

Reading Group: Shapley Values and SHAP

Machine Learning Interpretability

Manuel Quintero

October 21, 2026

1 Introduction

We take the following example from [1, Chapter 17]. A model has been trained to predict apartment prices. For an apartment of 50 m², on the 2nd floor, with a park nearby and where cats are banned, the model predicts \$300,000, while the average prediction over all apartments is \$310,000, see [1, Figure 17.1]. We want to know how much each of the four features contributed to the difference of $-\$10,000$.

That is, we want to write

$$\hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{X})] = \phi_1 + \phi_2 + \cdots + \phi_p,$$

where ϕ_j is the part of the difference that we attribute to feature j . If the model is linear, each feature enters as a separate term and there is not much to discuss. The difficulty comes from interactions: if the park and the area only matter *together*, who gets the credit?

This is the same problem as splitting the payout of a joint project among the people who worked on it, which is the question that cooperative game theory studies. The Shapley value [2] is its classical answer. In these notes we define it for a general cooperative game, prove that it is the unique rule satisfying four axioms, and then apply it to model predictions. In the second part we cover SHAP [4], which writes Shapley values as an additive model and gives faster ways to compute them.

2 Cooperative Games

Game theory studies situations where the outcome depends on the decisions of several agents, called **players**. In many games each player chooses an action and the question is which actions will be played. In a **cooperative game** we do not model actions at all. Instead, we assume players can form **coalitions**, and each coalition has a payout that it can secure by itself.

Definition 2.1 (Cooperative Game). A cooperative game (with transferable utility¹) is a tuple $G = (I, v)$ such that:

¹Transferable utility (TU) means that the payout of a coalition is a single quantity, like money, that its members can split among themselves in any way, with one unit given up by one player being worth exactly one unit to another. This is why the worth of a coalition can be described by a single number $v(S)$, and why any vector ϕ with $\sum_{i \in S} \phi_i = v(S)$ is an achievable split.

1. $I = \{1, 2, \dots, N\}$ is the set of players.
2. $v : 2^I \rightarrow \mathbb{R}$ is the **characteristic function**, which assigns to each coalition $S \subseteq I$ a payout $v(S)$, with $v(\emptyset) = 0$.

We call $S \subseteq I$ a **coalition** and I the **grand coalition**.

Definition 2.2 (Marginal contribution). In a cooperative game $G = (I, v)$, the **marginal contribution** of player i to a coalition $S \subseteq I \setminus \{i\}$ is

$$\Delta_i(S) = v(S \cup \{i\}) - v(S).$$

2.1 Solution concepts:

What does it mean to solve a cooperative game? We assume the grand coalition forms and earns $v(I)$. A solution is an allocation $\phi(v) = (\phi_1(v), \dots, \phi_N(v)) \in \mathbb{R}^N$, that is, how much of $v(I)$ each player receives.

Paying each player its marginal contribution to a fixed coalition does not work, since $\Delta_i(S)$ depends on S : the same player can add a lot to a small coalition and nothing to a large one. Moreover, if we build the grand coalition by adding players one at a time and pay each one its marginal contribution at the moment it joins, the payments add up to $v(I)$, but they depend on the order we chose. Shapley's solution is to average over all orders.

Example 2.1 (The order matters). Two players, with $v(\{1\}) = 10$, $v(\{2\}) = 20$ and $v(\{1, 2\}) = 60$. Together they earn more than separately: $60 > 10 + 20$.

Player 1 adds 10 to the empty coalition but $60 - 20 = 40$ to $\{2\}$, so there is no single "contribution of player 1". If we build the grand coalition one player at a time:

order	player 1 receives	player 2 receives	total
1 joins first, then 2	10	$60 - 10 = 50$	60
2 joins first, then 1	$60 - 20 = 40$	20	60
average	25	35	60

Both orders distribute all of $v(I) = 60$, but whoever joins second collects the extra 30 that the two players only produce together. Averaging over the two orders splits that extra 30 equally, giving $\phi_1 = 25$ and $\phi_2 = 35$: each player gets what it earns alone plus half of the joint gain.

3 The Shapley Value

3.1 Permutation form

Definition 3.1 (Shapley value). Let $\Pi(I)$ denote the set of all $N!$ orderings of I and, for $\pi \in \Pi(I)$, let P_i^π be the set of players that come before i in π . The **Shapley value** of player i is

$$\phi_i(v) = \frac{1}{N!} \sum_{\pi \in \Pi(I)} \left[v(P_i^\pi \cup \{i\}) - v(P_i^\pi) \right]. \quad (1)$$

Equivalently, $\phi_i(v) = \mathbb{E}_\pi [\Delta_i(P_i^\pi)]$ where π is a uniformly random ordering. We will use this probabilistic reading when we estimate Shapley values by sampling.

How to read (1):

- π is one arrival order, e.g. “2, then 3, then 1”. There are $N!$ orders, and $\Pi(I)$ is the set of all of them.
- P_i^π is the set of players already there when i arrives in order π . If $\pi = (2, 3, 1)$ and $i = 1$, then $P_1^\pi = \{2, 3\}$.
- $v(P_i^\pi \cup \{i\}) - v(P_i^\pi)$ is the payout with i minus the payout just before i arrived, that is, what i adds when it walks in under this order.
- $\sum_\pi$ adds this over every order, and $\frac{1}{N!}$ divides by the number of orders, so the sum becomes an average.

So ϕ_i is the average contribution of player i over all possible arrival orders.

Example 3.1. In Example 2.1 ($N = 2$, so $2! = 2$ orders), for player 1:

- Order (1, 2): nobody is there before player 1, so $P_1^\pi = \emptyset$ and the contribution is $v(\{1\}) - v(\emptyset) = 10 - 0 = 10$.
- Order (2, 1): player 2 is already there, so $P_1^\pi = \{2\}$ and the contribution is $v(\{1, 2\}) - v(\{2\}) = 60 - 20 = 40$.

Thus $\phi_1 = \frac{1}{2}(10 + 40) = 25$, which is the “player 1” column of the table in Example 2.1. Equation (1) says: fill in that table for every order and average the column.

3.2 Subset form

Different orderings can give player i the same set of predecessors, so the sum in (1) has many repeated terms. Collecting them gives the formula used in the book.

Proposition 3.1.

$$\phi_i(v) = \sum_{S \subseteq I \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} [v(S \cup \{i\}) - v(S)]. \quad (2)$$

Proof. Fix $S \subseteq I \setminus \{i\}$. An ordering π satisfies $P_i^\pi = S$ if and only if it lists the elements of S first, then i , then the rest. There are $|S|!$ ways to order S and $(N - |S| - 1)!$ ways to order the rest, so the term $\Delta_i(S)$ appears $|S|!(N - |S| - 1)!$ times in (1). $\square$

Remark: Writing $s = |S|$,

$$\frac{s!(N - s - 1)!}{N!} = \frac{1}{N} \cdot \frac{1}{\binom{N-1}{s}}.$$

So the weights correspond to choosing the size $s \in \{0, \dots, N - 1\}$ uniformly, and then a coalition of size s uniformly. In particular they sum to one, and each size class gets total weight $1/N$. A coalition that is alone in its size class ($S = \emptyset$ or $S = I \setminus \{i\}$) gets weight $1/N$ by itself.

Example 3.2. Consider $I = \{1, 2, 3\}$ with

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
$v(S)$	0	10	20	30	40	50	60	90

Permutation form. For each of the $3! = 6$ orderings we record the marginal contribution of each player when it joins:

ordering	player 1	player 2	player 3
(1, 2, 3)	10	30	50
(1, 3, 2)	10	40	40
(2, 1, 3)	20	20	50
(2, 3, 1)	30	20	40
(3, 1, 2)	20	40	30
(3, 2, 1)	30	30	30
average	20	30	40

Each row adds up to $v(I) = 90$, hence so does the last one.

Subset form. For $N = 3$ the weights are $\frac{1}{3}, \frac{1}{6}, \frac{1}{3}$ for $|S| = 0, 1, 2$. For player 1,

$$\phi_1 = \frac{1}{3}(10 - 0) + \frac{1}{6}(40 - 20) + \frac{1}{6}(50 - 30) + \frac{1}{3}(90 - 60) = 20.$$

3.3 Axioms and uniqueness

Definition 3.2 (Shapley axioms). An allocation rule $v \mapsto \phi(v)$ satisfies:

1. **Efficiency:** $\sum_{i=1}^N \phi_i(v) = v(I)$.
2. **Symmetry:** if $v(S \cup \{i\}) = v(S \cup \{k\})$ for all $S \subseteq I \setminus \{i, k\}$, then $\phi_i(v) = \phi_k(v)$.
3. **Dummy:** if $v(S \cup \{i\}) = v(S)$ for all $S \subseteq I$, then $\phi_i(v) = 0$.
4. **Additivity:** for any two games v, w on I , $\phi_i(v + w) = \phi_i(v) + \phi_i(w)$.

Theorem 3.1 (Shapley, 1953). The Shapley value is the unique allocation rule that satisfies Efficiency, Symmetry, Dummy and Additivity.

Proof. The Shapley value satisfies the axioms. For a fixed ordering the marginal contributions telescope to $v(I)$, so averaging over orderings gives Efficiency. Symmetry and Dummy can be read off from (2), and Additivity holds since (2) is linear in v .

Uniqueness. For each non-empty $T \subseteq I$ define the **unanimity game**

$$u_T(S) = \begin{cases} 1, & T \subseteq S, \\ 0, & \text{otherwise.} \end{cases}$$

The games on I form a vector space of dimension $2^N - 1$, and $\{u_T\}_{T \neq \emptyset}$ is a basis of it (the matrix $[u_T(S)]$ is triangular when coalitions are ordered by size). Thus every game can be written as $v = \sum_{T \neq \emptyset} c_T u_T$. Now fix T and $c \in \mathbb{R}$ and consider the game $c u_T$. Players outside T are dummies, so they get 0. Any two players in T are symmetric, so they get the

same amount. By Efficiency that amount is $c/|T|$. Hence any rule satisfying the axioms agrees with the Shapley value on every cu_T , and by Additivity on every game:

$$\phi_i(v) = \sum_{T \ni i} \frac{c_T}{|T|}.$$

□

4 Shapley Values for a Prediction

From now on we follow the book's notation: p features, indexed by j . Let $\hat{f} : \mathbb{R}^p \rightarrow \mathbb{R}$ be a trained model and $\mathbf{x} = (x_1, \dots, x_p)$ the instance we want to explain. We take the players to be the features of $\mathbf{x}$ and need a characteristic function: what is the “prediction” of a coalition S of features? The model requires all p inputs, so we fix the features in S at their values in $\mathbf{x}$ and average the model over the remaining ones.

Definition 4.1 (Value function of a prediction). *Let $C = \{1, \dots, p\} \setminus S$. The value function of $\mathbf{x}$ is*

$$v_{\mathbf{x}}(S) = \mathbb{E}_{X_C} [\hat{f}(x_S, X_C)] - \mathbb{E}[\hat{f}(\mathbf{X})], \quad (3)$$

where the expectation is over the marginal distribution of X_C .

Note that $v_{\mathbf{x}}(\emptyset) = 0$ and $v_{\mathbf{x}}(\{1, \dots, p\}) = \hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{X})]$. Thus, by Efficiency, the Shapley values of the game $v_{\mathbf{x}}$ give exactly the decomposition we asked for in the introduction.

Example 4.1 (Apartment, from [1]). *Go back to the apartment of the introduction. The players are the four feature values*

$$I = \{\text{park-nearby}, \text{area-50}, \text{floor-2nd}, \text{cat-banned}\},$$

$$\hat{f}(\mathbf{x}) = \$300k, \mathbb{E}[\hat{f}(\mathbf{X})] = \$310k, \text{ and the payout to split is } v_{\mathbf{x}}(I) = -\$10k.$$

A possible answer, given in [1], is

	park-nearby	area-50	floor-2nd	cat-banned
ϕ_j	+\$30k	+\$10k	\$0	-\$50k

which adds up to $-\$10k$, as Efficiency requires. Let us see what is needed to compute, for instance, ϕ_{cat} . With $p = 4$, the coalitions S not containing **cat-banned** are the $2^3 = 8$ subsets of $\{\text{park}, \text{area}, \text{floor}\}$, see [1, Figure 17.3], and by the remark in Section 3.2 their weights are $\frac{1}{4}$ for $S = \emptyset$ and $S = \{\text{park}, \text{area}, \text{floor}\}$, and $\frac{1}{12}$ for each of the other six:

$$\begin{aligned} \phi_{\text{cat}} &= \frac{1}{4} [v_{\mathbf{x}}(\{\text{cat}\}) - v_{\mathbf{x}}(\emptyset)] + \frac{1}{12} \sum_{|S|=1} [v_{\mathbf{x}}(S \cup \{\text{cat}\}) - v_{\mathbf{x}}(S)] \\ &\quad + \frac{1}{12} \sum_{|S|=2} [v_{\mathbf{x}}(S \cup \{\text{cat}\}) - v_{\mathbf{x}}(S)] + \frac{1}{4} [v_{\mathbf{x}}(I) - v_{\mathbf{x}}(\{\text{park}, \text{area}, \text{floor}\})]. \end{aligned}$$

Each term needs the expectation in (3). In [1, Figure 17.2] one term, for $S = \{\text{park}, \text{area}\}$, is estimated with a single random apartment that supplies the missing values (the floor, and the

cat rule when `cat-banned` is not in the coalition): the prediction is \$310k with `cat-banned` and \$320k with the random apartment's value, so this sample of $v_{\mathbf{x}}(\{\text{park}, \text{area}, \text{cat}\}) - v_{\mathbf{x}}(\{\text{park}, \text{area}\})$ is $-\$10\text{k}$. Repeating with many random apartments and averaging estimates the term.

Remarks:

- Three objects that are easy to confuse: the feature value x_j , the value function $v_{\mathbf{x}}$, and the Shapley value ϕ_j .
- The choice of distribution in (3) is part of the definition. Using the marginal distribution of X_C is the *interventional* choice. Using $X_C \mid X_S = x_S$ is the *conditional* choice. They coincide for independent features (Section 8).
- The reference population is also a choice. Replacing $\mathbf{X}$ by a subpopulation changes $v_{\mathbf{x}}$ and the Shapley values: each ϕ_j is a statement relative to that population.

5 Examples

5.1 Linear model

Proposition 5.1. *Let $\hat{f}(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$. Then*

$$\phi_j = \beta_j(x_j - \mathbb{E}[X_j]).$$

Proof. By linearity of expectation,

$$\mathbb{E}_{X_C}[\hat{f}(x_S, X_C)] = \beta_0 + \sum_{k \in S} \beta_k x_k + \sum_{k \in C} \beta_k \mathbb{E}[X_k].$$

Adding j to S only replaces $\beta_j \mathbb{E}[X_j]$ by $\beta_j x_j$, so $v_{\mathbf{x}}(S \cup \{j\}) - v_{\mathbf{x}}(S) = \beta_j(x_j - \mathbb{E}[X_j])$ for every S . Since the weights in (2) sum to one, this is ϕ_j . $\square$

Example 5.1. *Let $\hat{f}(x_1, x_2) = 2 + 3x_1 - x_2$ with $\mathbb{E}[X_1] = 0$, $\mathbb{E}[X_2] = 2$, and $\mathbf{x} = (1, 4)$.*

Then $\hat{f}(\mathbf{x}) = 1$ and $\mathbb{E}[\hat{f}(\mathbf{X})] = 0$. The Shapley values are $\phi_1 = 3(1 - 0) = 3$ and $\phi_2 = -1(4 - 2) = -2$, which add up to 1.

Remark: The attribution is not β_j , and not $\beta_j x_j$: a feature with a large coefficient gets little credit if x_j is close to its mean.

5.2 A pure interaction

Example 5.2. *Let $\hat{f}(x_1, x_2) = x_1 x_2$, with X_1, X_2 independent with means μ_1, μ_2 , so $\mathbb{E}[\hat{f}(\mathbf{X})] = \mu_1 \mu_2$.*

The value function is

$$\begin{aligned} v_{\mathbf{x}}(\emptyset) &= 0, & v_{\mathbf{x}}(\{1\}) &= (x_1 - \mu_1)\mu_2, \\ v_{\mathbf{x}}(\{2\}) &= \mu_1(x_2 - \mu_2), & v_{\mathbf{x}}(\{1, 2\}) &= x_1 x_2 - \mu_1 \mu_2. \end{aligned}$$

With $p = 2$ both weights are $\frac{1}{2}$, so

$$\phi_1 = \frac{1}{2} v_{\mathbf{x}}(\{1\}) + \frac{1}{2} [v_{\mathbf{x}}(\{1, 2\}) - v_{\mathbf{x}}(\{2\})] = \frac{1}{2} (x_1 - \mu_1)(x_2 + \mu_2),$$

and $\phi_2 = \frac{1}{2} (x_2 - \mu_2)(x_1 + \mu_1)$ by symmetry.

Remark: The sign of ϕ_j does not tell us what happens if x_j increases. With $\mu_1 = \mu_2 = 0$, the instance $\mathbf{x} = (1, -3)$ gives $\phi_1 = -\frac{3}{2}$ and $\mathbf{x} = (1, 3)$ gives $\phi_1 = +\frac{3}{2}$, with the same x_1 . The Shapley value describes how the prediction at $\mathbf{x}$ is split, not the local behavior of $\hat{f}$ around $\mathbf{x}$.

Example 5.3. Let $\hat{f}(x_1, x_2, x_3) = x_1 x_2$. The model does not depend on x_3 , so $v_{\mathbf{x}}(S \cup \{3\}) = v_{\mathbf{x}}(S)$ for every S and $\phi_3 = 0$ by Dummy. The values of ϕ_1, ϕ_2 are unchanged.

6 Estimation

Computing (2) exactly needs 2^{p-1} values of $v_{\mathbf{x}}$ per feature, and each one is an expectation over the data. We approximate both by sampling. Using the permutation form, $\phi_j = \mathbb{E}_{\pi} [v_{\mathbf{x}}(P_j^{\pi} \cup \{j\}) - v_{\mathbf{x}}(P_j^{\pi})]$, and the expectation inside $v_{\mathbf{x}}$ can be replaced by a single random instance $\mathbf{z}$ from the data:

Definition 6.1 (Monte Carlo estimator, [3]). For $m = 1, \dots, M$, draw an instance $\mathbf{z}^{(m)}$ from the data and a uniformly random ordering $\pi^{(m)}$. Let $\mathbf{x}_{+j}^{(m)}$ take the values of $\mathbf{x}$ on $P_j^{\pi^{(m)}} \cup \{j\}$ and the values of $\mathbf{z}^{(m)}$ elsewhere, and let $\mathbf{x}_{-j}^{(m)}$ be the same but with $z_j^{(m)}$ in coordinate j . Then

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^M [\hat{f}(\mathbf{x}_{+j}^{(m)}) - \hat{f}(\mathbf{x}_{-j}^{(m)})].$$

Since $\pi^{(m)}$ and $\mathbf{z}^{(m)}$ are independent and X_C is drawn from its marginal, each term has expectation ϕ_j , so $\hat{\phi}_j$ is unbiased with variance of order $1/M$.

7 Shapley Values on Real Data

We now compute Shapley values for an actual model. We use the California housing data (available in `scikit-learn`): each of the 20,640 instances is a district, with $p = 8$ features (median income, house age, average number of rooms and bedrooms, population, average occupancy, latitude and longitude), and the target is the median house value of the district, in thousands of dollars. We train a random forest with 100 trees on 80% of the data ($R^2 = 0.79$ on the remaining 20%).

Example 7.1 (Exact Shapley values). Take the district at latitude 37.76, longitude -122.44 (San Francisco). The model predicts \$415k, and the average prediction is \$206k, so the payout to split is \$209k.

Since $p = 8$, we can compute the Shapley values exactly from (2): there are $2^8 = 256$ coalitions, and for each one we estimate $v_{\mathbf{x}}(S)$ in (3) by averaging the model over 100 random districts from the training data. This takes $256 \times 100 = 25,600$ predictions. The

result is in Figure 1. The values add up to \$209k, as Efficiency requires. Note that the largest contribution comes from the average occupancy (1.7 people per household, which is low), not from the median income.

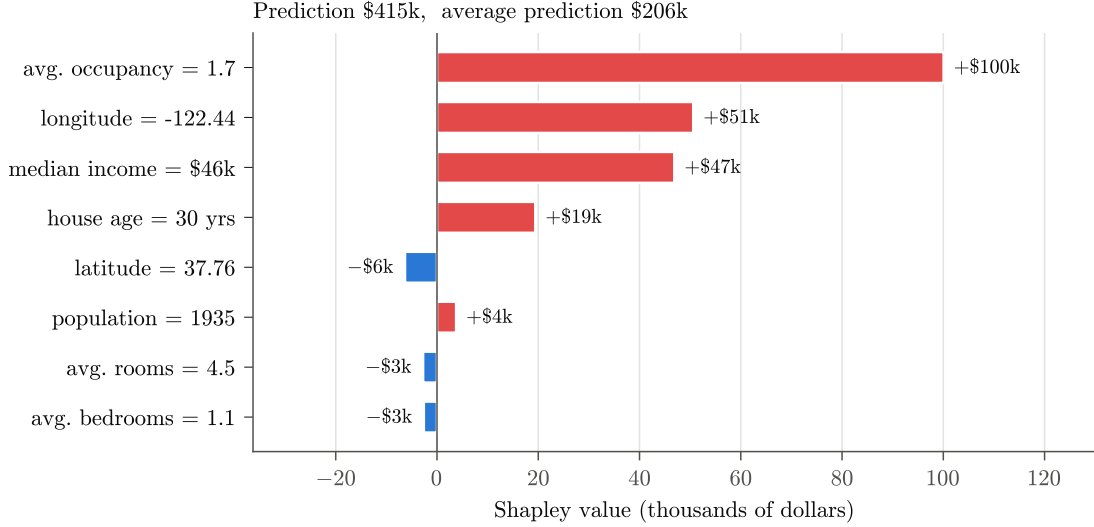


Figure 1: Exact Shapley values for the San Francisco district, random forest model. Red bars increase the prediction and blue bars decrease it, relative to the average prediction.

Now we estimate the same values with the Monte Carlo estimator of Section 6, for the two largest ones. Figure 2 shows $\hat{\phi}_j$ as a function of the number of samples M , for three independent runs, together with the exact values. For small M the estimates are far off (with $M = 1$ we get a single marginal contribution), and they get within a few thousand dollars of the exact value only after about a thousand samples.

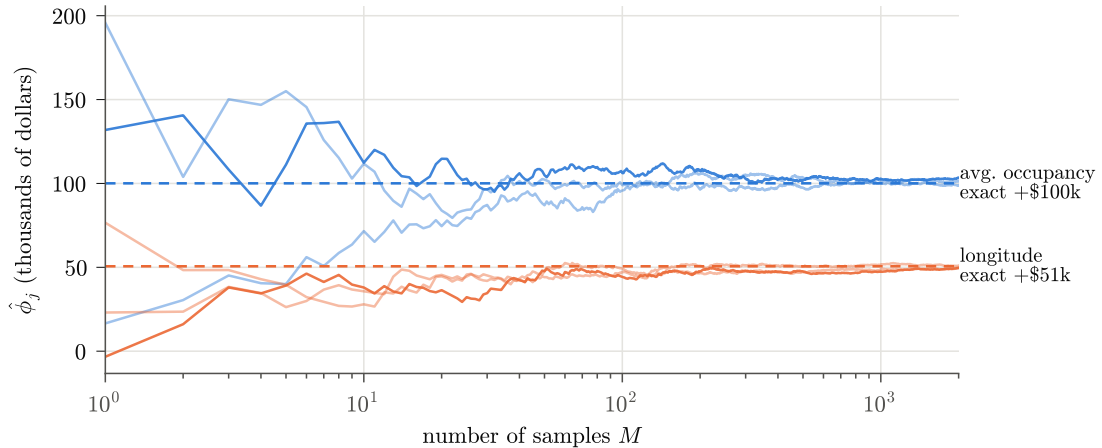


Figure 2: Monte Carlo estimates of two Shapley values as a function of the number of samples M (three runs each, log scale). Dashed lines: exact values from Figure 1.

Remark: As a check of the exact computation, the `shap` package (interventional TreeSHAP, Section 9, with the same 100 background districts) gives the same values up to \$20.

Example 7.2 (Changing the reference population). *Same model and same district, but now the expectation in (3) is taken over districts in the Bay Area only (latitude between 37.2 and 38.3, longitude between -122.8 and -121.7).*

The average prediction over the Bay Area is \$281k, so the payout is now \$134k instead of \$209k. Figure 3 compares the two explanations. The values of the location features change (latitude goes from $-\$6k$ to $+\$12k$ and longitude from $+\$51k$ to $+\$28k$), since the district is now compared with other districts in the Bay Area instead of all of California. More interestingly, the median income (\$46k) goes from $+\$47k$ to $-\$5k$: it is high compared to California, but not compared to the Bay Area. Both explanations are correct; they answer different questions.

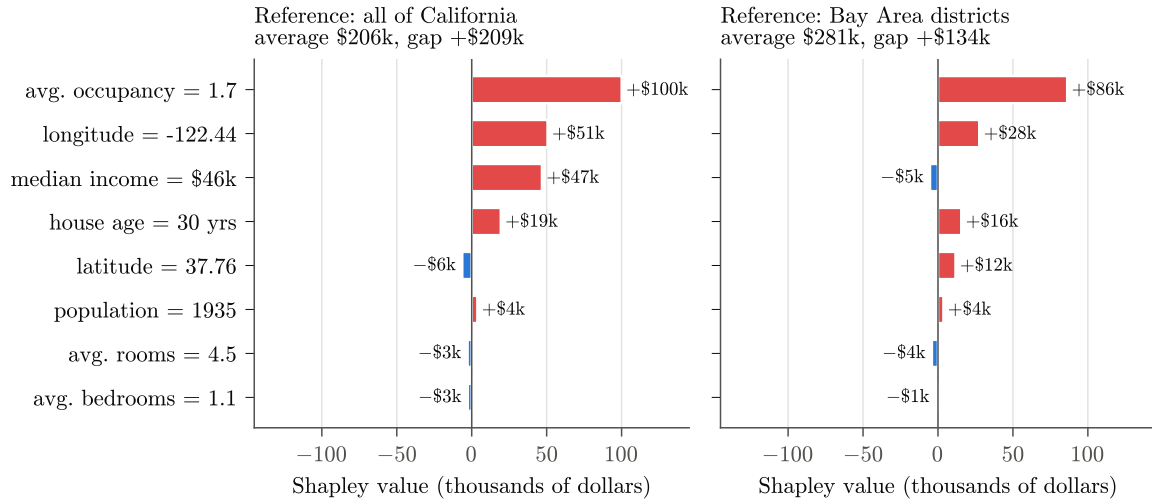


Figure 3: Shapley values of the same prediction with two reference populations. Rows are in the same order in both panels.

8 Pitfalls

Remarks:

- ϕ_j is not the change in prediction if we retrain without feature j . It is defined for a fixed model and a fixed reference population.
- ϕ_j is not a derivative (Section 5.2).
- **Correlated features.** In (3) we combine x_S with values of X_C drawn independently of x_S . If the features are dependent, these combinations can be far from the data (a 20 m² apartment with six bedrooms), and v_x depends on how $\hat{f}$ behaves where it was never trained. Conditioning on $X_S = x_S$ avoids this, but then a feature that $\hat{f}$ does not use can still get $\phi_j \neq 0$ through its correlation with features that are used, so Dummy no longer holds for the model [6].

9 SHAP

SHAP [4] keeps the Shapley value as the target and changes two things: it writes the explanation as a linear model on binary variables, and it gives estimators that are faster than the Monte Carlo estimator above. Following the book, in this section we write M for the number of features.

9.1 Additive explanation model

Definition 9.1 (Additive feature attribution). Let $\mathbf{z}' \in \{0, 1\}^M$ be a **coalition vector**, with $z'_j = 1$ if feature j is in the coalition. An **additive feature attribution** is a function

$$g(\mathbf{z}') = \phi_0 + \sum_{j=1}^M \phi_j z'_j.$$

To compare g with the model we need a map $h_{\mathbf{x}} : \{0, 1\}^M \rightarrow \mathbb{R}^p$ that sends a coalition to an input: coordinates with $z'_j = 1$ are set to x_j , the others are filled from the data. Write $\hat{f}_{\mathbf{x}}(\mathbf{z}') = \hat{f}(h_{\mathbf{x}}(\mathbf{z}'))$. With the filled-in values averaged out, $\hat{f}_{\mathbf{x}}(\mathbf{z}') - \phi_0$ is the value function $v_{\mathbf{x}}$ of Section 4.

See [1, Figures 18.1 and 18.2] for the map $h_{\mathbf{x}}$ on tabular data and on images, where the features are superpixels.

Definition 9.2 (SHAP properties). Let $\mathbf{x}' = (1, \dots, 1)$.

1. **Local accuracy:** $\hat{f}(\mathbf{x}) = g(\mathbf{x}') = \phi_0 + \sum_j \phi_j$, with $\phi_0 = \mathbb{E}[\hat{f}(\mathbf{X})]$.
2. **Missingness:** if $x'_j = 0$ then $\phi_j = 0$.
3. **Consistency:** if for two models $\hat{f}, \hat{f}'$,

$$\hat{f}'_{\mathbf{x}}(\mathbf{z}') - \hat{f}'_{\mathbf{x}}(\mathbf{z}'_{-j}) \geq \hat{f}_{\mathbf{x}}(\mathbf{z}') - \hat{f}_{\mathbf{x}}(\mathbf{z}'_{-j}) \quad \forall \mathbf{z}' \in \{0, 1\}^M,$$

where $\mathbf{z}'_{-j}$ is $\mathbf{z}'$ with $z'_j = 0$, then $\phi_j(\hat{f}', \mathbf{x}) \geq \phi_j(\hat{f}, \mathbf{x})$.

Local accuracy is Efficiency. Consistency is a monotonicity condition on marginal contributions, called strong monotonicity in [7], where Young shows that the Shapley value is the unique rule satisfying Efficiency, Symmetry and strong monotonicity, without assuming Additivity. The proof of the next theorem in [4] relies on this characterization.

Theorem 9.1 (Lundberg and Lee [4]). The only additive feature attribution satisfying Local accuracy, Missingness and Consistency is given by the Shapley values of $v_{\mathbf{x}}$.

9.2 KernelSHAP

KernelSHAP computes the ϕ_j as the solution of a weighted least squares problem. Given a set Z of coalitions, minimize

$$L(\phi) = \sum_{\mathbf{z}' \in Z} \left[\hat{f}_{\mathbf{x}}(\mathbf{z}') - g(\mathbf{z}') \right]^2 \pi_{\mathbf{x}}(\mathbf{z}'), \quad \pi_{\mathbf{x}}(\mathbf{z}') = \frac{M-1}{\binom{M}{|\mathbf{z}'|} |\mathbf{z}'| (M - |\mathbf{z}'|)},$$

subject to $g(\mathbf{0}) = \hat{f}_{\mathbf{x}}(\mathbf{0})$ and $g(\mathbf{1}) = \hat{f}_{\mathbf{x}}(\mathbf{x})$. (The kernel is infinite at $|\mathbf{z}'| \in \{0, M\}$; in practice these two coalitions are imposed as constraints.)

Theorem 9.2 ([4]). *If $Z = \{0, 1\}^M$, the solution of this problem is the vector of Shapley values.*

Example 9.1 ($M = 2$). *With two features, the only coalitions that are not constrained are $(1, 0)$ and $(0, 1)$, and both have the same weight. Writing $v_1 = v_{\mathbf{x}}(\{1\})$, $v_2 = v_{\mathbf{x}}(\{2\})$ and $\Delta = \hat{f}_{\mathbf{x}}(\mathbf{x}) - \phi_0$, the problem is*

$$\min_{\phi_1, \phi_2} (v_1 - \phi_1)^2 + (v_2 - \phi_2)^2 \quad \text{subject to} \quad \phi_1 + \phi_2 = \Delta.$$

The solution moves (v_1, v_2) orthogonally onto the line $\phi_1 + \phi_2 = \Delta$, so each coordinate is corrected by half the missing amount:

$$\phi_1 = v_1 + \frac{1}{2}(\Delta - v_1 - v_2) = \frac{1}{2}v_1 + \frac{1}{2}(\Delta - v_2),$$

which is the Shapley value for $N = 2$. For the model x_1x_2 this gives back $\phi_1 = \frac{1}{2}(x_1 - \mu_1)(x_2 + \mu_2)$.

Remark: For $M = 4$ the kernel gives weight $\frac{1}{4}$ to coalitions of size 1 and 3, and $\frac{1}{8}$ to coalitions of size 2. As with the Shapley weights in Section 3.2, the extreme sizes get more weight. This is the opposite of LIME, where the weight depends on the distance between the perturbed input and $\mathbf{x}$. When Z is a sample, the coalitions of size 1 and $M - 1$ are usually all included because they have the largest weights.

9.3 TreeSHAP

For tree ensembles the value function can be evaluated by walking down the trees, and the exact Shapley values can be computed in $O(TLD^2)$ time instead of $O(TL2^M)$, where T is the number of trees, L the number of leaves and D the depth [5]. By Additivity it is enough to do this tree by tree and combine the results the same way the ensemble combines predictions. There are two versions. The *interventional* one computes the Shapley values of (3). The *path-dependent* one replaces the marginal expectation by the proportions of training points going left and right at each split, which is closer to the conditional value function and has the same issue as in Section 8.

9.4 Permutation estimator

Instead of sampling one feature at a time, sample an ordering π and add the features one by one. One pass of $M + 1$ model evaluations gives a marginal contribution for every feature. The contributions along one ordering telescope to $\hat{f}_{\mathbf{x}}(\mathbf{x}) - \hat{f}_{\mathbf{x}}(\mathbf{0})$, so Local accuracy holds for every sample size. Using each ordering together with its reverse reduces the variance.

9.5 Global explanations

Given SHAP values $\phi_j^{(i)}$ for instances $i = 1, \dots, n$, all in the units of the prediction, we can aggregate:

- **Feature importance:** $I_j = \frac{1}{n} \sum_{i=1}^n |\phi_j^{(i)}|$. See [1, Figure 18.4].
- **Summary plot:** all points $\phi_j^{(i)}$, one row per feature, colored by $x_j^{(i)}$. See [1, Figure 18.5].
- **Dependence plot:** the points $\{(x_j^{(i)}, \phi_j^{(i)})\}_{i=1}^n$. If ϕ_j only depended on x_j (no interactions) these points would lie on a curve; vertical spread at a fixed x_j comes from interactions. See [1, Figure 18.6].
- **Clustering:** cluster instances by their vectors $(\phi_1^{(i)}, \dots, \phi_M^{(i)})$. See [1, Figure 18.8].

The force plots in [1, Figure 18.3] show the decomposition $\phi_0 + \sum_j \phi_j = \hat{f}(\mathbf{x})$, drawn as a bar starting at $\phi_0 = 0.47$.

SHAP interaction values. For $i \neq j$,

$$\phi_{i,j} = \sum_{S \subseteq I \setminus \{i,j\}} \frac{|S|!(M - |S| - 2)!}{2(M - 1)!} \delta_{ij}(S),$$

where

$$\delta_{ij}(S) = v_{\mathbf{x}}(S \cup \{i, j\}) - v_{\mathbf{x}}(S \cup \{i\}) - v_{\mathbf{x}}(S \cup \{j\}) + v_{\mathbf{x}}(S).$$

For $\hat{f} = x_1 x_2$ ($M = 2$) the only term is $S = \emptyset$ with weight $\frac{1}{2}$, and $\delta_{12}(\emptyset) = (x_1 - \mu_1)(x_2 - \mu_2)$. Hence $\phi_{1,2} = \phi_{2,1} = \frac{1}{2}(x_1 - \mu_1)(x_2 - \mu_2)$, and the main effects are $\phi_{1,1} = \phi_1 - \phi_{1,2} = \mu_2(x_1 - \mu_1)$ and $\phi_{2,2} = \mu_1(x_2 - \mu_2)$. See [1, Figure 18.7] for a dependence plot colored by the strongest interaction.

9.6 Example: TreeSHAP on the housing data

Example 9.2 (SHAP for a gradient boosting model). *On the same data as Section 7 we train a gradient boosting model with 300 trees of depth 4 ($R^2 = 0.84$ on the test set), and compute SHAP values with interventional TreeSHAP for 1,000 test districts, using the same 100 background districts.*

Local explanation. Figure 4 shows the San Francisco district again, now drawn as a waterfall: we start at $\phi_0 = \$205\text{k}$ and add the SHAP values one at a time until we reach the prediction $\hat{f}(\mathbf{x}) = \$431\text{k}$. This is Local accuracy. Note that the explanation is not the same as in Figure 1: here longitude gets $+\$187\text{k}$ and latitude $-\$89\text{k}$. The two models predict similar values for this district but use the features differently, and the Shapley values explain the model, not the housing market.

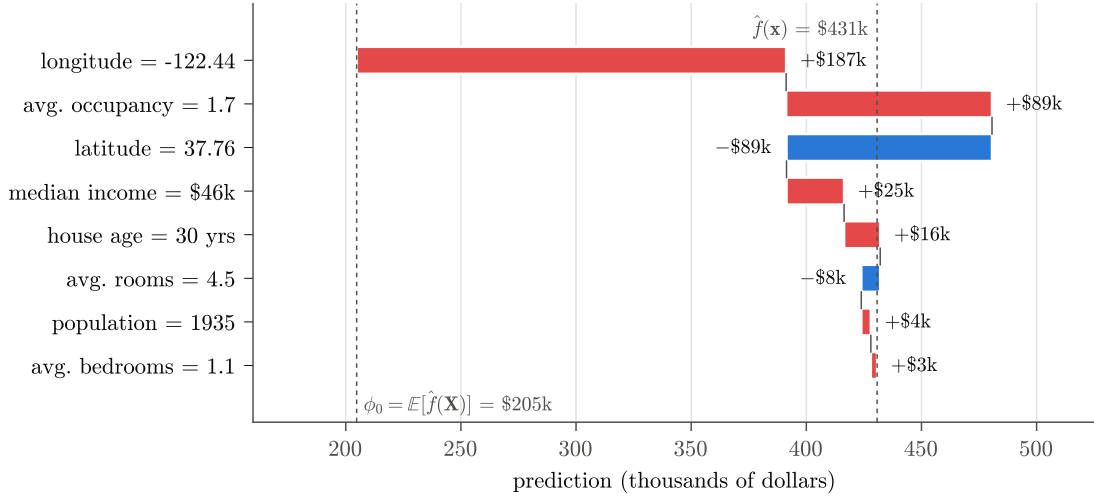


Figure 4: SHAP values for the San Francisco district, gradient boosting model, sorted by absolute value.

Global explanation. Figure 5 aggregates the 1,000 local explanations. In (a), latitude and longitude are the most important features, with a mean absolute effect of about \$91k and \$80k. In (b), each point is one district: for median income, high values (red) have positive SHAP values and low values (blue) negative ones, so the effect is increasing. For latitude it is the opposite, high latitude (north) lowers the prediction.

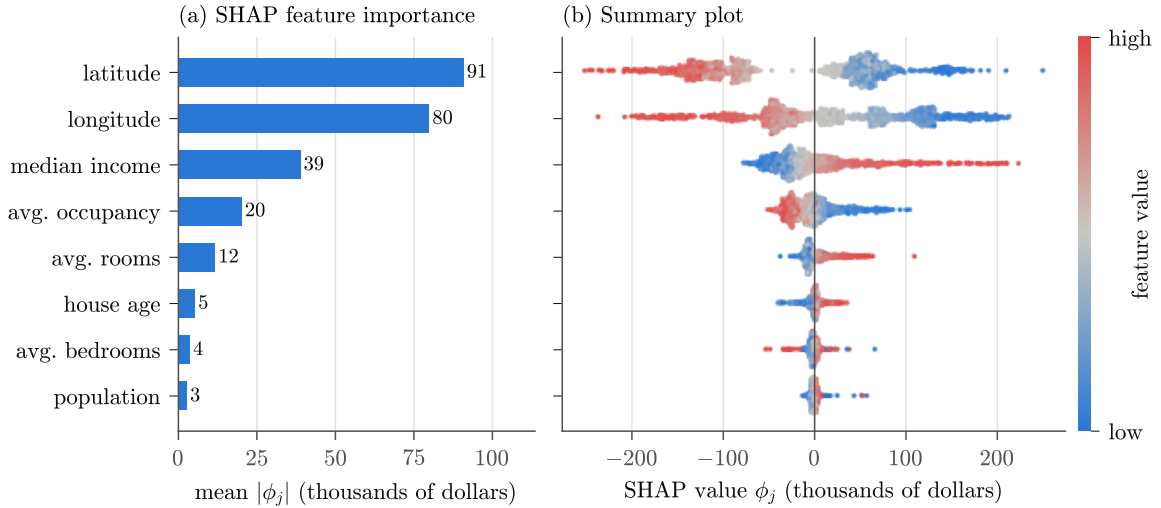


Figure 5: (a) SHAP feature importance I_j . (b) Summary plot: one point per district and feature, colored by the quantile of the feature value.

Figure 6 shows two dependence plots. For median income (a) the points lie close to an increasing curve that flattens for the highest incomes, so this feature acts almost on its own. For latitude (b) there is a lot of vertical spread at a fixed latitude, and the color shows that districts at the same latitude but different longitudes get different values: latitude and

longitude together encode the location, so the effect of one depends on the other. This is the interaction that the SHAP interaction values quantify.

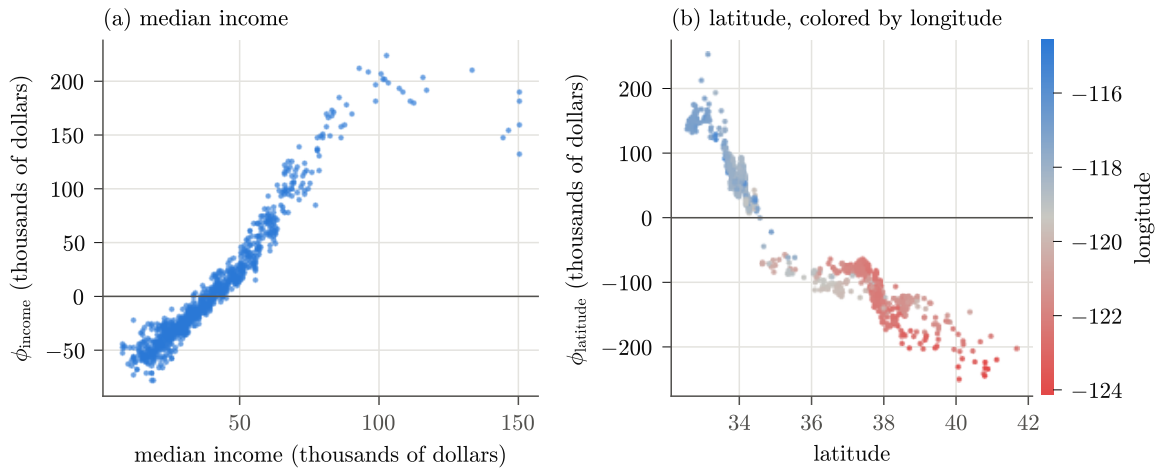


Figure 6: SHAP dependence plots. (a) Median income. (b) Latitude, colored by longitude.

10 Discussion Questions

1. Figures 17.4 and 17.5 give different explanations of the same prediction. Is one of them wrong?
2. If x_2 is an exact copy of x_1 and the model only uses x_1 , what are the interventional and the conditional Shapley values?
3. Can SHAP feature importance rank a feature high while permutation feature importance ranks it low?

References

- [1] C. Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Online edition, <https://christophm.github.io/interpretable-ml-book/>, accessed October 20, 2026. Chapter 17: <https://christophm.github.io/interpretable-ml-book/shapley.html>. Chapter 18: <https://christophm.github.io/interpretable-ml-book/shap.html>.
- [2] L. S. Shapley. A Value for n -Person Games. In H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games II*, Annals of Mathematics Studies 28, pp. 307–317. Princeton University Press, 1953.
- [3] E. Štrumbelj and I. Kononenko. Explaining Prediction Models and Individual Predictions with Feature Contributions. *Knowledge and Information Systems*, 41(3):647–665, 2014. doi:10.1007/s10115-013-0679-x.

- [4] S. M. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4765–4774, 2017. [arXiv:1705.07874](#).
- [5] S. M. Lundberg, G. G. Erion and S.-I. Lee. Consistent Individualized Feature Attribution for Tree Ensembles. [arXiv:1802.03888](#), 2018 (revised 2019).
- [6] M. Sundararajan and A. Najmi. The Many Shapley Values for Model Explanation. In *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119:9269–9278, 2020.
- [7] H. P. Young. Monotonic Solutions of Cooperative Games. *International Journal of Game Theory*, 14(2):65–72, 1985. [doi:10.1007/BF01769885](#).