Consistent Individualized Feature Attribution for Tree Ensembles

Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee

University of Washington

arXiv: 1802.03888

Reviewed by : Bill Zhang

University of Virginia

https://qdata.github.io/deep2Read/

Outline

Introduction

Basic Premise and Motivation

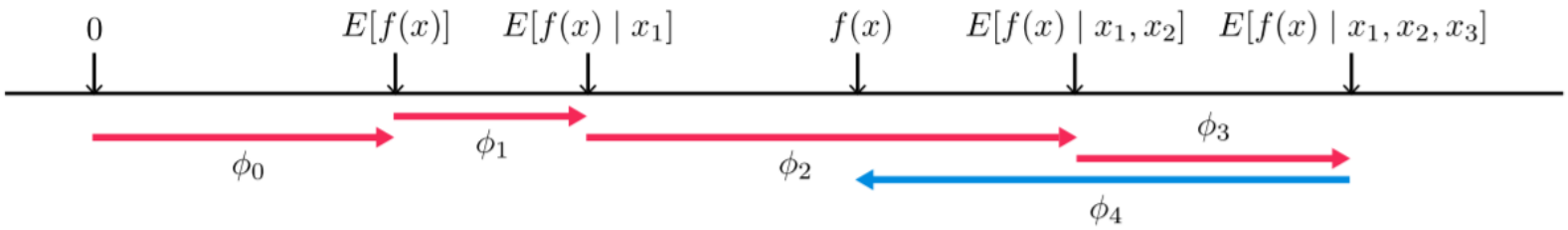
- Popular feature attribution methods for tree ensembles are inconsistent
- SHAP values (SHapley Additive exPlanation) theoretically optimal
- Propose method to reduce $O(TL2^M)$ to $O(TLD^2)$ where T is number of trees, L is max leaves in any tree, D is max depth
- Also propose Shapley interaction values for pairwise interactions

Inconsistencies

- Gain: total reduction of loss or impurity contributed by all splits for a given feature
- Split count: Count how many times a feature is used to split
- Permutation: Randomly permute value of a feature and observe change in model error
- Sabbas (tree-specific, rest are agnostic): similar to gain, but measure change in model's expected output
- All shown to be inconsistent; only SHAP consistent (detailed proof in omitted, uses additive feature attribution methods)

SHAP Overview

- $\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)]$



Tree SHAP

Exponential Estimation

- Challenges: Estimating $E\left[f(x)|x_S\right]$ efficiently, exponential complexity of SHAP equation

Algorithm 1 Estimating $E[f(x) \mid x_S]$

procedure EXPVALUE(x, S, $tree = \{v, a, b, t, r, d\}$)
  procedure G(j, w)
    if $v_j \neq internal$ **then**
      return $w \cdot v_j$
    else
      if $d_j \in S$ **then**
        return G(a_j, w) **if** $x_{d_j} \leq t_j$ **else** G(b_j, w)
      else
        return G(a_j, wr_{a_j}/r_j) + G(b_j, wr_{b_j}/r_j)
      end if
    end if
  end procedure
  return G(1, 1)
end procedure

Tree SHAP

Polynomial Estimation

- $O(TLD^2)$ time and $O(D^2 + M)$ memory
- Recursively keep track of what possible subsets flow down into each leaf of the tree
- Algorithm too long to include here (see paper)

SHAP Interaction Values

- Consider Shapley Interaction Matrix
- $\Phi_{i,j} = \sum_{S \subseteq N \setminus \{i,j\}} \frac{|S|!(M-|S|-2)!}{2(M-1)!} \nabla_{ij}(S)$ when $i \neq j$
- $$\begin{aligned} \nabla_{ij}(S) &= f_x(S \cup \{i,j\}) - f_x(S \cup \{i\}) - f_x(S \cup \{j\}) + f_x(S) \\ &= f_x(S \cup \{i,j\}) - f_x(S \cup \{j\}) - [f_x(S \cup \{i\}) - f_x(S)] \end{aligned}$$
- Then, can define main effects for a prediction as difference between Shapley value and all SHAP interaction values $\Phi_{i,i} = \phi_i - \sum_{j \neq i} \Phi_{i,j}$

Experiments and Applications

Agreement with Human Intuition

- Participants shown a tree model regarding risk for a certain disease; having both cough and fever increased baseline risk from 20 to 80 percent
- Participants attributed 60 point change to either cough or fever based on tree

Experiments and Applications

Agreement with Human Intuition

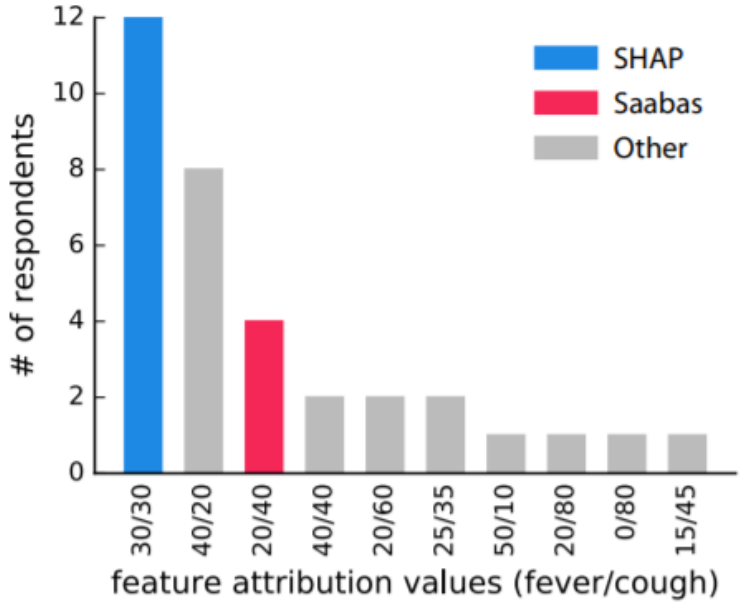


Figure 3: Feature attribution values from 34 participants shown the tree from Model A in Figure 1. The first number represents the allocation to the Fever feature, while the second represents the allocation to the Cough feature. Participants from Amazon Mechanical Turk were not selected for machine learning expertise. No constraints were placed on the feature attribution values users entered.

Experiments and Applications

Computational Performance

- Alg 2 provides significant speedup

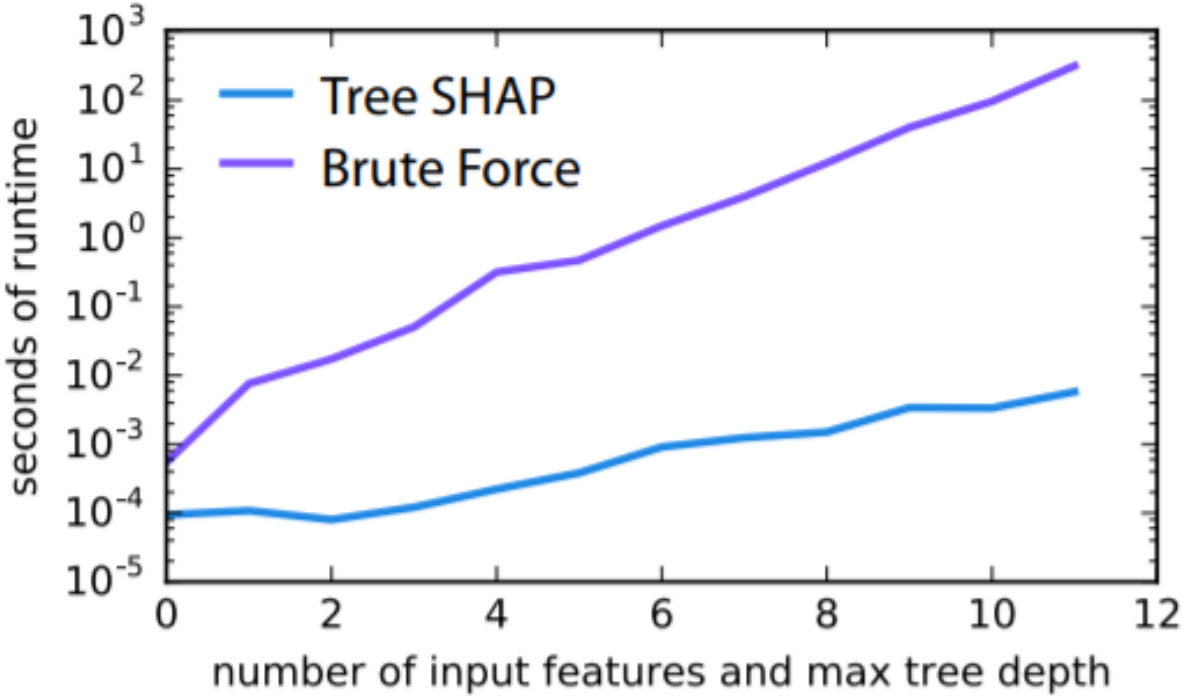


Figure 5: Runtime improvement of Algorithm 2 over using Equation 2 and Algorithm 1. An XGBoost model with 50 trees was trained using an equally increasing number of input features and max tree depths. The time to explain one input vector is reported.

Experiments and Applications

Supervised Clustering

- Supervised clustering with feature attributions to naturally convert all input features into values with same units as model output
- Test on UCI census dataset; use demographic data to predict if person is likely to make more than $50k annually

Experiments and Applications

Supervised Clustering

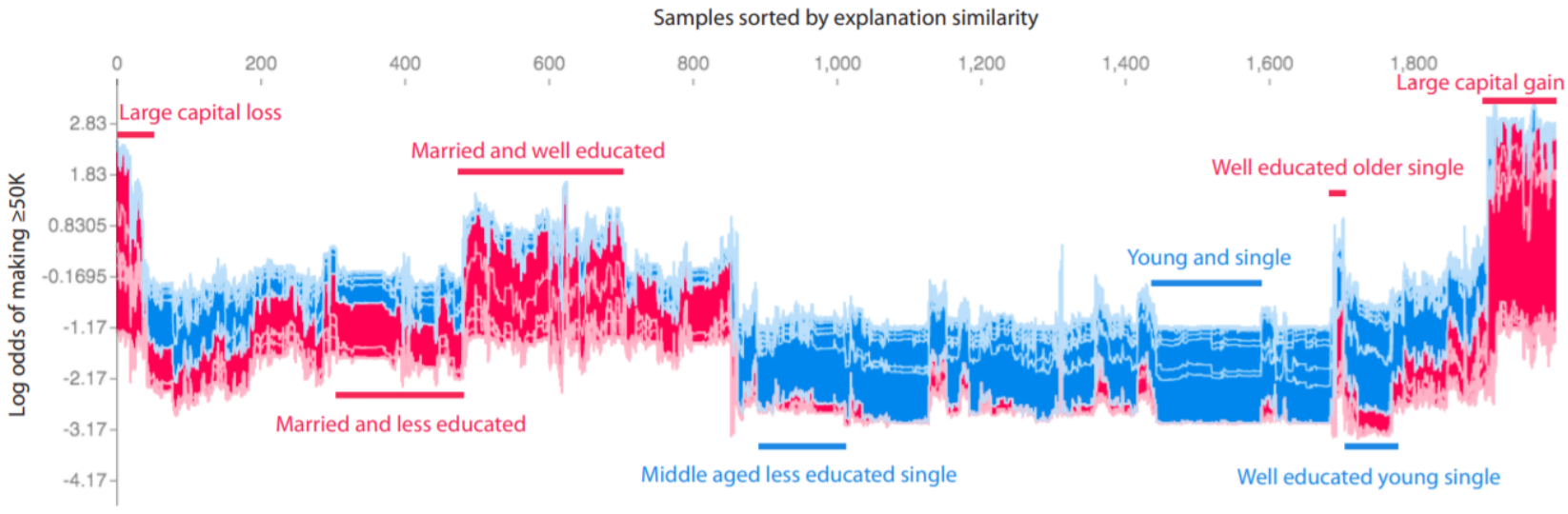
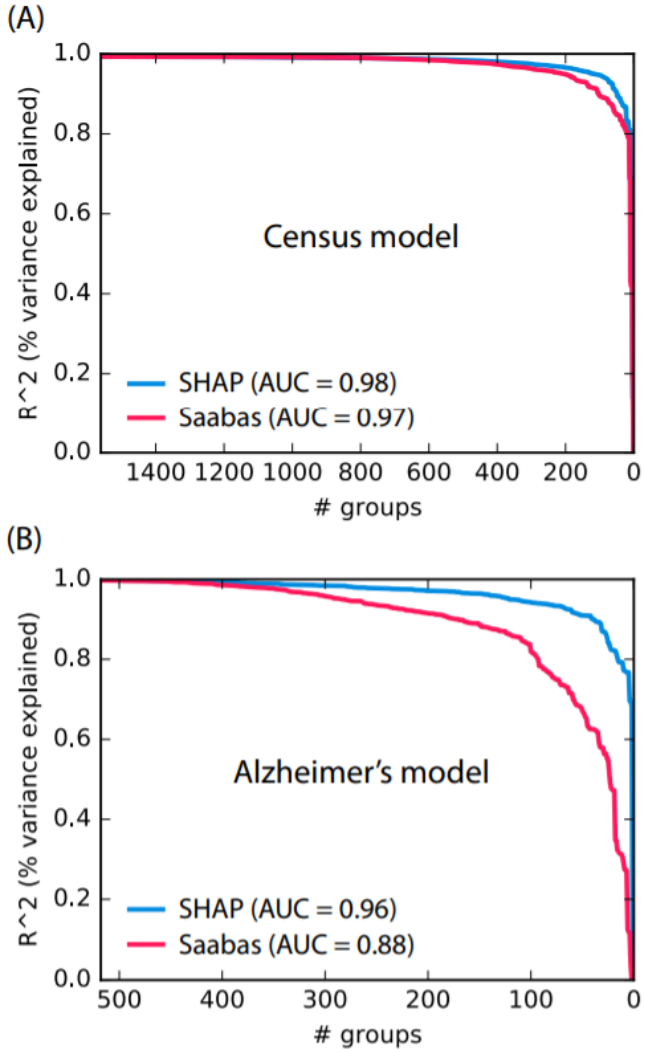


Figure 4: Supervised clustering with SHAP feature attributions in the UCI census dataset identifies among 2,000 individuals distinct subgroups of people that share similar reasons for making money. An XGBoost model with 500 trees of max depth six was trained on demographic data using a shrinkage factor of $\eta = 0.005$. This model was then used to predict the log odds that each person makes $\geq \$50K$. Each prediction was explained using Tree SHAP, and then clustered using hierarchical agglomerative clustering (imagine a dendrogram above the plot joining the samples). Red feature attributions push the score higher, while blue feature attributions push the score lower (as in Figure 2 but rotated 90°). A few of the noticeable subgroups are annotated with the features that define them.

Experiments and Applications

Supervised Clustering



Experiments and Applications

Identification of Influential Features

- Perturb most important feature and observe change in model prediction

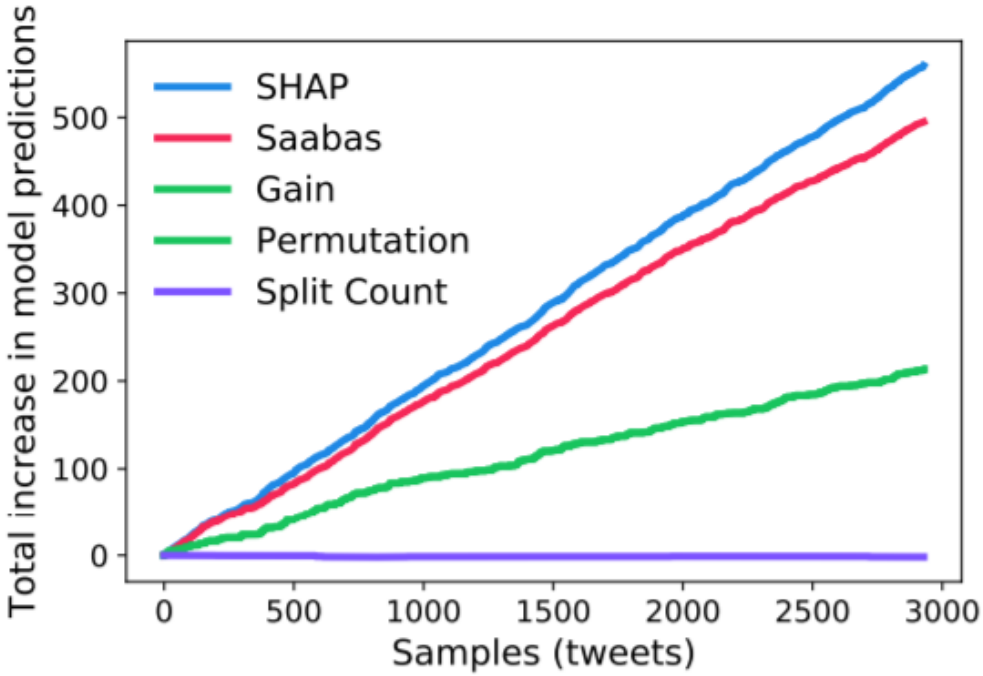


Figure 7: The total increase in a sentiment model's output when the most negative feature is replaced. Five different attribution methods were used to determine the most negative feature for each sample. The higher the total increase in model output, the more accurate the attribution method was at identifying the most influential negative feature.

Experiments and Applications

SHAP Plots

- SHAP values are individualized to predictions, not global feature attribution values
- Can have new, richer visualizations
- Summary plots to see global feature importance, distribution of data, and significance of each feature as its values changes
- Dependence plots to see how importance changes as value varies

Experiments and Applications

SHAP Summary Plots

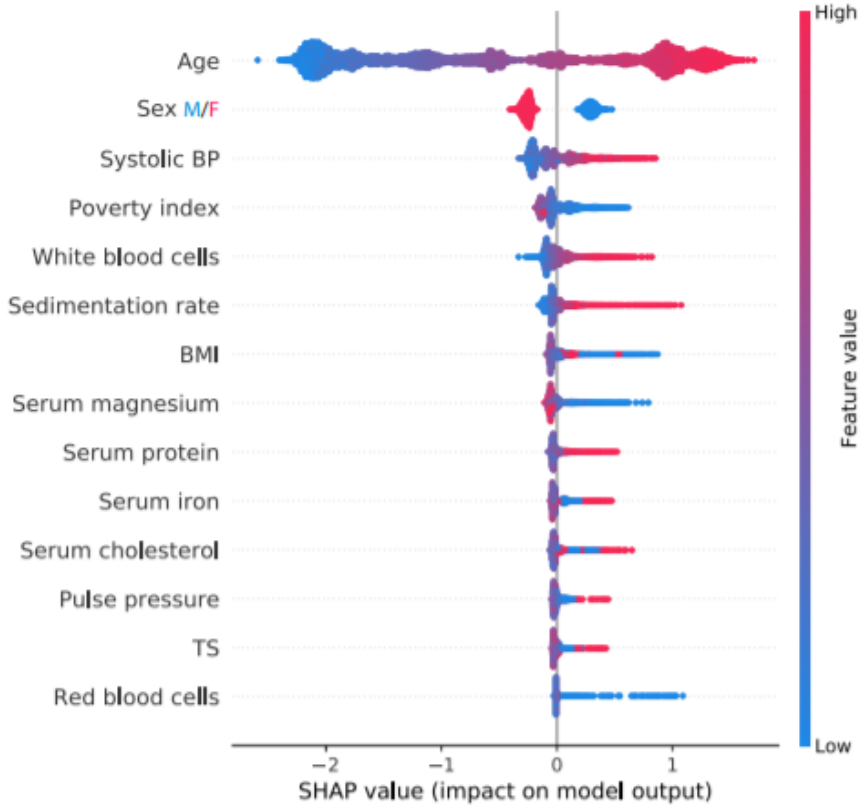


Figure 8: SHAP summary plot of a 14 feature XGBoost survival model on 20 year mortality followup data from NHANES I [18]. The higher the SHAP value of a feature, the higher your log odds of death in this Cox hazards model. Every individual in the dataset is run through the model and a dot is created for each feature attribution value, so one person gets one dot on each feature's line. Dot's are colored by the feature's value for that person and pile up vertically to show density.

Experiments and Applications

SHAP Dependence Plots

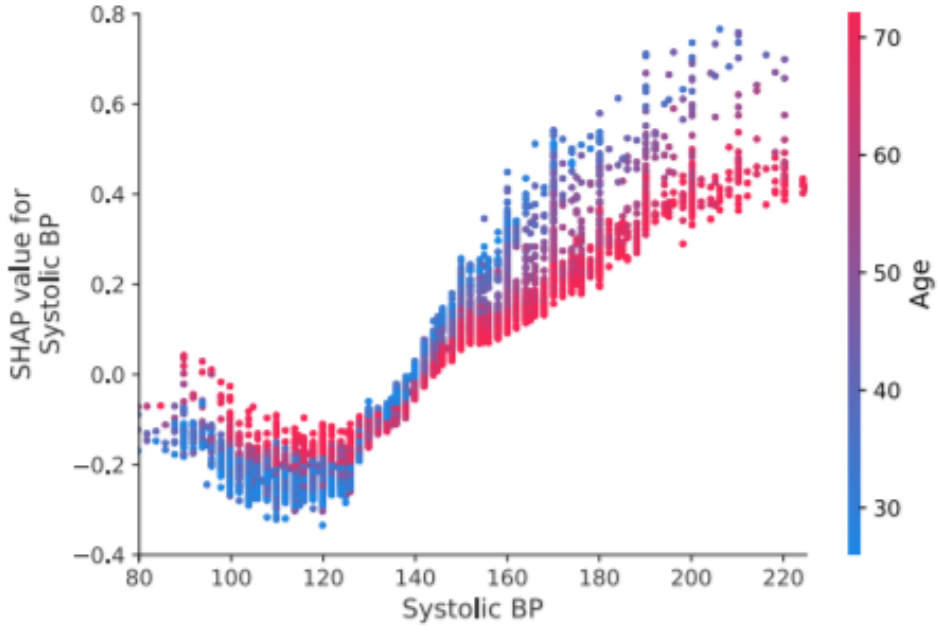


Figure 9: Each dot is a person. The x-axis is their systolic blood pressure and the y-axis is the SHAP value attributed to their systolic blood pressure. Higher SHAP values represent higher risk of death due to systolic blood pressure. Coloring each dot by the person's age reveals that high blood pressure is more concerning to the model when you are young (this represents an interaction effect).

Experiments and Applications

SHAP Interaction Plots

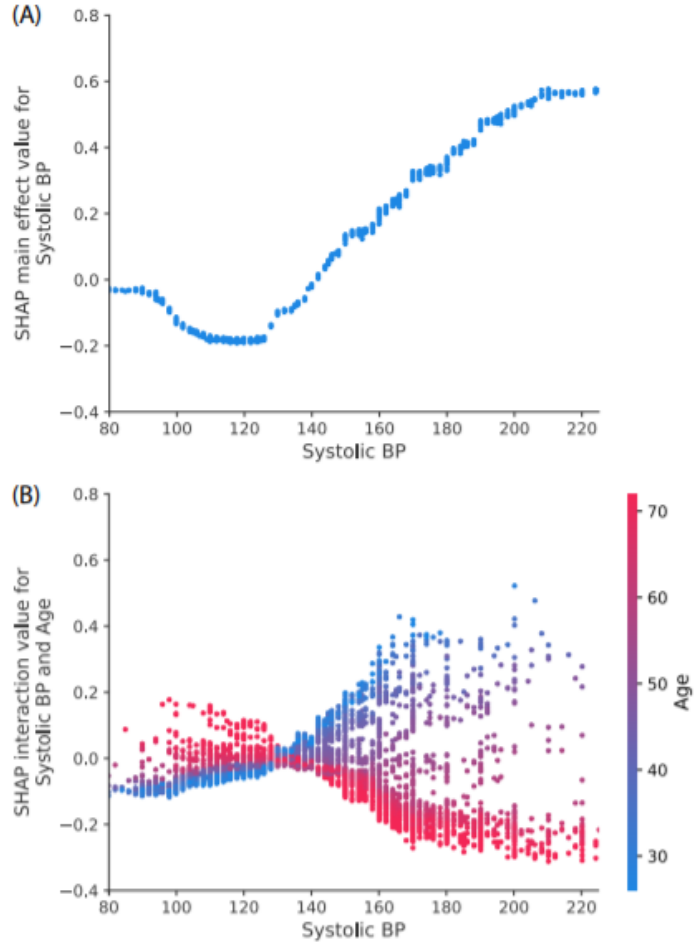


Figure 10: SHAP interaction values separate the impact of systolic blood pressure into main effects (A; Equation 6) and interaction effects (B; Equation 3). Systolic blood pressure has a strong interaction effect with age, so the sum of (A) and (B) nearly equals Figure 9. There is very little vertical dispersion in (A) since all the interaction effects have been removed.

Conclusion

- Showed SHAP as only consistent feature attribution method
- Proposed polynomial time estimation of SHAP value for tree ensembles
- Defined SHAP interaction values to measure pairwise relationships
- Opened up practical opportunities in supervised clustering, SHAP summary plots, and SHAP dependence plots for tree models

References

- https://arxiv.org/pdf/1802.03888.pdf