confidence interval, we can look up the two-sided z-value which is 1.96. If the actual average disclosure count for the group lies outside the confidence interval, we can state with 95% confidence that the difference between actual and expected is not due to any differences in the input variables X nor due to random noise in the disclosure count.

$$4.\ \sigma_e = \sqrt{\text{MSE}}$$

$$5.\ SE_j = \frac{\sigma_e}{\sqrt{n_j}}$$

$$6.\ CI_{lb} = \mu_{\hat{y}_j} - z \cdot SE_j$$

$$7.\ CI_{ub} = \mu_{\hat{y}_j} + z \cdot SE_j$$

σ_e	Standard deviation of the prediction error
n_j	Number of applicants in group (sample) j
SE_j	Standard error of the prediction error for group j
$\mu_{\hat{y}_j}$	Average predicted disclosure count for group j
z	Z-value corresponding to the confidence level
CL_{lb}	Lower bound of the confidence interval
CL_{ub}	Upper bound of the confidence interval

Table 1. Actual disclosure rates and expected (predicted) disclosure rates for five fictitious firms. Although three firms have lower than expected disclosure rates, only for the last firm this lower disclosure rate is statistically significant based on a 95% confidence level for our disclosure model.

Firm name	Applicants	Actual disclosure rate	Expected disclosure rate	A/E	95% Confidence interval
Distribution Firm 1	12	1.500	0.890	168.5%	[87.2%, 2200%]
Distribution Firm 2	15	1.133	1.488	76.2%	[50.9%, 151.3%]
Distribution Firm 3	68	1.147	1.221	93.9%	[73.3%, 130.8%]
Distribution Firm 4	75	1.293	1.109	116.7%	[90.1%, 165.6%]
Distribution Firm 5	3,399	1.225	1.287	95.2%	[91.7%, 98.9%]

From equation 5, we indeed see that the confidence interval is smaller for larger groups of applicants. Secondly, we see that the confidence interval will be smaller when our model has a smaller prediction error (a smaller MSE); i.e., the better our disclosure model is at predicting, the more confidence we have in those predictions being correct. Lastly, the size of the confidence interval is determined by the desired confidence level. The higher the confidence level, the larger z, and the larger the confidence interval. These effects on the size of the confidence interval can also be observed in the examples in Figure 4.

Model interpretation

In the section Model types, we explained a trade-off between quantitative model performance (high prediction accuracy) and qualitative characteristics such as model interpretability. We often treat machine learning models as black boxes that predict a useful quantity with a certain level of confidence. As noted earlier, some models may still be easy to understand. For example, for a linear model, we would only need to refer to each feature's coefficient to understand how that feature influences the target variable. However, at SCOR, we often rely on Shapley Additive Explanations (SHAP) to explain any type of machine learning model. SHAP quantifies the role of each variable in the final decision of the model. This type of model interpretation is a key step to building confidence in the model and using the predictions to guide business decision-making.

Explaining SHAP is beyond the scope of this whitepaper, but in summary we define some useful properties:

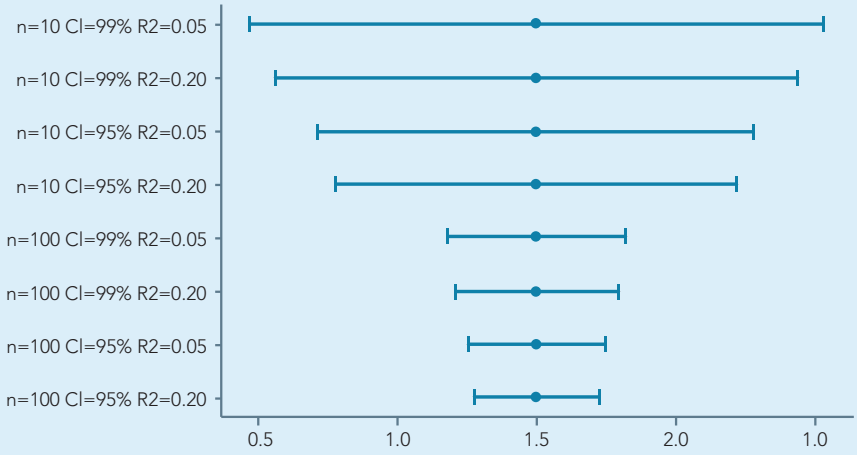
Property 1: Every sample (row) in the data has one SHAP value per feature.

Property 2: This SHAP value represents the amount to which the prediction of the target variable has been influenced by this sample's value of the feature. To explain how to calculate this SHAP value in simplified terms: it compares the prediction of models that include the specific feature for the sample, with the predictions of a group of models that exclude the feature.

Property 3: SHAP values are additive at a sample level: the sum of the SHAP values for a sample equals the difference between the prediction of the target variable and the target value's mean.

Property 4: SHAP values are also additive at a feature level: the absolute average of a feature's SHAP values represents the total average impact that this feature had on the prediction of the target variable.

Figure 4. Example confidence intervals for a predicted mean disclosure count of 1.5 for a group of n applicants with different levels of confidence (CI) and different levels of model accuracy (R2).



Property 4 can be seen in Figure 5, which shows us the global feature importance of a disclosure model using SHAP values. Compared with a plot of a linear model's coefficients, Figure 5 has two clear advantages. Firstly, the unit of measurement is in the target variable's domain; for example, the availability of the BMI feature changes the model's prediction of the disclosure rate on average by 0.25 disclosures. Secondly, Figure 5 balances the impact with the occurrence, so even though smokers have a higher disclosure rate than non-smokers of about 1.0 disclosure, given that only 15% of applicants smoke, the feature is less useful than BMI in predicting the disclosure rate.

Property 1 above is illustrated in Figure 6, which shows individual SHAP values for the feature AgeAtEntry. Figure 6 tells us that after controlling for all the other features, age has a mostly positive correlation with the disclosure rate (higher age leads to a higher disclosure rate). At closer inspection, we see that the impact of age is constant and negative for ages under 30, and there is a more or less linear, positively correlated effect from age 30 upwards. Note that the SHAP values for a linear model would always lie on a straight line, but because XGBoost can model non-linear effects (the shape of the relationship) and interaction effects (the width of the distribution of SHAP values for a single value of the feature), the distribution of SHAP values could show any pattern.

With Figure 7, the last SHAP visual that is useful for the analysis of disclosure rates is a strong example of the property of additivity. This waterfall chart can be used to understand why applicants at a certain firm have a higher or lower expected disclosure rate. In the example in Figure 7, we can see that the applicants at this firm have a lower expected disclosure rate of 1.234 disclosures, compared with the mean rate of 1.255. Averaged over all the applicants within the firm, the most important feature that drives the firm's predicted disclosure rate is gender, which lowers the expected disclosure rate by 0.03 disclosures (the firm has more male applicants, who have a lower disclosure rate). Other features have smaller effects, some in different directions.

Figure 5. Global feature importance of a disclosure model using SHAP values. A larger mean absolute SHAP value means that the feature has had a larger impact on predicting the target variable. An XGBoost model was used to predict the disclosure rate.

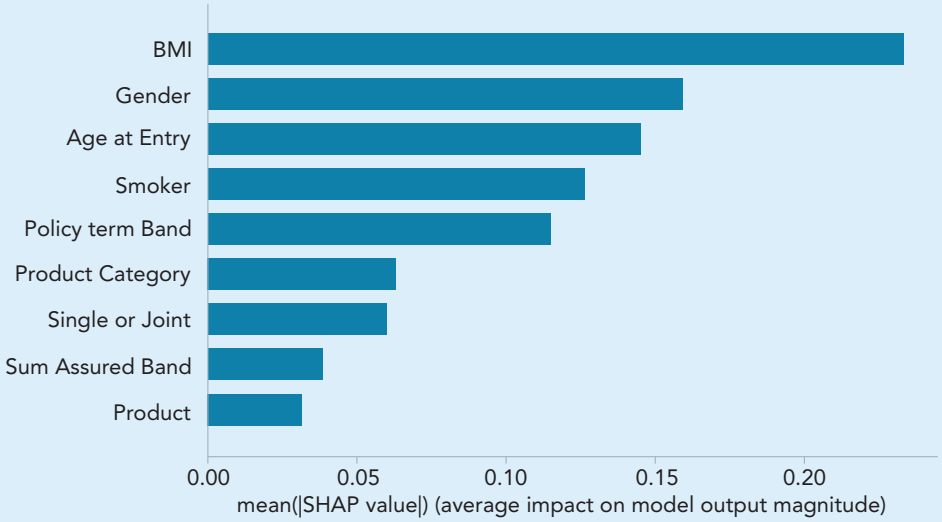


Figure 6. SHAP values for each sample for the feature AgeAtEntry (the applicant's age when applying for insurance). Positive values mean that the predicted disclosure rate was higher than if the model had not taken age into account. An XGBoost model was used to predict the disclosure rate.

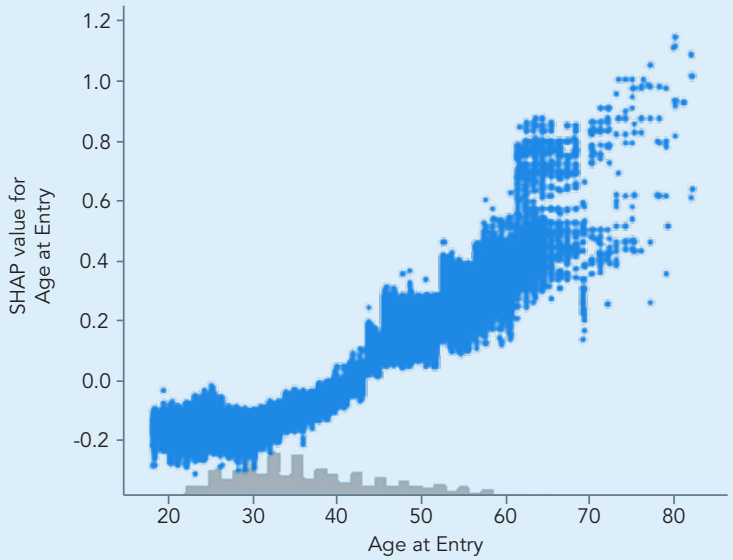
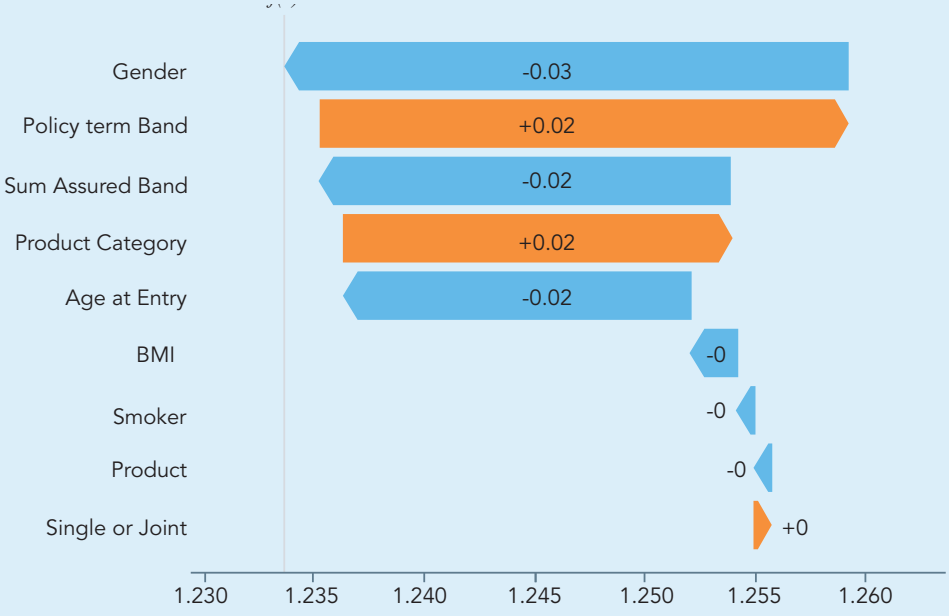


Figure 7. Waterfall chart of SHAP values for a group of applicants within a single firm. The bars show the mean SHAP value for that feature for the group of applicants. The waterfall shows the cumulative contribution of the features on the predicted mean disclosure rate for the group.



About the SCOR Data Analytics Solutions team

The Data Analytics Solutions (DAS) team is a global team that leverages the power of data to improve all aspects of the consumer journey for both Property & Casualty and Life & Health insurance products. This is achieved by combining SCOR's expert knowledge in insurance, machine learning, and software development with the business knowledge of insurance companies. For life insurers, the team has delivered end-to-end solutions to clients, including:

1. VITAE: biometric risk calculators that use machine learning to automate underwriting for specific conditions such as cardiovascular disease
2. Biological Age Model (BAM): leveraging wearable data to continuously engage with policyholders while empowering them to live healthier lives
3. Sylvanus: streamlined underwriting and claims processes through automated document analysis using Natural Language Processing

For all projects, the Data Analytics Solutions team works closely together with SCOR's local market. This whitepaper is the result of close collaboration between the UK market and DAS teams.

This article is written by:

Xander van den Eelaart
Core Data Scientist,
Amsterdam
xvandeneelaart-external@scor.com

SCOR
The Art & Science of Risk

November 2022