A UNIFIED APPROACH TO INTERPRETING MODEL PREDICTIONS

Gökalp Coşgun & Muhammed Berkay Şan

Trade-off Between Interpretability and Accuracy

- The ability to correctly interpret the output of a prediction model is very important.
- In many situations, knowing why a model produces a certain forecast can be just as important as knowing if the prediction is accurate. There is a conflict between accuracy and interpretability since the maximum accuracy for big current datasets is sometimes attained by complex models that even experts have trouble understanding, such ensemble or deep learning models.
- In short, there is a trade-off between making a prediction and its interpretability.

A Unified Framework for Interpreting Predictions

- Recently, a variety of approaches have been offered out to clarify things in understanding the predictions of complicated models, but it is sometimes unclear how these methods relate to one another and when one way is better than another.
- SHAP(SHapley Additive exPlanations) is introduced in the paper, a unified framework for understanding predictions, to overcome this issue.
- The approach leads to three results that bring clarity to the growing space of methods:

1. Explanation Model
2. Unique solution to the entire class of additive feature attribution methods
3. SHAP value estimation methods

1) Explanation Model

- The model itself is the best explanation for a simple model; it perfectly represents itself and is simple to understand. We cannot use the original model as the best explanation for complex models such as ensemble methods or deep networks because it is difficult to understand.
- We must instead employ a simpler explanation model, which we define as any interpretable approximation of the original model. Six current explanation methods from the literature are all shown below to use the same explanation model. This previously overlooked unity has intriguing implications, which we will discuss in later sections.

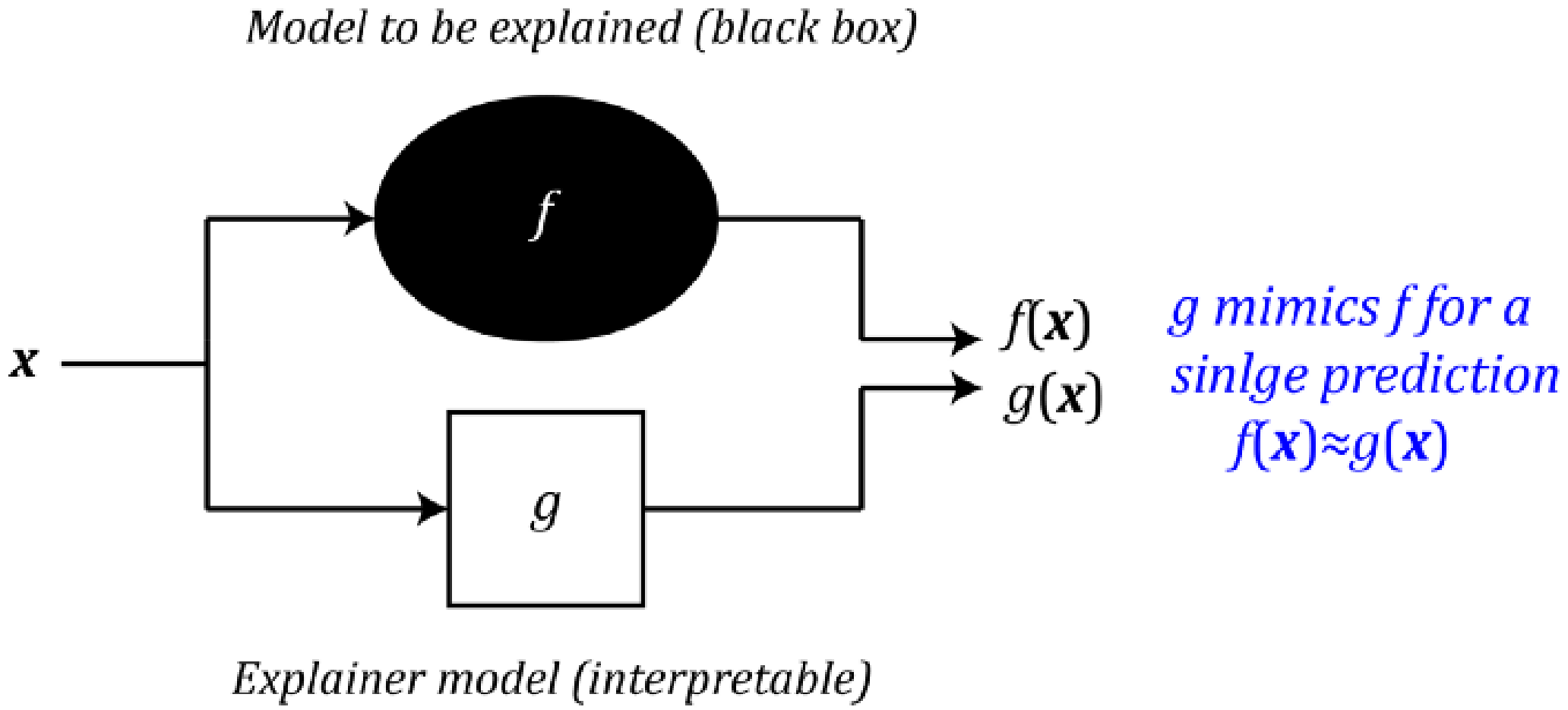
1) Explanation Model

- Let f be the original prediction model to be explained and g the explanation model. Explanation models often use simplified inputs x' that map to the original inputs through a mapping function $x = h_x(x')$. Local methods try to ensure $g(z') \approx f(h_x(z'))$ whenever $z' \approx x'$ (Note that $h_{x(x')} = x$ even though x' may contain less information than x because h_x is specific to the current input x.)

Definition 1 Additive feature attribution methods *have an explanation model that is a linear function of binary variables:*

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \tag{1}$$

where $z' \in \{0,1\}^M$, M *is the number of simplified input features, and* $\phi_i \in \mathbb{R}$.


Model to be explained (black box)
f
x
f(x)
g(x)
g
Explainer model (interpretable)
g mimics f for a
sinlge prediction
f(x)≈g(x)

1.1) LIME

- The LIME method interprets individual model predictions based on locally approximating the model around a given prediction [1]. The local linear explanation model that LIME uses adheres to Equation 1 exactly and is thus an additive feature attribution method. LIME refers to simplified inputs x' as "interpretable inputs," and the mapping $x = h_x(x')$ converts a binary vector of interpretable inputs into the original input space. Different types of h_x mappings are used for different input spaces.

To find ϕ, LIME minimizes the following objective function:

$$\xi = \underset{g \in \mathcal{G}}{\arg\min}\ L(f, g, \pi_{x'}) + \Omega(g). \tag{2}$$

1.2) DeepLIFT

- DeepLIFT was recently proposed as a recursive prediction explanation method for deep learning [4, 3]. It attributes to each input xi a value $C_{\Delta x_i y_{\Delta y}}$ that represents the effect of that input being set to a reference value as opposed to its original value. This means that for DeepLIFT, the mapping $x = h_x(x')$ converts binary values into the original inputs, where 1 indicates that an input takes its original value, and 0 indicates that it takes the reference value. The reference value, though chosen by the user, represents a typical uninformative background value for the feature.

DeepLIFT uses a "summation-to-delta" property that states:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o, \tag{3}$$

where $o = f(x)$ is the model output, $\Delta o = f(x) - f(r)$, $\Delta x_i = x_i - r_i$, and r is the reference input. If we let $\phi_i = C_{\Delta x_i \Delta o}$ and $\phi_0 = f(r)$, then DeepLIFT's explanation model matches Equation 1 and is thus another additive feature attribution method.

1.3) Layer-wise Relevance Propagation

- The layer-wise relevance propagation method interprets the predictions of deep networks [2]. As noted by Shrikumar et al., this menthod is equivalent to DeepLIFT with the reference activations of all neurons fixed to zero. Thus, $x = h_x(x')$ converts binary values into the original input space, where 1 means that an input takes its original value, and 0 means an input takes the 0 value. Layer-wise relevance propagation's explanation model, like DeepLIFT's, matches Equation 1.

1.4) Classic Shapley Value Estimation

- Shapley regression values are feature importances for linear models in the presence of multicollinearity. This method requires retraining the model on all feature subsets $S \subseteq F$, where F is the set of all features. It assigns an importance value to each feature that represents the effect on the model prediction of including that feature.

2) Simple Properties Uniquely Determine Additive Feature Attributions

- A surprising attribute of the class of additive feature attribution methods is the presence of a single unique solution in this class with three desirable properties (described below).

1. **Property 1 (Local accuracy)**

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{5}$$

The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$.

2) Simple Properties Uniquely Determine Additive Feature Attributions

2. **Property 2 (Missingness)**

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{5}$$

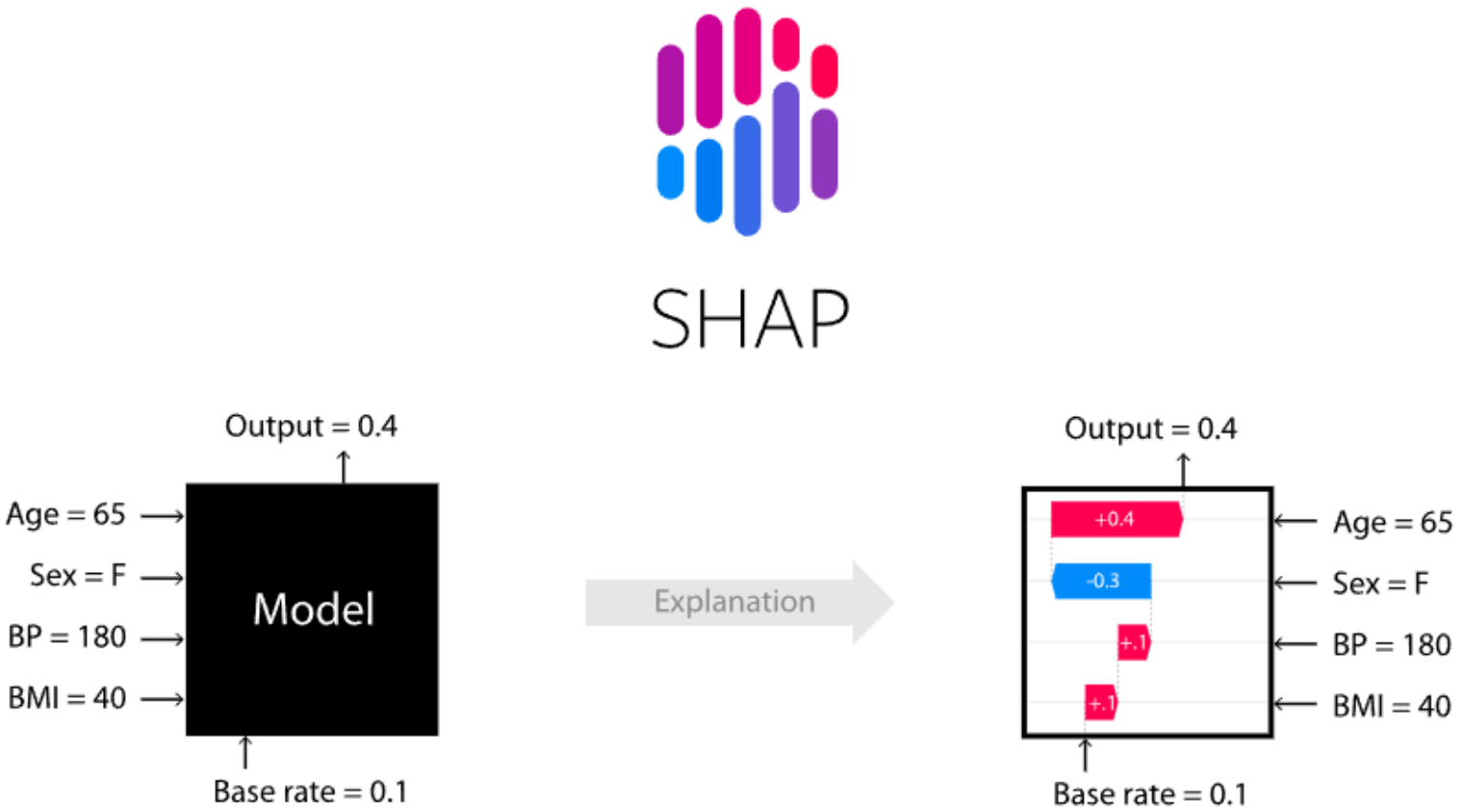
The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$.

3. **Property 3 (Consistency)**

Consistency states that if a model changes so that some simplified input's contribution increases or stays the same regardless of the other inputs, that input's attribution should not decrease.

Theorem 1 : Only one possible explanation model g follows Definition 1 and satisfies Properties 1, 2, and 3**.** (Proved in paper)

3) SHAP (Shapley Additive Explanation) Values



3) SHAP (Shapley Additive Explanation) Values

- We propose SHAP values as a unified measure of feature importance.
- The explainer model proposed in the article shows how the features contribute to the output of the original model, using SHAP values.

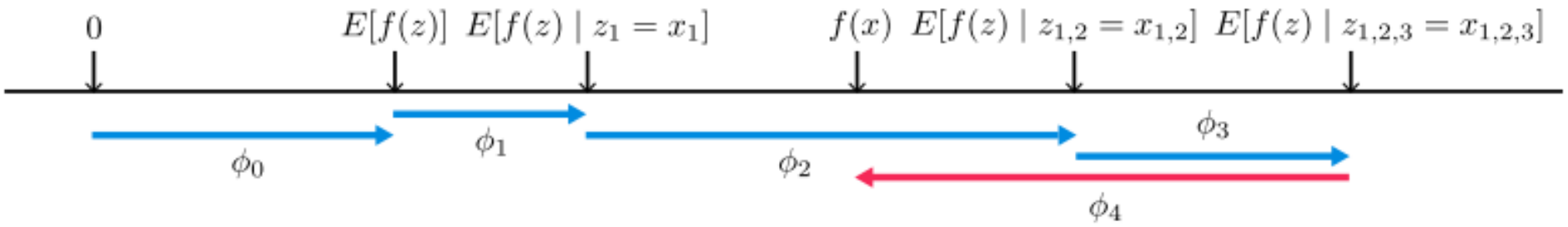


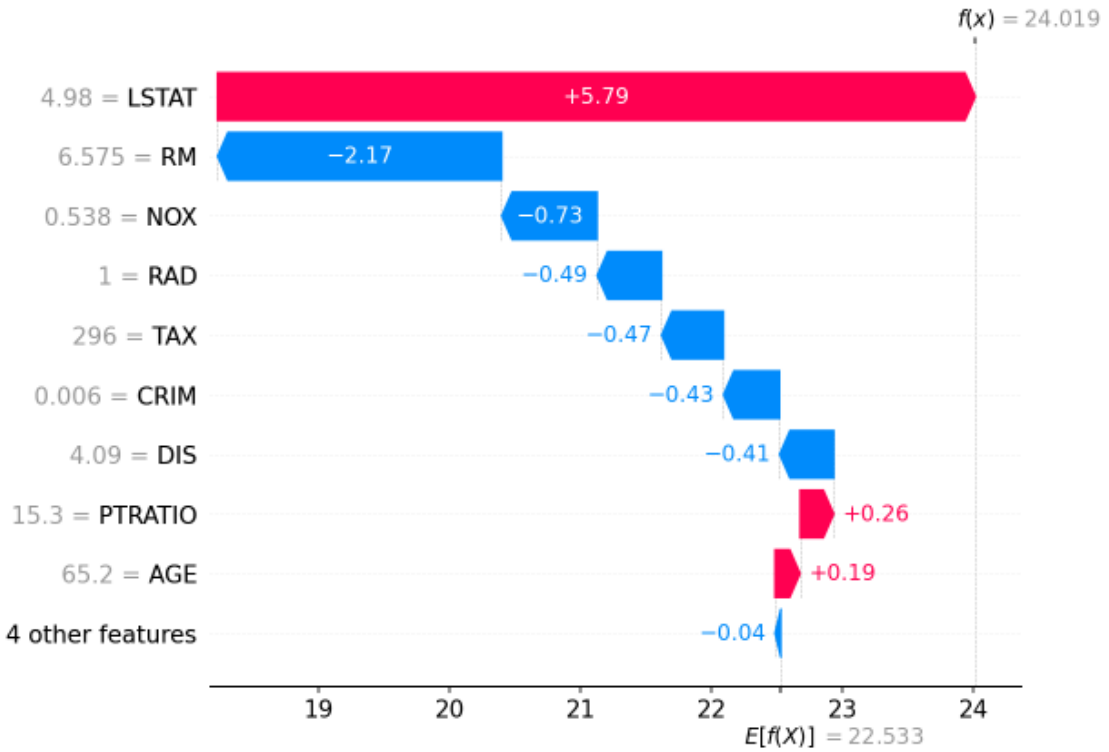
Figure 1: SHAP (SHapley Additive exPlanation) values attribute to each feature the change in the expected model prediction when conditioning on that feature. They explain how to get from the base value $E[f(z)]$ that would be predicted if we did not know any features to the current output $f(x)$. This diagram shows a single ordering. When the model is non-linear or the input features are not independent, however, the order in which features are added to the expectation matters, and the SHAP values arise from averaging the ϕ_i values across all possible orderings.

```python
import xgboost
import shap

# train an XGBoost model
X, y = shap.datasets.boston()
model = xgboost.XGBRegressor().fit(X, y)

# explain the model's predictions using SHAP
# (same syntax works for LightGBM, CatBoost, scikit-learn, transformers, Spark, etc.)
explainer = shap.Explainer(model)
shap_values = explainer(X)

# visualize the first prediction's explanation
shap.plots.waterfall(shap_values[0])
```



Conclusion

- The growing tension between the accuracy and interpretability of model predictions has motivated the development of methods that help users interpret predictions. The SHAP framework identifies the class of additive feature importance methods (which includes six previous methods) and shows there is a unique solution in this class that adheres to desirable properties.

REFERENCES

- [1] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why should i trust you?: Explaining the predictions of any classifier”. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. 2016, pp. 1135–1144.
- [2] Sebastian Bach et al. “On pixel-wise explanations for non-linear classifier decisions by layerwise relevance propagation”. In: PloS One 10.7 (2015), e0130140.
- [3] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning Important Features Through Propagating Activation Differences”. In: arXiv preprint arXiv:1704.02685 (2017).
- [4] Avanti Shrikumar et al. “Not Just a Black Box: Learning Important Features Through Propagating Activation Differences”. In: arXiv preprint arXiv:1605.01713 (2016)
- Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).