
Generalized SHAP: Generating multiple types of explanations in machine learning

Dillon Bowen*

Operations, Information, and Decisions Department
The Wharton School of Business, University of Pennsylvania
Philadelphia, PA 19104
dsbowen@wharton.upenn.edu

Lyle Ungar

Department of Computer and Information Science, University of Pennsylvania
Philadelphia, PA 19104
ungar@cis.upenn.edu

Abstract

Many important questions about a model cannot be answered just explaining how much each feature contributes its output. To answer a broader set of questions, we generalize a popular, mathematically well-grounded explanation technique, Shapley Additive Explanations (SHAP). Our new method - Generalized Shapley Additive Explanations (G-SHAP) - produces many additional types of explanations including: 1) *General classification explanations*; Why is this sample more likely to belong to one class rather than another? 2) *Intergroup differences*; Why do our model's predictions differ between groups of observations? 3) *Model failure*; Why does our model perform poorly on a given sample? We formally define these types of explanations and illustrate their practical use on real data.

*Dillon Bowen is credited with the original concept, code, data analysis, and initial paper draft. Lyle Ungar is credited with contributions to the draft and mathematical notation. Documentation can be found at <https://dsbowen.github.io/gshap/>.

1 Introduction

It is increasingly important to explain how machine learning models work [1]. For example, machine learning models are currently used to assess criminals' risk of recidivism [2], assist in medical diagnosis [3], and recruit employees [4]. Researchers have proposed a variety of methods to measure which features most affect a model's output [5, 6, 7]. However, many important questions about a model cannot be answered just by measuring feature importance with respect to its output. Rather, we need to measure feature importance with respect to *some function* of the model's output. Considering the following examples:

Example 1: General classification. Suppose we have a black-box model which diagnoses patients with COVID-19, the flu, or the common cold based on their symptoms. Existing explanatory methods can tell us why our model diagnosed a patient with COVID-19. However, medical practitioners, public health officials, and patients may also want to know, *why was this patient diagnosed with COVID-19, rather than the flu?* and *how do the symptoms which distinguish COVID-19 from the flu differ from those which distinguish COVID-19 from the common cold?*. These 'general classification explanations' are not provided by existing explainable AI methods.

Example 2: Intergroup differences. Suppose we have a black-box model which predicts a criminal's risk of recidivism to determine whether they are eligible for parole. Existing explanatory methods can tell us why our model predicted that a criminal has a high recidivism risk. However, lawyers and criminal justice policymakers may also want to know why our model predicts that Black criminals generally have a higher recidivism risk than White criminals. These 'intergroup difference explanations' are also not handled cleanly by existing feature importance methods.

Example 3: Model failure. Suppose we have a black-box model which forecasts GDP growth based on macroeconomic variables. Existing explanatory methods can tell us why our model forecast 3% GDP growth in a given year. However, fiscal policymakers, financial analysts, and central bankers may also want to know why our model failed to predict the 2008-2009 Great Recession. Such 'model failure' explanations are also not cleanly provided by existing methods.

To answer these questions, we generalize a popular, mathematically well-grounded explanation technique, Shapley Additive Explanations (SHAP). Our new method - Generalized Shapley Additive Explanations (G-SHAP) - produces three new types of explanations which current techniques do not cleanly provide: general classification explanations, intergroup difference explanations, and model failure explanations (see Table 1). We formally define these types of explanations and illustrate their practical use on real data.

Table 1: Types of explanations generated by G-SHAP

Explanation type	General formulation
General classification	
Question	Let $f_c(x_i)$ be the probability that observation x_i belongs to class c according to model f . Why are the observations in set X more likely to belong to a positive class $c \in C_+$ than a negative class $c \in C_-$?
$g(f, X, \Omega)$	$\frac{\prod_i \sum_{c \in C_+} f_c(x_i)}{\prod_i \sum_{c \in C_+} f_c(x_i) + \prod_i \sum_{c \in C_-} f_c(x_i)}$
Ω	$\{C_+, C_-\}$
Explanation	Let Pr_X [Pr_Z] be the probability that every observation in set X [the background set Z] belongs to a positive class rather than a negative class. ϕ_g^j is the number of percentage points of this difference $Pr_X - Pr_Z$ explained by feature j .
Intergroup differences	
Question	Why is there a difference d between the model output for $X_0 \subseteq X$ and the model output for $X_1 \subseteq X$?
$g(f, X, \Omega)$	$d(E_{x_i \in X_0}[f(x_i)], E_{x_i \in X_1}[f(x_i)])$
Ω	$\{X_0, X_1\}$
Explanation	ϕ_g^j is the amount of the intergroup difference explained by feature j .
Model failure	
Question	Why did our model perform poorly on a set of observations X with labels y , as measured by loss function L ?
$g(f, X, \Omega)$	$L(f, X, y)$
Ω	$\{y\}$
Explanation	Let $L(f, X, y)$ [$L(f, Z, y)$] be the loss over a set of observations X [the background set Z]. ϕ_g^j is the amount of this difference $L(f, X, y) - L(f, Z, y)$ explained by feature j .

2 Generalized Shapley Additive Explanations

Researchers have proposed several methods to measure which features most affect a model’s output [5, 6, 7]. Shapley Additive Explanations (SHAP), derived from Shapley values in game theory [8, 9], is a popular and mathematically well-grounded example of a feature importance measure [10, 6]. According to SHAP, the importance of feature j for the output of model f , $\phi^j(f)$, is a weighted sum of the feature’s contribution to the model’s output $f(x_i)$ over all possible feature combinations [11]:

$$\phi^j(f) = \sum_{S \subseteq \{x^1, \dots, x^p\} \setminus \{x^j\}} \frac{|S|!(p - |S| - 1)!}{p!} (f(S \cup \{x^j\}) - f(S)) \quad (1)$$

Where x^j is feature j , S is a subset of features, and p is the number of features in the model.

In practice, $f(S)$ is estimated by substituting in values for the remaining features, $\{x^1, \dots, x^p\} \setminus S$, from a randomly selected observation in a background dataset $z \in Z$. Suppose we compute the model output for an observation $f(x_i)$ and a background observation $f(z)$. Each SHAP value ϕ^j is the amount of this difference $f(x_i) - f(z)$ due to feature j .

Shapley feature importance measures have many practical applications. These applications include using SHAP to identify patient risk factors in tree-based medical diagnostic models [12] and determine which features of satellite images are most important in generating poverty maps [13]. Some researchers have modified Shapley values for specific use cases. For example, [14] uses counterfactual-adjusted Shapley values for attributing sales to advertisements, and [15] uses a Shapley variant to measure which features lead deep neural networks to be ‘confident’ in their output.

Here we present a unifying framework for SHAP and its variants; Generalized Shapley Additive Explanations (G-SHAP). G-SHAP allows us to compute the feature importance of any function g of a model’s output. Define a G-SHAP value $\phi_g^j(f)$ as:

$$\phi_g^j(f) = \sum_{S \subseteq \{x^1, \dots, x^p\} \setminus \{x^j\}} \frac{|S|!(p - |S| - 1)!}{p!} (g(f, S \cup \{x^j\}, \Omega) - g(f, S, \Omega)) \quad (2)$$

Where Ω is a set of additional arguments.

G-SHAP values have a similar interpretation as SHAP values. Suppose we compute a function of a model’s output for a sample $g(f, X, \Omega)$ and a background dataset $g(f, Z, \Omega)$. Each G-SHAP value ϕ_g^j is the amount of this difference $g(f, X, \Omega) - g(f, Z, \Omega)$ due to feature j .

In the following sections, we demonstrate how to use G-SHAP, along with select generalized functions g , to produce three new types of explanations not provided by existing techniques: general classification explanations, intergroup difference explanations, and model failure explanations.

3 General classification explanations

Example. In this example, we ask how the features which distinguish Versicolour from Setosa irises compare to those which distinguish Versicolour from Virginica irises. (This is analogous to asking how the symptoms which distinguish COVID-19 from the flu compare to those which distinguish COVID-19 from the common cold; however, a sufficient COVID-19 dataset is not publicly available.) We find that petal length is the most important feature. Additionally, petal length is relatively less important, and sepal length relatively more important, for distinguishing Versicolour from Setosa (as opposed to distinguishing Versicolour from Virginia).

General formulation. More generally, let $f_c(x_i)$ be the probability that observation x_i belongs to class $c \in C$ according to model f . Let $C_+ \subset C$ be a set of ‘positive classes’ and $C_- \subset C$ be a set of ‘negative classes’. For a single observation x_i , we can ask why our model predicts it comes from a positive class rather than a negative class. For a randomly selected set of observations X from positive classes, we can ask which features best distinguish observations which belong to positive classes from observations which belong to negative classes. Here, the positive class is Versicolour, and the negative class is Setosa or Virginica.

Table 2: Comparing Versicolour sample X to background data Z in terms of g

	Comparison	
	Versicolour-Setosa	Versicolour-Virginica
$g(f, X, \Omega)$	1.00	1.00
$g(f, Z, \Omega)$	0.65	0.42
$g(f, X, \Omega) - g(f, Z, \Omega)$	0.35	0.58

Model. We train Scikit-learn’s support vector classifier with default settings on the Scikit-learn iris dataset ($n = 150$), using a 75%-25% train-test split [16]. The purpose of this dataset is to predict an iris’s species (Versicolour, Setosa, or Virginica) based on its petal length, petal width, sepal length, and sepal width. Our classifier achieves 97% test accuracy.

G-SHAP values. We compute the G-SHAP values, where g measures the probability that every observation in set X belongs to a positive class given that every observation belongs to a positive class or every observation belongs to a negative class.

$$g(f, X, \Omega) = \frac{\prod_i \sum_{c \in C_+} f_c(x_i)}{\prod_i \sum_{c \in C_+} f_c(x_i) + \prod_i \sum_{c \in C_-} f_c(x_i)} \quad (3)$$

Where the additional arguments to g are the sets of positive and negative classes; $\Omega = \{C_+, C_-\}$.

Note that traditional SHAP values are a specific instance of $g(f, X, \Omega)$ in which set X contains a single observation x_i , the set of positive classes is the model’s classification of x_i ; $C_+ = \{\arg \max_{c \in C} f_c(x_i)\}$, and the negative classes are the complement of the positive classes $C_- = C \setminus C_+$. Also note that, while our generalized function $g(f, X, \Omega)$ uses only two subsets of classes, C_+ and C_- , it easily extends to accommodate any number of subsets.

In our example, g measures the probability that every iris in this sample is a Versicolour, given that every iris in this sample is a Versicolour or every iris in this sample is a Setosa. Specifically, the set of positive classes C_+ is {Versicolour}, and the set of negative classes C_- is {Setosa}.

$$g(f, X, \Omega) = \frac{\prod_i f_{\text{Versicolour}}(x_i)}{\prod_i f_{\text{Versicolour}}(x_i) + \prod_i f_{\text{Setosa}}(x_i)} \quad (4)$$

Similarly, to discover which features distinguish Versicolour from Virginica, C_- is {Virginica}. As before, $C_+ = \{\text{Versicolour}\}$.

Interpretation. To discover which features distinguish Versicolour from Setosa or Virginica, our set of observations X is a random sample of test observations classified as Versicolour (Figure 1). Note that our analysis will not work if X contains a mixture of iris species. Intuitively, we are asking what features lead our model to believe that X contains only Versicolour irises. If X contains a mixture of iris species, the G-SHAP values will be 0 for all features, indicating that our model does not believe X contains only Versicolour irises.

For interpretation, we compare our Versicolour sample X to a background dataset Z (the shuffled training data) in terms of g . For the Versicolour-Setosa comparison, our support vector classifier correctly predicts a 100% chance that our sample contains only Versicolour, as opposed to only Setosa (see Table 2). It also predicts a 65% chance that our background dataset contains only Versicolour, as opposed to only Setosa. Each G-SHAP value ϕ_g^j is the number of percentage points of this difference (35 percentage points) explained by feature j . We interpret the G-SHAP values for the Versicolour-Virginica comparison similarly.

For general classification explanations more broadly, our model predicts that each observation in our sample belongs to a positive class, rather than a negative class, with probability Pr_X . It also predicts that each observation in the background dataset comes from a positive class, rather than a

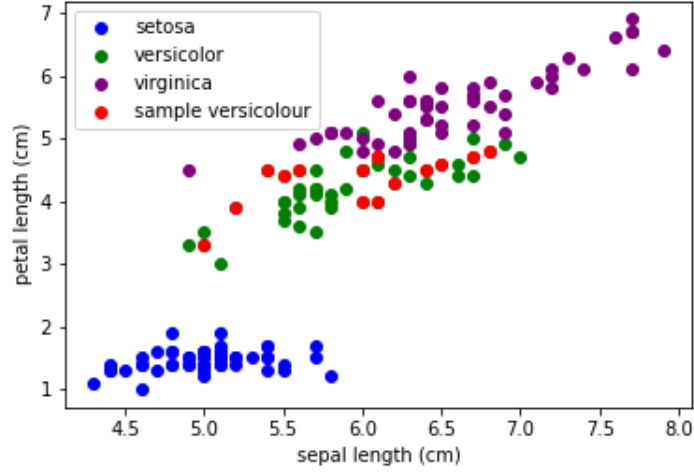


Figure 1: Plot of iris sepal length and petal length by species. Our sample Versicolour irises are highlighted in red.

negative class, with probability Pr_Z . Each G-SHAP value ϕ_g^j is the number of percentage points of this difference $Pr_X - Pr_Z$ explained by feature j .

Results. We plot our results in Figure 2, normalizing the G-SHAP values to sum to 1 for both comparisons (Versicolour-Setosa and Versicolour-Virginica). We find that petal length is the most important feature. Additionally, petal length is relatively less important, and sepal length relatively more important, for distinguishing Versicolour from Setosa (as opposed to distinguishing Versicolour from Virginia).

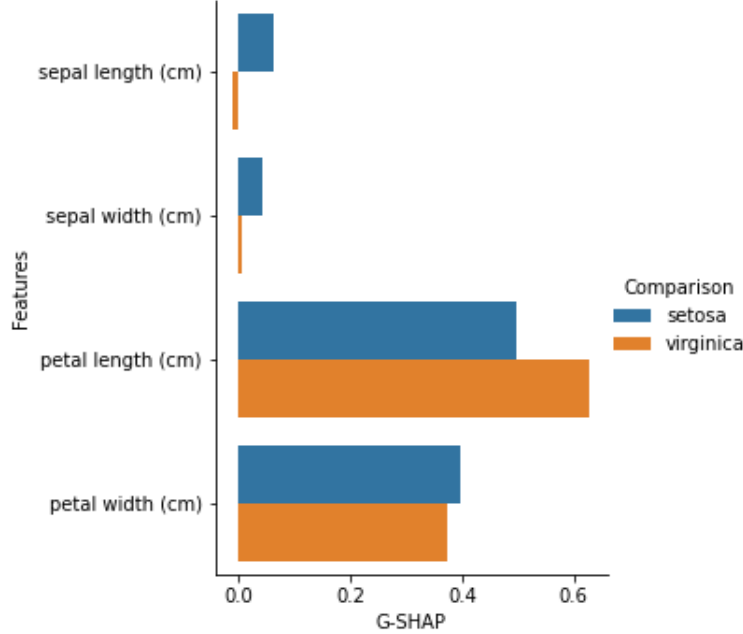


Figure 2: Normalized G-SHAP values for distinguishing Versicolour from Setosa (blue) and Versicolour from Virginica (orange). Note that petal length is relatively more important for the latter distinction, and sepal width relatively more important for the former.

4 Intergroup differences

Related work. Machine learning models are often used to study intergroup differences. For example, [12] describes a method to identify groups which differ in feature importances with respect to a model’s output. The method begins by computing feature importances with respect to a model’s output using SHAP, then clusters observations in a feature importance embedding space. By contrast, we present a method to identify features which explain why a model’s output differs between groups. Our method begins with groups of observations for which a model makes different predictions, then computes feature importances with respect to the difference.

Example. In this example, we create a model to predict whether a criminal will recidivate within two years of release. We find that our model predicts higher recidivism rates for Black criminals than for White criminals, then use G-SHAP to identify which features are most responsible for this difference. We find that prior convictions, age, and race are the most important variables.

General formulation. More generally, suppose we have two groups of observations, $X_0 \subseteq X$ and $X_1 \subseteq X$, and a measure d of how different our model output is between groups. We can ask why our model makes different predictions for different groups. Here, groups X_0 and X_1 are White and Black criminals respectively, the model output is a classification which indicates that a criminal will recidivate within two years, and the difference measure is the relative difference in recidivism rates.

Model. We train Scikit-learn’s support vector classifier [16] with default settings on the Correctional Offender Management Profiling for Alternative Sanctions dataset [17], selecting only Black and White criminals ($n_{\text{Black}} = 3565$, $n_{\text{White}} = 1904$), using a 75%-25% train-test split. The purpose of this dataset is to predict whether a criminal will recidivate within two years of release based on demographic information and criminal history. Our classifier achieves 68% test accuracy. The predicted recidivism rate is 63% higher for Black criminals (65%) than for White criminals (40%) (Figure 3).

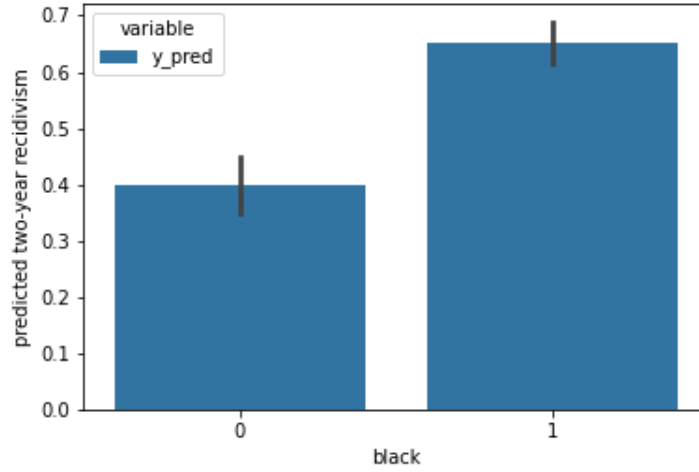


Figure 3: Difference in predicted recidivism rates between White and Black criminals.

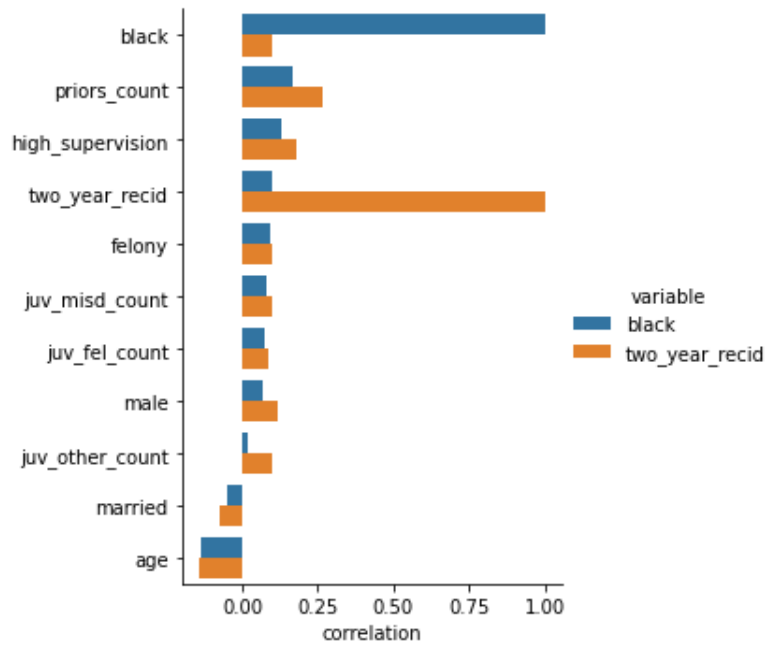


Figure 4: Plot of the correlations of all variables in our dataset with race (Black) and two-year recidivism.

Figure 4 plots the correlations of all variables in our dataset with race (Black) and two-year recidivism. Visually, it is unclear which variables lead our model to believe that Black criminals are at higher risk for recidivism.

G-SHAP values. We compute the G-SHAP values, where g measures the difference in model outputs between groups. In this case, d is the relative difference between group means. Intuitively, this measures the relative difference in predicted recidivism rates between Black and White criminals.

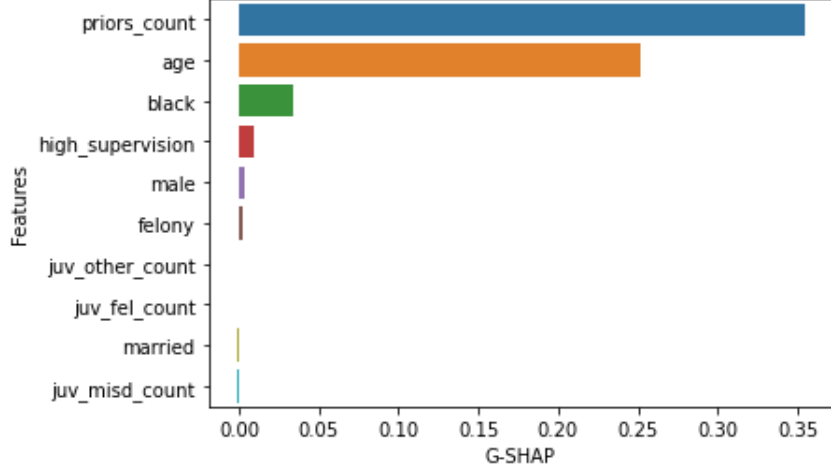


Figure 5: G-SHAP values for why our model predicts higher recidivism rates for Black criminals than for White criminals.

In the binary case, relative mean difference is known as *disparate impacts* in the algorithm fairness literature [18].

$$g(f, X, \Omega) = d(E_{x \in X_0}[f(x)], E_{x \in X_1}[f(x)]) = \frac{n_{\text{Black}}^{-1} \sum_{x_i \in X_{\text{Black}}} \mathbf{1}(f(x_i) = \text{Recidivate})}{n_{\text{White}}^{-1} \sum_{x_i \in X_{\text{White}}} \mathbf{1}(f(x_i) = \text{Recidivate})} - 1 \quad (5)$$

Where additional arguments to g indicate group membership; $\Omega = \{X_0, X_1\}$.

Although we express the intergroup difference measure as a function of group means, our method easily extends to intergroup difference measures more generally; for example, a difference between group medians or a Wasserstein distance between sample distributions.

Interpretation. For interpretation, we compare our test data X to a background dataset Z (the shuffled training data) in terms of g . Consistent with our previous result, the relative difference in predicted recidivism rates is 63%. The relative difference for our shuffled background data is 0% (because the background data are shuffled, the model has no way to statistically discriminate between groups). Each G-SHAP value ϕ_g^j is the number of percentage points of this difference (63 percentage points) explained by feature j .

For intergroup difference explanations more broadly, our model predicts the distance between groups is $d(E_{x \in X_0}[f(x)], E_{x \in X_1}[f(x)])$ for our sample, and 0 for the shuffled background data. Each G-SHAP value ϕ_g^j is the amount of this difference explained by feature j .

Results. We plot our results in Figure 5. This plot tells us which features explain the relative difference in predicted recidivism rates between Black and White criminals. We see that the most important features are number of prior convictions, age, and race.

5 Model failure

Related work. Models which perform well during training often perform poorly in real-world applications. The ubiquity of this problem is reflected in recent efforts to make machine learning more adaptable and robust. These efforts include transfer learning [19] and lifelong learning [20]. When, however, our models fail, we demand to know why. In the wake of the 2008-2009 financial crisis, for example, politicians, central bankers, and ordinary citizens demanded to know why economic models failed to predict the most serious economic downturn since the Great Depression.

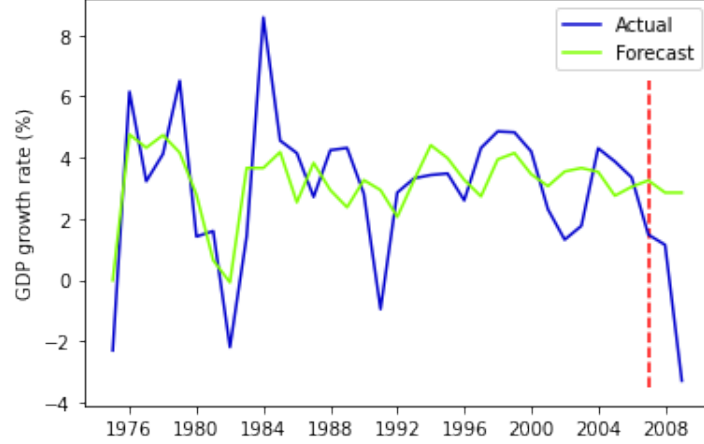


Figure 6: GDP growth forecasts from 1975 to 2009. Our model was trained on years 1975-2007, indicated by the red vertical line.

Example. In this example, we create a model to forecast GDP growth based on macroeconomic variables, then ask why it fails to predict the 2008-2009 financial crisis. We find that a high volume of real estate loans as a percent of GDP in 2007-2008 was detrimental to model performance.

General formulation. More generally, we can ask why our model performed poorly on sample X , as measured by loss function L . Here, our sample is the years 2008-2009, and our loss function is R^2 .

Model. We train a model on a GDP dataset with annual data, in which we produce 1-year ahead GDP growth forecasts based on 6 macroeconomic variables: growth in government expenditure, investment, consumption, exports, and imports, and real estate loans from commercial banks as a percent of GDP [21]. Our model first uses principal components analysis to reduce our data to 5 dimensions, then applies a K-nearest neighbors algorithm with 4 neighbors [16]. The number of principal components and neighbors were tuned using Scikit-learn’s grid search, training on data from 1975-1990 and validating on data from 1991-2007. Our model achieves a validation R^2 of .26. Retraining the model on the full training dataset (1975-2007), we achieve a training R^2 of .40. However, our forecasts for 2008-2009 perform worse than chance: $R^2 = -3.13$. From Figure 6, it is clear that our model fails to forecast the financial crisis.

G-SHAP values. We compute the G-SHAP values, where g is a loss function. In our case, the loss function is R^2 .

$$g(f, X, \Omega) = L(f, X, y) = 1 - \frac{\sum_t (\% \Delta \text{GDP}_t - f(x_{t-1}))^2}{\sum_t (\% \Delta \text{GDP}_t - \overline{\% \Delta \text{GDP}})^2} \quad (6)$$

Where additional arguments to g are the labels for X ; $\Omega = \{y\}$.

Interpretation. For interpretation, we make two comparisons. First, we compare the training data $X_{\text{train}} = \{x_{t-1} : t = 1975 \dots 2007\}$ to a background dataset Z (the shuffled training data) in terms of g using training labels; $y_{\text{train}} = \{\% \Delta \text{GDP}_t : t = 1975 \dots 2007\}$. Second, we compare the test data $X_{\text{test}} = \{x_{t-1} : t = 2008, 2009\}$ to Z in terms of g using test labels; $y_{\text{test}} = \{\% \Delta \text{GDP}_t : t = 2008, 2009\}$. By making these two comparisons, we can see the extent to which feature j contributed to model performance for the training data, and the extent to which feature j contributed to model performance for the test data. If feature j ’s contribution diminished between the training and test data, we can conclude it was responsible for our model’s poor test performance. A negative G-SHAP value indicates that feature j degraded performance.

Table 3: Model performance for training (1975-2007) and test (2008-2009) data

	Years	
	1975-2007	2008-2009
$g(f, X, \Omega)$	0.40	-3.13
$g(f, Z, \Omega)$	-0.28	-3.94
$g(f, X, \Omega) - g(f, Z, \Omega)$	0.68	0.81

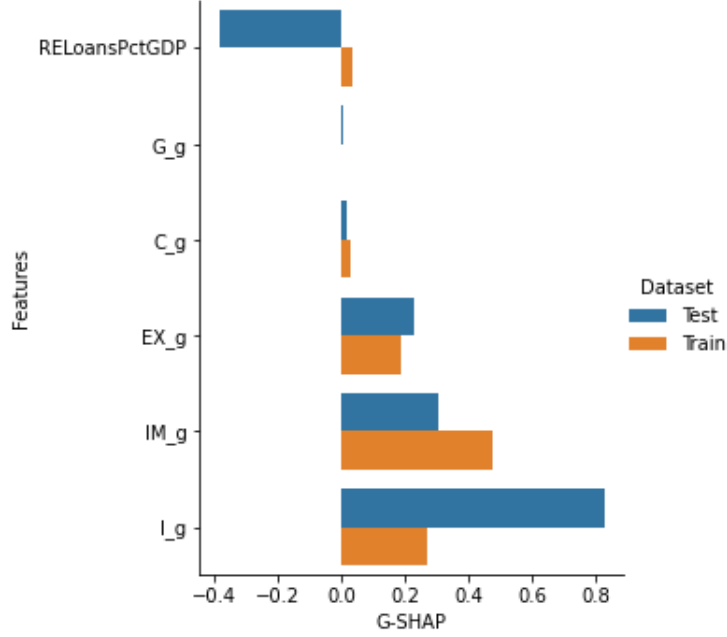


Figure 7: Normalized G-SHAP values for model performance on training data (1975-2007) and test data (2008-2009). The model's features 1-year lagged growth rates in government expenditure G_g , investment I_g , consumption C_g , exports EX_g , import IM_g , and 1-year lagged real estate loans as a percent of GDP $RELoansPctGDP$.

In the first comparison, our model's R^2 is .40 for the training data and -.28 for the background data (see Table 3). Each G-SHAP value ϕ_g^j is the amount of this difference (.68) explained by feature j . We interpret the G-SHAP values for the test data comparison comparison similarly.

For model failure explanations more broadly, our model has a loss $L(f, X, y)$ for our test sample and $L(f, Z, y)$ for the background data. Each G-SHAP value ϕ_g^j is the amount of this difference $L(f, X, y) - L(f, Z, y)$ explained by feature j .

Results. We plot our results in Figure 7. This plot tells us which features contribute most to model performance. Because the performance of the background dataset differs in terms of g when using the training versus test labels, we normalize the G-SHAP values to sum to 1 in order to compare performance. We see that real estate loans as a percent of GDP in 2007-2008 decreased the R^2 of our model's 2008-2009 GDP growth forecast by 38%.

6 Broader impact

It is increasingly important to explain how machine learning models work. Researchers have proposed a variety of methods to measure which features most affect a model’s output. However, many important questions about a model cannot be answered just by measuring feature importance with respect to its output. Rather, we need to measure feature importance with respect to *some function* of the model’s output. To answer this broader set of questions, we introduced Generalized Shapley Additive Explanations (G-SHAP) to compute generalized feature importances.

Explainability techniques such as G-SHAP may have negative impacts. For example, increased model transparency could exacerbate the threat of adversarial attacks. It is also possible that certain types of explanations, such as model failure explanations, could decrease user trust in reliable algorithms by drawing attention to rare errors.

G-SHAP also has significant potential for positive impacts. We illustrated these impacts by discussing three paradigmatic applications of G-SHAP: explaining *general classification* differences to improve transparency in medical diagnosis, explaining *intergroup differences* to detect potential sources of bias in models, and explaining *model failure* to predict a crisis in economic forecasting. Researchers and practitioners can easily extend our method to measure feature importance with respect to other generalized functions. We hope and expect that G-SHAP will provide a clean, principled way to measure feature importance for the purpose of improving algorithmic transparency, fairness, and performance.

References

- [1] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.
- [2] *Practitioners Guide to COMPAS*, 2012.
- [3] Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future healthcare journal*, 6(2):94, 2019.
- [4] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *Proceedings on privacy enhancing technologies*, 2015(1):92–112, 2015.
- [5] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.
- [6] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.
- [7] 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*, pages 1135–1144, 2016.
- [8] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.
- [9] Alvin E Roth. Introduction to the shapley value. *The Shapley value*, pages 1–27, 1988.
- [10] Shay B Cohen, Eytan Ruppín, and Gideon Dror. Feature selection based on the shapley value. In *IJCAI*, volume 5, pages 665–670, 2005.
- [11] Christoph Molnar. *Interpretable Machine Learning*. 2019. <https://christophm.github.io/interpretable-ml-book/>.
- [12] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. Explainable ai for trees: From local explanations to global understanding. *arXiv preprint arXiv:1905.04610*, 2019.
- [13] Kumar Ayush, Burak Uzkent, Marshall Burke, David Lobell, and Stefano Ermon. Generating interpretable poverty maps using object detection in satellite images. *arXiv preprint arXiv:2002.01612*, 2020.
- [14] Raghav Singal, Omar Besbes, Antoine Desir, Vineet Goyal, and Garud Iyengar. Shapley meets uniform: An axiomatic framework for attribution in online advertising. In *The World Wide Web Conference*, pages 1713–1723, 2019.
- [15] Susmit Jha, Sunny Raj, Steven Fernandes, Sumit K Jha, Somesh Jha, Brian Jalaian, Gunjan Verma, and Ananthram Swami. Attribution-based confidence metric for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 11826–11837, 2019.
- [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [17] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. Machine bias. 2016.
- [18] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.
- [19] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [20] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.
- [21] U.S. Bureau of Economic Analysis. Real gross domestic product [gdpc1], real government consumption expenditures and gross investment [gcec1], real gross private domestic investment [gpdic1], real personal consumption expenditures [pccc96], real exports of goods and services [expgsc1], real imports of goods and services [impgsc1], real estate loans, all commercial

banks [relacbw027sbog], 2020. data retrieved from Federal Reserve Bank of St. Louis, <https://fred.stlouisfed.org/>.