Understanding Global Feature Contributions Through Additive Importance Measures

Ian C. Covert [1] **Scott Lundberg** [2] **Su-In Lee** [1]

Abstract

Understanding the inner workings of complex machine learning models is a long-standing problem, with recent research focusing primarily on local interpretability. To assess the role of individual input features in a global sense, we propose a new feature importance method, Shapley additive global importance (SAGE), a model-agnostic measure of feature importance based on the predictive power associated with each feature. SAGE relates to prior work through the novel framework of *additive importance measures*, a perspective that unifies numerous other feature importance methods and shows that only SAGE properly accounts for complex feature interactions. We define SAGE using the Shapley value from cooperative game theory, which leads to numerous intuitive and desirable properties. Our experiments apply SAGE to eight datasets, including MNIST and breast cancer subtype classification, and demonstrate its advantages through quantitative and qualitative evaluations.

1. Introduction

Our lack of understanding about the inner workings of complex machine learning models, a long-standing problem, could impede the adoption of machine learning in many domains. Most recent research focuses on *local* interpretability, which explains individual predictions, e.g., the role of each input feature in a single patient diagnosis (Simonyan et al., 2013; Ribeiro et al., 2016; Sundararajan et al., 2017; Lundberg & Lee, 2017). Yet, users may require concise summaries of the *global* role of each feature to understand their importance across the entire dataset. For this, we require different model interpretation tools.

In this work we seek to understand models by measuring how much they rely on each feature, often referred to as the problem of *global feature importance*. While the idea of feature importance can be interpreted in multiple ways, we define a feature's importance based on the amount of predictive power it contributes. This perspective raises the challenge of dealing with complex feature interactions: features contribute different amounts of predictive power when introduced in isolation versus when introduced into a larger set of features. We aim to provide a feature importance measure that accounts for complex feature interactions, such as correlation, redundancy and complementary behavior.

To that end, we present the framework of additive importance measures, a view that unifies other methods that define feature importance in terms of predictive power (Section 2). Then, we present our measure for calculating feature importance, Shapley additive global importance (SAGE), which assigns importance values in a unique way that accounts for complex feature interactions (Section 3). SAGE is model-agnostic, and it is the only feature importance method that satisfies a number of desirable properties, such as accounting for complex feature interactions while remaining tractable to estimate.

This paper makes the following contributions:

1. *We derive SAGE* by applying Shapley values to a function representing the predictive power of subsets of features. Many desirable properties result from this definition of feature importance, including invariance to invertible feature transformations, and a relationship with SHAP (Lundberg & Lee, 2017).
2. We introduce the framework of *additive importance measures* to unify the prior literature, showing that numerous methods define feature importance in terms of predictive power, but only SAGE does so while accounting for complex interactions.
3. To confront tractability challenges with SAGE values, we propose an efficient *sampling-based approximation algorithm* that is much faster than calculating them naively via SHAP values.

We evaluate SAGE on eight datasets, including MNIST, breast cancer subtype classification, and six UCI datasets

[1]Paul G. Allen School of Computer Science & Engineering, University of Washington [2]Microsoft Research, Redmond. Correspondence to: Ian C. Covert <icovert@cs.washington.edu>.

(Dua & Graff, 2017). Quantitative metrics show that SAGE assigns feature importance values that are more representative of the predictive power associated with each feature.

2. Unifying Additive Importance Measures

We first discuss two notions of predictive power and then introduce the framework of *additive importance measures*, to provide necessary context for understanding our method (Section 3). The framework unifies numerous other methods that define feature importance in terms of predictive power.

2.1. Predictive Power of Feature Subsets

Consider a supervised learning task in which a model f is used to predict the response variable y given an input x, where x consists of individual features $(x^1, x^2, \ldots, x^d)$. We use uppercase symbols (e.g., X) to denote random variables and lowercase symbols (e.g., x) to denote values. Such models can be difficult to interpret, particularly complex ones such as neural networks or decision forests. Global feature importance methods provide a way to understand f by assigning scores $\phi_i \in \mathbb{R}$ to indicate how much f relies on each feature $i \in D \equiv \{1, \ldots, d\}$ across the entire dataset.

The notion of *feature importance* is open to different interpretations. Some methods use heuristics to measure how much f relies on each feature, such as with decision trees (Friedman et al., 2001). In this work, we assess the importance of each feature X^i by examining its role in enabling f to make accurate predictions, with "important" features defined as those whose absence degrades f's performance.

Although f is trained using all features, we examine its performance when given access only to subsets of features $X^S \equiv \{X^i \mid i \in S\}$ for different $S \subseteq D$. We therefore require a convention for evaluating f when it is deprived of the features $\bar{S} \equiv D \setminus S$. To accommodate the missing features $X^{\bar{S}}$, we define the *restricted model* f_S as

$$f_S(x^S) = \mathbb{E}\big[f(X) \;\big|\; X^S = x^S\big] \tag{1}$$

so that the missing features $X^{\bar{S}}$ are marginalized out using their conditional distribution $p(x^{\bar{S}}|X^S = x^s)$. Two special cases are $S = \varnothing$ and $S = D$, which are respectively the mean prediction $f_\varnothing(x^\varnothing) = \mathbb{E}\big[f(X)\big]$ and the full model $f_D(x) = f(x)$. We use this convention because f_S is closest to f on average and does not consider f's behavior off the data manifold (see Appendix A). We also note that it is common in recent work (Lundberg & Lee, 2017).

Using this convention for handling subsets of features, we next consider how much f's performance depends on each feature. Given a loss function ℓ, the population risk of f_S is defined as

$$\mathbb{E}\Big[\ell\big(f_S(X^S), Y\big)\Big], \tag{2}$$

with the expectation taken over the true data distribution $p(x, y)$. To analyze a function for *predictive power* that increases with improved performance, we consider the *reduction* in risk over the mean prediction $f_\varnothing(x^\varnothing) = \mathbb{E}\big[f(X)\big]$. We define a function on the power set $\mathcal{P}(D)$, denoted $v_f : \mathcal{P}(D) \mapsto \mathbb{R}$, as

$$v_f(S) = \underbrace{\mathbb{E}\Big[\ell\big(f_\varnothing(X^\varnothing), Y\big)\Big]}_{\text{Mean prediction}} - \underbrace{\mathbb{E}\Big[\ell\big(f_S(X^S), Y\big)\Big]}_{\text{Using features in } S}. \tag{3}$$

The left term is the loss achieved with the mean prediction $\mathbb{E}\big[f(X)\big]$, and the right term is the loss achieved using the features X^S. We see that $v_f(\varnothing) = 0$, and we generally expect that including more features in S will make $v_f(S)$ larger. Intuitively, $v_f(S)$ quantifies the amount of predictive power that f derives from the features X^S.

While v_f provides a *model-based* notion of the predictive power of X^S through f, we can also define a notion of *universal predictive power*. For this, we define the function v as the risk reduction from X^S when using optimal models

$$v(S) = \underbrace{\min_{\hat{y}} \; \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big]}_{\text{Optimal constant } \hat{y}} - \underbrace{\min_{g} \; \mathbb{E}\Big[\ell\big(g(X^S), Y\big)\Big]}_{\text{Optimal model using } X^S}, \tag{4}$$

where the left term is the loss achieved with an optimal constant prediction $\hat{y}$, and the right term is the loss achieved with an optimal model g from the unrestricted class of all functions (e.g., the Bayes classifier). Intuitively, v represents the maximum amount of predictive power that could hypothetically be derived from X^S.

The model-based notion of predictive power v_f approximates the universal predictive power v, particularly when f is nearly optimal, and the two coincide exactly in certain cases where f is optimal (see Appendix B).

2.2. Additive Importance Measures

We now introduce a framework that lets us unify numerous existing feature importance methods that either explicitly or implicitly define feature importance in terms of predictive power.

In certain very simple cases, such as a regression task with (X, Y) from a multivariate Gaussian where X has diagonal

covariance, features contribute predictive power in an additive manner: we have $v(S\cup\{i\})-v(S) = v(T\cup\{i\})-v(T)$ for all S, T where $i \notin S, T$. In such cases with additive feature contributions, we could define the importance of X^i as its contributed predictive power, $\phi_i = v(\{i\}) - v(\varnothing)$.

More generally, a feature's contribution is not additive because it depends on which features X^S are already present. We therefore propose a class of *additive importance measures*, which includes feature importance methods whose scores $\phi_1, \ldots, \phi_d$ can be understood as additive performance gains associated with each feature. These scores collectively approximate the predictive power function v. The class of methods is defined as follows.

Definition 1. *An* additive importance measure *is an importance measure $\phi_i \in \mathbb{R}$ for features $i \in D$ for which there exists a constant $\phi_0 \in \mathbb{R}$ so that the additive function*

$$u(S) = \phi_0 + \sum_{i \in S} \phi_i \tag{5}$$

provides a proxy for predictive power, i.e., $u(S) \approx v(S)$.

In this definition, u approximates v up to a constant value ϕ_0 by summing the values ϕ_i for each included feature $i \in S$. Each ϕ_i can be considered a feature importance value because it represents the performance gain associated with including X^i.

For most prediction problems, v will exhibit non-additive behavior, so u can provide only a crude approximation. Several existing feature importance methods therefore make tradeoffs by providing higher quality approximations in certain regions of the domain $\mathcal{P}(D)$. We need not model v perfectly, but a closer approximation gives a more accurate representation of each feature's contribution.

Additive importance measures use u to approximate v; however, methods that approximate v_f should also be understood as additive importance measures because approximations of v_f implicitly approximate v. The function v_f is a tool for understanding the model f, while v is a tool for understanding intrinsic properties of the data. Those problems are related, particularly when f is nearly optimal. This view allows the inclusion of more methods in the additive importance measure framework.

2.3. Existing Additive Importance Measures

We now unify parts of the literature on feature importance by identifying several methods that can be understood as additive importance measures. These methods can be divided into three categories, representing parts of the domain $\mathcal{P}(D)$ for which they make u model v most accurately. We provide a table in Appendix G that summarizes the methods in each category.

The first category of methods characterize predictive power when no more than one feature is excluded, providing an additive function u that accurately approximates v or v_f in the subdomain $\{D\} \cup \{D \setminus \{i\} \mid i \in D\} \subset \mathcal{P}(D)$.

The canonical method for this is a feature ablation study (e.g., Bengtson & Roth 2008), where, in addition to a model f trained on all features, separate models $f_1, f_2, \ldots, f_d$ are trained to account for the exclusion of each feature. Importance values are then assigned based on the degradation in performance. The importance values are

$$\phi_i = \mathbb{E}\Big[\ell\big(f_i(X^{D\setminus\{i\}}), Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] \tag{6}$$

for $i \in D$. A natural choice for ϕ_0 here is

$$\phi_0 = \min_{\hat{y}} \ \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] - \sum_{i \in D} \phi_i \tag{7}$$

because we then have $u(D), u(D \setminus \{i\})$ serving as estimators for $v(D), v(D \setminus \{i\})$, respectively. However, note that u does not account for v's behavior in the rest of $\mathcal{P}(D)$.

While feature ablation studies measure feature importance by approximating v, several other methods provide model-based notions feature importance for a model f via v_f. Permutation tests measure performance degradation when each column of the data is permuted (Breiman, 2001). Since permutation tests break feature dependencies, one variation advocates for a conditional permutation scheme (Strobl et al., 2008). Similarly, in a method we refer to as "mean importance," performance degradation is measured after mean imputing each feature (Setiono & Liu, 1997). These three methods are analogous to feature ablation studies, but they quantify how important each feature is to the model f by approximating v_f (see Appendix G for more details).

The second category of methods describe v when no more than one feature is included, providing an additive function u that accurately describes v in the subdomain $\{\varnothing\} \cup \{\{i\} \mid i \in D\} \subset \mathcal{P}(D)$. The methods in this category model the bivariate association between X^i and Y, quantifying the stand-alone predictive power of X^i.

Studying bivariate associations is common in computational biology (e.g., Liu et al. 2009) and can be used to identify sensitive features (Saltelli et al., 2004). For example, the squared correlation $\text{Corr}(X_i, Y)^2$ is equivalent to the variance reduction from a univariate linear model (up to a constant factor). More generally, one can measure the performance of univariate models trained to predict Y given X^i (Guyon & Elisseeff, 2003). Given a model g_i for each

feature $i \in D$, the importance values assigned are

$$\phi_i = \min_{\hat{y}} \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(g_i(X^i), Y\big)\Big], \quad (8)$$

with a natural choice for ϕ_0 being $\phi_0 = 0$. With these scores, we see that $u(\varnothing) = v(\varnothing) = 0$ and that $u(\{i\})$ is an estimator for $v(\{i\})$. However, note that u may not accurately represent v when multiple features are included.

Both of the previous categories of methods provide imperfect notions of feature importance due to an inability to account for feature interactions, such as redundancy or complementary behavior. For example, two perfectly correlated features with significant predictive power would both be deemed unimportant by a feature ablation study; and two complementary features would have their importance underestimated by univariate predictive models. The final category of methods addresses these types of issues.

The third category of methods account for complex feature interactions by attempting to model v across its entire domain $\mathcal{P}(D)$. By considering all feature subsets, such methods supersede the two other categories, which either exclude or include individual features. Our method, SAGE, belongs to this category. We show that SAGE assigns scores ϕ_i so that u models v_f optimally via weighted least squares.

3. Shapley Additive Global Importance

Here, we introduce our method, Shapley additive global importance (SAGE), for quantifying how much a model f depends on each feature. We first present SAGE as an application of the game theoretic Shapley value to the function v_f, and we then examine its properties, including how it can be understood as an additive importance measure. Finally, we propose a sampling-based approximation algorithm.

3.1. Shapley Values for Credit Allocation

Recall that the function v_f describes the amount of predictive power that a model f derives from subsets of features $S \subseteq D$. By defining feature importance through v_f, we quantify how critical each feature X^i is for enabling f to make accurate predictions.

It is natural to view the function v_f on $\mathcal{P}(D)$ as a cooperative game, representing the profit (predictive power) when each player (feature) participates (is made available to the model). Research in game theory has extensively analyzed credit allocation for cooperative games; we therefore apply a game theoretic solution known as the Shapley value, which is the unique credit allocation scheme that satisfies a set of fairness axioms (Shapley, 1953).

For any cooperative game $w : \mathcal{P}(D) \mapsto \mathbb{R}$, such as v or v_f, the scores $\phi_i(w)$ assigned to each player satisfy the following properties:

1. (Efficiency) They sum to the total improvement over the empty set, $\sum_{i=1}^{d} \phi_i(w) = w(D) - w(\varnothing)$.
2. (Symmetry) If $w(S \cup \{i\}) = w(S \cup \{j\})$ for all S, then $\phi_i(w) = \phi_j(w)$.
3. (Linearity) The game $w(S) = \sum_{k=1}^{n} c_k w_k(S)$, which is a linear combination of games $(w_1, \ldots, w_n)$, has scores $\phi_i(w) = \sum_{k=1}^{n} c_k \phi_i(w_k)$.
4. (Monotonicity) If for two games w, w' we have $w(S \cup \{i\}) - w(S) \geq w'(S \cup \{i\}) - w'(S)$ for all S, then $\phi_i(w) \geq \phi_i(w')$.
5. (Dummy) If $w(S) = w(S \cup \{i\})$ for all S, then $\phi_i(w) = 0$.

The Shapley value is the unique credit allocation scheme that satisfies these properties, and it is defined as:

$$\phi_i(w) = \frac{1}{d} \sum_{S \subseteq D \setminus \{i\}} \binom{d-1}{|S|}^{-1} \Big[w(S \cup \{i\}) - w(S)\Big]. \quad (9)$$

The expression in Eq. 9 shows that the Shapley value $\phi_i(w)$ is a weighted average of the incremental changes from including i. In SAGE, we assign feature importance values using the Shapley values $\phi_i(v_f)$ for $i = 1, 2, \ldots, d$.

3.2. Properties of SAGE Values

SAGE values satisfy many intuitive and desirable properties. Some arise from the way the cooperative game v_f is defined, and others arise from the properties of Shapley values. Below, we enumerate these properties.

1. Due to the efficiency property, SAGE values sum to the total improvement in performance over the mean prediction, $\sum_{i=1}^{d} \phi_i(v_f) = \mathbb{E}\big[\ell\big(f_\varnothing(x^\varnothing), y\big)\big] - \mathbb{E}\big[\ell\big(f(x), y\big)\big]$. That is, the feature importance values add up to X's total predictive power $v_f(D)$.
2. Due to the symmetry property, features X^i, X^j with a deterministic relationship (e.g., perfect correlation) always have equal importance. To see this, remark that $f_{S\cup\{i\}}(x^{S\cup\{i\}}) = f_{S\cup\{j\}}(x^{S\cup\{j\}})$ for all (S, x), so that $v_f(S \cup \{i\}) = v_f(S \cup \{j\})$.
3. Due to the linearity property, SAGE values are the expectation of per-instance SHAP values applied to the model loss (Lundberg et al., 2020). By this, we mean the Shapley values $\phi_i(v_{f,x,y})$ of the cooperative game

$$v_{f,x,y}(S) = \ell\big(f_\varnothing(x^\varnothing), y\big) - \ell\big(f_S(x^S), y\big). \quad (10)$$

From the interpretation of $p(x, y)$ as a compound distribution, we see that $\phi_i(v_f) = \mathbb{E}_{XY}\big[\phi_i(v_{f,X,Y})\big]$. While the values $\mathbb{E}\big[\phi_i(v_{f,X,Y})\big]$ were used in prior work, they were not analyzed in depth and were very costly to calculate via local explanations $\phi_i(v_{f,x,y})$.

4. Due to the monotonicity property, if we have two response variables Y, Y' with models f, f' and we have $v_f(S \cup \{i\}) - v_f(S) \geq v_{f'}(S \cup \{i\}) - v_{f'}(S)$ for all S, so that X^i contributes more predictive power for Y than for Y', then the respective SAGE values satisfy $\phi_i(v_f) \geq \phi_i(v_{f'})$.

5. Due to the dummy property, we have $\phi_i(v_f) = 0$ if $f_S(x^S) = f_{S\cup\{i\}}(x^{S\cup\{i\}})$ for all (S, x). Perhaps surprisingly, f being invariant to X^i is not sufficient for this to hold. Features may receive non-zero importance even if they are not used by f if they are proxies for other features. The sufficient condition for $\phi_i(v_f) = 0$ is that X^i must be conditionally independent of $f(X)$ given all possible subsets of features X^S. In terms of directed graphical models, X^i must belong to a subgraph disjoint from the one containing $f(X)$.

6. Due to our definition of v_f, SAGE values are invariant to invertible mappings of the features. If we apply an invertible function h to feature X^i, defining $Z^i = h(X^i)$, and use a model f' that applies the inverse h^{-1} to Z^i before f, then the SAGE values of the new model under the new data distribution are unchanged. For example, SAGE values do not depend on whether gene counts or log gene counts are used.

Lastly, SAGE values have an elegant interpretation when the loss function is cross entropy or mean squared error (MSE) and the model f is optimal. For brevity, we consider only the classification case (see Appendix C for more details). With cross entropy loss, the optimal model is the Bayes classifier, which predicts the conditional distribution $f^*(x) = p(y|X = x)$. The cooperative game is then $v_{f^*}(S) = I(Y; X^S)$, where I denotes mutual information. The resulting SAGE values are therefore

$$\phi_i(v_{f^*}) = \frac{1}{d} \sum_{S \subseteq D\setminus\{i\}} \binom{d-1}{|S|}^{-1} I(Y; X^i \mid X^S). \quad (11)$$

The expression in Eq. 11 represents a weighted average of the conditional mutual information, i.e., the reduction in uncertainty about Y from incorporating X^i into different subsets X^S. An analogous result arises in the MSE case, and in both cases the SAGE values satisfy $\phi_i(v_{f^*}) \geq 0$. Through this we see that although SAGE is a tool for model interpretation, it provides insight into intrinsic relationships of the data (e.g., mutual information) when applied to optimal models (e.g., the Bayes classifier).

3.3. SAGE as an Additive Importance Measure

Though not immediately obvious, SAGE is in fact an additive importance measure (Section 2). Prior work has shown that Shapley values (Eq. 9) can be understood as the solution to a weighted least squares problem (Charnes et al., 1988; Lundberg & Lee, 2017). From this work, we see that SAGE provides an additive approximation to v_f with $u(S) = \sum_{i\in S} \phi_i$, where the values $\phi_1, \phi_2, \ldots, \phi_d$ are solutions to the optimization problem

$$\min_{\phi_1,\ldots,\phi_d} \sum_{S\subseteq D} \frac{d-1}{\binom{d}{|S|}|S|(d-|S|)} \Big(\sum_{i\in S} \phi_i - v_f(S)\Big)^2. \quad (12)$$

Note that the weights in Eq. 12 force $\sum_{i\in D} \phi_i = v_f(D)$. Interpreting SAGE values as the solution to Eq. 12 reveals that SAGE attempts to describe the behavior of v_f across its whole domain, modeling it optimally via weighted least squares. Although the weighting is complex, this exact weighting scheme yields the importance values that satisfy SAGE's desirable properties (Section 3.2).

Using this interpretation of Shapley values, we observe that two other methods can be categorized as additive importance measures. The mean SHAP value of the loss (Lundberg et al., 2020) and Shapley Net Effects for linear models (Lipovetsky & Conklin, 2001) are both related to SAGE. However, SAGE admits much faster estimation, without explaining every individual prediction or fitting an exponential number of models. Neither of these methods have previously been connected to existing work through the framework of additive importance measures.

3.4. SAGE Approximation

We next address the question of how to calculate SAGE values $\phi_i(v_f)$. Obtaining these values efficiently is challenging for two reasons. First, there are an exponential number of subsets $S \subseteq D$. Second, evaluating the restricted models f_S is often difficult, requiring a Monte Carlo estimate with $X^{\bar{S}}$ sampled from $p(x^{\bar{S}}|X^S = x^S)$.

Both challenges have been confronted by prior work using Shapley values (Štrumbelj & Kononenko, 2014; Lundberg & Lee, 2017). Like those studies, we sidestep the exponential complexity of exact calculation using an approximation algorithm, addressing the first challenge by evaluating the loss for random subsets of features $S \subseteq D$ and the second by sampling $X^{\bar{S}}$ from its marginal distribution.

Algorithm 1 presents the approximation procedure, in which $\phi_i(v_f)$ is estimated by attempting to average many samples

Algorithm 1 Sampling-based approximation for SAGE.

Input: data $\{x_i, y_i\}_{i=1}^N$, model f, loss function ℓ, outer loop samples n, inner loop samples m
$\hat{\phi}_1 = 0, \hat{\phi}_2 = 0, \ldots, \hat{\phi}_d = 0$
$marginalPred = \frac{1}{N}\sum_{i=1}^N f(x_i)$
for $i = 1$ **to** n **do**
  Sample (x, y) from data
  Sample π, a permutation of D
  $S = \varnothing$
  $lossPrev = \ell(marginalPred, y)$
  for $j = 1$ **to** d **do**
    $S = S \cup \{\pi[j]\}$
    $\hat{y} = 0$
    for $k = 1$ **to** m **do**
      Sample $x_k^{\bar{S}} \sim q(x^{\bar{S}}|X^S = x^S)$
      $\hat{y} = \hat{y} + f(x^S, x_k^{\bar{S}})$
    end for
    $loss = \ell(\frac{\hat{y}}{m}, y)$
    $\Delta = lossPrev - loss$
    $\hat{\phi}_{\pi[j]} = \hat{\phi}_{\pi[j]} + \Delta$
    $lossPrev = loss$
  end for
end for
Return: $\frac{\hat{\phi}_1}{n}, \frac{\hat{\phi}_2}{n}, \ldots, \frac{\hat{\phi}_d}{n}$

of the form $\ell\big(f_S(x^S), y\big) - \ell\big(f_{S\cup\{i\}}(x^{S\cup\{i\}}), y\big)$. In each sample, we draw (x, y) from the empirical data distribution, determine S based on a random permutation π of feature indices D, and obtain $f_S(x^S)$ via Monte Carlo approximation with a distribution $q(x^{\bar{S}}|X^S = x^S)$ substituted for $p(x^{\bar{S}}|X^S = x^S)$.

In practice we sample from the marginal distribution $q(x^{\bar{S}}|x^S) = p(x^{\bar{S}})$, which corresponds to an assumption of feature independence. Another option is to mean impute the missing features, which corresponds to a further assumption of model linearity. Prior work has used sampling from the marginal distribution in a similar manner (Štrumbelj & Kononenko, 2014; Lundberg & Lee, 2017), but doing so alters some of SAGE's properties (see Appendix D).

We note that Algorithm 1 resembles the sampling algorithm from Interactions-based Method for Explanation (IME). However, it differs by aiming at a global explanation and wrapping a loss function around the model output (Štrumbelj & Kononenko, 2014). It also resembles a sampling-based algorithm for assessing the sensitivity of functions to their various inputs (Song et al., 2016).

We make two claims regarding the estimates from Algorithm 1, which are stated in Theorems 1 and 2 (with proofs in Appendix E). First, we show that Algorithm 1 converges to the correct values when run under the right conditions.

Theorem 1. *The SAGE value estimates $\hat{\phi}_i(v_f)$ from Algorithm 1 converge to the correct values $\phi_i(v_f)$ when run with $n \to \infty$, $m \to \infty$, with an arbitrarily large dataset $\{(x_i, y_i)\}_{i=1}^N$, and with sampling from the correct conditional distribution $q(x^{\bar{S}}|X^S = x^S) = p(x^{\bar{S}}|X^S = x^S)$.*

The next result shows that the estimates have variance that reduces at a linear rate.

Theorem 2. *The SAGE value estimates $\hat{\phi}_i(v_f)$ from Algorithm 1 have variance that reduces at the rate of $O(\frac{1}{n})$.*

In practice, the algorithm can run only for a finite number of iterations, so it is important to monitor the values of n, m that lead to approximate convergence. For a given value of m, one can keep a running variance estimate and terminate the algorithm when it falls below a threshold value.

4. Related Work

Section 2 described prior work that we unify under the framework of additive importance measures. There are also methods that do not fit into our framework. These are often model-specific heuristics that do not directly relate to the predictive power associated with each feature. For linear models, a simple heuristic is calculating the magnitude of each coefficient (Guyon et al., 2002). For tree-based models, options include Gini importance and counting splits based on each feature (Friedman et al., 2001). For neural networks, one can examine the magnitude of weights or aggregate local explanations (Horel et al., 2018), such as integrated gradients (Sundararajan et al., 2017).

As noted above, Shapley values have been studied extensively in game theory (Shapley, 1953) and have been applied to machine learning for both local (Štrumbelj & Kononenko, 2014; Lundberg & Lee, 2017) and global interpretability. One early study on Shapley Net Effects proposed training linear models with every combination of features (Lipovetsky & Conklin, 2001), which is similar to SAGE except for its use of linear models. Extending Shapley Net Effects to other model classes is straightforward, but efficient estimation would be impractical without a sampling-based approximation algorithm like the one we proposed. The literature has thus far overlooked the unification of these methods with other work on feature importance (Section 2), and it has not identified the specific connection with SHAP (Lundberg & Lee, 2017).

Some prior work has considered the application of Shapley values to function sensitivity, a subtly different problem than explaining the performance of machine learning models (Owen, 2014; Song et al., 2016; Owen & Prieur, 2017; Benoumechiara & Elie-Dit-Cosaque, 2019). See Appendix F for more details on how these problems differ.

SHAP was proposed for local interpretability, but it has

Table 1. Comparison of feature importance methods. The table is separated into four groups. The first contains methods that are not additive importance measures, and the remaining three correspond to the categories outlined in Section 2.3. Each column represents an attribute that methods may or may not satisfy. *Agnostic:* whether the method works with any model class. *Performance:* whether the scores are related to the performance gains associated with each feature. *Interactions:* whether complex feature interactions are considered. *Missingness:* whether held out features are accounted for properly (e.g., by training a new model, or marginalizing them out). *Tractable:* whether the method is computationally efficient. Check marks ($\surd$) show that a property is satisfied, crosses ($\times$) that it is not, and tildas ($\sim$) that it is to some extent.

Method	AGNOSTIC	PERFORMANCE	INTERACTIONS	MISSINGNESS	TRACTABLE
Coeff. Size in Linear Models	$\times$	$\times$	$\times$	$\times$	$\surd$
Gini Importance	$\times$	$\times$	$\times$	$\times$	$\surd$
Number of Splits	$\times$	$\times$	$\times$	$\times$	$\surd$
Neural Network Weights	$\times$	$\times$	$\times$	$\times$	$\surd$
Aggregated Local Saliency	$\times$	$\times$	$\times$	$\times$	$\surd$
Mean Abs. Output SHAP	$\surd$	$\times$	$\surd$	$\sim$	$\times$
Feature Ablation	$\surd$	$\surd$	$\times$	$\surd$	$\sim$
Permutation Test	$\surd$	$\surd$	$\times$	$\sim$	$\surd$
Conditional Permutation Test	$\surd$	$\surd$	$\times$	$\surd$	$\sim$
Mean Importance	$\surd$	$\surd$	$\times$	$\times$	$\surd$
Univariate Predictors	$\times$	$\surd$	$\times$	$\surd$	$\surd$
Squared Correlation	$\times$	$\surd$	$\times$	$\surd$	$\surd$
Shapley Net Effects	$\times$	$\surd$	$\surd$	$\surd$	$\times$
Mean Loss SHAP	$\surd$	$\surd$	$\surd$	$\sim$	$\times$
SAGE	$\surd$	$\surd$	$\surd$	$\sim$	$\sim$

been applied heuristically to global importance by calculating the mean absolute attribution value (Lundberg & Lee, 2017) and using SHAP on the loss instead of the model output (Lundberg et al., 2020). Our work shows that SAGE values are the expectation of SHAP values of the model loss (Section 3.2). It also provides a thorough analysis of the properties of SAGE values, and proposes an efficient approximation algorithm that calculates global importance directly instead of by averaging many local explanations.

Table 1 compares a large number of feature importance methods. The table is separated into four groups. The first contains methods that are not additive importance measures, and the remaining three correspond to the categories outlined in Section 2.3. Only our method, SAGE, accounts for complex feature interactions while remaining tractable.

5. Experiments

We now demonstrate and evaluate the use of SAGE for analyzing feature importance in eight datasets. To conserve space, we focus on results for only two datasets in the main text, and place the remaining results in Appendices H-I. Our code is available online.[1]

For the main text experiments, we performed MNIST digit recognition (LeCun & Cortes, 2010) and breast cancer (BRCA) subtype classification from gene microarray data (Tomczak et al., 2015). MNIST has 784 pixels and 70,000 samples, and the microarray data has 17,814 genes, 556 samples, and 4 BRCA subtypes. To avoid overfitting, we analyzed a subset of only 50 genes. Both datasets offer the opportunity for quantitative and qualitative evaluation. The remaining six data sets were from the UCI repository (Dua & Graff, 2017) and are described in Appendix I.

First, we trained prediction models for each dataset. We used a multi-layer perceptron (MLP) for MNIST and logistic regression for BRCA classification, as well as MLPs, decision forest models and support vector machines for the other datasets. We then calculated feature importance using several competing methods. We estimated SAGE values using $m \in \{32, 128, 512\}$ inner loop samples and with n increasing until the point of convergence. For baseline methods, we ran permutation tests until results converged; we performed feature ablations; we used the mean importance method; and we trained univariate prediction models to assess bivariate associations.

Visual inspection of MNIST feature importance (Figure 1 top) shows that the important features are generally located near the center. SAGE assigns the highest importance near the center, with scores decreasing with distance from the center. Feature ablation assigns meaningless values because removing single features has no impact on model performance. Mean importance and permutation tests erroneously assign negative (red) importance to some pixels, including some near the center, while the univariate predictors assign

[1] https://github.com/icc2115/sage

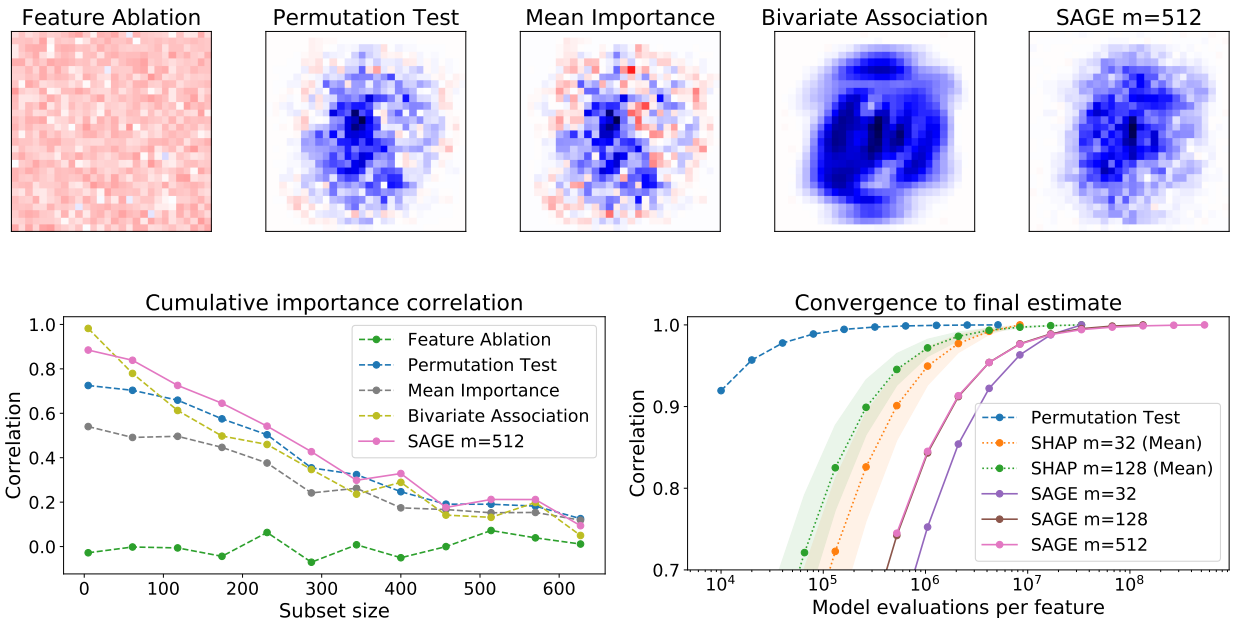

Figure 1. SAGE evaluation on MNIST digit recognition. Top: importance values from five methods, with positive values in blue and negative values in red. Bottom left: correlation of cumulative importance with performance of feature subsets (higher is better). Bottom right: convergence of importance estimators.

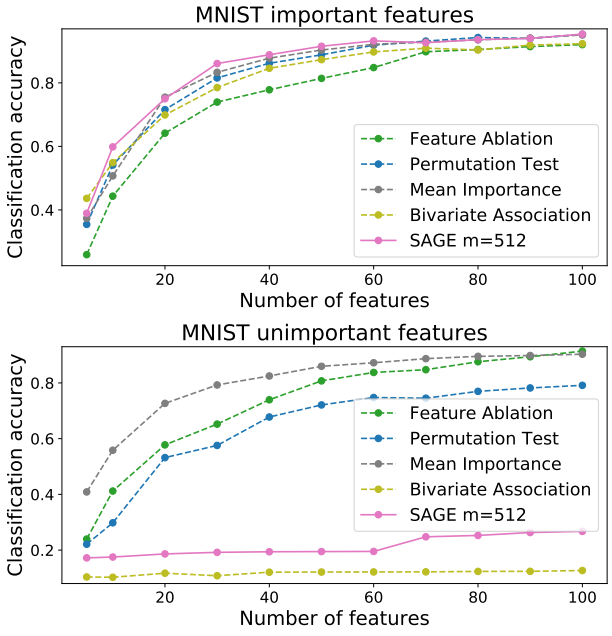

Figure 2. MNIST performance with subsets of the most important features (top) and least important features (bottom). Important features should provide high accuracy, while unimportant ones should provide low accuracy.

importance that is too uniform. Beyond these observations, differences are best analyzed through quantitative metrics.

Figure 3 (top) shows that genes previously known to be associated with BRCA receive the highest SAGE values. The most important gene (BCL11A) has a known association with a particularly aggressive form of BRCA (Khaled et al., 2015), and both BRCA1 and BRCA2 are also highly ranked. Among the important genes not associated with BRCA, some have documented associations with other cancers (e.g., SLC22A1, PDLIM4).

Next, we evaluated the feature importance values with quantitative experiments. To examine whether $u(S) = \phi_0 + \sum_{i \in S} \phi_i$ was a good proxy for $v(S)$, as it should be for all additive importance measures (Section 2), we randomly selected many subsets of features for different subset sizes $|S|$ (with linear spacing) and trained separate models for each subset S to approximate $v(S)$. We then measured the correlation between $u(S)$ with the test loss of the corresponding models. Intuitively, this experiment measures whether the cumulative importance $u(S)$ has a relationship with the amount of predictive power of X^S.

The results (Figures 1, 3 bottom left) show that SAGE is either the best or near best for all $|S|$. We attribute this to the fact that SAGE values attempt to model v_f accurately everywhere in $\mathcal{P}(D)$ (Section 3.3). Bivariate association provides

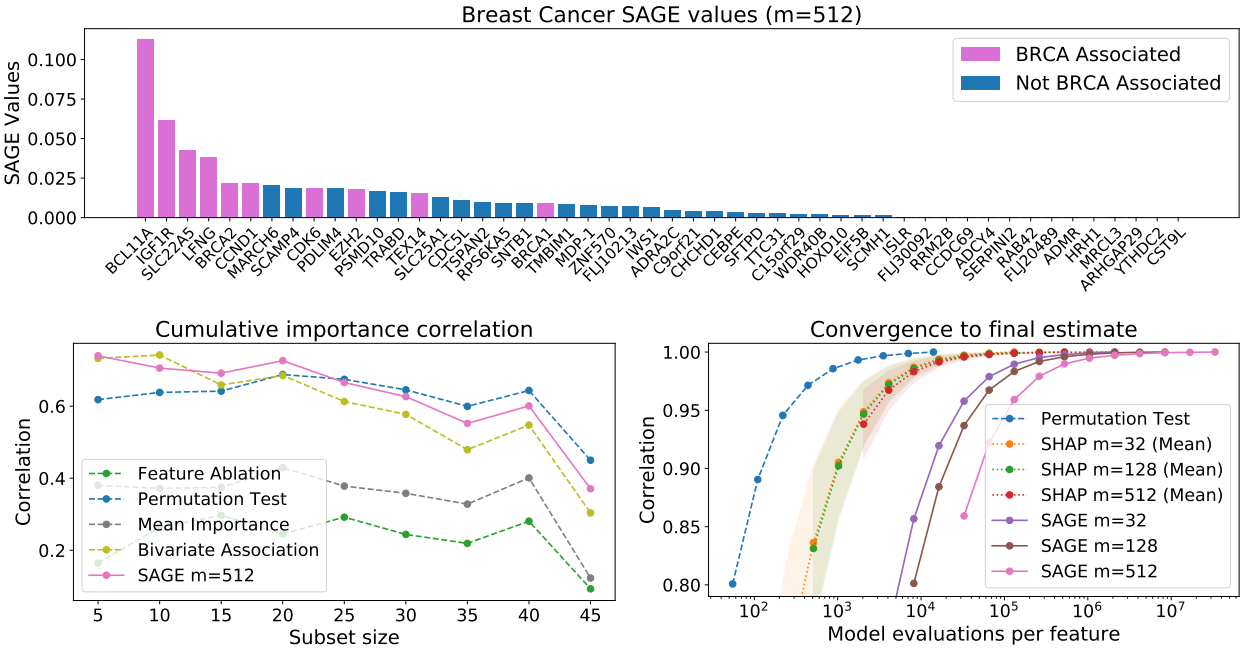

Figure 3. SAGE evaluation on BRCA subtype classification. Top: importance values from SAGE with $m = 512$. Bottom left: correlation of cumulative importance with performance of feature subsets (higher is better). Bottom right: convergence of importance estimators.

a good proxy for small $|S|$, and permutation tests are effective for larger $|S|$, while feature ablation fails because of the significant redundancy in both datasets. The pattern of SAGE having the highest correlation for most values of $|S|$ was replicated across the remaining six datasets.

We then trained models with the most and least important features, intending to achieve high (low) performance with the most (least) important ones. The results for MNIST (Figure 2) show that SAGE is highly effective at both tasks, unlike the baselines, because it is not biased towards either. In contrast, permutation tests are effective only for identifying features with significant signal, while bivariate association is best suited for identifying features that contain no signal. An identical result is replicated on BRCA and the remaining six datasets (see Appendix I).

Finally, since SAGE values are costly to calculate, we examined the number of model evaluations necessary for convergence. We compared SAGE to permutation tests, which may not converge until they are run many times, and to SHAP values of the loss, which are calculated for individual predictions (Lundberg et al., 2020). Here, SHAP values were estimated for multiple instances (32 for MNIST, 128 for BRCA) using a single-sample variant of Algorithm 1.

The results (Figures 1, 3 bottom right) show the mean correlation of intermediate estimates with the fully converged estimates, along with one standard deviation confidence intervals for SHAP. For these two datasets, SAGE takes roughly two orders of magnitude more model evaluations than permutation tests to converge. Compared to SHAP, SAGE takes about one order of magnitude more model evaluations; however, this means that a global explanation through SAGE can be calculated at the cost of only $\approx$10 local explanations, which is very efficient. In Appendix I, we compare the run-time of SHAP and SAGE across all eight datasets. SAGE proves to be a much more tractable method to get feature importance scores because many SHAP values (i.e., for the entire dataset) would need to be averaged to obtain the same values. Although SAGE requires many model evaluations, Algorithm 1 can be highly parallelized.

6. Conclusion

We presented a new framework of additive importance measures to unify a large body of work on quantifying global feature importance. We also proposed a model-agnostic importance measure that accounts for complex feature interactions, SAGE, which satisfies many desirable and intuitive properties. Our quantitative experiments show that SAGE values are more representative of feature contributions than importance values assigned by several baseline methods. Our future work will focus on efficient estimation of SAGE values, as well as approximations that properly model the conditional distributions of held out features.

References

Bengtson, E. and Roth, D. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 294–303, 2008.

Benoumechiara, N. and Elie-Dit-Cosaque, K. Shapley effects for sensitivity analysis with dependent inputs: bootstrap and kriging-based algorithms. *ESAIM: Proceedings and Surveys*, 65:266–293, 2019.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Charnes, A., Golany, B., Keane, M., and Rousseau, J. Extremal principle solutions of games in characteristic function form: core, chebychev and shapley value generalizations. In *Econometrics of Planning and Efficiency*, pp. 123–133. Springer, 1988.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009.

Datta, A., Sen, S., and Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 598–617. IEEE, 2016.

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.-J., Sandhu, S., Guppy, K. H., Lee, S., and Froelicher, V. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5): 304–310, 1989.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Fanaee-T, H. and Gama, J. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2-3):113–127, 2014.

Friedman, J., Hastie, T., and Tibshirani, R. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, 2001.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

Horel, E., Mison, V., Xiong, T., Giesecke, K., and Mangu, L. Sensitivity based neural networks explanations. *arXiv preprint arXiv:1812.01029*, 2018.

Janzing, D., Minorics, L., and Blöbaum, P. Feature relevance quantification in explainable ai: A causality problem. *arXiv preprint arXiv:1910.13413*, 2019.

Khaled, W. T., Lee, S. C., Stingl, J., Chen, X., Ali, H. R., Rueda, O. M., Hadi, F., Wang, J., Yu, Y., Chin, S.-F., et al. Bcl11a is a triple-negative breast cancer gene with critical functions in stem and progenitor cells. *Nature Communications*, 6(1):1–10, 2015.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lipovetsky, S. and Conklin, M. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

Liu, Y.-Z., Pei, Y.-F., Liu, J.-F., Yang, F., Guo, Y., Zhang, L., Liu, X.-G., Yan, H., Wang, L., Zhang, Y.-P., et al. Powerful bivariate genome-wide association analyses suggest the sox6 gene influencing both obesity and osteoporosis phenotypes in males. *PloS One*, 4(8), 2009.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020. ISSN 2522-5839. doi: 10.1038/s42256-019-0138-9. URL https://doi.org/10.1038/s42256-019-0138-9.

Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Owen, A. B. Sobol' indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014.

Owen, A. B. and Prieur, C. On shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002, 2017.

Pearl, J. *Causality*. Cambridge university press, 2009.

Ribeiro, M. T., Singh, S., and Guestrin, C. "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*, pp. 1135–1144, 2016.

Sakar, C. O., Polat, S. O., Katircioglu, M., and Kastro, Y. Real-time prediction of online shoppers purchasing intention using multilayer perceptron and lstm recurrent neural networks. *Neural Computing and Applications*, 31 (10):6893–6908, 2019.

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M. *Sensitivity Analysis in Practice: a Guide to Assessing Scientific Models*, volume 1. Wiley Online Library, 2004.

Setiono, R. and Liu, H. Neural-network feature selector. *IEEE Transactions on Neural Networks*, 8(3):654–662, 1997.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Song, E., Nelson, B. L., and Staum, J. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1): 1060–1083, 2016.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307, 2008.

Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3): 647–665, 2014.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328. JMLR. org, 2017.

Tomczak, K., Czerwińska, P., and Wiznerowicz, M. The cancer genome atlas (TCGA): an immeasurable source of knowledge. *Contemporary Oncology*, 19(1A):A68, 2015.

A. Convention for Handling Missing Features

We require a convention for making predictions with arbitrary subsets of features $S \subseteq D$ in order to probe how a model f performs when deprived of certain features. We therefore define the restricted model f_S as

$$f_S(x^S) = \mathbb{E}\big[f(X) \;\big|\; X^S = x^S\big].$$

Although we use an approximation in practice (Section 3.4), we have several reasons for defining SAGE using this convention. The reasons are: 1) the model f_S is as close as possible to the full model f in expectation; 2) the convention f_S yields connections with intrinsic properties of the data distribution, such as mutual information; and 3) alternative conventions often involve evaluating the model off the manifold of real data examples. We elaborate on each of these reasons below.

Consider how to measure the deviation between a model f on all features X and a model g on a subset of features X^S. We consider this separately for regression tasks and classification tasks. For a regression model f that makes prediction in $\mathbb{R}^p$, a natural way to determine how much its prediction given x differs from that of g is with the squared Euclidean distance $\big|\big|f(x) - g(x^S)\big|\big|^2$. The mean squared deviation between f and g is:

$$\begin{aligned}
&\mathbb{E}_X\Big[\big|\big|f(X) - g(X^S)\big|\big|^2\Big] \\
&= \mathbb{E}_X\Big[\big|\big|f(X) - \mathbb{E}\big[f(X) \;\big|\; X^S\big]\big|\big|^2\Big] \\
&\quad + \underbrace{\mathbb{E}_{X^S}\Big[\big|\big|\mathbb{E}\big[f(X) \;\big|\; X^S\big] - g(X^S)\big|\big|^2\Big]}_{\substack{\text{Mean squared deviation between} \\ g(X^S) \text{ and } \mathbb{E}\big[f(X) \;\big|\; X^S\big]}}
\end{aligned}$$

It is clear from the above that the model $g(x^S) = \mathbb{E}\big[f(X) \;\big|\; X^S = x^S\big]$ deviates least from f on average.

Next, consider a classification model f that outputs probabilities for a p-class categorical variable. To be explicit that the prediction is a vector of probabilities, we denote the model output as $f(y|x)$, where we have $f(i|x) \geq 0$ for $i = 1, 2, \ldots, p$ and $\sum_{i=1}^p f(i|x) = 1$. The Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}\big(f(y|x) \;||\; g(y|x^S)\big)$ is a natural way to measure the deviation of the predictions from g and f. Their mean deviation can then be expressed as:

$$\begin{aligned}
&\mathbb{E}_X\Big[D_{\mathrm{KL}}\big(f(y|X) \;||\; g(y|X^S)\big)\Big] \\
&= \mathbb{E}_X\Big[\mathbb{E}_{y\sim f(y|X)}\big[-\log g(y|X^S)\big]\Big] - \mathbb{E}_X\Big[H\big(f(y|X)\big)\Big] \\
&= \mathbb{E}_{X^S}\Big[\mathbb{E}_{y\sim \mathbb{E}[f(y|X)\;|\;X^S]}\big[-\log g(y|X^S)\big]\Big] \\
&\quad - \mathbb{E}_X\Big[H\big(f(y|X)\big)\Big] \\
&= \mathbb{E}_{X^S}\Big[H\big(\mathbb{E}\big[f(y|X) \;\big|\; X^S\big]\big)\Big] - \mathbb{E}_X\Big[H\big(f(y|X)\big)\Big] \\
&\quad + \underbrace{\mathbb{E}_{X^S}\Big[D_{\mathrm{KL}}\big(\mathbb{E}\big[f(y|X) \;\big|\; X^S\big] \;||\; g(y|X^S)\big)\Big]}_{\substack{\text{Mean KL divergence between} \\ g(y|X^S) \text{ and } \mathbb{E}\big[f(y|X) \;\big|\; X^S\big]}}
\end{aligned}$$

It is clear from this way of rewriting the KL divergence that the model $g(y|x^S) = \mathbb{E}\big[f(y|X) \;\big|\; x^S\big]$ is closest to f in expectation. These derivations show that in both the regression and classification cases, our convention for f_S yields the model that is closest to f on average. We argue that when analyzing performance differences when f is deprived of certain features, it is most conservative to use a convention for restricted models f_S that is as faithful to the full model f as possible; to do otherwise may result in inflated losses in model performance.

Handling missing features with f_S yields connections with intrinsic properties of the data distribution, such as mutual information and conditional variance (Section C). That happens because when our convention is applied to an optimal model (e.g., the Bayes classifier), it preserves the model's optimality. For example, the Bayes classifier $f^*(x) = p(y|X = x)$ becomes the Bayes classifier $f_S^*(x^S) = p(y|X^S = x^S)$, and the conditional expectation $f^*(x) = \mathbb{E}[Y|X = x]$ becomes the conditional expectation $f_S^*(x^S) = \mathbb{E}[Y|X^S = x^S]$. Our definition of f_S is the unique convention for which this holds.

Finally, for feature values x^S that have support under $p(x^S)$, the convention f_S only considers values of $x^{\bar{S}}$ such that $x = (x^S, x^{\bar{S}})$ has support under the data distribution $p(x)$. That property is a benefit of handling missing features using their conditional distribution $p(x^{\bar{S}}|X^S = x^S)$. By contrast, other choices of conventions may involve implausible combinations of features. As an example, one alternative is to intervene on the observed features by computing

$$\mathbb{E}\big[f(X) \;\big|\; \mathrm{do}(X^S = x^S)\big],$$

where we use the notation of Judea Pearl's do-calculus (Pearl, 2009). This effectively calculates the mean prediction when the missing features are drawn from their joint marginal distribution $p(x^{\bar{S}})$, which is what we do in practice (Section 3.4). Unfortunately, this breaks feature dependen-

cies and may result in combinations of values $(x^S, x^{\bar{S}})$ that are off-manifold (e.g., if there are two perfectly correlated features and one is removed). We view this as an undesirable property, and encourage work that removes the need for this approximation in practice.

Another option would be to use the mean prediction when the missing features are drawn from the product of their marginal distributions, as in the Quantitative Input Influence method (Datta et al., 2016). This convention is even more likely to result in off-manifold examples, because in addition to breaking dependencies between X^S and $X^{\bar{S}}$, it also breaks dependencies within $X^{\bar{S}}$.

B. Model-Based and Universal Predictive Power

In the main text, we use two set functions to represent different notions of predictive power. The function v represents *universal predictive power* and quantifies the amount of signal that can hypothetically be derived from a set of features X^S. It is defined as

$$v(S) = \min_{\hat{y}} \; \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \min_{g} \; \mathbb{E}\Big[\ell\big(g(X^S), Y\big)\Big].$$

By contrast, the function v_f represents a *model-based* notion of predictive power, and it quantifies how much signal f derives from a given set of features. It is defined as

$$v_f(S) = \mathbb{E}\Big[\ell\big(f_\varnothing(X^\varnothing), Y\big)\Big] - \mathbb{E}\Big[\ell\big(f_S(X^S), Y\big)\Big].$$

The two quantities are different, but related. To estimate $v(S)$, a natural approach would be to train a model using X^S, learn the optimal constant prediction $\hat{y}$, and then use the performance of those models as plug-in estimators for the two terms in $v(S)$. $v_f(S)$ can be viewed as an single-model approximation to this, where, instead of training a model from scratch on X^S, we obtain the model via an existing model f trained using all features.

Under certain circumstances when the model f^* is optimal, we see that v and v_{f^*} coincide exactly for all $S \subseteq D$. Two simple cases where this holds are 1) for a regression task that uses the model $f^*(x) = \mathbb{E}[Y|X = x]$ and mean squared error (MSE) loss, and 2) for a classification task that uses the Bayes classifier $f^*(x) = p(y|X = x)$ and cross entropy loss. We show equality in the first case as follows:

$$\begin{aligned} v_{f^*}(S) &= \mathbb{E}\Big[\big|\big|Y - f^*_\varnothing(X^\varnothing)\big|\big|^2\Big] - \mathbb{E}\Big[\big|\big|Y - f^*_S(X^S)\big|\big|^2\Big] \\ &= \mathbb{E}\Big[\big|\big|Y - \mathbb{E}[Y]\big|\big|^2\Big] - \mathbb{E}\Big[\big|\big|Y - \mathbb{E}[Y|X^S = x^S]\big|\big|^2\Big] \\ &= v(S) \end{aligned}$$

Similarly, we show equality in the second case as follows:

$$\begin{aligned} v_{f^*}(S) &= \mathbb{E}\big[-\log f^*_\varnothing(Y|X^\varnothing)\big] - \mathbb{E}\big[-\log f^*_S(Y|X^S)\big] \\ &= \mathbb{E}\big[-\log p(Y)\big] - \mathbb{E}\big[-\log p(Y|X^S = x^S)\big] \\ &= v(S) \end{aligned}$$

Equality between v and v_{f^*} holds for optimal models f^* with a specific class of loss functions: it holds for loss functions ℓ with the property that an optimal model f^* for X yields an optimal model f^*_S for X^S. Besides MSE and cross entropy loss, this property holds for all *strictly proper scoring rules*, which are defined by the characteristic that the unique optimal model under a strictly proper scoring rule is the probabilistic forecast (e.g., $f^*(x) = p(y|X = x)$) (Gneiting & Raftery, 2007).

C. SAGE Properties with Optimal Models

C.1. Properties with Bayes Classifier

Here, we derive the properties of SAGE when it is applied to the Bayes classifier with cross entropy loss. We derive the claim from scratch, beginning with a proof that the Bayes classifier is optimal. To be explicit that the prediction is a vector of probabilities, we denote the model output as $f(y|x)$, where we have $f(i|x) \geq 0$ for $i = 1, 2, \ldots, p$ and $\sum_{i=1}^p f(i|x) = 1$. We also use H to denote entropy, and I to denote mutual information.

For a classification model trained with cross entropy loss, we decompose the true risk as follows to reveal the optimal classifier:

$$\begin{aligned} &\mathbb{E}\big[\ell\big(f(y|X), Y\big)\big] \\ &= \mathbb{E}\big[-\log f(Y|X)\big] \\ &= \mathbb{E}_X\Big[\mathbb{E}_{Y|X}\big[-\log f(Y|X)\big]\Big] \\ &= \mathbb{E}_X\big[D_{\mathrm{KL}}\big(p(y|X) \;||\; f(y|X)\big) + H(Y|X)\big] \end{aligned}$$

The entropy term inside the expectation is constant, and only the KL divergence term depends on f. The optimal prediction model is therefore the Bayes classifier $f^*(x) = p(y|X = x)$. We now consider the application of SAGE to the model f^*. The restricted models f^*_S are the following:

$$
\begin{aligned}
f_S^*(y|x^S) &= \mathbb{E}\big[f^*(y|X) \;\big|\; X^S = x^S\big] \\
&= \mathbb{E}\big[p(y|X) \;\big|\; X^S = x^S\big] \\
&= p(y|X^S = x^S)
\end{aligned}
$$

The risk incurred by the restricted model f_S^* is then:

$$
\begin{aligned}
\mathbb{E}\big[\ell\big(f_S^*(y|X^S), Y\big)\big] &= \mathbb{E}\big[-\log f_S^*(Y|X^S)\big] \\
&= \mathbb{E}\big[-\log p(Y|X^S)\big] \\
&= H(Y|X^S)
\end{aligned}
$$

We can now see that the cooperative game v_{f^*} is:

$$
\begin{aligned}
v_{f^*}(S) &= \mathbb{E}\big[\ell\big(f_\varnothing^*(X^\varnothing), Y\big)\big] - \mathbb{E}\big[\ell\big(f_S^*(X^S), Y\big)\big] \\
&= H(Y) - H(Y \mid X^S) \\
&= I(Y; X^S)
\end{aligned}
$$

In the expression for Shapley values, the weighted summation has terms of the following form:

$$
\begin{aligned}
v(S \cup \{i\}) - v(S) &= I(Y; X^{S\cup\{i\}}) - I(Y; X^S) \\
&= H(Y \mid X^S) - H(Y \mid X^{S\cup\{i\}}) \\
&= I(Y; X^i \;\big|\; X^S)
\end{aligned}
$$

This completes the derivation of Eq. 11 of the main text, because we see that the Shapley values are equal to

$$
\phi_i(v_{f^*}) = \frac{1}{d} \sum_{S \subseteq D\setminus\{i\}} \binom{d-1}{|S|}^{-1} I(Y; X^i \mid X^S).
$$

C.2. Properties with Conditional Expectation

We now show a similar result for optimal regression models when using mean squared error (MSE) loss. We assume that the predictions are in $\mathbb{R}$, although a similar result holds for predictions in $\mathbb{R}^p$. We first decompose the true risk to determine the optimal model:

$$
\begin{aligned}
&\mathbb{E}\big[\ell\big(f(X), Y\big)\big] \\
&= \mathbb{E}\big[\big(f(X) - Y\big)^2\big] \\
&= \mathbb{E}\big[\big(f(X) - \mathbb{E}[Y|X]\big)^2 + \big(\mathbb{E}[Y|X] - Y\big)^2\big]
\end{aligned}
$$

It is clear from this way of rewriting the MSE that the conditional expectation function $f^*(x) = \mathbb{E}[Y|X = x]$ is optimal. We now consider the application of SAGE to the model f^*. The restricted models f_S^* are the following:

$$
\begin{aligned}
f_S^*(x^S) &= \mathbb{E}\big[f^*(X) \;\big|\; X^S = x^S\big] \\
&= \mathbb{E}\big[\mathbb{E}[Y|X] \;\big|\; X^S = x^S\big] \\
&= \mathbb{E}[Y|X^S = x^S]
\end{aligned}
$$

The last line follows from the law of iterated expectations. The risk incurred by the restricted model f_S^* is then:

$$
\begin{aligned}
\mathbb{E}\big[\ell\big(f_S^*(X^S), Y\big)\big] &= \mathbb{E}\big[\big(\mathbb{E}[Y \mid X^S] - Y\big)^2\big] \\
&= \mathbb{E}[\mathrm{Var}(Y|X^S)]
\end{aligned}
$$

We now see that the cooperative game v_{f^*} is:

$$
\begin{aligned}
v_{f^*}(S) &= \mathrm{Var}(Y) - \mathbb{E}\big[\mathrm{Var}(Y|X^S)\big] \\
&= \mathrm{Var}\big(\mathbb{E}[Y|X]\big)
\end{aligned}
$$

The last line follows from the law of total variance. The difference terms in the Shapley summation are the following:

$$
\begin{aligned}
&v_{f^*}(S \cup \{i\}) - v_{f^*}(S) \\
&= \mathbb{E}\big[\mathrm{Var}(Y|X^S)\big] - \mathbb{E}\big[\mathrm{Var}(Y|X^{S\cup\{i\}})\big] \\
&= \mathbb{E}_{X^S}\Big[\mathrm{Var}\big(\mathbb{E}[Y|X^S, X^i] \mid X^S\big)\Big]
\end{aligned}
$$

The last line also follows from the law of total variance. Intuitively, these terms quantify the average amount of variation left in the random variable $\mathbb{E}[Y \mid X^S, X^i]$ from X^i being unknown, but distributed according to $p(x^i|X^S)$. If the amount of variation is high, then i contains significant incremental information about Y. The above expression is conceptually analogous to $I(Y; X^i \mid X^S)$ from the classification case.

Finally, we see that the SAGE values are equal to

$$
\phi_i(v_{f^*}) = \frac{1}{d} \sum_{S \subseteq D\setminus\{i\}} \binom{d-1}{|S|}^{-1} \mathbb{E}\Big[\mathrm{Var}\big(\mathbb{E}[Y|X^S, X^i]\big|X^S\big)\Big].
$$

D. SAGE Properties with Marginal Sampling

Here, we describe how the properties of SAGE change when Algorithm 1 is run with sampling from the marginal distribution $q(x^{\bar{S}}|X^S = x^S) = p(x^{\bar{S}})$. Sampling from the

marginal instead of the conditional changes the underlying cooperative game, so that we no longer estimate the Shapley values $\phi_i(v_f)$, but rather we estimate the Shapley values of a different game.

In the inner loop of Algorithm 1, the Monte Carlo approximation is an approximation of $\mathbb{E}\big[f(X) \mid \text{do}(X^S = x^S)\big]$, where we use the notation from Judea Pearl's do-calculus (Pearl, 2009). The notation means that X is drawn from the marginal distribution $p(x)$ and the features in S are then changed to the values x^S. We adopt the notation $\tilde{f}_S$ to denote an alternative restricted model, which is defined as

$$\tilde{f}_S(x^S) = \mathbb{E}[f(X) \mid \text{do}(X^S = x^S)].$$

We then adopt the notation $\tilde{v}_f$ to denote the new cooperative game using $\tilde{f}_S$, which is

$$\tilde{v}_f(S) = \mathbb{E}\Big[\ell\big(\tilde{f}_\varnothing(X^\varnothing), Y\big)\Big] - \mathbb{E}\Big[\ell\big(\tilde{f}_S(X^S), Y\big)\Big].$$

Sampling from the marginal distribution in practice means that we estimate the Shapley values $\phi_i(\tilde{v}_f)$. We now provide the properties of the importance values $\phi_i(\tilde{v}_f)$, which differ from the properties described in Section 3.2.

1. Due to the efficiency property, the scores satisfy $\sum_{i=1}^d \phi_i(\tilde{v}_f) = \tilde{v}_f(S) - \tilde{v}_f(\varnothing)$. The alternative restricted model $\tilde{f}_S$ is identical to f_S with all features included or excluded, i.e., we have $\tilde{f}_\varnothing(x^\varnothing) = f_\varnothing(x^\varnothing)$ and $\tilde{f}_D(x) = f_D(x)$ for all x. We therefore also have $\tilde{v}_f(S) = v_f(S)$ and $\tilde{v}_f(\varnothing) = v_f(\varnothing)$, so that the Shapley values sum to the same total, $\sum_{i=1}^d \phi_i(\tilde{v}_f) = \sum_{i=1}^d \phi_i(v_f)$.

2. Due to the symmetry property, we have $\phi_i(\tilde{v}_f) = \phi_j(\tilde{v}_f)$ when $\tilde{v}_f(S \cup \{i\}) = \tilde{v}_f(S \cup \{j\})$ for all S. That holds if we have $\tilde{f}_{S\cup\{i\}}(x^{S\cup\{i\}}) = \tilde{f}_{S\cup\{j\}}(x^{S\cup\{j\}})$ for all (S, x). Given our definition of $\tilde{f}_S$, there is no convenient sufficient condition for this to hold. Unlike in the original formulation, perfectly correlated features do not receive equal importance.

3. Due to the linearity property, the values $\phi_i(\tilde{v}_f)$ are the expectation of per-instance loss SHAP values computed with sampling from the marginal distribution. If we define the cooperative game $\tilde{v}_{f,x,y}$ as

   $$\tilde{v}_{f,x,y}(S) = \ell\big(\tilde{f}_\varnothing(x^\varnothing), y\big) - \ell\big(\tilde{f}_S(x^S), y\big),$$

   then we have $\phi_i(\tilde{v}_f) = \mathbb{E}_{XY}\big[\phi_i(\tilde{v}_{f,X,Y})\big]$. These are loss SHAP values that are computed by algorithms that assume feature independence (Lundberg & Lee, 2017).

4. Due to the monotonicity property, if we have two response variables Y, Y' with models f, f', and we have $\tilde{v}_f(S \cup \{i\}) - \tilde{v}_f(S) \geq \tilde{v}_{f'}(S \cup \{i\}) - \tilde{v}_{f'}(S)$ for all S, then we have $\phi_i(\tilde{v}_f) \geq \phi_i(\tilde{v}_{f'})$. This says that if X^i contributes more predictive power to Y than to Y', then it receives more importance for Y.

5. Due to the dummy property, we have $\phi_i(\tilde{v}_f) = 0$ if $\tilde{f}_S(x^S) = \tilde{f}_{S\cup\{i\}}(x^{S\cup\{i\}})$ for all (S, x). A sufficient condition for this to hold is that the model f is invariant to X^i. That means that the value $\phi_i(\tilde{v}_f)$ for a sensitive attribute X^i (e.g., race) may be zero even if the model depends on correlated features.

6. As in the original formulation, the values $\phi_i(\tilde{v}_f)$ are invariant to invertible mappings of the features.

One elegant aspect of the original formulation of SAGE is the connection with intrinsic properties of the data distribution when applied to optimal models. Under the formulation with sampling from the marginal distribution, we lose these connections because the restricted models $\tilde{f}_S^*$ based on optimal models f^* (e.g., the Bayes classifier) are no longer optimal for X^S. In this sense, the alternative formulation of SAGE is ill-suited for understanding properties of the data distribution.

However, one recent work has advocated for advantages of sampling from the marginal distribution in SHAP (Janzing et al., 2019). The most appealing property is that features which are not used by the model always receive zero attribution. We showed above that the same holds for SAGE. This formulation satisfies the seemingly obvious property that unused features should receive zero importance, although that property conflicts with the equally intuitive notion that a model interpretation tool should uncover the use of sensitive attributes, even when they are used indirectly.

E. Proofs

The two results from Section 3.4 of the main text are restated and proved below.

Theorem 1. *The SAGE value estimates $\hat{\phi}_i(v_f)$ from Algorithm 1 converge to the correct values $\phi_i(v_f)$ when run with $n \to \infty$, $m \to \infty$, with an arbitrarily large dataset $\big\{(x_i, y_i)\big\}_{i=1}^N$, and with sampling from the correct conditional distribution $q(x^{\bar{S}}|X^S = x^S) = p(x^{\bar{S}}|X^S = x^S)$.*

Proof. At a high level, the algorithm has an outer loop that contributes one sample to each of the SAGE value estimates $\hat{\phi}_i(v_f)$. Each estimate can be interpreted as a sample mean that converges to its expectation as n becomes large. Our proof proceeds by considering the value of the expectation under the assumptions that $m \to \infty$ and $q(x^{\bar{S}}|X^S = x^S) = p(x^{\bar{S}}|X^S = x^S)$.

Each estimate $\hat{\phi}_i(v_f)$ is the average of many samples of the random variable $\Delta^{i,m}_{X,Y,S}$ which we define here as

$$\Delta^{i,m}_{x,y,S} = \ell\big(\frac{1}{m}\sum_{k=1}^{m} f(x^S, x_k^{\bar{S}}), y\big) - \ell\big(\frac{1}{m}\sum_{l=1}^{m} f(x^{S\cup\{i\}}, x_l^{\bar{S}\setminus\{i\}}), y\big). \tag{13}$$

Specifically, we have

$$\hat{\phi}_i(v_f) = \frac{1}{n}\sum_{j=1}^{n} \Delta^{i,m}_{x_j,y_j,S_j} \tag{14}$$

where i, m are fixed and x_j, y_j and S_j are determined by each iteration of Algorithm 1. Even for fixed i, m, x, y, S, note that $\Delta^{i,m}_{x,y,S}$ is a random variable because each $x_k^{\bar{S}}$ and $x_l^{\bar{S}\setminus\{i\}}$ are independent samples from the distributions $q(x^{\bar{S}}|X^S = x^S)$ and $q(x^{\bar{S}\setminus\{i\}}|X^{S\cup\{i\}} = x^{S\cup\{i\}})$ respectively. We begin by analyzing the random variable $\Delta^{i,m}_{x,y,S}$ and what it converges to as $m \to \infty$.

Consider the first term in Eq. 13. The mean prediction $\frac{1}{m}\sum_{k=1}^{m} f(x^S, x_k^{\bar{S}})$ provides a Monte Carlo approximation of

$$\mathbb{E}_{q(x^{\bar{S}}|X^S=x^S)}[f(x^S, X^{\bar{S}})].$$

We assume that samples are from the true conditional distribution $p(x^{\bar{S}}|X^S = x^S)$, so the average prediction $\frac{1}{m}\sum_{k=1}^{m} f(x^S, x_k^{\bar{S}})$ is in fact an approximation of $f_S(x^S)$. When we let $m \to \infty$ the law of large numbers says that

$$\frac{1}{m}\sum_{k=1}^{m} f(x^S, x_k^{\bar{S}}) \xrightarrow{p} f_S(x^S), \tag{15}$$

where $\xrightarrow{p}$ denotes convergence in probability. By an identical argument for the second term in Eq. 13, because of sampling from $p(x^{\bar{S}\setminus\{i\}}|X^{S\cup\{i\}} = x^{S\cup\{i\}})$, we see that

$$\frac{1}{m}\sum_{l=1}^{m} f(x^{S\cup\{i\}}, x_l^{\bar{S}\setminus\{i\}}) \xrightarrow{p} f_{S\cup\{i\}}(x^{S\cup\{i\}}) \tag{16}$$

as $m \to \infty$. This lets us conclude that the random variable $\Delta^{i,m}_{x,y,S}$ converges as $m \to \infty$, with

$$\Delta^{i,m}_{x,y,S} \xrightarrow{p} \ell\big(f_S(x^S), y\big) - \ell\big(f_{S\cup\{i\}}(x^{S\cup\{i\}}), y\big). \tag{17}$$

With this result, we define $\Delta^{i}_{x,y,S} \equiv \lim_{m\to\infty} \Delta^{i,m}_{x,y,S}$. We now consider the fact that the SAGE estimates $\hat{\phi}_i(v_f)$ are the average of many samples $\Delta^{i,m}_{X,Y,S}$, or many samples $\Delta^{i}_{X,Y,S}$ in the limit $m \to \infty$. We will determine the expected value of $\hat{\phi}_i(v_f)$, and argue that it converges to this value as $n \to \infty$.

We first consider the distribution from which S is drawn implicitly. In Algorithm 1, S is determined by a permutation π of the feature indices, and it contains indices that are already included when we arrive at feature i. The number of preceding indices $|S|$ is uniformly distributed between 0 and $d-1$, and the preceding indices are chosen uniformly at random among the $\binom{d-1}{|S|}$ possible combinations. We can therefore write a probability mass function $p(S)$ for subsets S that may be included by the time when i is added,

$$p(S) = \frac{1}{d}\binom{d-1}{|S|}^{-1}.$$

When we take the expectation of $\Delta^{i}_{x,y,S}$ over S, we have

$$\begin{aligned}
&\mathbb{E}_{p(S)}\Big[\Delta^{i}_{x,y,S}\Big] \\
&= \mathbb{E}_{p(S)}\Big[\ell\big(f_S(x^S), y\big) - \ell\big(f_{S\cup\{i\}}(x^{S\cup\{i\}}), y\big)\Big] \\
&= \sum_{T\subseteq D\setminus\{i\}} \frac{1}{d}\binom{d-1}{|T|}^{-1}\Big(\ell\big(f_T(x^T), y\big) - \ell\big(f_{T\cup\{i\}}(x^{T\cup\{i\}}), y\big)\Big).
\end{aligned} \tag{18}$$

The expression above already resembles the Shapley value because of the weighted summation over subsets. We can then incorporate an expectation over (x, y) pairs drawn from the data distribution $p(x, y)$, and see that the Shapley value $\phi_i(v_f)$ arises naturally:

$$\begin{aligned}
&\mathbb{E}_{XY}\mathbb{E}_{p(S)}\Big[\Delta^{i}_{X,Y,S}\Big] \\
&= \mathbb{E}_{XY}\mathbb{E}_{p(S)}\Big[\ell\big(f_S(X^S), Y\big) - \ell\big(f_{S\cup\{i\}}(X^{S\cup\{i\}}), Y\big)\Big] \\
&= \mathbb{E}_{p(S)}\big[v_f(S\cup\{i\}) - v_f(S)\big] \\
&= \phi_i(v_f)
\end{aligned} \tag{19}$$

Finally, we invoke the law of large numbers again to conclude that in the limit of an arbitrarily large dataset $\big\{(x_i, y_i)\big\}_{i=1}^{N}$ drawn from $p(x, y)$, we have

$$\hat{\phi}_i(v_f) \xrightarrow{p} \phi_i(v_f) \tag{20}$$

as $n \to \infty, m \to \infty$.

In summary, our proof is the following:

$$\begin{aligned}
\hat{\phi}_i(v_f) &= \frac{1}{n}\sum_{j=1}^{n} \Delta^{i,m}_{x_j,y_j,S_j} \\
\Delta^{i}_{x,y,S} &\equiv \lim_{m\to\infty} \Delta^{i,m}_{x,y,S} \\
&= \ell\big(f_S(x^S), y\big) - \ell\big(f_{S\cup\{i\}}(x^{S\cup\{i\}}), y\big) \\
\lim_{m\to\infty} \hat{\phi}_i(v_f) &= \frac{1}{n}\sum_{j=1}^{n} \Delta^{i}_{x_j,y_j,S_j} \\
\lim_{n\to\infty}\lim_{m\to\infty} \hat{\phi}_i(v_f) &= \mathbb{E}_{XY}\mathbb{E}_{p(S)}\big[\Delta^{i}_{X,Y,S}\big] = \phi_i(v_f)
\end{aligned}$$

□

We now prove the second theorem.

Theorem 2. *The SAGE value estimates $\hat{\phi}_i(v_f)$ from Algorithm 1 have variance that reduces at the rate of $O(\frac{1}{n})$.*

Proof. At a high level, the algorithm has an outer loop that contributes one sample $\Delta^{i,m}_{X,Y,S}$ (Eq. 13) to each of the SAGE estimates $\hat{\phi}_i(v_f)$, where randomness arises from the sampling of x, y and S (see Eq. 14). Regardless of how $q(x^S|X^S = x^S)$ is chosen and the number of inner loop samples m, the central limit theorem says that as n becomes large, the sample mean $\hat{\phi}_i(v_f)$ converges in distribution to a Gaussian with mean $\mathbb{E}[\Delta^{i,m}_{X,Y,S}]$ and variance equal to

$$\frac{\mathrm{Var}(\Delta^{i,m}_{X,Y,S})}{n},$$

where both terms have randomness arising from the inner loop samples (Eq. 13) and also from X, Y and S. Although we do not have access to the numerator $\mathrm{Var}(\Delta^{i,m}_{X,Y,S})$, we can conclude that the variance of the estimates behaves as $O(\frac{1}{n})$.

□

Consider how we can use Theorem 2 to create a stopping criterion for Algorithm 1. While running the algorithm, we can keep track of an empirical estimate $\hat{\sigma}_i^2$ of $\mathrm{Var}(\Delta^{i,m}_{X,Y,S})$ so that the algorithm can stop when $\sqrt{\frac{\hat{\sigma}_i^2}{n}}$ falls below a threshold value t, or specifically when

$$\max_{i\in D} \sqrt{\frac{\hat{\sigma}_i^2}{n}} < t.$$

Since the SAGE values are roughly expected to be non-negative and to sum to $v_f(D) - v_f(\varnothing)$, so that the true SAGE values are roughly between 0 and $v_f(D) - v_f(\varnothing)$, a natural stopping value is perhaps $t = 0.01\big(v_f(D) - v_f(\varnothing)\big)$.

F. Function Sensitivity

Several recent papers considered the related question of sensitivity of functions to their various inputs (Owen, 2014; Song et al., 2016; Owen & Prieur, 2017). We provide a brief presentation of the problem in order to illustrate how it differs from our work.

For a scalar real-valued function f defined on multiple features $x = (x_1, x_2, \ldots, x_d)$, that work assigns sensitivity measures to each feature. This is done with a variance-based measure of the dependence of f on each feature, through an analysis of the following cooperative game

$$\begin{aligned}
w_f(S) &= \mathrm{Var}\big(f(X)\big) - \mathbb{E}\big[\mathrm{Var}(f(X) \mid X^S)\big] \\
&= \mathrm{Var}\big(\mathbb{E}\big[f(X) \mid X^S\big]\big). \qquad (21)
\end{aligned}$$

The Shapley values $\phi_i(w_f)$ serve as sensitivity measures for each feature $i = 1, 2, \ldots, d$. In the prior work, Owen (2014) connected this measure of feature importance to the two Sobol' indices, Owen & Prieur (2017) considered special cases with closed form solutions, and Song et al. (2016) provided a sampling-based approximation algorithm.

Our work considers the related problem of assessing feature importance for black-box machine learning models. In contrast with this work, we allow for a response variable Y that is jointly distributed with X, which is not necessarily in $\mathbb{R}$, and we consider how predictive each feature is of Y rather than of the model output $f(X)$. The work on function sensitivity is equivalent to an application of SAGE to the special case where $Y \equiv f(X)$, where the model output is real-valued $f(X) \in \mathbb{R}$, and where the loss ℓ is MSE loss.

For cases with a response variable $Y \in \mathbb{R}$, a natural question is whether there is a relationship between $\phi_i(v_f)$ and $\phi_i(w_f)$. The only case when these values coincide is when the loss is MSE and the model f is the conditional expectation, i.e., $f^*(x) = \mathbb{E}[Y|X = x]$. Equality of the Shapley values follows from equality of the cooperative games:

$$\begin{aligned}
w_{f^*}(S) &= \mathrm{Var}\big(\mathbb{E}[f^*(X) \mid X^S]\big) \\
&= \mathrm{Var}\big(\mathbb{E}[\mathbb{E}[Y \mid X] \mid X^S]\big) \\
&= \mathrm{Var}\big(\mathbb{E}[Y \mid X^S]\big) \\
&= \mathrm{Var}(Y) - \mathbb{E}[\mathrm{Var}(Y \mid X^S)] \\
&= \mathbb{E}\big[\big(Y - f^*_{\varnothing}(X^{\varnothing})\big)\big] - \mathbb{E}\big[\big(Y - f^*_S(X^S)\big)^2\big] \\
&= v_{f^*}(S)
\end{aligned}$$

The cooperative games are equal, so they have the same Shapley values. However, outside of this special case, we do not have sensitivity values $\phi_i(w_f)$ equal to SAGE values $\phi_i(v_f)$.

G. Summary of Additive Importance Measures

Table 2 provides a summary of the additive importance measures described in Section 2.3. For each method, we indicate which part of the subdomain of $\mathcal{P}(D)$ it prioritizes, and we show whether it approximates v for universal predictive power or v_f for model-based predictive power.

Remark that permutation tests, conditional permutation tests and mean importance assign similar scores. Each of these methods make different assumptions to approximate

$$
\begin{aligned}
& v_f(D \setminus \{i\}) - v_f(D) \\
&= \mathbb{E}\Big[\ell\big(f_{D\setminus\{i\}}(X^{D\setminus\{i\}}), Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] \\
&= \mathbb{E}_{X^{D\setminus\{i\}}Y}\Big[\ell\big(\mathbb{E}_{p(x^i|X^{D\setminus\{i\}})}[f(X^i, X^{D\setminus\{i\}})], Y\big)\Big] \\
&\quad - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big].
\end{aligned}
$$

Conditional permutation tests make the closest approximation, but they have the expectation over X^i outside the loss function instead of inside it. Permutation tests make an assumption of feature independence, sampling X^i from its marginal distribution $p(x^i)$ instead of from its conditional distribution $p(x^i|X^{D\setminus\{i\}} = x^{D\setminus\{i\}})$. And finally, mean importance makes a further assumption of model linearity, avoiding taking an expectation by simply using the marginal mean $\mathbb{E}[X^i]$.

H. Breast Cancer Feature Selection Results

As in the experiment with MNIST, we evaluated the different feature importance measures for the breast cancer (BRCA) data by training models with the most important and least important features. Due to the small size of the dataset and sensitivity of results to different splits, we evaluated the performance using leave-one-out cross validation, and trained separate models for each data point using all the other data points.

The results in Figure 4 show the same pattern as the MNIST data. SAGE strikes a balance by assigning feature importance values so that the most important features contain significant signal and usually outperform the most important features from the baselines, while the least important features contain minimal signal. The least important features identified by univariate prediction models contain even less signal, but the most important features do not perform as well.

I. Additional Datasets

Here, we show results from six additional datasets. Table 3 provides a summary of the dataset, including the size of the dataset, the nature of the prediction task, and the model used. We used a variety of model classes for these tasks, including multi-layer perceptrons (MLP), gradient boosting machines (GBM), random forests (RF), and support vector regression (SVR). We also describe each dataset briefly below.

- The Bank Marketing dataset is from the direct marketing campaigns of a Portuguese bank, and the task is to predict whether a call will be successful based on information about the customer (Moro et al., 2014).
- The Bike Rental dataset contains information from one hour time periods, including date, time and weather, and the task is to predict the number of bikes rented during each period (Fanaee-T & Gama, 2014).
- The German Credit Default dataset provides information about customers, and the task is to predict whether the customer has a high credit risk (Dua & Graff, 2017).
- The Heart Disease dataset contains information about patients, and we performed a binary classification of their disease status (Detrano et al., 1989). We used the subset of data from Cleveland Clinic, and we ignored a small number of patients with missing values.
- The Online Shopping Dataset contains information about users' behavior and the pages they visit, and the task is to predict whether they will make a purchase (Sakar et al., 2019).
- The Wine Quality dataset provides physiochemical properties from many different wines, and the task is to predict a numerical score for each wine's quality (Cortez et al., 2009). We examined only the white wines, for which there are more examples than the red wines.

Figures 5-10 show the results. As in the main text, we used feature ablations, permutation tests and univariate predictors as baselines for all tasks, and we used mean importance on tasks with only continuous features.

Qualitative examinations of the feature importance values (Figures 5-10 top) reveal that in most cases there are significant differences between the importance values assigned by SAGE and by the baselines. To demonstrate that SAGE assigns importance values in a more correct manner, we performed the same quantitative experiments as in the main text.

For each dataset, we replicated the experiment that measures the correlation between the cumulative importance of

Table 2. Summary of additive importance measures. *Approximates* indicates whether the method approximates v or v_f, to assess universal feature importance or model-based feature importance, respectively. *Importance values* indicates the values that are assigned in expectation, or as each method is run until convergence (in the case of permutation test and conditional permutation tests), with ϕ_0 indicating the value for which $u(S)$ closely approximates $v(S)$ or $v_f(S)$ in the specified subdomain. For Shapley Net Effects and SAGE $\phi_i(\cdot)$ denotes the Shapley value (see Eq. 9 of main text).

SUBDOMAIN	APPROXIMATES	METHOD	IMPORTANCE VALUES
$\{D\} \cup \{\{D \setminus \{i\}\} \mid i \in D\}$	v	Feature Ablation	$\phi_i = \mathbb{E}\Big[\ell\big(f_i(X^{D\setminus\{i\}}), Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big]$ $\phi_0 = \min_{\hat{y}} \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] - \sum_{i\in D} \phi_i$
	v_f	Permutation Test	$\phi_i = \mathbb{E}_{X^{D\setminus\{i\}}Y}\Big[\mathbb{E}_{p(x^i)}\Big[\ell\big(f(X^i, X^{D\setminus\{i\}}), Y\big)\Big]\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big]$ $\phi_0 = \min_{\hat{y}} \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] - \sum_{i\in D} \phi_i$
		Conditional Permutation Test	$\phi_i = \mathbb{E}_{X^{D\setminus\{i\}}Y}\Big[\mathbb{E}_{p(x^i \mid X^{D\setminus\{i\}})}\Big[\ell\big(f(X^i, X^{D\setminus\{i\}}), Y\big)\Big]\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big]$ $\phi_0 = \min_{\hat{y}} \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] - \sum_{i\in D} \phi_i$
		Mean Importance	$\phi_i = \mathbb{E}\Big[\ell\big(f(\mathbb{E}[X^i], X^{D\setminus\{i\}}), Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big]$ $\phi_0 = \min_{\hat{y}} \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(f(X), Y\big)\Big] - \sum_{i\in D} \phi_i$
$\{\varnothing\} \cup \{\{i\} \mid i \in D\}$	v	Univariate Predictors	$\phi_i = \min_{\hat{y}} \mathbb{E}\Big[\ell\big(\hat{y}, Y\big)\Big] - \mathbb{E}\Big[\ell\big(f_i(X^i), Y\big)\Big]$ $\phi_0 = 0$
		Squared Correlation	$\phi_i = \text{Corr}(X_i, Y_i)^2$ $\phi_0 = 0$
$\mathcal{P}(D)$	v	Shapley Net Effects	$\phi_i = \phi_i(v)$ (Shapley value) $\phi_0 = 0$
	v_f	SAGE	$\phi_i = \phi_i(v_f)$ (Shapley value) $\phi_0 = 0$

feature subsets $u(S) = \phi_0 + \sum_{i\in S} \phi_i$ and the performance of new models trained on X^S. Figures 5-10 (middle left) show that SAGE has the best, or near best correlation for most feature subsets sizes $|S|$. Feature ablations narrowly outperform SAGE on the Bank and Wine datasets, but performs very poorly in several datasets; its poor performance is in some cases due to high dimensionality, where there is greater redundancy. Table 4 provides a summary of the results across all eight datasets by averaging the correlation values across all subset sizes $|S|$; it shows that SAGE provides the most accurate representation of each feature's contribution in $6/8$ datasets, and is the second best in the remaining $2/8$ datasets.

We next trained models with the most important and least important features. The results (Figures 5-10 bottom) show that SAGE is consistently able to identify important features that contain significant signal, and unimportant features that contain minimal signal. None of the baseline methods are able to do both consistently.

Finally, we examine the convergence speed of SAGE in comparison with permutation tests and loss SHAP values (Figures 5-10 middle right). For SHAP, we used the same single-sample variant of Algorithm 1 and computed loss SHAP values for 128 instances in each dataset. We again found that SAGE takes many more model evaluations than permutation tests to converge, but in most cases not significantly more model evaluations than SHAP. Table 5 provides a summary of the relative speed of SAGE and SHAP, where we identify the point of convergence as the number of model evaluations necessary for intermediate estimates to have a

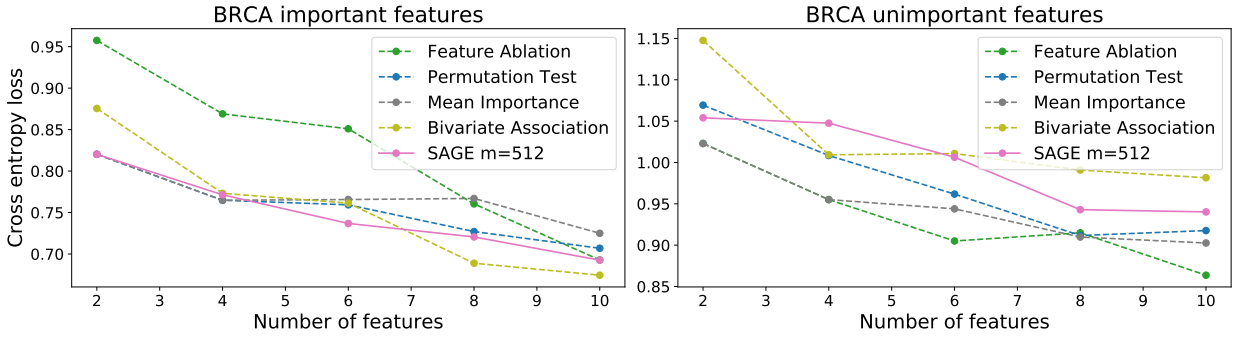

Figure 4. BRCA performance with subsets of the most important (left) and least important (right) features. Important features should lead to low loss, while unimportant ones should lead to high loss.

mean correlation of 0.99 with the final estimate. The results show that in all cases, SAGE was computed for the cost of at most ≈ 90 local SHAP explanations, though in most cases far fewer. Across all the datasets that we considered, the cost of computing SAGE was significantly lower than the cost of computing SHAP values for every example in the dataset.

Table 3. Summary of datasets. *Split size* indicates number of examples used for training, validation and testing, respectively. *Classes* indicates the number of output classes, where applicable. For datasets where we use MLPs, the width of hidden layers is indicated.

Dataset	Features	Examples	Split Size	Classes	Loss	Model Class
Bank Marketing	16	45,211	(36,169, 4,521, 4,521)	2	Cross Entropy	MLP (256, 256)
Bike Rental	12	10,886	(8,710, 1,088, 1,088)	–	MSE	MLP (256, 256, 256)
Breast Cancer	50	556	(446, 55, 55)	4	Cross Entropy	Logistic Regression
Credit Default	20	1,000	(800, 100, 100)	2	Cross Entropy	GBM
Heart Disease	13	297	(239, 29, 29)	2	Cross Entropy	RF
MNIST	784	70,000	(54,000, 6,000, 10,000)	10	Cross Entropy	MLP (256)
Online Shopping	17	12,330	(9,864, 1,233, 1,233)	2	Cross Entropy	MLP (64, 64)
Wine Quality	11	4,898	(3,920, 489, 489)	–	MSE	SVR

Table 4. Mean correlation of cumulative feature importance with model performance. Cumulative importance is defined as the sum of importance values for the features in S, and model performance is the loss of a model trained on X^S. Correlation values are calculated individually for each subset $|S|$ with linear spacing (see Figure 5 middle left, for example) and then averaged.

Method	Bank	Bike	BRCA	Credit	Heart	MNIST	Shopping	Wine
Feature Ablation	**0.930**	0.827	0.233	0.784	0.677	-0.001	0.893	**0.761**
Permutation Test	0.828	0.840	0.622	0.887	0.847	0.398	0.887	0.411
Mean Importance	–	–	0.349	–	–	0.302	–	0.305
Univariate Predictors	0.797	0.774	0.593	0.907	0.817	0.394	0.923	0.674
SAGE	0.919	**0.852**	**0.631**	**0.929**	**0.866**	**0.449**	**0.925**	0.715

Table 5. Speed of SHAP and SAGE convergence in terms of number of model evaluations per feature. Convergence of each method is determined by the number of model evaluations for the mean correlation of intermediate estimates with the final estimate to reach 0.99.

	SAGE			SHAP			SAGE / SHAP Ratio		
Inner loop samples	32	128	512	32	128	512	32	128	512
Bank Marketing	1.0×10^5	2.9×10^5	8.4×10^5	3.1×10^3	4.4×10^3	9.2×10^3	32.4	66.3	90.4
Bike Rental	1.0×10^4	2.2×10^4	5.7×10^4	9.8×10^3	1.7×10^4	4.8×10^4	1.0	1.3	1.2
Breast Cancer	1.4×10^5	2.2×10^5	5.3×10^5	1.1×10^4	1.2×10^4	1.4×10^4	12.9	18.6	37.1
Credit Default	8.8×10^4	2.3×10^5	8.0×10^5	1.5×10^4	2.9×10^4	4.1×10^4	5.7	8.0	19.5
Heart Disease	1.5×10^4	4.0×10^4	1.6×10^5	1.6×10^4	3.3×10^4	5.0×10^4	1.0	1.2	3.2
MNIST	1.9×10^7	1.9×10^7	2.1×10^7	3.6×10^6	2.9×10^6	–	5.3	6.4	–
Online Shopping	2.3×10^4	4.9×10^4	1.5×10^5	3.5×10^3	4.3×10^3	7.0×10^3	6.6	11.2	20.8
Wine Quality	1.4×10^5	3.4×10^5	1.3×10^6	3.2×10^4	8.9×10^4	1.9×10^5	4.2	3.8	6.8

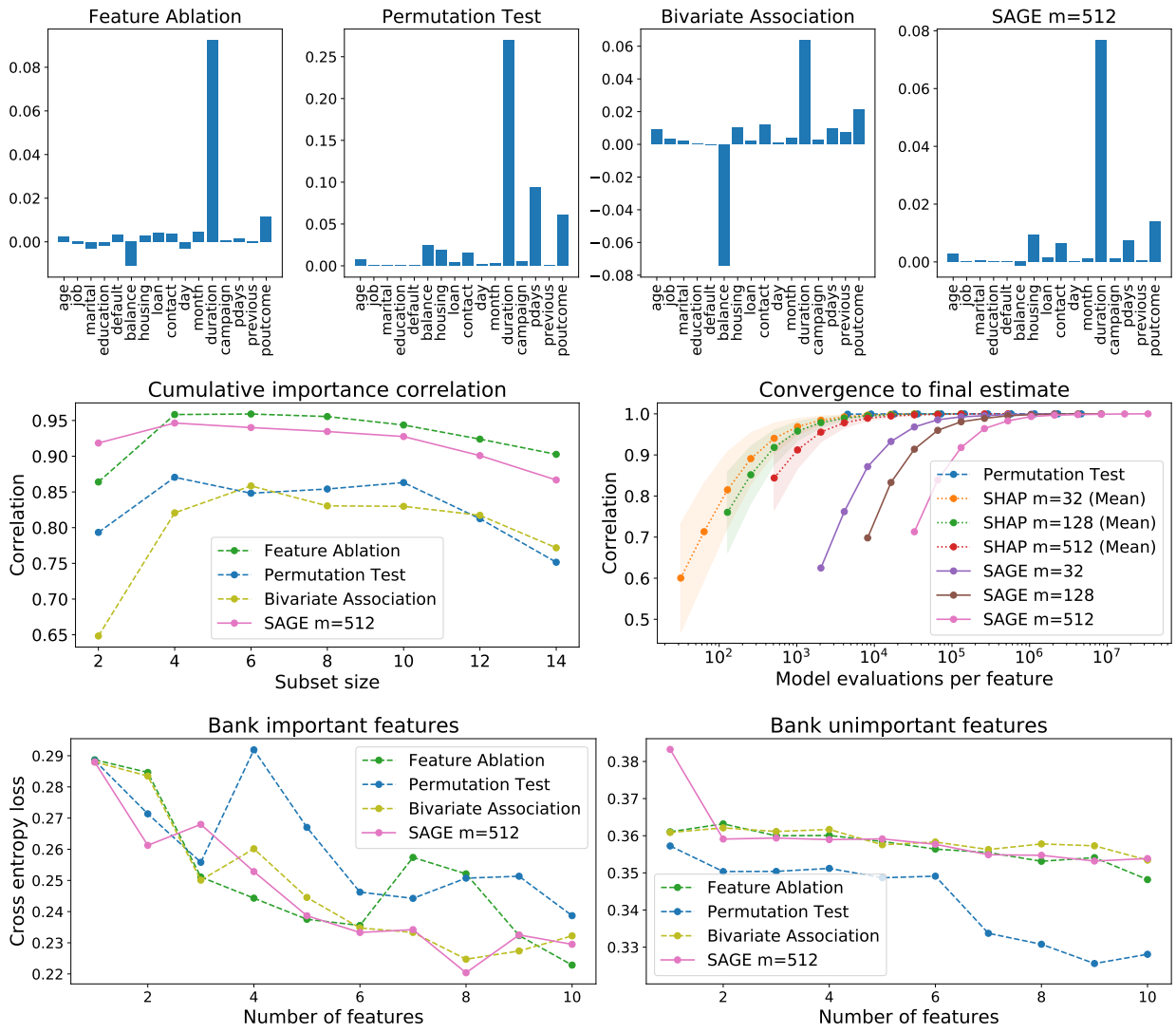

Figure 5. SAGE evaluation on the bank marketing dataset. Top: comparison of feature importance values. Middle left: correlation of cumulative importance with performance of feature subsets (higher is better). Middle right: convergence of importance estimators. Lower left: model performance with most important features (lower is better). Lower right: model performance with least important features (higher is better).

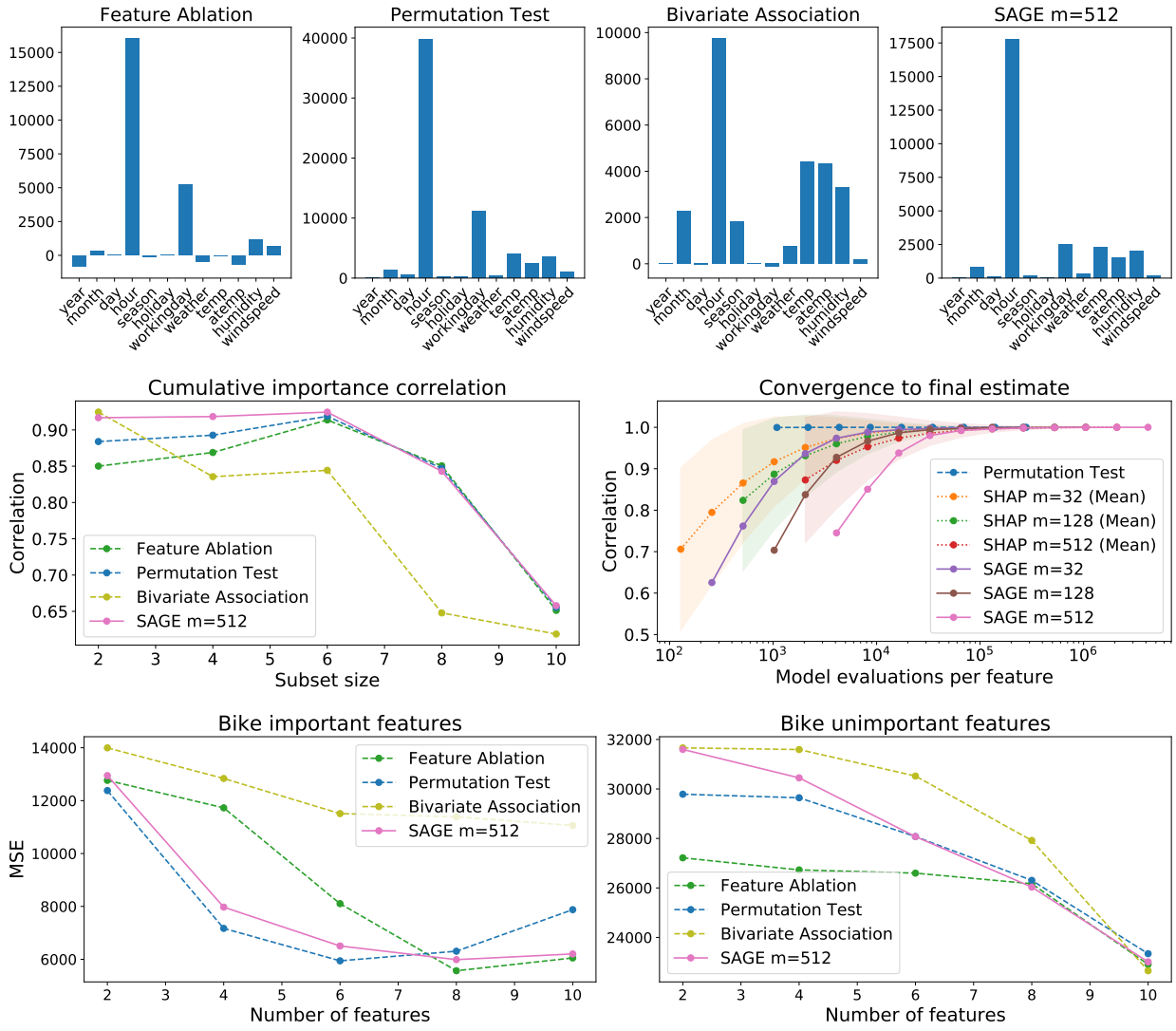

Figure 6. SAGE evaluation on the bike rental dataset. Top: comparison of feature importance values. Middle left: correlation of cumulative importance with performance of feature subsets (higher is better). Middle right: convergence of importance estimators. Lower left: model performance with most important features (lower is better). Middle right: model performance with least important features (higher is better).

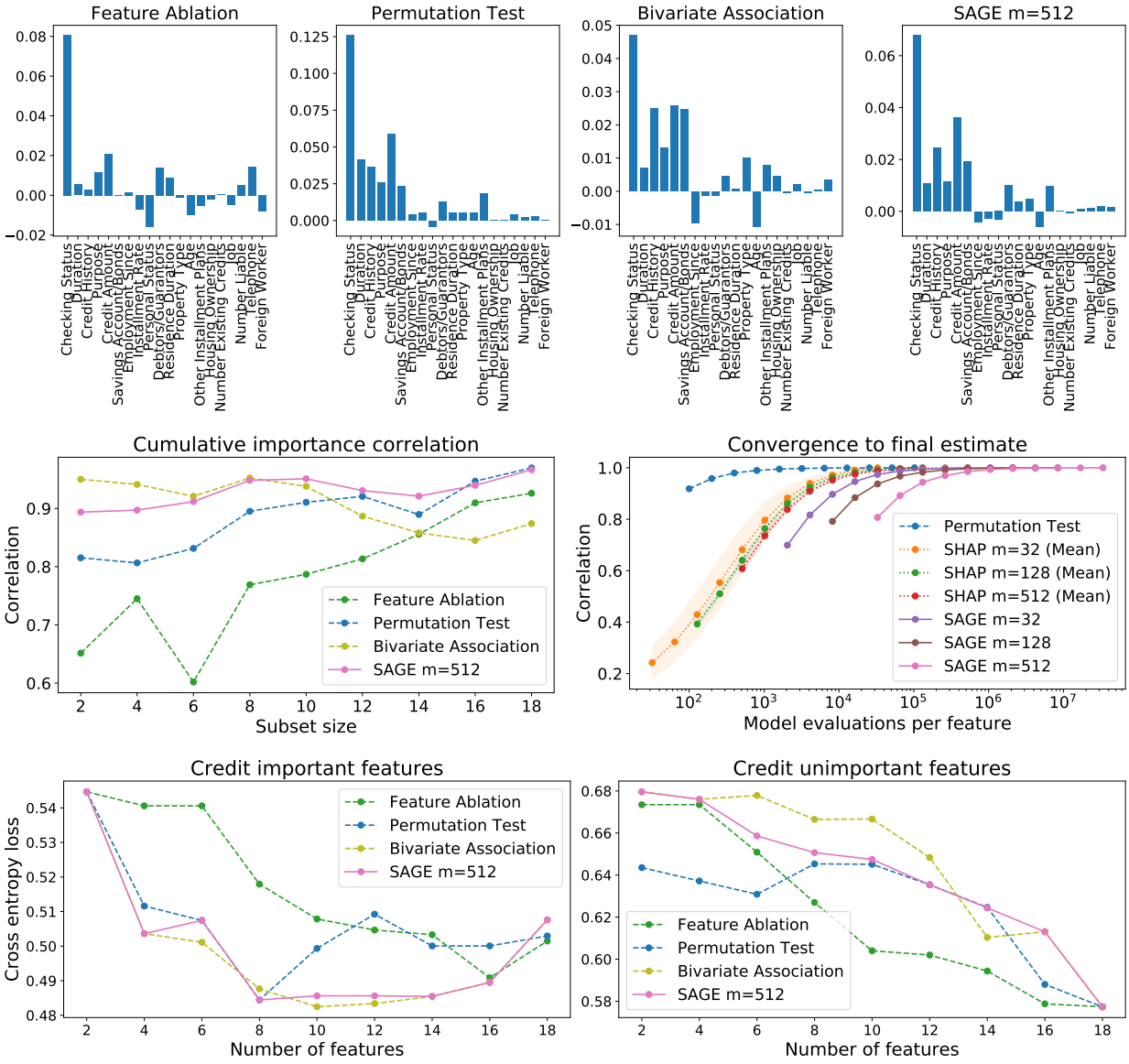

Figure 7. SAGE evaluation on the credit default dataset. Top: comparison of feature importance values. Middle left: correlation of cumulative importance with performance of feature subsets (higher is better). Middle right: convergence of importance estimators. Lower left: model performance with most important features (lower is better). Lower right: model performance with least important features (higher is better).

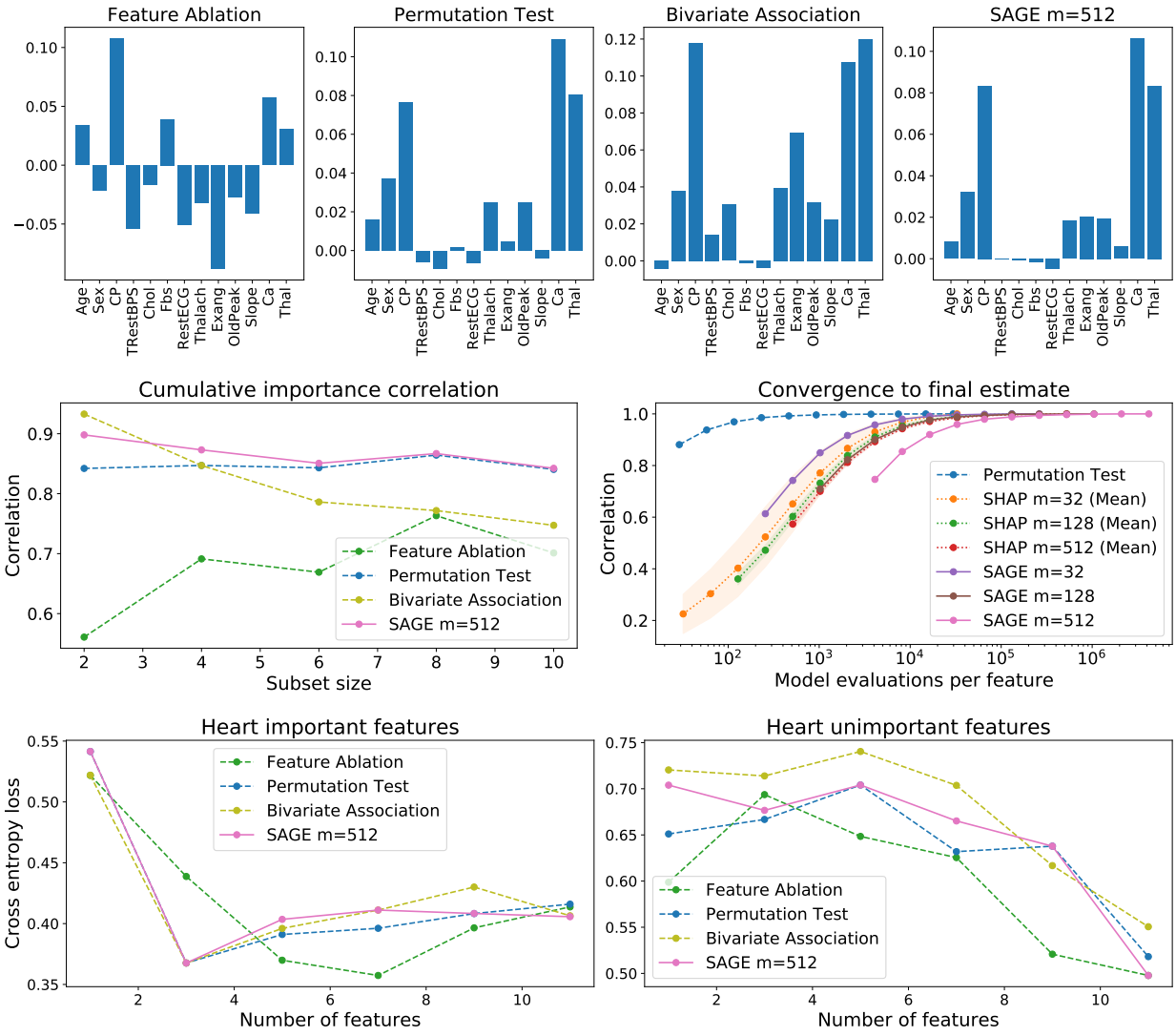

Figure 8. SAGE evaluation on the heart disease dataset. Top: comparison of feature importance values. Middle left: correlation of cumulative importance with performance of feature subsets (higher is better). Middle right: convergence of importance estimators. Lower left: model performance with most important features (lower is better). Middle right: model performance with least important features (higher is better).

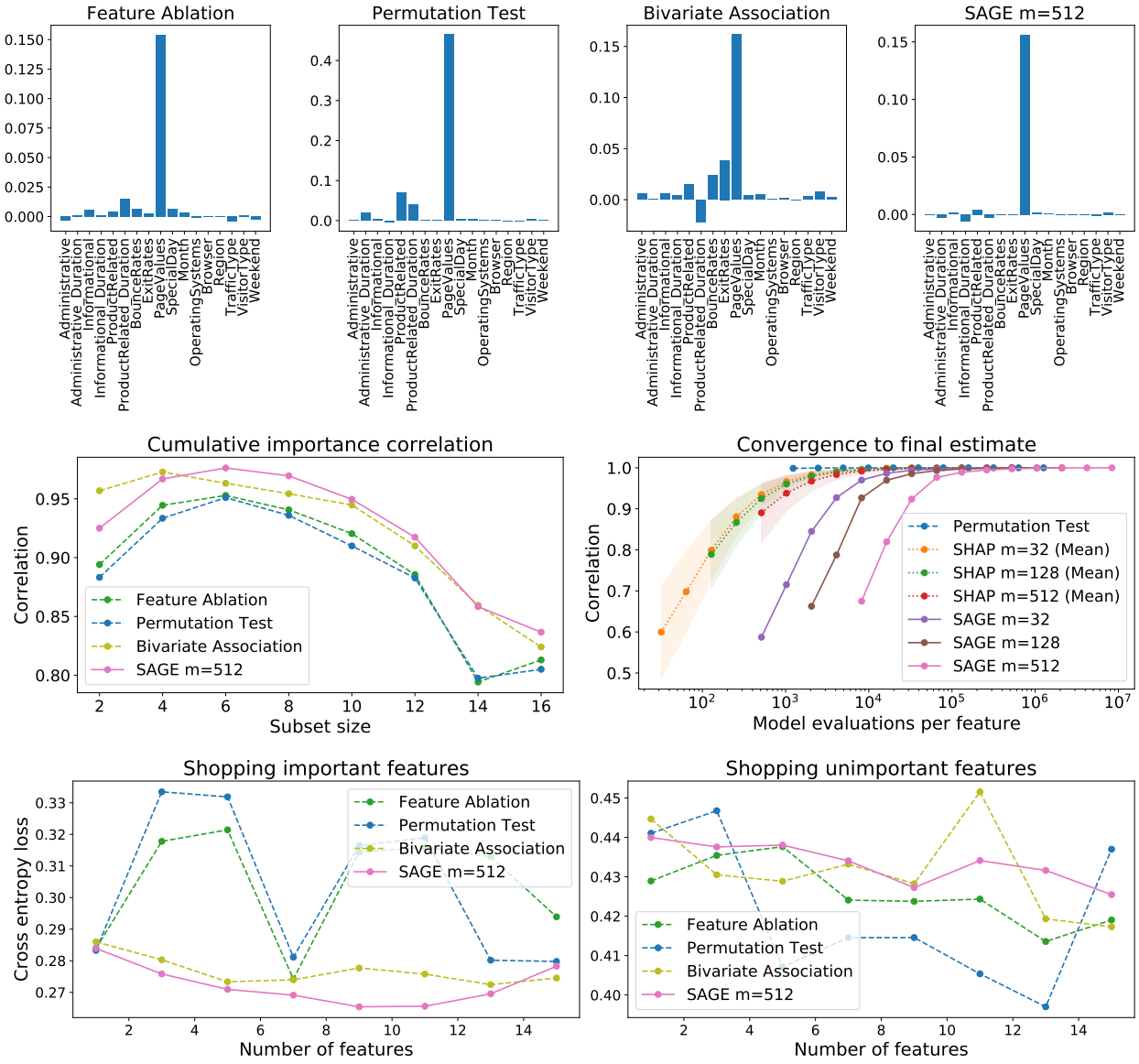

Figure 9. SAGE evaluation on the online shopping dataset. Top: comparison of feature importance values. Middle left: correlation of cumulative importance with performance of feature subsets (higher is better). Middle right: convergence of importance estimators. Lower left: model performance with most important features (lower is better). Lower right: model performance with least important features (higher is better).

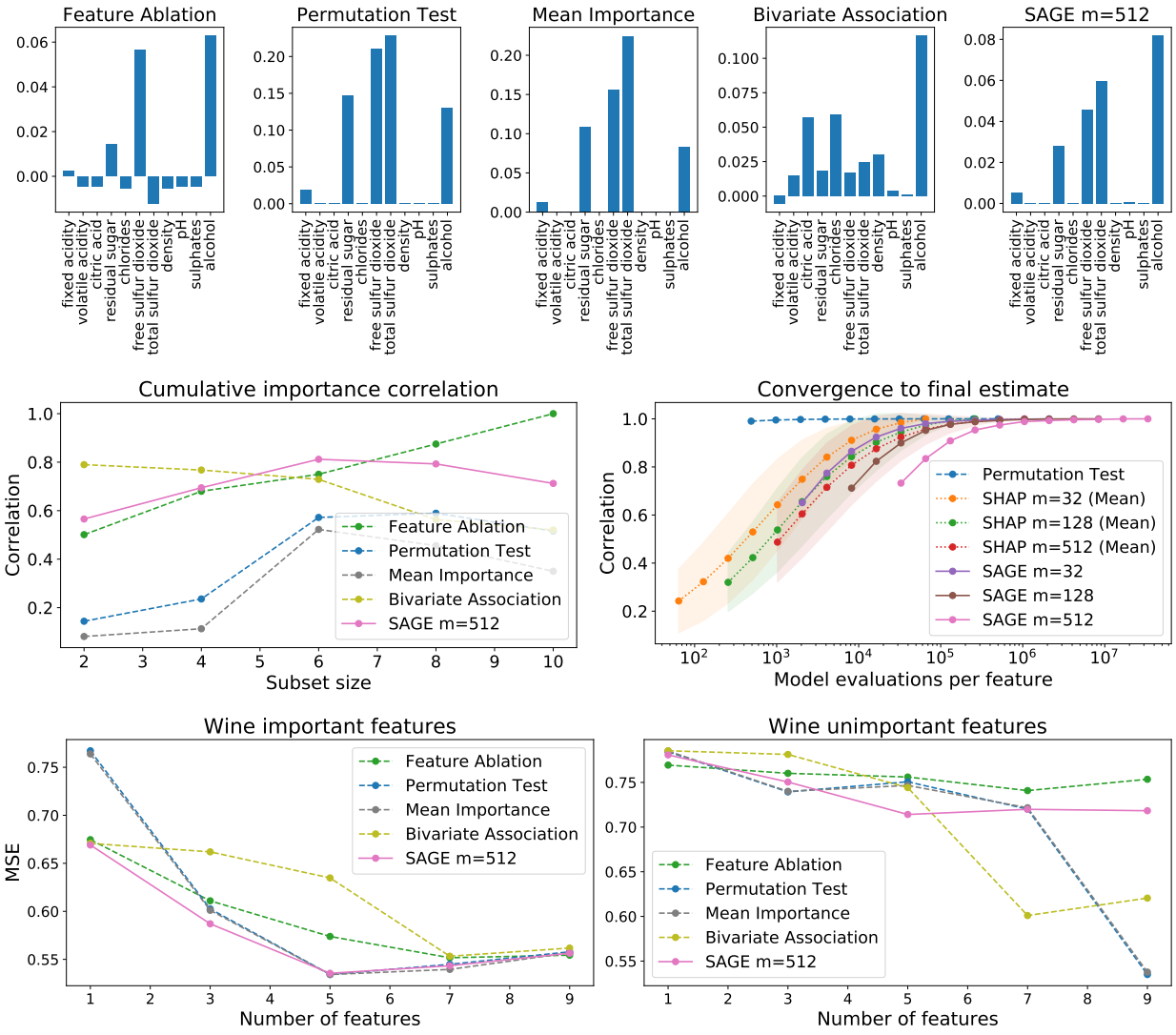

Figure 10. SAGE evaluation on the wine quality dataset. Top: comparison of feature importance values. Middle left: correlation of cumulative importance with performance of feature subsets (higher is better). Middle right: convergence of importance estimators. Lower left: model performance with most important features (lower is better). Lower right: model performance with least important features (higher is better).