Received: 14 August 2024 | Revised: 1 October 2024 | Accepted: 13 October 2024

DOI: 10.1111/cts.70056

TUTORIAL

Practical guide to SHAP analysis: Explaining supervised machine learning model predictions in drug development

Ana Victoria Ponce-Bobadilla | Vanessa Schmitt | Corinna S. Maier | Sven Mensing | Sven Stodtmann

AbbVie Deutschland GmbH & Co. KG, Ludwigshafen, Germany

Correspondence
Sven Stodtmann, AbbVie Deutschland GmbH & Co. KG, Ludwigshafen, Germany.
Email: sven.stodtmann@abbvie.com

Abstract

Despite increasing interest in using Artificial Intelligence (AI) and Machine Learning (ML) models for drug development, effectively interpreting their predictions remains a challenge, which limits their impact on clinical decisions. We address this issue by providing a practical guide to SHapley Additive exPlanations (SHAP), a popular feature-based interpretability method, which can be seamlessly integrated into supervised ML models to gain a deeper understanding of their predictions, thereby enhancing their transparency and trustworthiness. This tutorial focuses on the application of SHAP analysis to standard ML black-box models for regression and classification problems. We provide an overview of various visualization plots and their interpretation, available software for implementing SHAP, and highlight best practices, as well as special considerations, when dealing with binary endpoints and time-series models. To enhance the reader's understanding for the method, we also apply it to inherently explainable regression models. Finally, we discuss the limitations and ongoing advancements aimed at tackling the current drawbacks of the method.

INTRODUCTION

The use of Artificial Intelligence (AI) and Machine Learning (ML) models in drug discovery and development offers great potential.[1–4] Several ML models have been developed for predicting the absorption, distribution, metabolism, and excretion (ADME) properties of small molecules,[5,6] pharmacokinetic-pharmacodynamic modeling,[7,8] understanding exposure-response relationships,[9] patient subpopulation detection,[10–14] patient stratification,[15,16] and biomarker detection.[17] To inform clinical decisions in drug development and build trust for these tools, it is crucial to understand how the predictors influence the model predictions and which ones are the most impactful.

The research area of explainable AI or interpretable ML aims to trace the decision-making process of a ML model and understand the key features driving its predictions.[18] Model interpretability can be achieved either by considering models that are inherently interpretable (e.g., linear, logistic regression, decision trees) or considering post hoc "explainability" methods that can be applied to black-box models (e.g., neural networks, random forests, XGBoost, gradient boosting machines). It is interesting to note that post hoc methods can be applied to both interpretable and black-box models, making them model-agnostic. In a recent perspective, Imrie et al.[19] classified the different model-agnostic methods according to the type of explanation they provide: (1) feature-based (How relevant are the data features to model predictions?), (2) model-based (Can the model predictions be explained using auxiliary meta-models?), (3) example-based (What other samples' model predictions are similar to a particular subset of samples?), (4) concept-based (Does the model appear to adhere to concepts derived from domain knowledge?), and (5) counterfactual explanations (What does the model predict when considering synthetic samples under particular scenarios?). This tutorial provides a practical guide to one of the most popular feature-based ML interpretability methods: SHapley Additive exPlanations (SHAP) analysis.

SHAP analysis is a feature-based interpretability method that has gained popularity thanks to its versatility which provides local and global explanations. It also provides values that are easy to interpret and can be easily implemented thanks to its easy-to-use packages that implement this method. Several publications in this field have used this methodology, and some of them have started to highlight some important considerations on its use.[9,10,20–23]

In this tutorial, we first briefly review the game-theoretical background of Shapley values, and their connection to SHAP analysis, and give remarks on the calculation of SHAP values. Then we describe the datasets used for this tutorial along with the supervised learning problems that we considered to exhibit the SHAP methodology. Next, we discuss the implementation and practical considerations of SHAP value analysis: visualization plots, special considerations when applying SHAP to classification endpoints, special considerations when dealing with time-series models, and available software to perform the analysis. Subsequently, we walk through a SHAP analysis in common statistical frameworks (linear regression and logistic regression) and present how to apply this analysis to the most common ML frameworks (random forest, XGBoost, neural networks) for both a regression and classification problem in Python. Furthermore, the application of SHAP analysis to time-dependent models is demonstrated. Finally, we summarize the limitations and mention other extensions that should be considered for this popular technique.

THEORETICAL BACKGROUND OF SHAP ANALYSIS

Mathematical background

SHAP analysis is rooted in Shapley values, a concept from collaborative game theory. Shapley values provide a fair distribution of a payout among players in a collaborative game where players work together for a common goal, even if the players may have contributed unequally. We will use the example of combination drug therapy and calculate the contribution of each drug on the response to illustrate the concepts. Let's assume Drug A, Drug B, and Drug C, when administered together, have a response rate of 90%. The individual drugs' response rates are:

Drug A:	40%
Drug B:	50%
Drug C:	60%

Let's assume we also know the response rates of the two-drug combinations, or in the language of game theory, coalitions of two (subsets of the group of size 2):

Drug A and Drug B:	70%
Drug A and Drug C:	65%
Drug B and Drug C:	80%

Given the single drug and dual combination response rates, what would be the fair individual drug contribution to the triple combination response rate of 90%?

In 1953, Lloyd Shapley[24] defined a set of properties that a fair measurement of contribution in a collaborative game should fulfill:

- Efficiency: The sum of all players' contributions must equal the payout (90% in our example).
- Symmetry: If two players contribute the same to all coalitions (subsets of players), they should receive an equal payout. (e.g., a generic of Drug A would be assumed to work the same as Drug A in all coalitions).
- Additivity: In a game with multiple subgames, each having a separate payout, the contribution of a player to the combined game is equal to the sum of contributions to each individual subgame. (e.g., if there are two subpopulations of patients, then Drug A's contribution to the total population would equal the sum of the contributions to each of the subpopulations).
- Null player: If a player doesn't contribute to any coalitions, their share of the payout is 0. (e.g., if a drug does not add responders in any combination, it is assigned no contribution).

Lloyd Shapley derived a mathematical formula of the values ϕ_j that satisfies these properties and proved that these values are the only ones capable of satisfying all these properties. These values are the so-called Shapley values. Considering a payout, V, each player j is assigned a Shapley value, ϕ_j, that corresponds to their fair share of the payout based on their individual contribution. The mathematical formula for ϕ_j is the following:

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} \left(V\left(\mathrm{S} \bigcup \{j\}\right) - V(S) \right) \quad (1)$$

where S is a coalition, N is the set of all players, $\left(V\left(\mathrm{S} \bigcup \{j\}\right) - V(S)\right)$ quantifies the marginal contribution of player j to coalition S, $\frac{|S|!(|N|-|S|-1)!}{|N|!}$ is the weight of the marginal contribution and $\sum_{S \subseteq N \setminus \{j\}}$ sums over all possible coalitions without j. The weights are the inverse of the number of coalitions of size $|S|$ excluding player j.

Shapley values can be interpreted as the weighted average of a player's marginal contributions across all possible coalitions.

Let's calculate the Shapley values for the combination therapy example. First, for each drug we consider all the possible coalitions that can be created without that specific drug. For example, for Drug A, these coalitions are: one coalition of size 0: $\{\oslash\}$, two coalitions of size 1: $\{B\}, \{C\}$ and one coalition of size two: $\{B, C\}$. We calculate the marginal contribution when adding Drug A to each of these coalitions and use Formula 1 to calculate its SHAP value. We then do the same for Drug B and Drug C. In Table 1 we include these calculations along with each drug's marginal contribution and their corresponding Shapley values. The sum of all Shapley values equals the total response: $20.83\% + 33.33\% + 35.83\% = 90\%$.

Connection between Shapley values and SHAP analysis

The use of Shapley values for ML interpretability was first introduced by Štrumbelj and Kononenko in 2010.[25] The authors showed how Shapley values can be used to fairly quantify the contribution of features in a ML model. The method was then popularized and extended by Lundberg et al.[26] to the so-called SHAP analysis. In their paper, the authors were able to show how Shapley values unified other feature-based interpretability methods and provided an open-source package that sparked a widespread adoption of the method.

The connection between Shapley values and SHAP analysis can be understood by considering the potential model features (covariates) as "players" in a collaborative game: working together to determine each individual predicted value. Table 2 illustrates the terminology from game theory and ML that establishes this connection. For each term in game theory, we also include (in parenthesis) the equivalent term for our combination therapy example. By establishing this link, SHAP values inherit all the properties of Shapley values, such as efficiency, symmetry, additivity, and null player. Consequently, SHAP values provide a fair and unbiased measure of each feature's contribution to the predicted value of each sample. To provide more clarity to the reader, Table S1 establishes a connection between the terminology used in ML and the corresponding terminology in statistics.

From the efficiency property, a property of SHAP values that is crucial for their interpretation can be derived: consider the prediction for a sample i, $f(i)$. The SHAP values for all features j of sample i, $\phi_{\mathrm{i,j}}$, satisfy the following formula:

$$\underset{\substack{\text{Model prediction} \\ \text{for sample } i}}{f(i)} = \underset{\substack{\text{Average} \\ \text{prediction}}}{\phi_o} + \underset{\substack{\text{Sum of all SHAP} \\ \text{values for sample } i}}{\sum_{\text{features}} \phi_{i,j}} \quad (2)$$

This means that for a given sample i, the SHAP values assigned to each feature describe the extent to which that feature contributed to the difference between the individual prediction of this sample from the average model prediction. A derivation of this formula from the efficiency axiom is presented in Molnar et al.[27]

TABLE 1 Marginal contributions for drug combination example illustrating calculation of Shapley values.

Coalition size of S	Drug A's marginal contributions	Drug B's marginal contributions	Drug C's marginal contributions
0	V(A) – V(⊘) = 40	V(B) – V(⊘) = 50	V(C) – V(⊘) = 60
1	V(B,A) – V(B) = 70–50 = 20 V(C,A) – V(C) = 65–60 = 5	V(A,B) – V(A) = 70–40 = 30 V(C,B) – V(C) = 80–60 = 20	V(A,C) – V(A) = 65–40 = 25 V(B,C) – V(B) = 80–50 = 30
2	V (B,C,A) – V(B,C) = 90–80 = 10	V(A,C,B) – V(A,C) = 90–65 = 25	V(A,B,C) – V(A,B) = 90–70 = 20
Shapley values	$\frac{1}{3}40 + \frac{1}{6}20 + \frac{1}{6}5 + \frac{1}{3}10 = 20.83\%$	$\frac{1}{3}50 + \frac{1}{6}30 + \frac{1}{6}20 + \frac{1}{3}25 = 33.33\%$	$\frac{1}{3}60 + \frac{1}{6}25 + \frac{1}{6}30 + \frac{1}{3}20 = 35.83\%$

TABLE 2 Linking the terminology from game theory and ML for Shapley values.

Game theory	Machine learning
Players (drugs)	Features
Payout (response rate)	Model prediction of an observation, known as sample
Marginal Contribution (response each drug delivers when considered in a possible combination)	Value of the feature for the specific sample
Shapley value of player *j* (fair individual contribution of drug *j* for the triple combination response rate)	SHAP value of feature *j*

Calculation of SHAP values

Typically, using Formula 1 to calculate SHAP values for feature contributions of a ML model prediction is too computationally expensive; therefore, different implementation and approximation methods have been considered to efficiently calculate SHAP values for common ML models. An exception to this rule are tree-based models, for example, XGBoost, random forest, LightGBM, CatBoost, for which it is possible to efficiently calculate the exact SHAP values by exploiting the tree structure of models.[28]

For neural network-based models, approximation methods can leverage the computational graph or exploit the model differentiability to calculate the SHAP values. In the popular Python package SHAP, examples of such approaches are implemented in DeepExplainer and GradientExplainer. Finally, an approximation method that can be applied to any supervised model, (i.e., a model-agnostic method) is the kernel SHAP method. Kernel SHAP approximates the Shapley values by using a weighted sampling approach. One needs to define a representative subset of the dataset often called *background dataset* (which should capture the range and distribution of the features in the dataset) from which subsets of features are sampled. A kernel function is applied to compute the contributions of these subsets to the prediction. These contributions are then weighted based on the number of times each feature appears in the subsets. By repeating this process multiple times, Kernel SHAP provides an estimate of the Shapley values.

TUTORIAL DATASETS

To introduce the SHAP methodology and allow the reader to reproduce our tutorial we utilized both public datasets and in-silico datasets.

To illustrate the use of SHAP analysis in different black-box regression models, we utilized the National Health and Nutrition Examination Survey (NHANES).[29] These surveys are designed to assess the health and nutritional status of US adults and children. The NHANES survey includes demographics, socioeconomic, dietary, laboratory, and health-related questions. We create a dataset using a subset of this survey data. We consider the features of sex, body mass index (BMI), age, estimated number of days over the past year that the participant drank alcoholic beverages, and the number of days in a typical week that the participant does moderate or vigorous physical activity to predict blood pressure.

For showcasing the use of SHAP analysis in different black-box classification models, we employed a breast cancer diagnostic dataset. This dataset consists of features computed from digitized images of a fine needle aspirate of breast masses. These features describe the characteristics of the cell nuclei present in the image. Based on the features of the different samples, we are interested in classifying the tumor mass as benign or malignant. A more detailed description of the dataset is available here.[30]

To show the application of SHAP values to a time-dependent ML model, we utilized in-silico data from a pharmacokinetic (PK) model. This PK model consisted of a first-order absorption and linear clearance. We assumed clearance exhibits interindividual variability. Further details on the PK model and model parameters can be found in Supporting Information. We considered the regression problem of predicting individual clearances using a Long Short-Term Memory (LSTM) model from drug concentration-time profiles. We simulated 1000 drug concentration-time profiles during a 48-h period, assuming a daily infusion of 200 mg lasting for 1 h. The drug concentration was sampled every 4 h, starting 1 h after the first dose was administered, resulting in 12 recorded concentrations for each simulated subject.

A description of where to find the datasets and how we pre-process them before training the ML models is available in the Supporting Information. When training the models, we consider a 70:30 train:test split. The code used for performing all the SHAP analyses can be found in the Supporting Information.

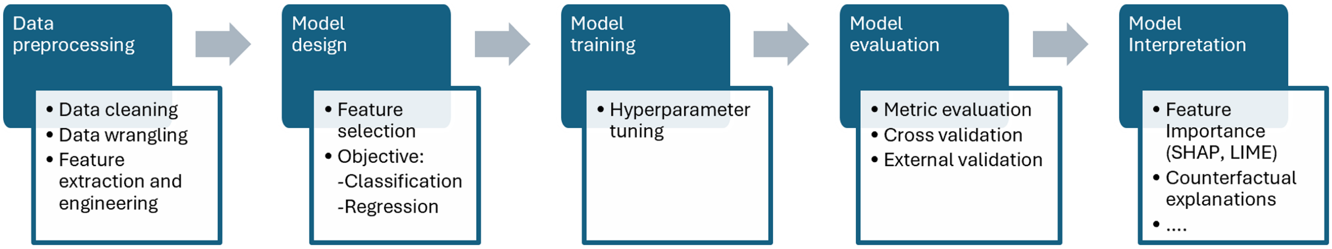


FIGURE 1 Standard supervised ML workflow.

IMPLEMENTATION AND PRACTICAL CONSIDERATIONS

Consider the standard supervised ML workflow depicted in Figure 1.

SHAP analysis is part of the model interpretation phase in a ML workflow and should be only performed if the model demonstrates adequate performance. Therefore, the reported SHAP values should usually be the SHAP values that correspond to the test dataset.

As pointed out by Lu et al.,[23] a good practice in ML consists of performing the model validation using k-fold cross-validation. Analyzing the distributions of SHAP values across the different folds allows us to understand if the SHAP value trends observed in the visualization plots are consistent across the data and not just present in a subset of the dataset.

Overview of visualization plots

Different plots can be used to visualize SHAP values, which allow either local or global explanations and each has a specific interpretation, purpose, advantages, and disadvantages. In this section, we describe the most common plots that are offered by the popular Python package SHAP.[26,28]

To illustrate the use of these plots, we first trained an XGBoost model for the blood pressure regression problem introduced in the tutorial datasets. After assessing that the model has an adequate fit, we used the SHAP package to calculate the SHAP values for both the train and test dataset. If not otherwise specified, the plots show the SHAP values for the test dataset. In Figure 2 we show the most common visualization plots for SHAP values. In this section, we describe the different parts of the plots, their purpose, advantages, and disadvantages. The code to create these plots can be found in the Supporting Information.

Bar plot

The bar plot, a global explanation method, displays the mean absolute SHAP value for each feature across all predictions, serving as a measure of feature importance. It quantifies, on average, the magnitude of each feature's contribution to the model prediction; thus, it provides a ranking of features.

An advantage of this measurement for determining feature importance is that the SHAP values are expressed in the same units as the model predictions making them more intuitive. For example, in our regression example, the average impact of the variable age on the predictions of the model is 5.86 mm Hg. To illustrate this point, we refer to Formula 2, where we see that if the prediction is a blood pressure (e.g., mm Hg), all SHAP values will also have the same units, mm Hg. This contrasts with other feature importance measurements like permutation feature importance scores. A disadvantage of this representation is the lack of nuance on the directionality of the impact, or whether the relationship is monotonous. Also, selecting a cut-off based on the SHAP values, to identify the most important features, depends on domain knowledge and should be determined on a case-by-case basis and does not follow a general rule.

In Figure 2a we observe that when considering the XGBoost model for predicting blood pressure, age appears to be the most important feature followed by BMI and then sex.

Beeswarm plot

The beeswarm plot provides a global overview of SHAP values for selected features, with rows representing each feature ranked by the mean absolute SHAP value. The individual dots in each row are the SHAP values for that particular feature in each sample of the dataset. Two properties of the dot encode the information of the SHAP values:

- The point color indicates the feature value (e.g., age) according to the color bar. The color bar is normalized according to the feature range such that high feature values appear in red and low values appear in blue. In the case where the ML model can handle missing data (e.g., xgboost), the point color would be shown in gray.

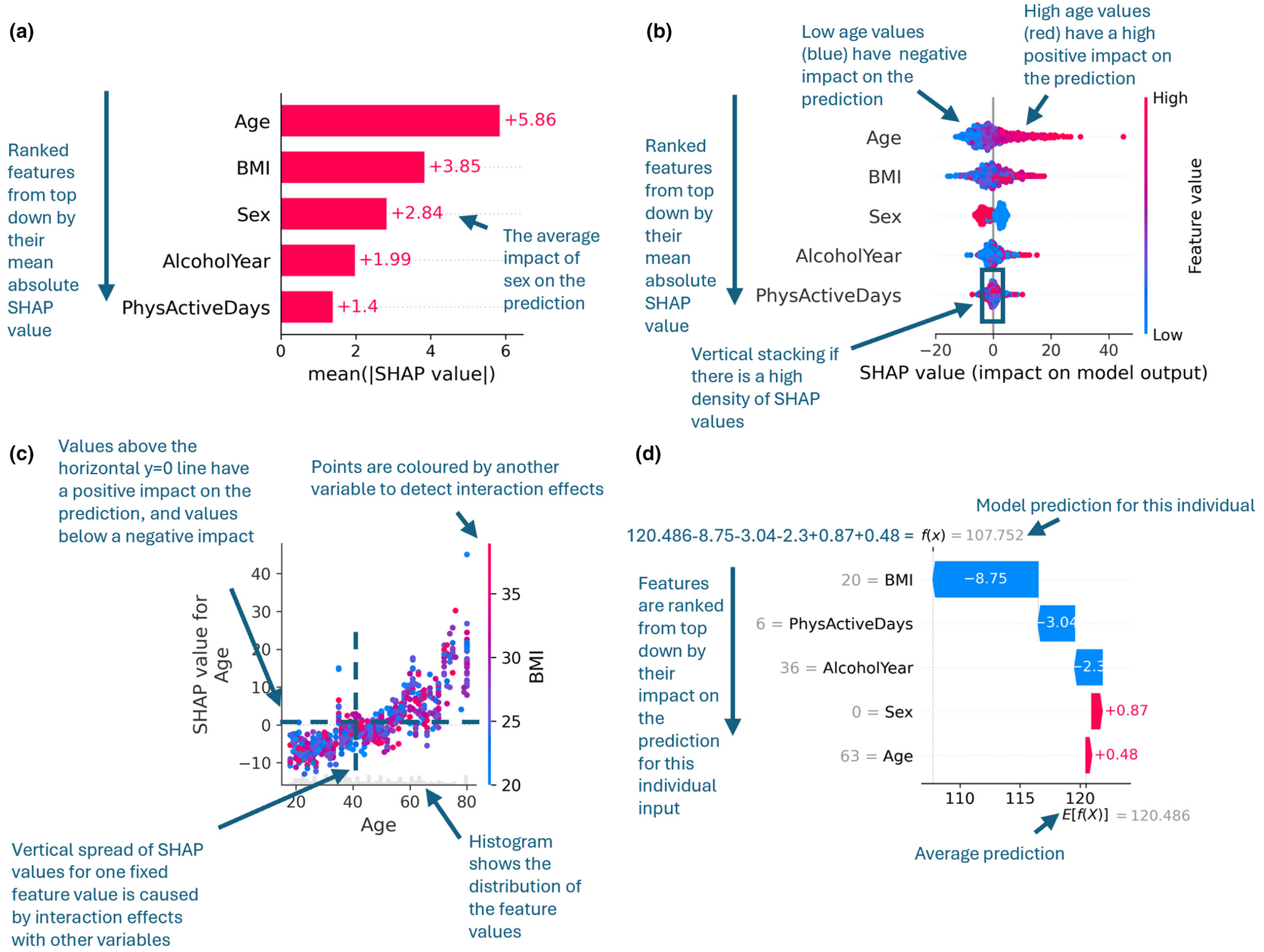


FIGURE 2 Different visualization plots of SHAP values from an XGBoost model when predicting blood pressure: (a) Bar plot; (b) Beeswarm plot; (c) A scatter plot for the feature age colored by each subject's BMI; (d) Waterfall plot for an example subject.

- The point position along the x axis is determined by their SHAP value.

Inspection of the color distribution for each feature along the xaxis can provide global insights into the relationship between the feature values and its SHAP values:

- From an initial inspection, can we determine the relationship between the feature values and SHAP values for each feature? Do they all show monotonous increasing/decreasing trends, for example, increasing feature value is related to a higher prediction? Can we detect any non-linear trends?
- Are there features whose SHAP values distribution are alike? Do they have the same spread? How does the distribution of the SHAP values differ between the different features?
- Are there outlier SHAP values for some features?
- Are the missing values of different features random, or do they show an interpretation bias?

An advantage of this plot is the ability to show multiple features at once, ranked by their impact, and at the same time give an indication of directionality and shape of the impact. A disadvantage is the use of color as a scale since it can be difficult to extract the shape of a relationship from the color differences.

In Figure 2b the beeswarm plot shows a detailed overview of the SHAP values for our regression example. We observe that the feature age has a wide SHAP values range and has much higher values than the mean absolute value (5.86). Age appears to have a generally monotonous increasing trend. Age and BMI SHAP values appear to follow a similar pattern hinting toward a possible interaction. This can be further investigated in the following plot.

Scatter plot

The scatter plot in displays the relationship between one feature value (x axis) and the SHAP values (y axis). It consists of (1) the scatterplot in which each point represents the sample's feature value on the x axis and the SHAP value on the y axis and (2) the inset gray histogram above the x axis which displays the feature value distribution.

The distribution of the points informs us whether there is a general trend between the feature values and SHAP values and allows us to identify linear or non-linear relationships. It may be helpful to include trend lines to detect which type of relationship is present. The vertical spread of SHAP values at a fixed feature value is a sign of interaction effects with other features in the model. It is important to consider the inset histogram when interpreting the trends. A high number of samples present in a particular part of the plot indicates that the trend in that area is supported by many samples. Conversely, when the number of samples is limited, it is important to be cautious and refrain from overinterpreting the trend.

Scatter plots may be colored according to another feature's value, by either coloring according to another feature value or first binning the other feature's values and considering coloring according to the bins.[23] This allows to detect interaction effects between different features.

The advantage of this visualization is the level of detail which clearly indicates the direction, magnitude, and variability of the feature's impact on model predictions. This comes at the price of only allowing a single feature per figure or panel.

In Figure 2c we consider the scatter plot for the feature age colored by the subject's BMI. We observe that age above 50 generally has a positive SHAP value (higher blood pressure) and we observe some non-linearity in the relationship. However, we are unable to detect any interactions between the feature age and BMI.

An extension of this plot can be considered when doing k-fold cross-validation such that the points are colored to the different folds they belong to, and the plot is faceted by the train-test status. Trend lines can also be included per fold. This plot then allows us to corroborate if trends are consistent across different subsets of data.

In Figure 3, we consider this extension of the scatter plot for the feature age. We performed a fivefold cross-validation and calculated the SHAP values for each fold for the test and the train dataset. We can observe that the trend of the SHAP values for age is consistent across the folds and across the train-test split. This demonstrates that the feature impact is stable and trustworthy.

Waterfall plot

A waterfall plot is used to visualize the predicted SHAP values of a single sample showing all features. The predicted value for this specific sample is given in the plot title.

The model features are located along the y axis, and the value of that feature for that specific sample is denoted in gray. The SHAP value corresponding to each feature for this specific sample is written in the main

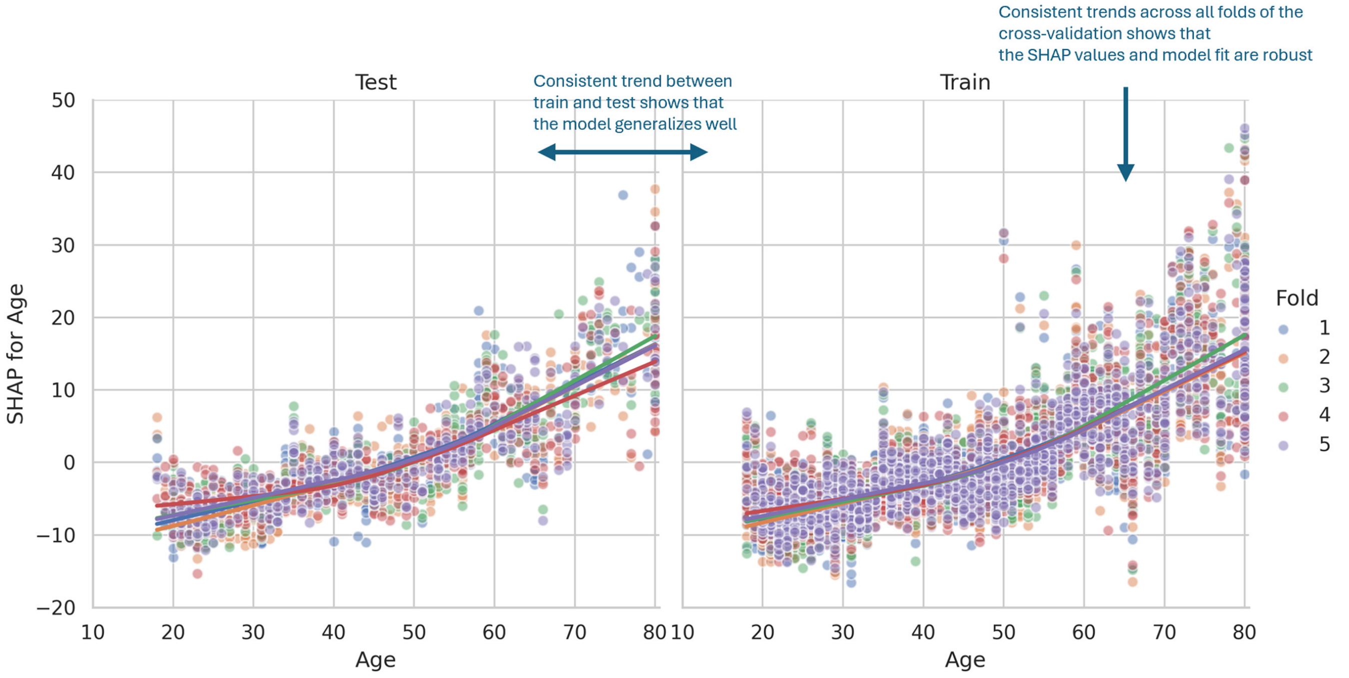


FIGURE 3 Scatter plot of SHAP values for feature age faceted by train-test status when considering cross-validation. The trends across the different folds are depicted in different lines corresponding to the fold.

panel within an arrow for each row or feature. The row is red (blue) if the SHAP value increases (decreases) the prediction $f(x_i)$ in comparison to the expected or average prediction.

This type of plot provides a local interpretation since it focuses on a single sample. This is useful when analyzing model outliers, SHAP values outliers, or specific subjects of interest.

In Figure 2d, we show the waterfall plot for an example subject. We can see that BMI is the feature that affects the difference between its model-predicted blood pressure the most when compared to the average prediction.

More references on other plots like heatmap plots, force-plots, decision plots, violin plots, stacked force-plots, interaction plots, to name a few and their interpretation can be found in other publications.[27,31,32]

Binary endpoints

For classification ML models, special care needs to be taken when calculating SHAP values. Using SHAP for explaining a classification model prediction can be set up in two ways: (1) explaining the predicted probabilities or (2) explaining the predicted log-odds for each sample belonging to that class. Explaining probabilities makes it easier to identify the contribution of features to the prediction; however, explaining the predicted log-odds can provide a clearer picture of the relationship between the model features, making the log-odds sometimes easier to interpret. This can be due to nonlinear interactions and implicit interactions. It is worth considering both explanations.

To illustrate the application of SHAP analysis for classification models, we trained an XGBoost model using the breast cancer diagnostic dataset. After assessing that the model has an adequate fit, we used the SHAP package to calculate the SHAP values for the test dataset. In Figure 4 we show the bar plot, beeswarm plot, and the scatter plot for the most important feature (concave points).[33] On the left plots, we display the SHAP values when explaining the predicted probabilities, while on the right, we present the corresponding SHAP values when explaining predicted log-odds.

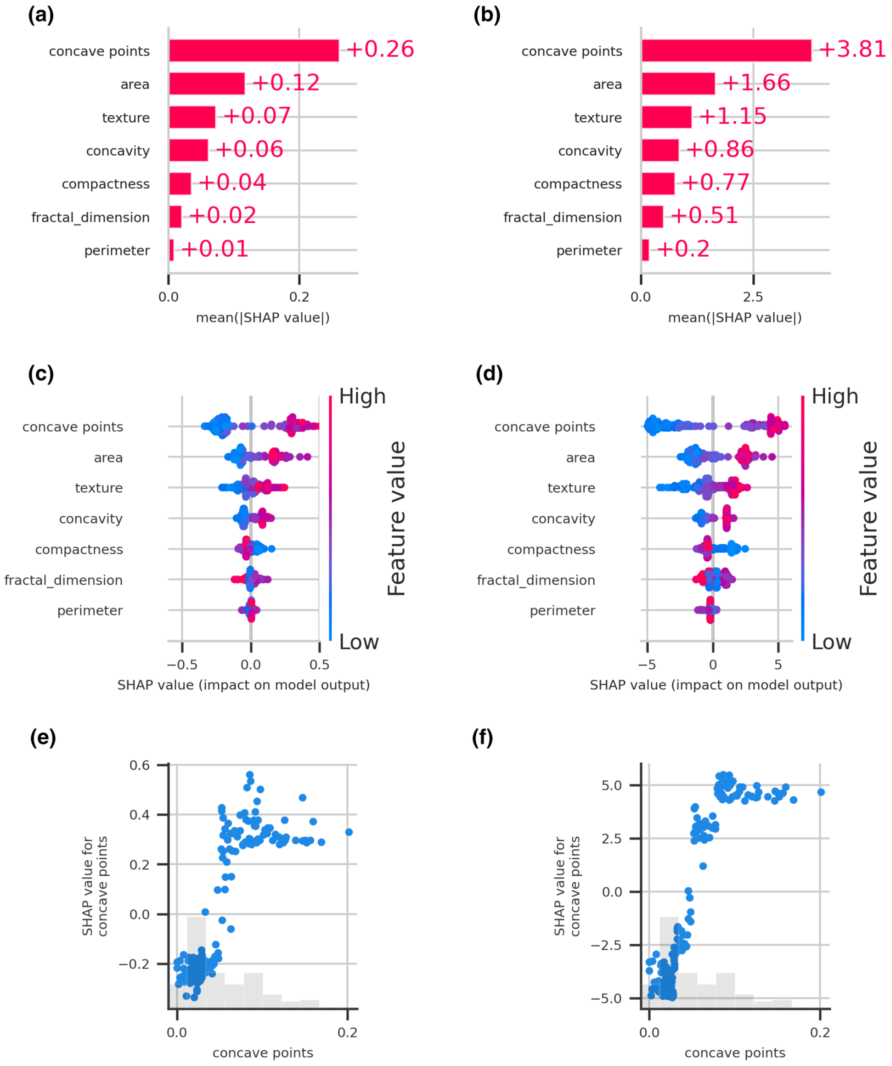
When comparing the SHAP plots for explaining the predicted probabilities and predicted log-odds, the result is quite similar. Both bar plots (Figure 4a,b) show that the feature concave points has the highest mean absolute SHAP value followed by area and texture. Note the different values of the SHAP values in probabilities (concave points have a mean absolute SHAP value of 0.26) and in log-odds (has a mean absolute SHAP value of 3.81). This signifies that concave points are the most influential feature in classifying the breast mass as benign or malignant. The beeswarm plots (Figure 4c,d) can give a better overview of the relationship between the feature values and its SHAP values. Both beeswarm plots reveal a positive relationship for concave points, area, texture, and concavity which means the higher the feature value, the higher the probability or log-odds. For the feature area, we can see that some samples with a middle feature value (purple) are associated with the highest SHAP values, indicating that the relationship is not linear. For a closer look at the relationship between a feature's values and the model's predicted outcomes, we consider the scatter plots. Figure 4e,f shows the scatter plots for the most important feature: concave points. Both scatter plots show an increase in SHAP values for increasing concave points up to 0.09. After that, increasing feature values the SHAP values are mostly flat, especially for predicting log-odds. In addition, at a feature value of 0.05, the model switches from predicting lower to higher probabilities or log-odds.

SHAP for time-dependent ML models

The use of SHAP analysis for time-dependent is a challenging[34,35] and currently an active area of research. Traditionally, researchers have performed SHAP analysis of a ML model prediction that considers time-dependent data by crafting features that capture the time-dependent aspects of the data and subsequently treating the problem as standard classification or regression models.

We consider the time-series in-silico data created from the PK model described in the tutorial dataset section. The drug concentration over time of this model is shown in Figure 5a for an example subject. The samples are labeled according to the cycle they belong to and the time since last dose (TSLD). Each concentration is denoted by $C_{\text{cycle,TSLD}}$. We trained a LSTM model to the simulated data and calculated the SHAP values using GradientExplainer. In Figure 5, we show different visualization plots of the SHAP values from this model. In Figure 5b, we can see that the most important features to predict the individual clearances are $C_{1,9}$ followed by $C_{1,13}$ and $C_{1,17}$. The beeswarm plot in Figure 5c shows a monotonous decreasing trend for the concentrations in the elimination phase except for the first concentrations after $t_{\max}$ (i.e., $C_{0,5}$ and $C_{1,5}$). The scatter plot in Figure 5d,e show that SHAP values for the drug concentrations with the same TSLD for the different cycles follow the same trend. The scatter plot Figure 5f shows that the SHAP values for the first drug concentration at cycle 1 have a monotonous increasing trend as expected.

FIGURE 4 Visualization plots of SHAP values derived from an XGBoost model for a classification problem, explaining the predicted probabilities (a,c,e) and the predicted log-odds (b,d,f).



Software

For Python, the most popular package that implements SHAP analysis is the SHAP package that is compatible with standard ML packages such as keras, tensorflow, scikit-learn, and pytorch. For R and specifically XGBoost there is the package SHAPforxgboost[36] available in CRAN. The shapper package is also available in R and it is a R wrapper of the SHAP Python library.[37]

EXAMPLES

In this section, we include the results of performing SHAP analyses for popular ML black-box models for classification and regression problems, specifically, we consider XGBoost,[36] random forest, and neural networks (multilayer perceptron model).

To further demystify how to interpret SHAP analyses on black-box models, we apply this methodology to linear regression and logistic regression models using in-silico data. This will help the reader gain intuition on how to interpret the different visualization plots within a familiar framework.

For both black-box models and interpretable models, we provide the Python code in the Supplementary Information for performing the SHAP analysis.

Common ML frameworks

We considered XGBoost, random forest, and a neural network model as regression models and trained them using the regression dataset. In Figure 6, we can see the bar plot, the beeswarm plot, and the scatter plot of the SHAP values of the variable age for the test dataset of the fitted models. When comparing the bar plots (Figure 6a,c,e), we observe that for all three models, age has the highest mean absolute SHAP value. However, for the XGBoost and the neural network, BMI has the second-highest value, followed by sex. In contrast, for the random forest model, sex has the second-highest

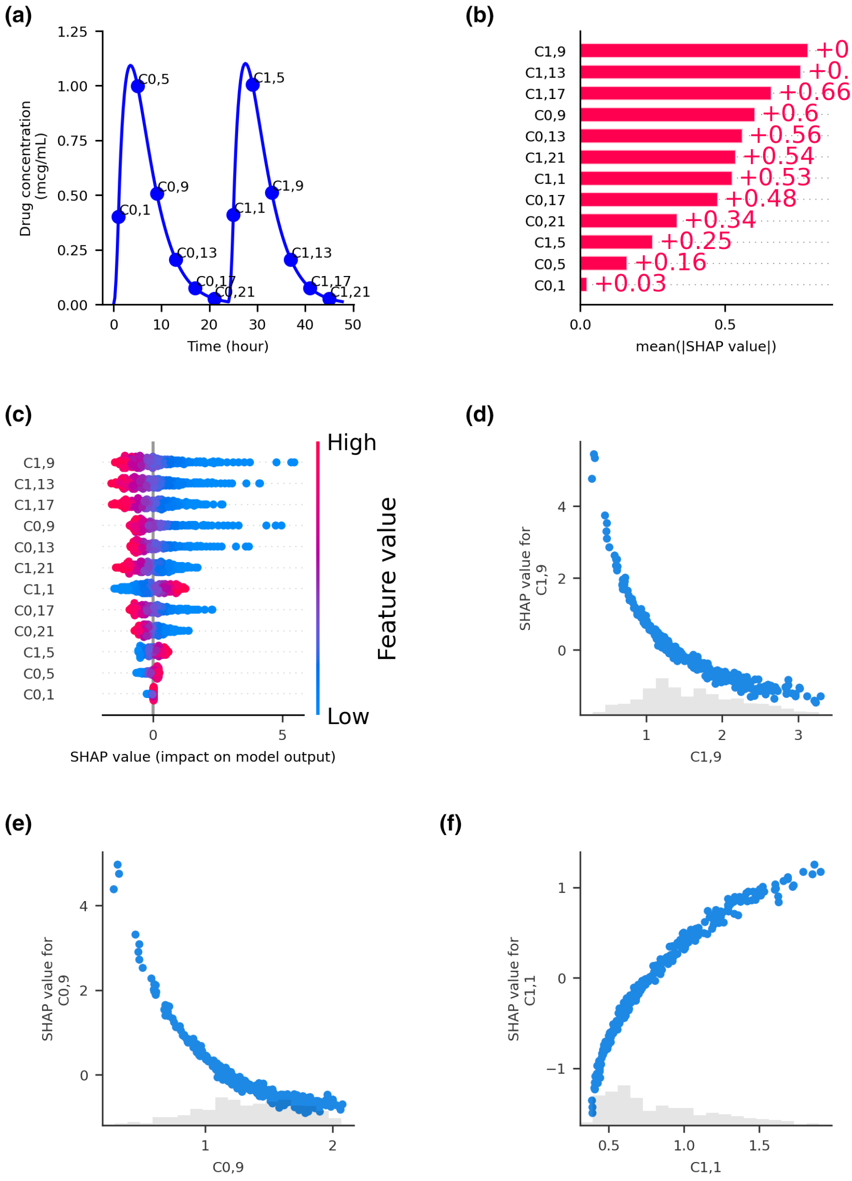


FIGURE 5 (a) Example time-course of the PK model considered to model drug concentration. Different visualization plots for the SHAP values explaining the predictions of individual clearances are depicted; (b) Bar plot; (c) Beeswarm plot; (d–f) Scatter plots of SHAP values corresponding to concentration at different times.

mean absolute SHAP values, followed by BMI. The beeswarm plots (Figure 6b,d,f) are also comparable between all three models. They show a positive relationship between age and BMI which means the higher the feature value, the higher the outcome. Especially for the neural network we can see the smooth color gradient that transitions from blue to red indicating that the positive relationship is nearly monotonous. For the random forest and XGBoost model, some mixed colors are visible around the SHAP value of 0. The categorical feature, sex, clearly shows a negative relationship for all three models. The beeswarm plot doesn't show a clear trend for AlcoholYear or PhysActiveDays, especially for the random forest model and neural network.

We considered XGBoost, random forest, and a neural network model as classification models and trained them using the breast cancer diagnostic dataset. In Figure S1, we can see the different visualization plots for the SHAP values for explaining the predicted probabilities for the different models. We can see that XGBoost and random forest ranked the different features according to the mean absolute SHAP value very similarly. This contrasts with the neural network model that only has positive mean absolute SHAP values for three features: area, perimeter, and texture. XGBoost and random forest have better and similar classification metrics in comparison to the neural network model (See Table S2 for accuracy metrics for all classification models). The beeswarm plots of XGBoost and random forest indicate a positive relationship for concave points, area, texture, and concavity. The feature perimeter has a negative relationship in the neural network model compared to random forest.

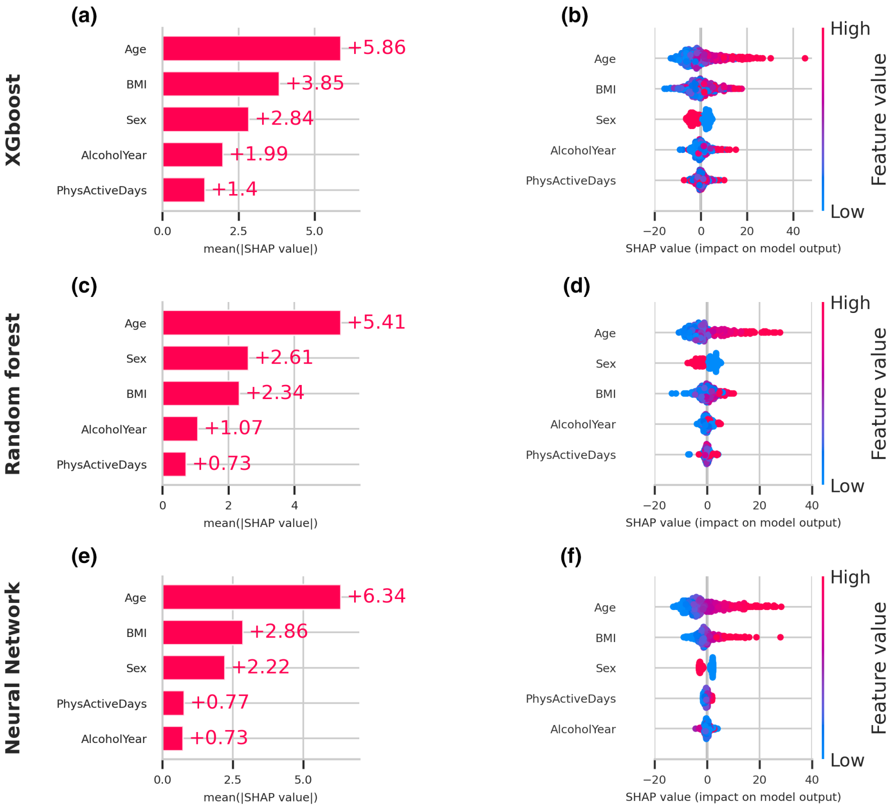


FIGURE 6 Visualizations plots for SHAP values of different ML regression models; (a,c,e) Bar plots; (b,d,f) Beeswarm plots.

Common statistical frameworks

Linear regression

We first simulate 2000 observations from a linear model (3) by considering two predictors x_1 and x_2 that follow an uniform distribution, that is, $x_1, x_2 \sim U_{[0,10]}$:

$$y = b_0 + b_1 x_1 + b_2 x_2 \tag{3}$$

where $b_0 = -10$, $b_1 = 6$ and $b_2 = -3$.

Now we assume the data generation process to be unknown and fit the generated data using a linear regression model.

For a linear regression model with no interaction terms, such as the one presented here, it can be shown that SHAP values for feature x_j correspond to

$$\phi_j = b_j \left(x_j - \mathbb{E}(x_j) \right) \tag{4}$$

where b_j is its corresponding linear coefficent and $\mathbb{E}(x_j)$ corresponds to the expected sample mean of the corresponding feature. The derivation of this formula can be found in Chapter 7 of the book by Christopher Molnar.[38]

When considering the in-silico dataset we obtain $\mathbb{E}(x_1)=5.05$ and $\mathbb{E}(x_2)=4.99$ for the test dataset which are close to the theoretical mean of 5 of the underlying distribution. Based on the conditions used to simulate the in-silico data, the SHAP values for a pair of model features (x_1, x_2) are:

$$\phi_1 = 6x_1 - 30.3 = 6(x_1 - 5.05) \text{ and } \phi_2 = -3x_2 + 14.97 = -3(x_2 - 4.99) \tag{5}$$

By considering kernelSHAP for estimating the SHAP values of this model, we can observe that the SHAP values are described by these formulas. In Figure S2, we show the different visualization plots that were considered for the SHAP values of this particular model. The bar plot in Figure S2A shows that the feature x_1 has a higher mean absolute SHAP value than feature x_2, as expected. Therefore, we can see that measuring the feature importance, based on the mean absolute SHAP values, matches to the intepretation that we can perform on this linear regression model.

When analyzing the beeswarm plot in Figure S2B, we can observe the general trends of the SHAP values for the two features of the linear regression model. The distribution of the SHAP values corresponding to the feature x_1 has a wider spread compared to the distribution of the SHAP values of feature x_2. This is expected since the absolute value of b_1 is twice the absolute value of b_2. Furthermore, in the beeswarm plot, we can observe there is a monotonous increasing relationship between feature x_1 and its SHAP values (blue to red) and a decreasing negative relationship between feature x_2 and its SHAP values (red to blue).

When inspecting the scatter plots for each feature (Figure S2C,D), we can see that the SHAP values for features x_1 and x_2 follow the formulas given in Formula 5. The waterfall plots of Figure S2E,F for two particular instances satisfy the formula given in Formula 4. Figure S2E shows that the feature values for this sample are $x_1 = 7.05$ and $x_2 = 0.614$. The SHAP value corresponding to feature x_1 is 11.07 which means that this feature increases the prediction $f(x_i)$ in comparison to the average prediction. Feature x_2 increases the prediction with a SHAP value of 12.55. Finally, the predicted value for this specific sample is $f(x_i) = 6.839 + 11.07 + 12.55 = 30.459$.

Logistic regression

We first simulate data from a logistic regression model by considering two predictors x_1 and x_2 that follow a uniform distribution, that is, $x_1, x_2 \sim U_{[0,10]}$, we then define a linear model:

$$\text{log\_odds} = b_0 + b_1 x_1 + b_2 x_2 \tag{6}$$

where $b_0 = 10$, $b_1 = 2$ and $b_2 = -6$. We then consider the transformation to the probability space

$$p = \frac{1}{1 + \exp(-\text{log\_odds})} \tag{7}$$

Using these probabilities we perform 2000 Bernoulli samples, that is, $\{(x_{1,i}, x_{2,i}, y_i)\}_{i=1}^{2000}$ where y_i is the outcome of Bernoulli trials considering the probabily p_i given by Formula 7.

We fit the resultant data using a logistic regression model and the estimated parameters match those of the model used to generate the data. Figure S3 illustrates the different visualization plots of SHAP values from a logistic regression, explaining the predicted probabilities and the predicted log-odds, side by side.

When comparing the bar plots we can see that feature x_2 has a higher mean absolute SHAP value in comparison to feature x_1 which signifies that feature x_2 is a more influential feature than feature x_1 (Figure S3A,B). This intepretation can also obtained in the classical way when inspecting the coefficient of features x_1 and x_2.

When comparing the beeswarm plots of the SHAP values of the logistic regression model, we can observe a non-uniform distribution for the SHAP values when explaining the predicted probabilities. This is in contrast to the uniform distribution of the SHAP values for both feature x_1 and x_2 when explaining the predicted log-odds. These observations can be seen in Figure S3C,D.

If we inspect the scatter plots for feature x_2, we notice that when explaining the predicted probabilities the relationship between the features and the SHAP values is non-linear (Figure S3E). When explaining the predicted log-odds, the relationship is linear (Figure S3F) as expected.

Generally, we also observe that the different visualization plots for the SHAP values of the predicted log-odds are similar to the visualization plots for the SHAP values of the linear regression. This is expected since in the log-odds scale the predictors follow a linear regression model.

FINAL REMARKS

In this tutorial, we provided a brief overview of the theoretical background of SHAP analysis, including Shapley values and their connection to SHAP analysis. We discussed the implementation and practical considerations of SHAP analysis, including visualization plots, special considerations for classification and time-series models, and available software. We showed the application of SHAP analysis to standard interpretable models and popular ML frameworks, both for regression and classification problems, as well as time-dependent models. We conclude by summarizing the limitations and potential extensions of this widely used technique.

While SHAP analysis is a powerful feature-based interpretability technique with numerous benefits, it is important to acknowledge its limitations and explore extensions for a comprehensive interpretability analysis of a ML model. Although the SHAP method does not assume feature independence, some of the approximation methods used to calculate the SHAP values (e.g., kernelSHAP) do assume it. In almost any real application this will not be the case, therefore, the reader needs to be careful when including features that are strongly correlated. A pre-processing step of analyzing the data and being aware of this situation can help to avoid situations where the model gives inaccurate predictions or give predictions that are at odds with domain knowledge of the topic. For both regression and classification problems, we performed this pre-processing step. For the regression models, we did not include weight or height as predictors since we included BMI as a predictor. For the classification models in the original dataset, the dataset for each breast mass' characteristics included the mean, standard deviation, highest and lowest values, and we only considered the mean values as predictors. An extension of kernelSHAP has recently been introduced[39] that takes into account the correlation between features. A Python package based on this extension was recently released[40] as well as an R package.[41] Feature engineering and selection can also help address the issue beforehand, enabling the use of the standard methodology. Other SHAP approximation methods like TreeSHAP explicitly consider feature dependence within the model structure.

Another limitation worth mentioning is that SHAP analysis does not quantify the importance of predictors in the real-world problem, but rather their importance to the model's predictions. SHAP values do not show how the features contribute to the observations, but rather how the features contribute to the models' predictions for the observations. Especially in the presence of model bias or in case of overfitting, this difference is important and should always be considered when interpreting SHAP values.

In this tutorial, we focused on SHAP analysis for supervised ML models that considered tabular data. However, SHAP analysis can also be applied to other ML models[42,43] relevant in drug development and one can also consider other type of data (e.g., electrocardiogram data,[44] MRI data[45]). A deeper dive into these other topics is necessary to understand the special considerations that need to be taken into account in each case. Also, this tutorial mostly focused on SHAP analysis; however, there exist other feature-based explainability methods with advantages and disadvantages in comparison to SHAP analysis such as LIME,[46] integrated gradients,[47] and permutation feature importance.[48] Considering these alternatives may be worthwhile to address some of the SHAP limitations. Also, as the authors in Imrie et al.[19] mentioned, ML interpretability has mostly focused on feature-based models; however, to fully understand a black-box model and provide explanations to the different stakeholders in clinical pharmacology, it is also important to embrace and combine other types of model-agnostic interpretation methods. In this tutorial, we also showed an example of how to perform SHAP analysis on a time-dependent ML model. However, there are several limitations to this approach. First, aggregating SHAP values over multiple time points can result in the loss of temporal information. Additionally, understanding the time-dependent impact of a feature can be difficult as its importance may vary over time. For the specific example shown, the SHAP values might not accurately capture the internal mechanisms of LSTM models, which are used to handle temporal dependencies. When considering feature engineering to account for time dependence, the chosen approach may not effectively capture time dynamics. Nevertheless, recent advancements have introduced new methodologies that incorporate the time-dependent structure[49,50] into the SHAP calculation. In particular, there are new packages available for SHAP calculation for deriving time-dependent explanation of ML survival models.[42]

The use of ML models in drug development is expected to continue growing and to become an integral part of this industry. In light of this, it is important to adopt ML interpretability methods to enhance the transparency and trustworthiness of black-box model predictions. This tutorial aims to offer a small yet meaningful step toward addressing this objective.

ACKNOWLEDGMENTS

Medical writing support was provided by Wesley Wayman, PhD, an employee of AbbVie.

FUNDING INFORMATION

This study was funded by AbbVie, Inc. AbbVie contributed to the study design, research, and interpretation of data, and the writing, reviewing, and approving of the publication.

CONFLICT OF INTEREST STATEMENT

All authors are employees of AbbVie and may hold AbbVie stock.

ORCID

Ana Victoria Ponce-Bobadilla https://orcid.org/0000-0002-0959-4058
Vanessa Schmitt https://orcid.org/0000-0003-4268-6306
Sven Mensing https://orcid.org/0000-0002-9434-647X
Sven Stodtmann https://orcid.org/0000-0002-7986-4447

REFERENCES

1. Liu Q, Zhu H, Liu C, et al. Application of machine learning in drug development and regulation: current status and future potential. *Clin Pharmacol Ther*. 2020;107:726-729.
2. Terranova N, Renard D, Shahin MH, et al. Artificial intelligence for quantitative modeling in drug discovery and development: an innovation and quality consortium perspective on use cases and best practices. *Clin Pharmacol Ther*. 2024;115:658-672.
3. Marques L, Costa B, Pereira M, et al. Advancing precision medicine: a review of innovative in silico approaches for drug development, clinical pharmacology and personalized healthcare. *Pharmaceutics*. 2024;16:332. https://doi.org/10.3390/pharmaceutics16030332
4. Bhhatarai B, Walters WP, Hop C, Lanza G, Ekins S. Opportunities and challenges using artificial intelligence in ADME/Tox. *Nat Mater*. 2019;18:418-422.
5. Zhang W, Roy Burman SS, Chen J, et al. Machine learning modeling of protein-intrinsic features predicts tractability of targeted protein degradation. *Genomics Proteomics Bioinformatics*. 2022;20:882-898.
6. Grebner C, Matter H, Kofink D, Wenzel J, Schmidt F, Hessler G. Application of deep neural network models in drug discovery programs. *ChemMedChem*. 2021;16:3772-3786.

7. Keutzer L, You H, Farnoud A, et al. Machine learning and Pharmacometrics for prediction of pharmacokinetic data: differences, similarities and challenges illustrated with rifampicin. *Pharmaceutics*. 2022;14:1530.
8. Lu J, Bender B, Jin JY, Guan Y. Deep learning prediction of patient response time course from early data via neural-pharmacokinetic/pharmacodynamic modelling. *Nature Machine Intelligence*. 2021;3:696-704.
9. Harun R, Lu J, Kassir N, Zhang W. Machine learning-based quantification of patient factors impacting remission in patients with ulcerative colitis: insights from Etrolizumab phase III clinical trials. *Clin Pharmacol Ther*. 2024;115:815-824.
10. Basu S, Munafo A, Ben-Amor AF, Roy S, Girard P, Terranova N. Predicting disease activity in patients with multiple sclerosis: an explainable machine-learning approach in the Mavenclad trials. *CPT Pharmacometrics Syst Pharmacol*. 2022;11:843-853.
11. Fisher CK, Smith AM, Walsh JR. Machine learning for comprehensive forecasting of Alzheimer's disease progression. *Sci Rep*. 2019;9:13622.
12. Young AL, Bragman FJS, Rangelov B, et al. Disease progression modeling in chronic obstructive pulmonary disease. *Am J Respir Crit Care Med*. 2020;201:294-302.
13. Pinto MF, Oliveira H, Batista S, et al. Prediction of disease progression and outcomes in multiple sclerosis with machine learning. *Sci Rep*. 2020;10:21038.
14. Qian Z, Zame W, Fleuren L, Elbers P, van der Schaar M. Integrating expert ODEs into neural ODEs: pharmacology and disease progression. *Adv Neural Inf Proces Syst*. 2021;34:11364-11383.
15. Fang C, Xu D, Su J, Dry JR, Linghu B. DeePaN: deep patient graph convolutional network integrating clinico-genomic evidence to stratify lung cancers for immunotherapy. *NPJ Digit Med*. 2021;4:14.
16. Cui ZL, Kadziola Z, Lipkovich I, Faries DE, Sheffield KM, Carter GC. Predicting optimal treatment regimens for patients with HR+/HER− breast cancer using machine learning based on electronic health records. *J Comp Eff Res*. 2021;10:777-795.
17. Xie Y, Meng WY, Li RZ, et al. Early lung cancer diagnostic biomarker discovery by machine learning methods. *Transl Oncol*. 2021;14:100907.
18. Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: a survey on methods and metrics. *Electronics*. 2019;8:832.
19. Imrie F, Davis R, van der Schaar M. Multiple stakeholders drive diverse interpretability requirements for machine learning in healthcare. *Nature Machine Intelligence*. 2023;5:824-829.
20. Janssen A, Hoogendoorn M, Cnossen MH, Mathôt RAA. Application of SHAP values for inferring the optimal functional form of covariates in pharmacokinetic modeling. *CPT Pharmacometrics Syst Pharmacol*. 2022;11:1100-1110.
21. Denney W. What is normal? A meta-analysis of phase 1 placebo data. *Population Approach Group in Europe*. 2014;23 Abstr 3190.
22. Zhu X, Zhang M, Wen Y, Shang D. Machine learning advances the integration of covariates in population pharmacokinetic models: Valproic acid as an example. *Front Pharmacol*. 2022;13:994665.
23. Harun R, Yang E, Kassir N, Zhang W, Lu J. Machine learning for exposure-response analysis: methodological considerations and confirmation of their importance via computational experimentations. *Pharmaceutics*. 2023;15. https://www.mdpi.com/about/announcements/784
24. Shapley LS. A value for n-person games. *Contribution to the Theory of Games*. 1953;2:307-318.
25. Strumbelj E, Kononenko I. An efficient explanation of individual classifications using game theory. *J Machine Learning Res*. 2010;11:1-18.
26. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Adv Neural Inf Proces Syst*. 2017;30:4766-4777.
27. Molnar C. *Interpreting Machine Learning Models with SHAP*. Lulu Com; 2023.
28. Lundberg SM, Erion G, Chen H, et al. From local explanations to global understanding with explainable AI for trees. *Nat Mach Intell*. 2020;2:56-67.
29. Centers for Disease Control and Prevention (CDC). National Center for Health Statistics (NCHS). National Health and Nutrition Examination Survey. Accessed July 25, 2024. https://www.cdc.gov/nchs/nhanes/
30. Street WN, Wolberg WH, Mangasarian OL. Nuclear feature extraction for breast tumor diagnosis. *Biomedical Image Processing and Biomedical Visualization*. 1993;1905:861-870.
31. Gianfagna L, Di Cecco A. *Explainable AI with Python*. Springer; 2021.
32. Masís S. *Interpretable Machine Learning with Python: Build Explainable, Fair, and Robust High-Performance Models with Hands-on, Real-World Examples*. Packt Publishing Ltd; 2023.
33. Wolberg WH, Street WN, Mangasarian OL. Importance of nuclear morphology in breast cancer prognosis. *Clin Cancer Res*. 1999;5:3542-3548.
34. Ismail AA, Gunady M, Corrada Bravo H, Feizi S. Benchmarking deep learning interpretability in time series predictions. *Adv Neural Inf Proces Syst*. 2020;33:6441-6452.
35. Turbé H, Bjelogrlic M, Lovis C, Mengaldo G. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*. 2023;5:250-260.
36. SHAPforxgboost. Accessed July 25, 2024. https://cran.r-project.org/web/packages/SHAPforxgboost/readme/README.html
37. Shapper. Accessed July 25, 2024. https://modeloriented.github.io/shapper/
38. Christoph M. *Interpretable machine learning: A guide for making black box models explainable* (Leanpub). 2020.
39. Aas K, Jullum M, Løland A. Explaining individual predictions when features are dependent: more accurate approximations to Shapley values. *Artif Intell*. 2021;298:103502.
40. Corr_shap. Accessed July 29, 2024. https://github.com/Fraunhofer-SCAI/corr_shap/tree/main
41. Shapr. Accessed July 29, 2024. https://github.com/NorskRegnesentral/shapr
42. Krzyziński M, Spytek M, Baniecki H, Biecek P. SurvSHAP (t): time-dependent explanations of machine learning survival models. *Knowl-Based Syst*. 2023;262:110234.
43. Duval A, Malliaros FD. Graphsvx: Shapley value explanations for graph neural networks. Machine Learning and Knowledge Discovery in Databases Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21. 2021: 302–318.
44. Ayano YM, Schwenker F, Dufera BD, Debelee TG. Interpretable machine learning techniques in ECG-based heart disease classification: a systematic review. *Diagnostics*. 2022;13:111.

45. Salih A, Galazzo IB, Cruciani F, Brusini L, Radeva P. Investigating explainable artificial intelligence for MRI-based classification of dementia: a new stability criterion for explainable methods. *2022 IEEE International Conference on Image Processing (ICIP)*. 2022;2022:4003-4007.
46. Ribeiro MT, Singh S, Guestrin C. "Why should i trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016: 1135–1144.
47. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. International Conference on Machine Learning 2017: 3319–3328.
48. Fisher A, Rudin C, Dominici F. All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. *J Mach Learn Res*. 2019;20:1-81.
49. Bento J, Saleiro P, Cruz AF, Figueiredo MA, Bizarro P. Timeshap: explaining recurrent models through sequence perturbations. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining 2021: 2565–2573.
50. Nayebi A, Tipirneni S, Reddy CK, Foreman B, Subbian V. WindowSHAP: an efficient framework for explaining time-series classifiers based on Shapley values. *J Biomed Inform*. 2023;144:104438.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Ponce-Bobadilla AV, Schmitt V, Maier CS, Mensing S, Stodtmann S. Practical guide to SHAP analysis: Explaining supervised machine learning model predictions in drug development. *Clin Transl Sci*. 2024;17:e70056. doi:10.1111/cts.70056