Explainability of Machine Learning Models using Co-operative Game Theory

Submitted by

Brigida Zhunio Cardenas

David Aguilera

Dibaloke Chanda

MSSC 6520 : Statistical Machine Learning

May 2024

Introduction

The Shapley value, introduced in 1951 by Lloyd Shapley, is a concept from cooperative game theory that ensures a fair distribution of payoff among players in a coalition based on their contributions. In machine learning, Shapley values can help to improve the explainability of complex models. With the development of machine learning, specifically deep learning, models have become opaque black boxes, more and more complex to explain to individuals not related to the field. A clear example of this is a credit card company that uses a sophisticated model to accept or decline credit card applications. A customer with a denied application calls customer service looking for an explanation for this denial. If the company does not count a tool that helps to quantify (explain) the contribution that each factor had on the evaluation process, the customer could believe his/her application has been treated unfairly, just because the company is unable to explain the outcome of the model in a more customer-friendly approach. Here is when Shapley values can become the key for model explainability. Our class group believes that the interpretation of Shapley values holds great potential in determining the influence of each feature within machine learning models. It's essential for both the tech industry and academia to recognize the importance of quantifying or measuring feature's contributions in prediction models. This verification process is fundamental for ensuring the accuracy and reliability of the model [1]. That's why, almost 70 years later, the Shapley value remains a key component in the prediction models, providing the following benefits:

- Provides insights into the influence of individual factors within complex models.
- Helps in understanding the contributions of each feature or parameter to the model's predictions.
- Facilitates model evaluation and verification by quantifying the importance of each attribution.
- Promotes transparency and trust in predictive models by revealing the underlying factors driving their outputs from the models.

Intuition and Idea of the Method

To understand the main idea behind Shapley Value, we first need to grab the concept for game theory, specifically, cooperative game theory. In cooperative game, "players" form coalitions to accomplish a common goal. Imagine the following scenario: For their class final project, a group of graduate students (players) work together (form a coalition) to finish the project and achieve the best score (payout) as result.

A problem in cooperative game theory is the fair distribution of the payout between the players. For our example, the final grade each student should be assigned (Fig. 1).

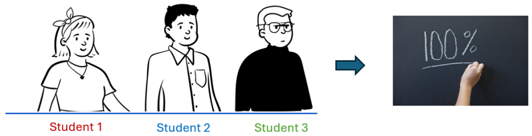

Figure 1: Three students participating in a project where the total grade of the project is graded out of 100

Their professor wants to assign their scores in the fairest way possible, but he faces some challenges. What if one student contributed to 60% of the work? What if one of the students didn't write any part of the report but did the search of all the sources used to do the project? The students may have variable levels of efficiency based on who they collaborate with and when they join the collaboration (different coalitions).

In summary, it's not straightforward for the professor to decide what grade each student should receive. He needs to estimate how essential is each student contribution to the total project and based on their contribution, assign them a fair grade. It's important to understand that in cooperative game theory, players can form different coalitions, coalitions that will have different outcomes. Following our example, in Fig. 2 we show the possible group scores (payout) of all the possible coalitions formed by the students.

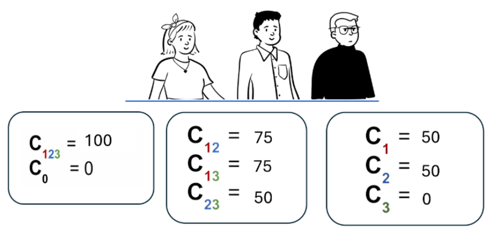

Figure 2: Visualization of different coalitions. As an example consider $C_{12} = 75$ if students 1 and 2 work together they could expect a score of 75 for the group.

To compute the contribution of each player (marginal contribution) to the outcome, Shapley Values evaluate the outcomes of all the possible scenarios (coalitions) where the player contributed. In Fig. 3 we see all the scenarios and their outcomes when student 1 contributed, the marginal contribution of student 1 will be calculated based on the different outcomes and how many times the different coalitions show up during the game.

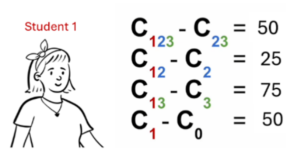

Figure 3: Marginal contribution calculation for student 1 for all the scenarios. The expected marginal contribution is computed by taking the average of all the marginal contributions

Applying this concept to the explainability of a machine learning model, the players will be replaced by the data feature values (predictors), and these features will work as the coalition members. The payout will be the prediction of the trained model. The Shapley values aim is computing the contribution of each feature (predictor) to the prediction and based on it, determine the part of the outcome that is attributed to the feature.

The advantages and disadvantages [4] are mentioned below:

Advantages

1. **Fair Distribution of Feature Importance:** The Shapley value fairly distributes the difference between the prediction and the average prediction among the feature values of the instance, ensuring a balanced explanation.

2. **Versatile Contrastive Explanations:** Shapley values allow for more detailed comparisons, such as contrasting with subsets or specific data points,

3. **Solid Theoretical Foundation:** The Shapley value is based on axioms like efficiency, symmetry, dummy, and additivity, offering a robust theoretical basis. This contrasts with other methods like LIME [5], which assume linear behavior in local contexts without a solid theory.

4. **Interesting Interpretative Perspective:** The Shapley value can be seen as explaining a prediction through a game played by feature values, providing a unique and engaging way to interpret model outcomes.

Disadvantages

1. **High Computational Cost:** Exact computation of Shapley values is computationally expensive due to the need to consider an exponential number of feature combinations. Approximations are often required, which can affect accuracy.

2. **Potential Misinterpretation:** Shapley values can be misunderstood. It's important to note that the Shapley value represents the contribution of a feature value to the difference between the actual prediction and the mean prediction, not the impact of removing the feature from the model training.

3. **Lack of Sparse Explanations:** Shapley values involve all features in the explanation, which can be overwhelming for users seeking simpler explanations.

4. **Limited Predictive Insight:** Shapley values provide a single value per feature without a model to predict how changes in input might affect output. This restricts the ability to explore "what-if" scenarios.

5. **Data Access Requirement:** To calculate Shapley values for a new instance, access to the original data is needed, as part of the process involves sampling from the training data. This can raise data privacy concerns.

6. **Risk of Unrealistic Instances with Correlated Features:** When features are correlated, sampling from the marginal distribution can produce unrealistic combinations, leading to misleading results. However conditional sampling can address this, it can violate Shapley value methodology.

Method

Mathematical Expression of the Model

Consider the set of features defined as $\mathcal{P} = \{x_1, x_2, x_3, \ldots, x_i, \ldots, x_p\}$ where the number of features are p. We denote a general element from the feature set as x_i where $x_i \in \mathcal{P}$.

Consider $\mathcal{S}$ a coalition set formed after a particular feature x_i is excluded from $\mathcal{P}$ i.e. $\mathcal{S} \subseteq \mathcal{P} \backslash \{x_i\}$. There are different ways to form $\mathcal{S}$ and to refer to any of the possible coalitions we use $\mathcal{S}$ as the general symbol. If the coalition is indexed like $\mathcal{S}_0, \mathcal{S}_{12}, \ldots$ it refers to a specific coalition where the index indicates which specific features form the coalition. As an example $\mathcal{S}_{12}$ means a coalition of feature x_1 and x_2.

The Shapely value for feature x_i is defined as $\phi(x_i)$ which is given by (1).

$$\phi(x_i) = \sum_{\mathcal{S} \subseteq \mathcal{P} \backslash \{x_i\}} \frac{|\mathcal{S}|!(p - |\mathcal{S}| - 1)!}{p!} m(\mathcal{S}, x_i) \tag{1}$$

For (1) the following are the symbol and expression meanings:

(i) $p!$ The number of ways to form a coalition of p features

(ii) $|\mathcal{S}|$ The number of features in a coalition

(iii) $|\mathcal{S}|!$ The number of ways coalition $\mathcal{S}$ can form.

(iv) $(p - |\mathcal{S}| - 1)!$ The number ways other features can be included after feature x_i is included.

(v) $m(\mathcal{S}, x_i)$ The expected marginal contribution of the feature x_i.

The fraction $\frac{|\mathcal{S}|!(p-|\mathcal{S}|-1)!}{p!}$ is a weighting factor. The expression for $m(\mathcal{S}, x_i)$ which is the marginal contribution for feature x_i in a particular coalition $\mathcal{S}$ is given by (2).

$$m(\mathcal{S}, x_i) = f(\mathcal{S} \cup \{x_i\}) - f(\mathcal{S}) \tag{2}$$

In (2) the $f(\cdot)$ is a machine learning model that takes in a set of features and outputs a prediction. Intuitively, $f(\mathcal{S} \cup \{x_i\}) - f(\mathcal{S})$ is measuring the contribution of feature x_i after it is included in the coalition.

For the ease of understanding (1) is re-written with annotation included.

$$\overbrace{\phi(x_i)}^{\text{Shapley value for feature } x_i} = \underbrace{\sum_{\mathcal{S}\subseteq\mathcal{P}\setminus\{x_i\}}}_{\text{Iterating over possible coalitions } \mathcal{S}_0, \mathcal{S}_{12}, \ldots} \frac{\overbrace{|\mathcal{S}|!}^{\text{Number of features in coalition } \mathcal{S}} \; \overbrace{(p-|\mathcal{S}|-1)!}^{\text{Number of ways to include features after including } x_i \text{ in } \mathcal{S}}}{\underbrace{p!}_{\text{Number of ways to permutate the feature set } \mathcal{P}}} \overbrace{\underbrace{m(\mathcal{S}, x_i)}_{\text{Marginal contribution of } x_i \text{ for coalition } \mathcal{S}}}^{f(\mathcal{S}\cup\{x_i\}) - f(\mathcal{S})}$$

Figure 4: Annotated equation for the computation of Shapley value of feature x_i

From Fig. 4, it is apparent that the Shapley value is just the expected marginal contribution of feature x_i. We are essentially computing a weighted average of the marginal contributions of $m(\mathcal{S}, x_i)$. The averaging is done over the possible coalitions and as this is a weighted average, the weight factors are used.

It is difficult to get the intuition behind the formula if we consider the general case. To get a better intuition, consider the special case where $\mathcal{P} = \{x_1, x_2, x_3\}$ shown in Fig. 5.

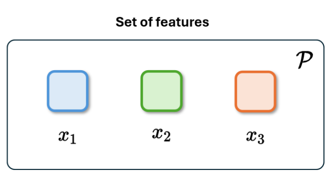

Figure 5: Visualization of the set of all features ($\mathcal{P}$) for the special case where $p = 3$

The number of ways this feature set can be permutated is $p! = 3! = 6$. A visualization is provided in Fig. 6. This is the denominator in the weight factor.

In Fig. 7, we provide visualization to the possible coalition set excluding feature x_i. Let's only consider the first case when we exclude the first feature x_1 and try to compute the Shapley value for feature value x_1. For this case, the components of the weight factor can be calculated as shown in Fig. 8.

We also summarize the weight factors in Table 1.

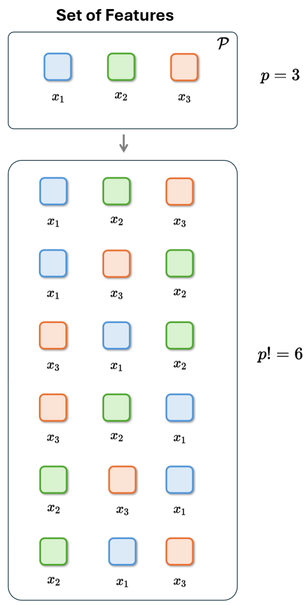

Figure 6: Possible permutations of features for the special case $\mathcal{P} = \{x_1, x_2, x_3\}$

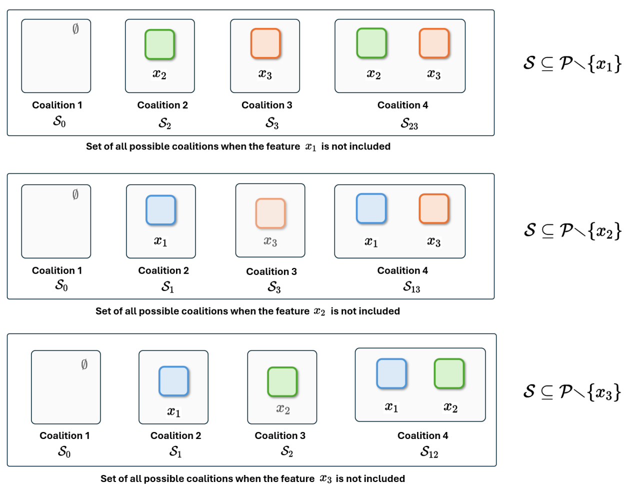

Figure 7: Visualization of the set $\mathcal{S}$ when a particular feature x_i is excluded. One of the possible coalitions is the empty set $\emptyset$

.

Coalitions $\mathcal{S}$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{S}\rvert!$	$(p - \lvert\mathcal{S}\rvert - 1)!$	Weight Factors $\frac{\lvert\mathcal{S}\rvert!(p-\lvert\mathcal{S}\rvert-1)!}{p!}$
$\mathcal{S}_1 = \emptyset$	0	$0! = 1$	$(3 - 0 - 1)! = 2! = 2$	$\frac{1 \cdot 2}{3!} = \frac{1}{3}$
$\mathcal{S}_2 = \{x_2\}$	1	$1! = 1$	$(3 - 1 - 1)! = 1! = 1$	$\frac{1 \cdot 1}{3!} = \frac{1}{6}$
$\mathcal{S}_3 = \{x_3\}$	1	$1! = 1$	$(3 - 1 - 1)! = 1! = 1$	$\frac{1 \cdot 1}{3!} = \frac{1}{6}$
$\mathcal{S}_{23} = \{x_2, x_3\}$	2	$2! = 2$	$(3 - 1 - 2)! = 0! = 1$	$\frac{2 \cdot 1}{3!} = \frac{1}{3}$

Table 1: Weight factors for the Shapley value with for $p = 3$ features i.e. $\mathcal{P} = \{x_1, x_2, x_3\}$ and the Shapley value is computed for the feature x_1.

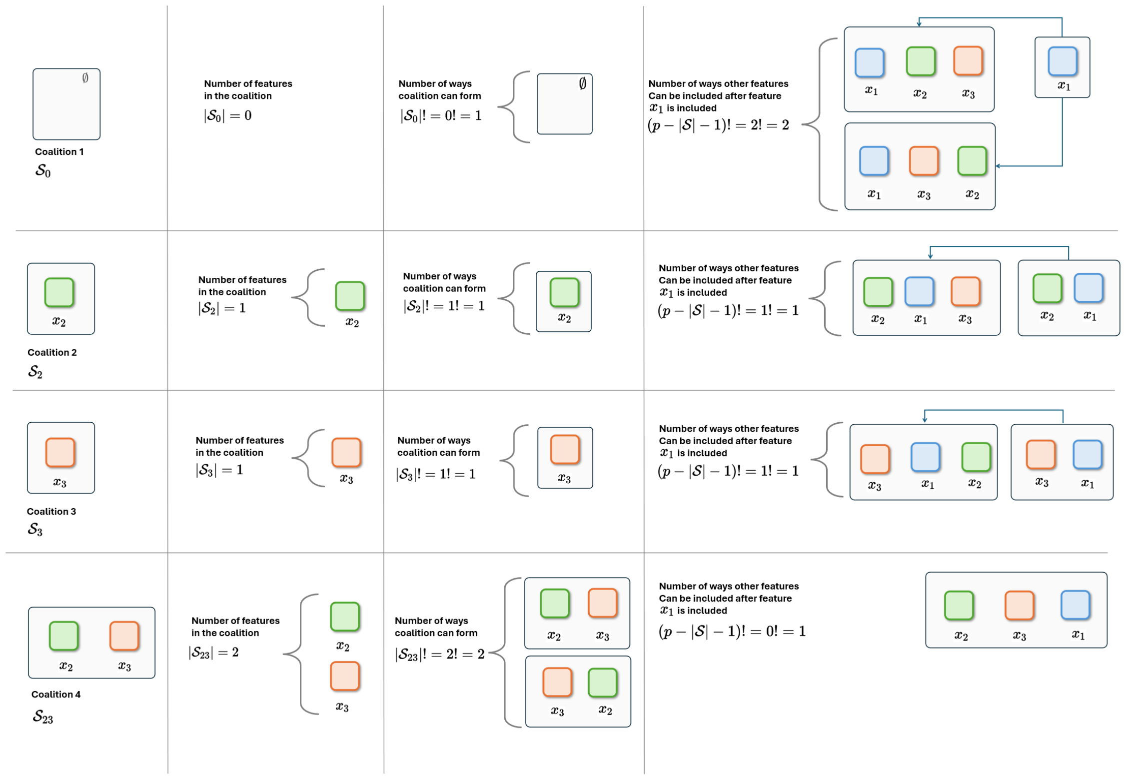

Figure 8: Components of weight factors for computing $\phi(x_1)$

Finally, the Shapley value can be computed as shown in (3).

$$\phi(x_1) = \left[\frac{1}{3}m(\mathcal{S}_0, x_1) + \frac{1}{6}m(\mathcal{S}_2, x_1) + \frac{1}{6}m(\mathcal{S}_3, x_1) + \frac{1}{3}m(\mathcal{S}_{23}, x_1)\right] \tag{3}$$

In a similar manner, $\phi(x_2)$ and $\phi(x_3)$ can be computed.

Explaination of the Method

The implementation of the model is slightly different from the mathematical formulation given in the previous section. Most machine learning models are not compatible with the varying number of features. Hence, it is not possible to evaluate the model performance with different-size coalitions. In practice, the feature that needs to be removed is replaced by a feature from a random sample. Fig. 9 shows a visualization of how the $x_1^{(1)}$ is replaced by another random sample. If the features are fairly uncorrelated across the samples, this will work as the replaced feature will be essentially a random feature with very little contribution to the prediction. Fig. 9 shows the Shapley computation for a single coalition, but it needs to be done for all possible coalitions. However, if there are p features, there will be 2^p possible coalitions. For large p, it becomes unfeasible to compute the marginal value for all possible coalitions. Therefore, the Monte-Carlo

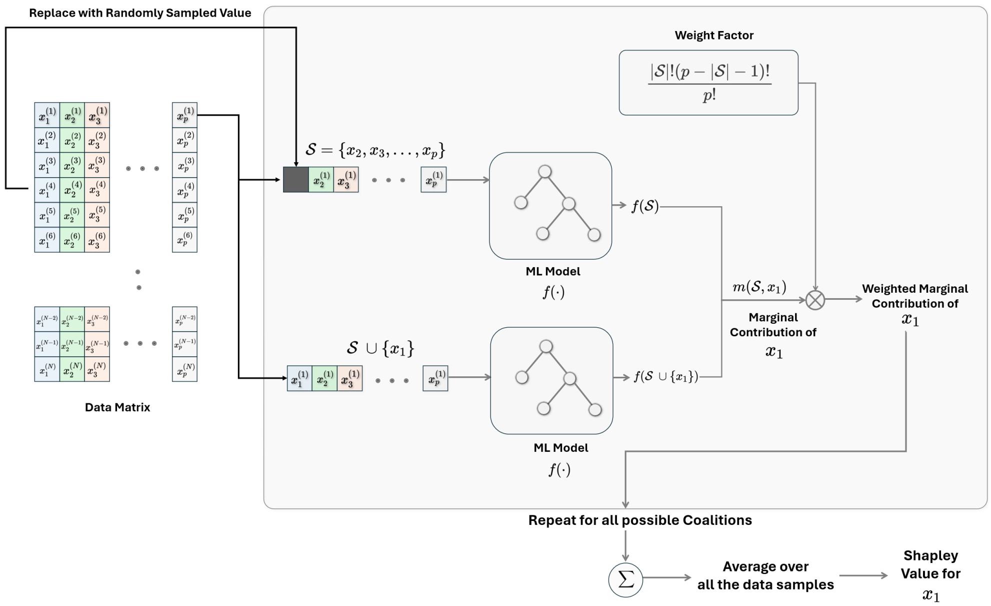

Figure 9: Shapley value computation visualization for a feature x_1 (Tabular Data). The machine learning model $f(\cdot)$ can be any model, for the sake of visualization we used decision trees.

approach is employed to approximate the value.

The Monte Carlo simulation [6] performs random sampling repeatedly, in each simulation, the model is run, generating multiple outcomes, finally, these outcomes are averaged to obtain an estimate of the real value. As we mentioned before, in the Shapley values calculation, as p increases, it becomes computationally unfeasible to obtain the marginal contribution for each feature for all possible coalitions. MC estimates Shapley values by sampling from all possible coalitions of the features, running the simulation for each sample coalition and averaging the multiple marginal values obtained to obtain the estimated contribution of the feature.

Properties of the Method

Following the Shapley principle of fairness, Shapley Values satisfy the following properties:

1. **Efficiency:** The sum of the Shapley values of each feature (payout for each player), equals the value of the total prediction that includes all the features. Mathematically, that can be expressed as follows:

$$\sum_{l=1}^{p} \phi(x_l^{(k)}) = f(\mathbf{x}^{(k)}) - E_{\mathbf{X}}(f(\mathbf{X})) \tag{4}$$

Here, $E_{\mathbf{X}}(f(\mathbf{X}))$ is the average prediction value overall all samples. The vector $\mathbf{x}^{(k)}$ is the feature vector for a single data point k and matrix $\mathbf{X}$ is the feature matrix that contains feature vectors for all the samples.

2. **Dummy:** If a feature does not change the predicted value, its Shapley value must be 0. Basically, because it's not being used by the model to perform the prediction. Consider a feature x_j that does not contribute anything to the prediction, which means adding it or removing it from any coalition should not change anything. Mathematically this can be written as (5).

$$f(\mathcal{S} \cup x_j) = f(\mathcal{S}) \tag{5}$$

for $\mathcal{S} \subseteq \mathcal{P}$, then (6) should hold true.

$$\phi(x_j) = 0 \tag{6}$$

3. **Symmetry:** If two features contribute equally to all the possible sub-coalitions, their Shapley Values should also be the same. Consider two features x_j and x_k and they contributed equally to all possible sub-coalitions. This means if (7) holds true:

$$f(\mathcal{S} \cup \{x_j\}) = f(\mathcal{S} \cup \{x_k\}) \tag{7}$$

for all coalitions $\mathcal{S} \subseteq \mathcal{P} \backslash \{x_j, x_k\}$, then (8) will also hold true by symmetry property.

$$\phi(x_j) = \phi(x_k) \tag{8}$$

4. **Linearity:** This property helps to distribute the payout of a model that is the result of the linear combination of two or more models. Computing Shapley values for a random forest model is a good example of this property. The model prediction is the average of multiple different decision trees. We can obtain the Shapley values for each tree and average them to compute the feature value for the random forest model. Consider a random forest consisting of n decision trees. The Shapley value for a feature x_i will be the linear combination of individual Shapley values for each decision tree. Mathematically that can be expressed as (9) where $\phi^{RF}(x_i)$ is the Shapley value of the random forest.

$$\phi^{RF}(x_i) = \phi^1(x_i) + \phi^2(x_i) + \ldots + \phi^n(x_i) \tag{9}$$

Simulation and Discussion

To keep it concise, we only focused on regression, however, Shapley values can easily be computed for classification problems. For the simulation, we made use of the SHAP [3] library in Python. This is a high-level library in Python that provides the functionality to perform generalized additive feature attributions. Additionally, we used R to perform Lasso regression.

Machine Learning Models

As for the machine learning models, we used XGBoost and Random Forest both of which are tree ensemble methods. XGBoost makes use of boosting and Random Forest makes use of bagging. The issue with such models is they are non-parametric and even though they achieve high accuracy it comes at the cost of loss of explainability. In contrast, fitting models like Lasso generate coefficients which provides intuition into the contribution of each features. Using Shapley makes it possible to generate feature attribution coefficients for tree-based ensemble models. To prove Shapley actually finds feature contribution we also fitted a Lasso model and reported the coefficients as a comparison.

Dataset

We generated the dataset using the following equation:

$$Y = 0.6X_1 + 0.6X_2 - 0.1X_3 - 0.05X_4 + X_5 + \epsilon \tag{10}$$

where, $\epsilon \sim \mathcal{N}(0, 1)$ is the noise component and X_1, X_2, X_3, X_4, X_5 are random variables. Individual data points $\mathbf{x}$ are sampled from a multivariate Gaussian distribution i.e. $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{10\times10})$. The covariance matrix $\mathbf{\Sigma}$ is parameterized by a correlation factor ρ. We used two different correlation values 0.05 and 0.07. The Y is the generated label for the data points. We generate a total of 1000 data points.

To help visualize the amount of information that Shapley Values can provide, we make use of some of the many plots that the SHAP library has available which include the beeswarm plot, barplot, waterfall plot and heatmap plot.

In Table 2 the comparison between Lasso and Shapley value is shown. The first column represents the true coefficient values according to equation (10). The next two columns show the Lasso coefficients. It is apparent these values are close to the true coefficient value irrespective of the correlation factor value. In addition, Lasso is able to pick up which features have zero contribution. The next columns show the mean absolute values for XGBoost and Random Forest. Even though these methods are ensemble approaches, the

Features	Actual Coefficients	Lasso Coefficients		Shapley Values (Mean Absolute Value)			
				XGBoost		Random Forest	
		$\rho = 0.05$ $\lambda = 0.02$	$\rho = 0.07$ $\lambda = 0.03$	$\rho = 0.05$	$\rho = 0.07$	$\rho = 0.05$	$\rho = 0.07$
Feature 1	0.6	0.54	0.55	0.51	0.51	0.48	0.49
Feature 2	0.6	0.46	0.44	0.46	0.47	0.43	0.45
Feature 3	-0.1	-0.05	-0.04	0.16	0.13	0.09	0.07
Feature 4	-0.05	-	-	0.07	0.06	0.03	0.03
Feature 5	1	0.88	0.83	0.81	0.78	0.84	0.77
Feature 6	-	-	-	0.07	0.08	0.03	0.03
Feature 7	-	-	-	0.08	0.09	0.03	0.02
Feature 8	-	-	-	0.09	0.08	0.03	0.03
Feature 9	-	-	-	0.09	0.09	0.03	0.03
Feature 10	-	-	-	0.09	0.08	0.03	0.03

Table 2: Comparison table between Lasso coefficients vs Shapley Value with correlation factor $\rho = 0.05$ and $\rho = 0.07$.

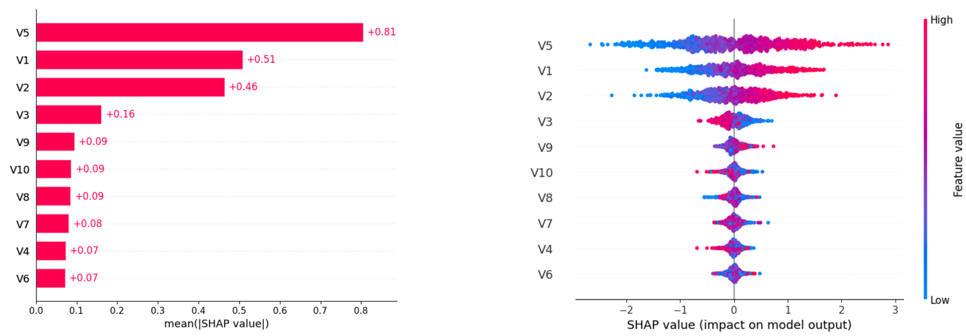

Figure 10: (**Left**) Global Shapley value (absolute) bar chart for individual features with XGBoost model and correlation factor $\rho = 0.05$. (**Right**) Beewarm plot of Shapley value all the data samples with XGBoost model and correlation factor $\rho = 0.05$.

Shapley values are similar to the original coefficient values.

In Fig. 10 (Left), the length of each bar represents the mean absolute Shapley value, indicating the importance of each feature. V5, with the longest bar and a Shapley value of +0.81, is the most important feature, while V6, with the shortest bar and a Shapley value of +0.07, is the least important. This chart visually represents the significance of each feature in the model. However, this plot ranks our features based on their absolute value, so the information on this plot is not enough to define if the impact of each feature in the outcome is positive or negative, which can be important for some models. For example, for a model designed to accept or decline credit card applications, it will be vital for the company to be able to explain to their customers which feature had a negative impact on their applications. In Fig. 11, The Waterfall plot helps us to explain the positive or negative influence of each feature in any outcome predicted by the model.

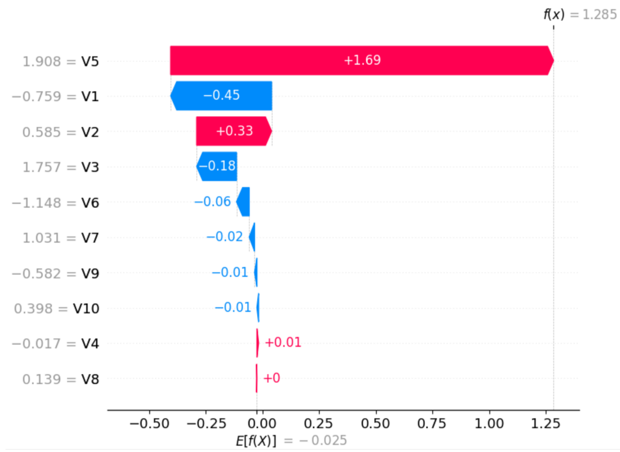

Figure 11: Waterfall chart for training sample 1 with XGBoost model and correlation factor $\rho = 0.05$

The sum of all the values in the waterfall plot will be equal to the outcome $f(x)$.

We can also extract valuable information from a beeswarm plot Fig. 10 (Right). The beeswarm plot has vertical axes labeled V1 through V10, each representing a different feature. On each axis, a series of individual points are spread horizontally. These points represent the SHAP values for each data sample, indicating the impact of that feature on the model's output. The most relevant features are V5, V1 and V2, where the dots, determined by the SHAP value, extend far in both directions—right (increases the model output) or left (decreases the model output)—from the center (0), indicating that these features have a significant impact on the model's output. Which is congruent with the coefficients from the formula for the simulated data (10). In a resume, the features are ranked along the y-axis according to their importance. The beeswarm plot offers an extra feature: the color of the points represents the feature values for each data point compared to the average for the entire population, with a color map ranging from blue (low) to red (high), with purple and pink indicating intermediate values. For continuous features the coloring is gradient, for categorical features the color would be red or blue.

Another diagram that provides a more holistic view of the generated Shapley value is shown in Fig. 12. In the x-axis, each tick represents a single instance from the data set. In the y-axis, the 10 features are represented. This diagram shows for each individual data point the Shapley value for all the features. The bar chart shown is actually the bar chart from Fig. 10 (Left). The $f(x)$ plot on top is the predicted value for individual data points. The dotted line is the average prediction value over all the data points. One important observation is for features 6 to 10 the Shapley value for individual instances is almost near zero which is consistent with the true coefficients.

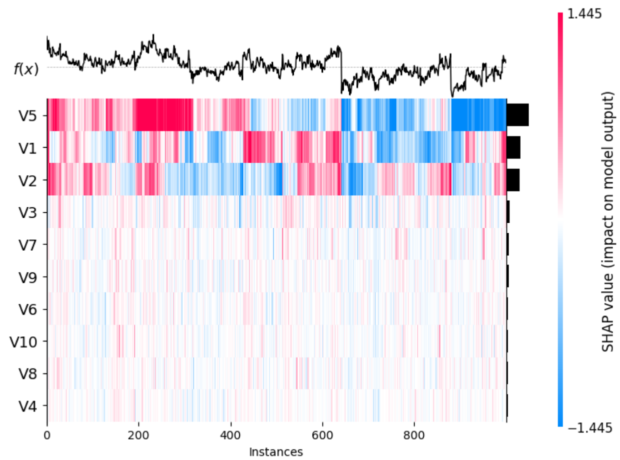

Figure 12: Shapley value visualization with a heatmap for all 10 features and all 1000 samples. The figure was generated with the XGBoost model and correlation factor $\rho = 0.05$.

Advantages and Disadvantages Based on Simulation

Without a doubt, the most powerful advantage of Shapley Values is its capacity to provide explainability to the outcomes of non-parametric models. Shapley Values can be considered a complement to any machine learning model. becoming a powerful tool for strong models that deliver accurate predictions, but due to their complexity, provide hard-to-explain outcomes, especially to people not related to the Data Analysis field. This can be a game changer, especially for fields where interpretability is a priority when selecting the most suitable model.

However, an important downside of Shapley Values is that the reliability depends on the accuracy of the machine learning model that was used to obtain the predictions and consequently the contributions of each feature. In resume, if the chosen model delivers a poor prediction, our Shapley values will clearly explain our outcome, but the values will be inaccurate. It is important to remember that Shapley values is not a prediction model, it's an interpretability tool that provides explainability to an already established machine learning model. Another drawback is due to the approximations used to compute Shapley values it does not strictly follow the properties mentioned. As an example, from Table 2 we can see, that it attributed some Shapley value for features that should have theoretically zero Shapley value. This means the property shown in (6) doesn't hold in practice. However, those attributed feature values are nearly zero.

Variants of Shapley Values

A variation of Shapley Value is LIME interpretability tool [5]. The main difference between Shapley and LIME is their explainability domain [2]. Shapley Values explains the global contribution of the predictors to the outcome, explaining the general behavior of the model across the whole data set. On the other hand, LIME, short for local interpretable model explanations, focuses as its name says on providing individual explainability to a specific outcome. To do that, LIME pays attention to how the model behaves in the vicinity of the instance being predicted. LIME uses a surrogate model to find the explanation for a single instance by considering the local feature space around that single instance. This surrogate model is usually a linear model like linear regression which is inherently interpretable. The mathematical formulation of LIME as an optimization problem is shown in (11).

$$\xi = \underset{g \in \mathcal{G}}{\arg\min}\ L\left(f, g, \pi_{x'}\right) + \Omega(g) \tag{11}$$

Here, f is the actual model that we are trying to generate the explanation for, g is the linear surrogate model from a set of possible models $\mathcal{G}$, $\Omega(g)$ is the regularizer term, $\pi_{x'}$ is the kernel weight term. Intuitively, by minimizing the loss term L, LIME produces a linear model g which can explain the contribution of the features for x'. There are also other variants of Shapley like Kernel Shap, linear Shap, Deep Shap, Tree Shap, etc.

References

[1] Soufiane Fadel. "Explainable machine learning, game theory, and shapley values: a technical review". In: *Statistics Canada: Ottawa* (2022).

[2] Thomas ten Heuvel. *Opening the Black Box of Machine Learning Models: SHAP vs LIME for Model Explanation.* https://medium.com/cmotions/opening-the-black-box-of-machine-learning-models-shap-vs-lime-for-model-explanation-d7bf545ce15f. Accessed: 2024-05-06. Mar. 2023.

[3] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[4] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable.* 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.* 2016, pp. 1135–1144.

[6] Sofiane Touati, Mohammed Said Radjef, and SAIS Lakhdar. "A Bayesian Monte Carlo method for computing the Shapley value: Application to weighted voting and bin packing games". In: *Computers & Operations Research* 125 (2021), p. 105094.