



Model-agnostic variable importance for predictive uncertainty

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Wood, D., Papamarkou, T., Benatan, M., & Allmendinger, R. (2023). *Model-agnostic variable importance for predictive uncertainty: an entropy-based approach*.

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



Model-agnostic variable importance for predictive uncertainty: an entropy-based approach

Danny Wood¹, Theodore Papamarkou², Matt Benatan^{1,3},
Richard Allmendinger^{1*}

¹Alliance Manchester Business School, The University of Manchester,
Manchester, M13 9PL, UK.

²Department of Mathematics, The University of Manchester,
Manchester, M13 9PL, UK.

³Sonos Experience Limited, London, SE1 3ER, UK.

*Corresponding author(s). E-mail(s):

richard.allmendinger@manchester.ac.uk;

Contributing authors: danny.wood@manchester.ac.uk;

theo.papamarkou@manchester.ac.uk; matt.benatan@sonos.com;

Abstract

In order to trust the predictions of a machine learning algorithm, it is necessary to understand the factors that contribute to those predictions. In the case of probabilistic and uncertainty-aware models, it is necessary to understand not only the reasons for the predictions themselves, but also the model's level of confidence in those predictions. In this paper, we show how existing methods in explainability can be extended to uncertainty-aware models and how such extensions can be used to understand the sources of uncertainty in a model's predictive distribution. In particular, by adapting permutation feature importance, partial dependence plots, and individual conditional expectation plots, we demonstrate that novel insights into model behaviour may be obtained and that these methods can be used to measure the impact of features on both the entropy of the predictive distribution and the log-likelihood of the ground truth labels under that distribution. With experiments using both synthetic and real-world data, we demonstrate the utility of these approaches in understanding both the sources of uncertainty and their impact on model performance.

Keywords: Entropy, feature importance, predictive uncertainty, variable importance

1 Introduction

A common criticism leveled at models in modern machine learning is that their complexity makes it difficult to understand the reasons for their decisions. This is especially problematic when applying these models in domains where decisions must be carefully justified, such as healthcare and judiciary decision making (Kelly et al, 2020), making explainability an important tool when developing trustworthy AI. Another key factor in the development of trustworthy models is uncertainty quantification: the ability of models to give accurate assessments of the uncertainty inherent in their predictions. While both explainability and uncertainty quantification are vital for the use of machine learning in high-stakes applications, remarkably little work has been done at their intersection.

Traditional explainable machine learning seeks to reveal the reasons for a model’s output or the reasons for its level of accuracy. These remain important pieces of information for understanding uncertainty-aware models, though equally important is the ability to determine the sources of uncertainty: the features that cause the model to be more or less confident in its prediction. Although there has been some recent work exploring this area for Bayesian neural networks (Depeweg et al, 2017; Chai, 2018; Antoran et al, 2021), to the best of our knowledge, there has been only one recent attempt at a truly model-agnostic approach (Watson et al, 2023).

In this paper, we demonstrate that a range of existing simple techniques for explaining model output and performance can be modified to explain both the uncertainty and likelihood of the predictive distributions of uncertainty-aware models. In particular, we introduce novel adaptations of permutation feature importance (PFI), partial dependence plots (PDP) and individual conditional expectation (ICE) plots to explain how each feature available to a model affects its predictive distribution. We explore two complementary approaches: one looking at feature importance for the negative log-likelihood (loss), the other looking at the predictive uncertainty. For the first approach, we introduce Likelihood-PFI, Likelihood-PDP and Likelihood-ICE; likelihood-based variants of the methods listed above, which allow feature importance to be measured in terms of the effect a feature has on the model’s negative log-likelihood. For the second approach, we measure the uncertainty of the model’s predictive distribution through that distribution’s entropy, introducing Entropy-PFI, Entropy-PDP and Entropy-ICE. We examine the properties of both sets of empirical measures through the use of carefully constructed synthetic datasets, and demonstrate how they can be used in a model-agnostic way to derive insights from models trained on real-world datasets in both classification and regression settings.

In recent years, methods like the ones listed above have faced scrutiny due to the potential for misleading results in the presence of statistical dependencies between features (Hooker et al, 2021). These results are due to the fact that permutation-based methods break dependencies, which can force the model to extrapolate. While this extrapolation can be an issue for traditional permutation-based methods, we will show that this is not the case for Entropy-PFI; instead, the effect that these dependencies have on model confidence is a critical component of what is being measured. From this interpretation, measuring Entropy-PFI alongside Likelihood-PFI can help

to detect issues when dealing with feature dependencies, while not requiring training of additional models.

The remainder of the paper is structured as follows. In Section 2, we briefly review the fields of uncertainty quantification and feature importance, as well as give a more in-depth introduction to the main feature importance methods of interest in this work. In Section 3, we introduce our novel likelihood and entropy-based feature importance methods, as well as demonstrate their properties through experiments on synthetic datasets. In Section 4, we demonstrate how these techniques can be used to gain new insights on real-world datasets. Finally, in Section 5, we summarise our findings and give suggestions for future work.

2 Background

This section briefly reviews the fields of uncertainty quantification and feature importance, before introducing notation and describing some of the key methods in more detail.

2.1 Uncertainty quantification

Accurate measures of uncertainty require that we are able to construct models that output a distribution over possible outcomes, as opposed to a single value deemed most likely by the model. To this end, several approaches have been proposed to create uncertainty-aware models, such as Gaussian processes (Williams and Rasmussen, 2006) and Bayesian neural networks. For Bayesian neural networks, several approaches have been developed, including fully Bayesian networks (Neal, 2012), approximations such as Monte Carlo dropout (Gal and Ghahramani, 2016), as well as a range of calibration methods (Guo et al, 2017). These uncertainty-aware models have their own unique strengths and weaknesses, though they all raise a set of common questions: Can we identify the sources of uncertainty for a model? And how do the features that increase model confidence differ from those that increase model performance? That is, how do we explain the uncertainty of a model?

Much previous work on explaining uncertainty revolves around decomposing the uncertainty into two types: *epistemic* uncertainty, i.e., uncertainty due to the finite amount of data available and limitations of the model; and *aleatoric* uncertainty, i.e., uncertainty that is inherent in the system that we are observing, which cannot be reduced by collecting more data (Depeweg et al, 2018). Epistemic uncertainty can be further decomposed into uncertainty about the suitability of the chosen model (structural uncertainty) and uncertainty in the choice of parameters given the specification of the chosen model (Liu et al, 2019). Various efforts have been made to identify how much uncertainty comes from each source: explicitly modelling aleatoric uncertainty via a noise parameter, estimating the role of epistemic uncertainty by modelling the density of the data in latent spaces (Mukhoti et al, 2023) and estimating aleatoric uncertainty by reducing epistemic uncertainty via ensembling (Shaker and Hüllermeier, 2020), while other work critically examines the validity of this approach of decomposing uncertainty (Wimmer et al, 2023).

However, we are interested in more fine-grained explanations for the sources of uncertainty: explaining how the particular values of given features in an example affect the uncertainty of the model output. Little work has been done in this area, although there are notable exceptions. [Depeweg et al \(2017\)](#) examined the role of features in the uncertainty of Bayesian neural networks via sensitivity analysis: examining the gradient of the uncertainty with respect to each feature in turn. [Antoran et al \(2021\)](#) look at how uncertainty estimates can be explained in Bayesian neural networks via counterfactuals: identifying which constellations of features are responsible for uncertainty by finding sets of minimal changes required to increase a model’s confidence in its prediction. Similarly, [Chai \(2018\)](#) and [Zhang et al \(2022\)](#) look at the importance of features through predictive difference, showing changes in the predictive distribution of Bayesian neural networks when features are replaced with non-informative features modelled from a conditional distribution. More recently, [Watson et al \(2023\)](#) have considered how Shapley values could be used to explain both aleatoric and epistemic uncertainty in a completely model-agnostic manner. This allowed for local explanations of the behaviour of model uncertainty.

Our work shares the same motivation as these works: to explain not just the predictions of these models, but also the uncertainty associated with those predictions. Unlike [Depeweg et al \(2017\)](#); [Chai \(2018\)](#); [Zhang et al \(2022\)](#) and [Antoran et al \(2021\)](#), we do not assume a particular structure for our model and data. Although similar in spirit to the work of [Watson et al \(2023\)](#), our work uses techniques that are conceptually and computationally simpler, while still able to offer insight into model behaviour.

2.2 Feature importance

Early feature importance methods were developed in an ad hoc way to describe models of interest, rather than being the subjects of research in their own right. Permutation feature importance (PFI) and partial dependence plots (PDPs) were first introduced in articles on random forests ([Breiman, 2001](#)) and gradient boosting ([Friedman, 2001](#)), respectively. Although in both cases, the advantages and disadvantages of the feature importance measures were discussed, such a discussion was secondary to the main objectives of the article. Despite this, both methods have been widely adopted, adapted, and extended in the literature. Most notably for our purposes, [Moosbauer et al \(2021\)](#) introduced the idea of including confidence bands on PDPs to measure the uncertainty associated with a cost function when applying PDPs to a cost function in Bayesian optimisation.

Although PFI and PDPs were introduced independently, they share many common features. Implicit in both is the assumption that we can break the dependence of the target variable from the feature of interest by sampling from that feature’s marginal distribution without generating an out-of-distribution sample. However, this is not a reasonable assumption—as was indeed observed in different respects in the original papers introducing the two measures ([Breiman, 2001](#); [Friedman, 2001](#)).

These shortcomings have been addressed for PFI using conditional methods ([Strobl et al, 2008](#); [Molnar et al, 2023](#)). In [Strobl et al \(2008\)](#), the issue is resolved for random forests by taking advantage of the tree structure to break the feature space into a grid,

with permutations carried out only between features which share a common region, allowing for an approximation of the conditional distribution. This issue is addressed in a model-agnostic way in (Molnar et al, 2023): by splitting the test set into sub-groups in which the features of interest are conditionally independent of the other features using an additional model then performing PFI with permutations restricted to only swap values for examples within the same sub-group. An alternative approach is to compare the model of interest with a second model in which the feature of interest has been completely removed, or the information in that feature has been destroyed by permutation in the training set (Hooker et al, 2021). These approaches are referred to as remove-and-relearn and permute-and-relearn, respectively. These approaches resolve many of the issues that permutation-based methods face but do so at greater computational cost, requiring at least one more model to be trained for each feature of interest.

For PDP, the issue of extrapolation can be partially resolved through the use of individual conditional expectation (ICE) plots (Goldstein et al, 2015). These de-aggregate the effects of individual test examples on PDP, allowing users to see how examples differ in their sensitivity to a particular feature. In this way, ICE plots can be used to reveal heterogeneity in the model behaviour for a given feature.

Another popular approach for local explanations is local interpretable model-agnostic explanations (LIME) (Ribeiro et al, 2016), in which a simple, easy-to-interpret model is used to approximate the more complex model of interest within a small region of interest. This local model is trained explicitly so that it is locally a good approximation of the target model, and therefore can be analysed to determine what factors affected the model’s decision at that point.

The SHAP algorithm is a method to derive point-wise explanations for a model’s output in a principled manner (Lundberg and Lee, 2017). SHAP is derived from the game-theoretic notion of Shapley values, and is provably unique in satisfying a specific set of desirable properties (Lundberg and Lee, 2017). Although SHAP values are theoretically sound, they suffer from problems in terms of tractability. Several approximations are suggested by Lundberg and Lee (2017), and there are model-specific variants that allow for more accurate calculation (Lundberg et al, 2020). A detailed discussion on the ways in which Shapley values can be computed and estimated can be found in (Chen et al, 2023). Like permutation-based methods, both LIME and SHAP suffer when forced to extrapolate. For these methods, this difficulty comes in the form of adversarial attacks, which have been shown to allow a malicious actor to create false/misleading explanations by creating perturbed examples that are separate from the true data distribution (Slack et al, 2020).

Recently, a common framework unifying many of the methods described above, amongst others, was developed under the paradigm of “explaining by removing” (Covert et al, 2021). Using this framework, Covert et al (2021) conducted a systematic exploration of the connections and differences between existing model explanation methods.

2.3 Permutation-based feature importance methods

In this section, we describe the key feature importance methods from the literature in more detail. These methods provide a tool for explaining how much a particular feature’s value is responsible for determining the performance of a model.

2.3.1 Permutation feature importance (PFI)

Define the random variables for the feature vector and the target as X and Y , respectively, and let P_{XY} denote the true joint distribution of the data, with $(X, Y) \sim P_{XY}$. Additionally, let P_X denote the marginal distribution of X . We write $X = (X_1, \dots, X_d)$, with X_j being the random variable corresponding to the j -th feature, and denote its marginal distribution P_{X_j} . Furthermore, we define X_{-j} as the vector of features with the j -th element omitted, that is, $X_{-j} = (X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_d)$, similarly denoting its joint distribution as $P_{X_{-j}}$. Throughout, we use the convention that i indexes examples in a test set (with $1 \leq i \leq n$), j indexes features in a feature vector ($1 \leq j \leq d$) and c indexes class labels ($1 \leq c \leq k$).

PFI works as follows: given a trained model and set of test examples $\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})\}$, with each $\mathbf{x}^{(i)}$, we construct the design matrix $X_{\text{test}} \in \mathbb{R}^{n \times d}$, where the i -th row is the transpose of $\mathbf{x}^{(i)}$. To get the PFI measurement for the j -th feature, we randomly permute the j -th column and use the rows with that feature permuted as the feature vectors for our new test set. We refer to this new test set with the notation $\{(\tilde{\mathbf{x}}^{(1)}, y^{(1)}), \dots, (\tilde{\mathbf{x}}^{(n)}, y^{(n)})\}$. With a cost function \mathcal{C} , the empirical PFI for this test set is given as

$$\widehat{\text{PFI}}_{\mathcal{C}}(j) = \frac{1}{n} \sum_{i=1}^n \mathcal{C}(y^{(i)}, f(\mathbf{x}_{-j}^{(i)}, \tilde{x}_j^{(i)})) - \mathcal{C}(y^{(i)}, f(\mathbf{x}^{(i)})). \quad (1)$$

Here, we allow ourselves to use the convention adapted from [Casalicchio et al \(2019\)](#) of writing $f(\mathbf{x}_{-j}, \tilde{x}_j)$ to mean the function f with an input vector where the j -th entry is \tilde{x}_j (i.e., the j -th entry of $\tilde{\mathbf{x}}$) and the other entries are populated using the entries of \mathbf{x}_{-j} . We may think of the PFI for the j -th feature as the difference in performance when the dependence of the model on the j -th feature is broken by permuting the j -th feature in the test examples.

The values obtained by PFI are specific to the test set considered. However, they are Monte Carlo approximations of a quantity defined in terms of the data distribution ([Casalicchio et al, 2019](#)). Consider $X, Y \sim P_{XY}$, with X_{-j} being a subset of the features of X . Let \tilde{X}_j have the same marginal distribution as X_j but independent of X and Y . We have that

$$\begin{aligned} \text{PFI}_{\mathcal{C}}(j) &= \mathbb{E}_{X_{-j}, \tilde{X}_j}[\mathcal{C}(Y, f(X_{-j}, \tilde{X}_j))] - \mathbb{E}_X[\mathcal{C}(y, f(X))] \\ &\approx \frac{1}{n} \sum_{k=1}^n \mathcal{C}(y^{(i)}, f(\mathbf{x}_{-j}^{(i)}, \tilde{x}_j^{(i)})) - \mathcal{C}(y^{(i)}, f(\mathbf{x}^{(i)})) \\ &= \widehat{\text{PFI}}_{\mathcal{C}}(j). \end{aligned} \quad (2)$$

Permutation feature importance is most often considered for the case of regression, where \mathcal{C} is the squared loss. The classification case has not been thoroughly explored in the literature. [Molnar \(2022\)](#) demonstrates the use of AUC as a metric for the classification case, although this is easily applicable only in a binary classification scenario.

A common modification of PFI is to condition the distribution of the feature of interest on the observed values of the other features. This way, conditional PFI (CPFI) is defined as

$$\text{CPFI}_{\mathcal{C}}(j) = \mathbb{E}_{X_{-j}, (\tilde{X}_j^c | X_{-j})}[\mathcal{C}(Y, f(X_{-j}, \tilde{X}_j^c))] - \mathbb{E}_X[\mathcal{C}(y, f(X))],$$

where \tilde{X}_j^c is constructed such that \tilde{X}_j^c follows the conditional distribution of $X_j | X_{-j}$ but $\tilde{X}_j^c | X_{-j}$ is independent of Y ([Strobl et al, 2008](#)).

2.3.2 Partial dependence plots (PDPs)

While PFI is a measure of the effect of a given feature on the model’s performance, given a ground-truth label, PDP gives a method of visualising a feature’s effect on the model output itself. In PDP, a single feature is kept constant, while all other features assume values from the test set. The average model output is then captured across the test set. For feature j , the PDP is found by plotting $\widehat{\text{PDP}}(x; j)$ for all values x , where $\widehat{\text{PDP}}$ is defined as

$$\widehat{\text{PDP}}(x; j) = \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_{-j}^{(i)}, x), \quad (3)$$

where implicitly x is the value assumed by the j -th feature. This is a Monte-Carlo approximation of the true value of interest, that being

$$\text{PDP}(x; j) = \mathbb{E}_{X_{-j}}[f(X_{-j}, x)].$$

2.3.3 Individual conditional expectations (ICEs)

A PDP is most effective when the features are independent; as shown by [Goldstein et al \(2015\)](#), a PDP can hide the real effects of varying a feature by considering only the average over the training distribution, rather than examining the effects on individual examples. To this end, [Goldstein et al \(2015\)](#) have introduced individual conditional expectation (ICE) plots, which show how the model outputs for individual examples change as the feature of interest is varied. For feature of interest j , the ICE plot for the test example i is given by plotting the function

$$\text{ICE}^{(i)}(x; j) = f(\mathbf{x}_{-j}^{(i)}, x). \quad (4)$$

The PDP curve is simply the average of ICE curves over the test set, as can be trivially observed from Equations 3 and 4.

3 Explaining likelihood and uncertainty

In this section, we propose modifications to the feature importance methods described in Sections 2.3.1 and 2.3.2 to capture explanations for the likelihood and uncertainty of the predictive distribution of uncertainty-aware models.

3.1 Likelihood-PFI

We begin with PFI for negative log-likelihood, which we refer to as Likelihood-PFI: this is a natural extension of the original PFI measure, with the only difference being that we exchange the loss function used in the traditional PFI setting with the negative log-likelihood of the target given the feature variables. Rather than defining a model output f , we now think of the model as giving a predictive *distribution* q and use the negative log-likelihood of the target given this distribution as our loss function.

Given a test set $\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})\}$ of size n , the empirical Likelihood-PFI is given by

$$\widehat{\text{PFI}}_{\mathcal{L}}(j) = \frac{1}{n} \sum_{i=1}^n \log q(y^{(i)} | X = \mathbf{x}^{(i)}) - \log q(y^{(i)} | X_{-j} = \mathbf{x}_{-j}^{(i)}, \tilde{X}_j = \tilde{x}_j^{(i)}). \quad (5)$$

In order to obtain the exact value of which this is an approximation, we write

$$\text{PFI}_{\mathcal{L}}(j) = \mathbb{E}_{X,Y}[\log q(Y|X)] - \mathbb{E}_{\tilde{X}_j, X_{-j}, Y}[\log q(Y|X_{-j}, X_j = \tilde{X}_j)], \quad (6)$$

where the $X_j = \tilde{X}_j$ in $q(Y|X_{-j}, X_j = \tilde{X}_j)$ denotes the fact that while under the model distribution, Y is conditional on X_j , not \tilde{X}_j , when we take the expectation, we treat X_j as being distributed according to the marginal, rather than the joint distribution. This is clarified further in Appendix A.

Note that the order of terms in Equations 5 and 6 are reversed in comparison to the terms in the original PFI definitions (Equations 1 and 2). This is because the function under consideration here is the *negative* log-likelihood, and reversing the order of terms allows us to avoid a double negative. However, the quantity of interest is still the loss given the permuted feature minus the loss given the original feature value.

3.2 Entropy-PFI

Likelihood-PFI gives a measure of how the model *performance* is affected by re-sampling a feature from its marginal distribution. However, it is also useful to know how the *uncertainty* of a model is affected by the value of each feature. To do this, we will look at how PFI can be performed for the entropy of its prediction rather than for its accuracy.

The Shannon entropy for a given categorical random variable Y , given a model distribution q , is given by

$$\mathcal{H}_q(Y) = - \sum_{y \in \mathcal{Y}} q(y) \log q(y),$$

where the q subscript in \mathcal{H}_q emphasises that we consider the entropy under the model’s predictive distribution (in contrast with the entropy of the true predictive posterior). Similarly, for a continuous random variable, the entropy is written as

$$\mathcal{H}_q(y) = - \int q(y) \log q(y) dy.$$

With this, we define the entropy permutation feature importance (Entropy-PFI) as

$$\text{PFI}_{\mathcal{H}}(j) = \mathbb{E}_{X, \tilde{X}_j} \left[\mathcal{H}_q(Y|X_{-j}, X_j = \tilde{X}_j) - \mathcal{H}_q(Y|X) \right],$$

and we get the Monte-Carlo approximation using a test set as

$$\widehat{\text{PFI}}_{\mathcal{H}}(j) = \frac{1}{n} \sum_{i=1}^n \mathcal{H}_q(Y|X_{-j} = \mathbf{x}_{-j}^{(i)}, X_j = \tilde{x}_j) - \mathcal{H}_q(Y|X = \mathbf{x}^{(i)}).$$

We can think of Entropy-PFI as measuring how much uncertainty increases on average when we replace the j -th feature of an example with a random sample from its marginal distribution. Intuitively, we would expect for this value to be non-negative: by replacing this feature, we will often be moving away from dense regions in the sample space, where the model will have low epistemic uncertainty, to sparser ones, where the model will have seen fewer examples and therefore exhibit lower confidence in its predictions. So, a high PFI score means that the value of a feature is helping to increase the confidence of the model in its prediction. Note that, unlike the Likelihood-PFI, the Entropy-PFI is independent of the ground-truth label.

3.3 PDP and ICE for entropy and likelihood

We can additionally adapt PDP to look at the effect that varying a feature value has on the uncertainty of a model. To this end, we define entropy partial dependence plots (Entropy-PDPs) as

$$\text{PDP}_{\mathcal{H}}(x; j) = \mathbb{E}_{X_{-j}} [\mathcal{H}_q(Y|X_{-j}, X_j = x)], \quad (7)$$

and we approximate this value using a test set in the following way:

$$\widehat{\text{PDP}}_{\mathcal{H}}(x; j) = \frac{1}{n} \sum_{i=1}^n \mathcal{H}_q(Y|X_{-j} = \mathbf{x}_{-j}^{(i)}, X_j = x). \quad (8)$$

We similarly define Entropy-ICE plots with

$$\text{ICE}_{\mathcal{H}}^{(i)}(x; j) = \mathcal{H}_q(Y|X_{-j} = \mathbf{x}_{-j}^{(i)}, X_j = x). \quad (9)$$

PDPs and ICE plots tend to be used on the model output itself. Entropy-PDPs and Entropy-ICEs can be seen as extensions of this idea, but using a statistic derived

from the model output distribution, as opposed to using the model output directly. We now further extend this notion, looking at the likelihood of the true label given the features. In this way we define the PDP for likelihood as

$$\text{PDP}_{\mathcal{L}}(x; j) = -\mathbb{E}_{Y, X_{-j}}[\log q(Y|X_{-j}, X_j = x)].$$

Note that this is defined in terms of the negative log-likelihood. The PDP for likelihood is approximated on a test set via averaging:

$$\widehat{\text{PDP}}_{\mathcal{L}}(x; j) = -\frac{1}{n} \sum_{i=1}^n \log q(y^{(i)}|X_{-j} = \mathbf{x}_{-j}^{(i)}, X_j = x).$$

3.4 Properties of Entropy-PFI

In interpreting Entropy-PFI, it is important to understand exactly what the quantity measures. For a given feature, it gives a measure of how much the value of that feature supports the model’s conclusion derived from the other features. If the feature shares task-relevant information with other features, the model will be more confident when the feature under consideration agrees with those features and less confident when the relationship between the feature values is destroyed. Entropy-PFI establishes the difference between these two levels of confidence by comparing the level of confidence under the true distribution against the level of confidence when the feature of interest follows the same marginal distribution, but is independent of the other features.

In Figure 1, we see a mock-up of how permuting features affects the entropy of a test set by moving examples from low-entropy regions to high-entropy ones. In the left panel, the contour plot shows the values of the entropy over the feature space, and each dot represents a member of the test set. We see that the test examples all occur in low-entropy areas: the combined information of the two features allows the model to be certain in its prediction. In the centre panel, we perform PFI on the second feature. Here, we see that there are test points which are in high-entropy areas: we can think of this as the model being surprised by the combination of features, and increasing its level of uncertainty as a result. On the right, we see the histogram of the entropies. We see that when permuted, there are now more high-entropy points, and the average have increased.

Intuitively, we can think of Entropy-PFI as answering the question “How much does the true value of this feature support the prediction given on the basis of the evidence provided by the other features?”. A consequence of this interpretation is that we would expect that if a feature did not share any information with other features, the Entropy-PFI would be zero. In the following proposition, we verify that this is the case.

Proposition 1. *If X_{-j} is independent of X_j , then the entropy-PFI is zero.*

Proof. Starting from the definition of $\text{PFI}_{\mathcal{H}}$, we use the fact that X_j is independent of X_{-j} to swap it with \tilde{X}_j :

$$\text{PFI}_{\mathcal{H}}(j) = \mathbb{E}_X \left[\mathbb{E}_{\tilde{X}_j} [\mathcal{H}_q(Y|X_{-j}, X_j = \tilde{X}_j)] - \mathcal{H}_q(Y|X) \right]$$

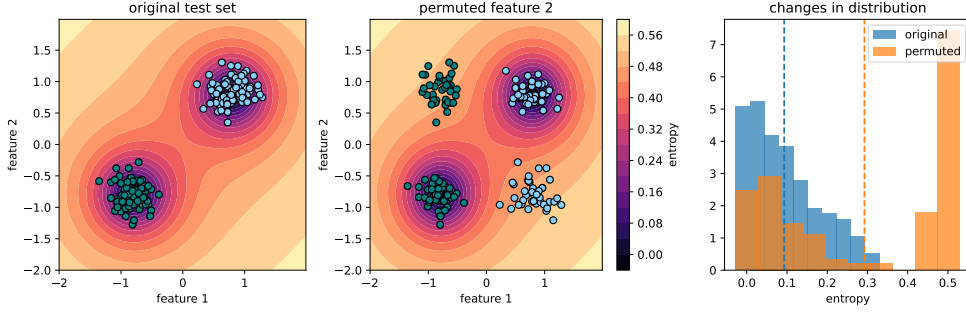


Fig. 1 Visualisation of effects of PFI. The colour of each point shows the cluster to which the original (unpermuted) test example belonged. In the left panel, the original test set is shown, along with the contour lines for the entropy of a (hypothetical) model’s predictive distribution. In the centre panel, the test set is shown after permuting feature 2. In the right panel, histograms of the entropy before and after permuting the second feature are shown.

$$\begin{aligned}
&= \mathbb{E}_X [\mathbb{E}_{X_j} [\mathcal{H}_q(Y|X_{-j}, X_j)] - \mathcal{H}_q(Y|X)] \\
&= \mathbb{E}_X [\mathcal{H}_q(Y|X) - \mathcal{H}_q(Y|X)] = 0.
\end{aligned}$$

□

We note that this is not the case for the Likelihood-PFI, or the PFI of any target-dependent measure in general, where dependencies between X_j and Y prevent the substitution used in the proof above. The above proposition means that when a feature is independent of the others, it does not globally affect the model confidence. However, this does not mean that the feature does not affect the model’s confidence locally, only that local effects cancel out in aggregate.

A feature being independent of the complementary set is not the only way that Entropy-PFI can be zero. It can also be zero if the predictive distribution does not depend on the feature of interest. This is analogous to how traditional PFI will be zero if a feature is not used in determining the model output.

Proposition 2. *If the predictive distribution is not dependent on feature j , i.e., $q(Y|X) = q(Y|X_{-j})$, then $\text{PFI}_{\mathcal{H}}(j) = 0$ and $\text{PFI}_{\mathcal{L}}(j) = 0$.*

Proof. By assumption and definition of \mathcal{H}_q , we have that $\mathcal{H}_q(Y|X_{-j}, X_j) = \mathcal{H}_q(Y|X_{-j})$. By definition of $\text{PFI}_{\mathcal{H}}$, we therefore have

$$\begin{aligned}
\text{PFI}_{\mathcal{H}}(j) &= \mathbb{E}_X [\mathbb{E}_{\tilde{X}_j} [\mathcal{H}_q(Y|X_{-j}, X_j = \tilde{X}_j)] - \mathcal{H}_q(Y|X)] \\
&= \mathbb{E}_X [[\mathcal{H}_q(Y|X_{-j}) - \mathcal{H}_q(Y|X_{-j})] = 0.
\end{aligned}$$

Similar reasoning gives the result for $\text{PFI}_{\mathcal{L}}(j)$:

$$\begin{aligned}
\text{PFI}_{\mathcal{H}}(j) &= \mathbb{E}_{X,Y} [q(Y|X) \ln q(Y|X) - \mathbb{E}_{\tilde{X}_j} [q(Y|X_{-j}, X_j = \tilde{X}_j) \ln q(Y|X_{-j}, X_j = \tilde{X}_j)]] \\
&= \mathbb{E}_{X,Y} [[q(Y|X_{-j}) \ln q(Y|X_{-j}) - q(Y|X_{-j}) \ln q(Y|X_{-j})] = 0.
\end{aligned}$$

□

By considering the Entropy-PFI and Likelihood-PFI jointly, we can distinguish between when a feature is informative but independent of others and when it is not informative to the predictive distribution. In the latter case, the Likelihood-PFI will also be zero (or at least very small in empirical approximations), whereas in the former case, the Likelihood-PFI can be large even when the Entropy-PFI is zero.

3.4.1 Examples of joint usage of Entropy-PFI and Likelihood-PFI

We now present two examples of synthetic datasets in which Entropy-PFI and Likelihood-PFI together can enhance interpretability in terms of feature importance and predictive uncertainty. In particular, one classification and one regression example are presented.

Classification experiment on synthetic data

In order to examine the interpretation of Entropy-PFI and Likelihood-PFI in a classification setting, we consider a toy binary classification dataset simulated according to the model

$$P(Y = 1|\mathbf{x}) = \epsilon + (1 - 2\epsilon) \mathbb{1} \left(\sum_{j=1}^J \mathbf{x}_j > \frac{J}{2} \right),$$

where \mathbf{x} is sampled uniformly from the unit hypercube $[0, 1]^d$ (Mease and Wyner, 2008). Here, ϵ is the amount of label noise, d is the total number of features, and J is the number of relevant features. For our experiments, we set $d = 10$, $J = 4$ and $\epsilon = 0.1$. This means that there are four features all with equal importance (features 1, 2, 3 and 4) and six features which are irrelevant in determining the target class. We consider three versions of this data: one that is exactly as described above, a second in which feature 10 is replaced with a copy of feature 1, and a third in which feature 10 is replaced with a copy of feature 5.

In Figure 2, we see how the Likelihood-PFI and Entropy-PFI are affected by adding redundancy to the dataset for a random forest with calibration (see Appendix B for details). We use 5000 examples with a train/test split of 3750/1250. In the original dataset, all four task-relevant features have similarly high Likelihood-PFI values. However, when feature 10 is replaced with a copy of feature 1, the first feature is now ranked as less important; this is because there is now an alternate source from which the model can get the same information.

In contrast, features 1 and 10 are the features that Entropy-PFI identifies as most important in making the model confident in its output, as shown in the right panel of the figure. This makes sense: in all the training data, these features have been strongly correlated (identical, in fact), and therefore examples where this relationship is broken should be treated as out-of-distribution, which should be reflected in greater predictive uncertainty.

We also see a small increase in Entropy-PFI for the redundant features in the third dataset. This is likely due to the fact that, despite features 5 and 10 not containing any

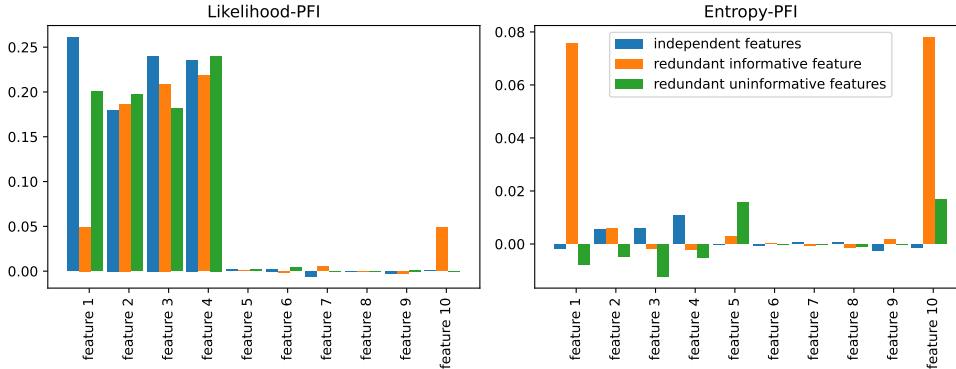


Fig. 2 Comparison of Likelihood-PFI and Entropy-PFI for three datasets, the second and third of which contain redundant features. When feature 10 is a copy of feature 1 (an informative feature), we see PFI-likelihood of feature 1 drop and PFI-entropy increase, and both PFI-likelihood and PFI-entropy increase for feature 10. When feature 10 is a copy of feature 5 (an uninformative feature), there is no effect on PFI-likelihood for either feature, and a small increase in PFI-entropy for both.

information about the target, spurious correlations in the training set may cause the model to use these features, and therefore the model is able to identify when it goes out-of-distribution due to disagreement between the two values, resulting in changes in entropy. However, we note that this effect is small in comparison to the effect in features that are informative and therefore are actually useful to the model. Indeed, as shown in Proposition 2, if the model (correctly) learns to disregard both features, the Entropy-PFI should be zero.

Regression example on synthetic data

A synthetic dataset is simulated from the regression model

$$Y = X_1 + X_2 + 0.9X_3^2 + X_4 + X_5 + \varepsilon,$$

where Y is a random variable dependent on feature variables X_1, \dots, X_5 . We sample the features from the following Gaussian distributions:

$$(X_1, X_2), (X_3, X_4) \sim \mathcal{N} \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \right), X_5 \sim \mathcal{N}(0, 1), \varepsilon \sim \mathcal{N}(0, 2).$$

Apart from the stated relationships, the features are otherwise all independent of each other. We train a Gaussian process regression model on 500 examples drawn from this distribution and generate an additional 500 test samples to be used for the importance measures. In Figure 3, we show the Entropy-PFI and Likelihood-PFI.

In Figure 3, we display the importance of each feature as measured by Likelihood-PFI and Entropy-PFI. For Likelihood-PFI, we observe many of the known properties of PFI under typical loss functions: having the same information shared between multiple features (e.g., having a large covariance between features X_1 and X_2) diminishes their

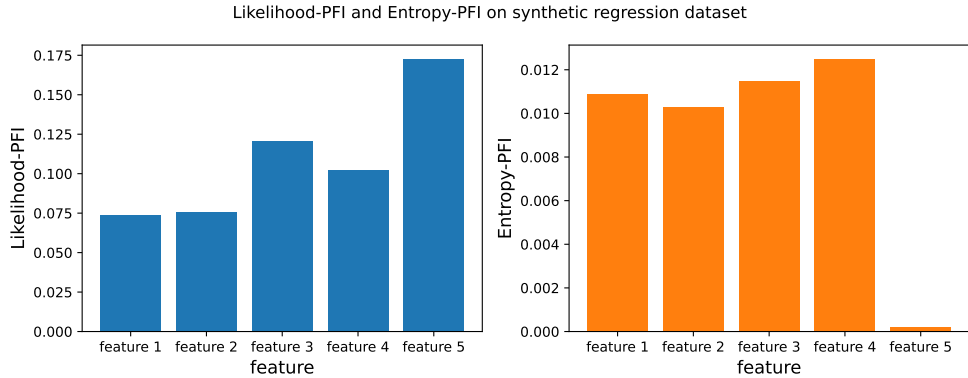


Fig. 3 Likelihood-PFI and Entropy-PFI for features in a synthetic dataset using a Gaussian process model. Since features 1-4 share information with each other, their Likelihood-PFI is reduced relative to the independent feature 5. In contrast, their shared information means that they have higher Entropy-PFI, where feature 5’s Entropy-PFI is negligible.

importance to the model relative to features that contain no shared information (e.g., feature X_5). In contrast, because features X_1 and X_2 contain shared information, their Entropy-PFI is high: when the connection between these features is broken by permuting one of their values in the test set, the model is forced to extrapolate, resulting in higher predictive uncertainty. We see similar behaviour in the measures for X_3 and X_4 , despite their non-linear relationship. Additionally, while we see that feature X_5 is considered important in determining the negative log-likelihood (i.e., the model uses information from feature X_5 in order to make an accurate prediction), it is not considered important in determining the uncertainty (i.e., on average, knowing feature X_5 neither increases nor decreases the model’s confidence in its prediction); again, this is due to the feature being independent from the others.

3.4.2 Why conditional PFI is not useful in the context of entropy

At their core, permutation-based feature importance methods rely on resampling features based on their marginal distribution, breaking the relationship between the feature and the set of all other variables. However, doing this leads to undesirable outcomes: by ignoring correlations and other relationships between feature variables, we can find ourselves evaluating the model on points outside the true data distribution. As discussed in depth by [Hooker et al \(2021\)](#), this can be problematic in that feature importances rely on the extrapolating behaviour of the model, giving importance measures that are dependent on the model’s behaviour in regions far from the training data, where the model’s behaviour is unlikely to reflect the true distribution of the data.

One of the proposed methods for dealing with this is through conditional approaches, where a variable \tilde{X}_j^c is constructed so that it retains the same conditional relationship with X_{-j} as X_j , but is independent of Y given the information

contained in X_j . However, such approaches are at odds with what Entropy-PFI measures: Entropy-PFI is non-zero only when features share task-dependent information and permuting one of the features breaks the dependency between one feature and the set of other features.

Say we have access to a random variable \tilde{X}_j^c such that (\tilde{X}_j^c, X_{-j}) has the same joint distribution as (X_j, X_{-j}) but \tilde{X}_j^c is independent of Y (either completely or when conditioned on X_{-j}). We can define Likelihood-PFI in the same way that conventional PFI is defined for this approach. However, if we attempt to define conditional Entropy-PFI in the same way, e.g.,

$$\text{CPFI}_{\mathcal{H}}(j) = \mathbb{E}_{X, \tilde{X}_j^c} [\mathcal{H}_q(Y|X_{-j}, X_j = \tilde{X}_j^c) - \mathcal{H}_q(Y|X)]. \quad (10)$$

we cannot use it to get estimates of the uncertainty caused by feature X_j , since this will always be zero, as shown in the following proposition.

Proposition 3. *The entropy version of conditional PFI, as defined in Equation (10), is zero for all features.*

Proof. Considering an arbitrary feature indexed by j , we look at the first term in more detail, finding

$$\mathbb{E}_{X, \tilde{X}_j^c} [\mathcal{H}_q(Y|X_{-j}, X_j = \tilde{X}_j^c)] = \mathbb{E}_{X, \tilde{X}_j^c} \left[\int u(y|X_{-j}, X_j = \tilde{X}_j^c) dy \right],$$

where $u(\cdot) = q(\cdot) \log q(\cdot)$ for the sake of brevity. Using the fact that (\tilde{X}_j^c, X_{-j}) has the same joint distribution as (X_j, X_{-j}) by definition, we can expand the expectations to give

$$\begin{aligned} & \mathbb{E}_{X, \tilde{X}_j^c} [\mathcal{H}_q(Y|X_{-j}, X_j = \tilde{X}_j^c)] \\ &= - \int \int \int u(y|X_{-j} = \mathbf{x}_{-j}, X_j = \tilde{x}_j) P_{X_{-j}, \tilde{X}_j^c}(\mathbf{x}_{-j}, \tilde{x}_j) dy d\mathbf{x}_{-j} d\tilde{x}_j \\ &= - \int \int \int u(y|X_{-j} = \mathbf{x}_{-j}, X_j = \tilde{x}_j) P_{X_{-j}, X_j}(\mathbf{x}_{-j}, \tilde{x}_j) dy d\mathbf{x}_{-j} d\tilde{x}_j \\ &= - \int \int \int u(y|X = \mathbf{x}) dy P_X(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_X [\mathcal{H}_q(Y|X)]. \end{aligned}$$

Plugging this into Equation (10) gives that the conditional Entropy-PFI is zero. \square

Therefore, we see that, by definition, conditional approaches for PFI are ineffective in measuring the importance of features in determining entropy. However, we make two arguments as to why this is not problematic. Firstly, one of the ways in which Entropy-PFI has utility is in identifying when shared information between features has the effect of boosting model confidence; in attempting to preserve shared information between the feature of interest and the other features via conditioning, we eliminate the very discrepancy that we aim to measure. Secondly, while relying on

the extrapolation behaviour of a model is in general undesirable, especially when the underlying data generating process is of interest, for an uncertainty-aware model, its ability to display increased (epistemic) uncertainty when extrapolating is one of the key desirable characteristics of the model.

3.5 A toy example for Entropy-PDP and Entropy-ICE

To offer an interpretation of Entropy-PDP plots, we consider a synthetic dataset, the distribution of (training) points for which is shown on the left of Figure 4. We observe that the data is distributed around the border of the feature space, with no examples lying on the interior. This means that an uncertainty-aware model should exhibit high epistemic uncertainty when both features are in the middle of their range (i.e., $-1.5 < X_1, X_2 < 1.5$), and lower uncertainty when either feature takes on a more extreme value (i.e., either $|X_1| > 1.5$ or $|X_2| > 1.5$). The target variable (not shown) is of the form $Y = (X_1 + X_2)^2 + 0.1\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. Note that by symmetry of the features in the data-generating process, both features should appear equally important to any reasonable model and importance measure; therefore, little insight is to be gained from PFI or similar methods. However, we can still gain understanding of our model’s uncertainty by looking at Entropy-PDP.

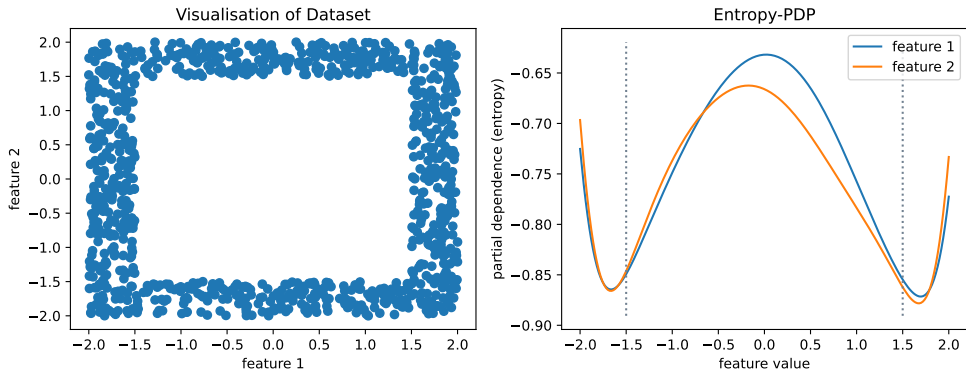


Fig. 4 Visualisation of the distribution of synthetic dataset (left) and Entropy-PDP plots for each feature (right). The vertical dotted lines on the Entropy-PDP plot show where the “interior” of the distribution begins.

In the right-hand plot of Figure 4, we see the Entropy-PDP plot for a Gaussian process model on a test set drawn from the same distribution as the training set. We observe high uncertainty both in the interior values and at the extremes. We can hypothesise that the uncertainty in the interior is caused by examples where the feature not under consideration is also mid-range, causing the model input to be out-of-distribution and, therefore, for the model to exhibit high epistemic uncertainty. Similarly, at the most extreme values, uncertainty may be higher, as there are fewer proximal training examples than in the centre of the bands.

This is a case where PDP fails to capture the heterogeneity of the data: we see that uncertainty increases for interior values, but have no information about whether this is true for all possible values of the other feature or just some values.

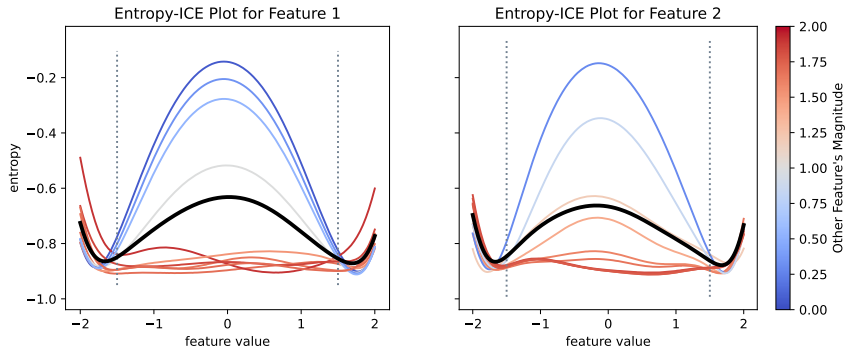


Fig. 5 Entropy-ICE plots for randomly sampled test points for each feature. The PDP curve is shown in black. All other lines are ICE plots, with the colour of the line showing the value of the constant features (i.e., $\mathbf{x}_{-j}^{(i)}$).

In order to better understand the model behaviour, we therefore additionally look at Entropy-ICE plots (Figure 5). In these plots, we can see two distinct behaviours: if we consider the Entropy-ICE plot for the first feature, for examples where feature X_2 has small magnitude, we observe higher uncertainty when the feature under consideration (X_1) is also small. As hypothesised, this is due to the example being out-of-distribution and, therefore, having high associated uncertainty. On the other hand, when the complementary feature takes a more extreme value, the example generated will be in-distribution and, therefore, have lower associated uncertainty.

We also observe that the behaviour of the ICE curves at extreme values is homogeneous: the model uniformly becomes more uncertain as the feature value approaches the edge of the distribution, in contrast with the heterogeneous behaviour for central values.

4 Experiments using real-world datasets

In this section, we examine how Entropy-PFI and Likelihood-PFI can be used in practice to gain insights into how various probabilistic models make their predictions. We consider a variety of models in both classification and regression settings.

4.1 Regression example: concrete dataset

In this example, we show how the proposed methods can be used to gain insight into the behaviour of models on a real-world regression dataset. Here, we demonstrate how Likelihood-PFI and Entropy-PFI give complementary explanations for the behaviour

of uncertainty-aware regression models. As an example, we consider the UCI concrete dataset (Yeh, 2007). We use two uncertainty-aware models: a Gaussian process with a radial basis function (RBF) kernel and a neural network using Monte-Carlo (MC) dropout (Gal and Ghahramani, 2016). Details of the configurations for both models can be found in Appendix B. We use 75% of the dataset for training and the other 25% for testing.

In Figure 6, we see the relative importance of each feature in terms of both Likelihood-PFI and Entropy-PFI. We observe that although *age* is the most important feature in terms of the likelihood, its (global) affect on the entropy is small. This suggests that *age* is important in accurately predicting the target variable (i.e., the value of the feature will often have a significant effect on the likelihood), but that it is not strongly related to any of the other features.

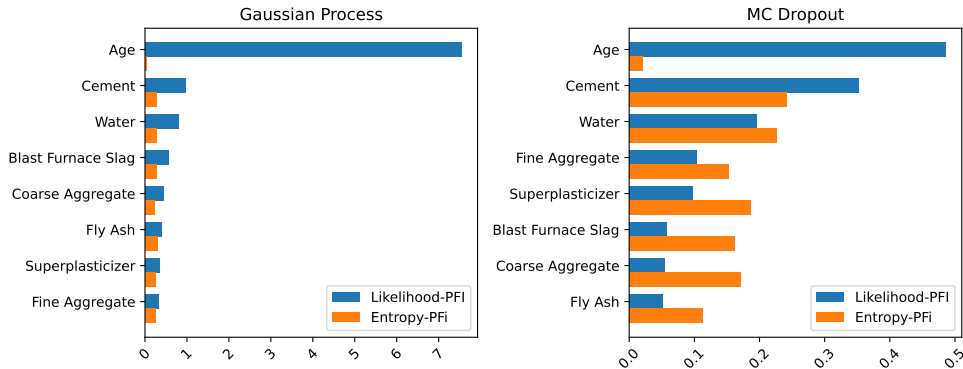


Fig. 6 Comparison of Entropy-PFI and Likelihood-PFI for neural networks with Monte-Carlo dropout and Gaussian processes fitted to the UCI concrete dataset.

We can verify this by examining how effectively we can train regression models to learn each feature in the dataset given the others. In Figure 7, the coefficient of determination when a random forest regression model is trained to predict one feature given that we observe all the other features (on the right, this includes the target). We see that indeed *age* appears to be independent of the other features: knowing all the other features does not give reliable information about *age*. However, the target variable contains information about *age*, and prediction of the *age* variable improves significantly when we have access to it, suggesting that the target and *age* share information.

Note that in concluding that *age* is independent of the other features, we draw on several observations. The fact that the Likelihood-PFI is high for the feature discounts the possibility that the Entropy-PFI being zero is simply a result of the model discarding the feature, not making use of it in determining the predictive distribution. Additionally, the fact that the Entropy-PFI is non-zero for other features means that

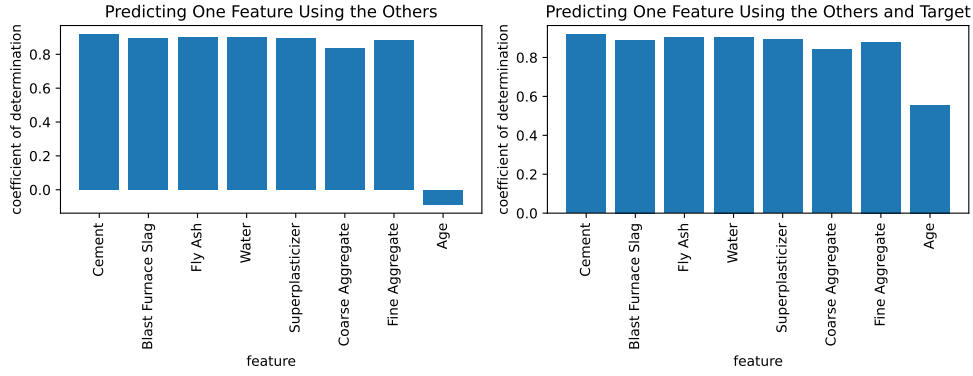


Fig. 7 Comparison of coefficients of determination for models predicting one feature’s value given the others (UCI concrete dataset).

the model is indeed uncertainty-aware, and is not just using the same distribution with shifted mean for each point.

However, as previously noted, just because the Entropy-PFI is small/zero, it does not mean that the entropy is not affected *locally* by the specific value of the feature. To better understand this, in Figure 8 we plot the Entropy-PDP curve for *age* for both models, along with a few randomly chosen ICE curves. This figure highlights the fact that Entropy-PFI is a global property: despite Entropy-PFI being near zero, we see that entropy varies not only as we change the feature value (shown by how the PDP curve changes as the value for *age* does), but is also affected by the values that other features take (shown by the variation in characteristics of the ICE curves).

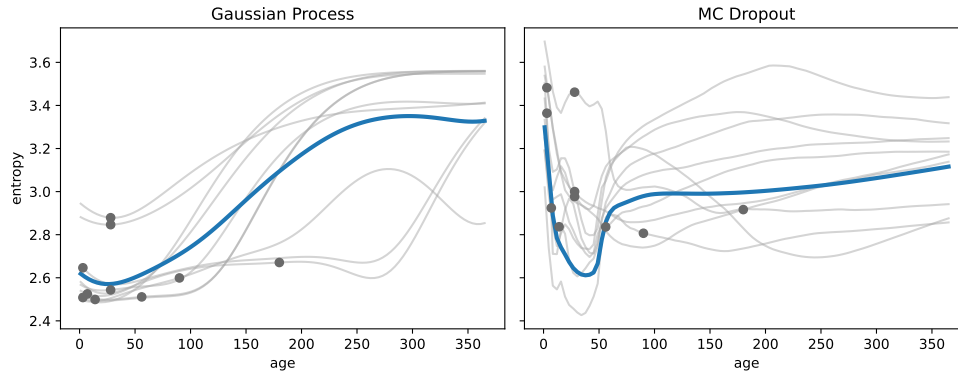


Fig. 8 Entropy-ICE and Entropy-PDP plots for *age* feature (UCI concrete dataset). The thicker blue curves are the PDP curves, with ICE curves for examples in the test set shown in grey. The original values for the feature for each ICE curve are shown as grey dots lying on each grey curve.

4.2 Classification example: diabetes dataset

To demonstrate the utility of our approach for uncertainty-aware models in a classification setting, we examine the importance of various features for models trained on the UCI diabetes dataset [Smith et al \(1988\)](#). In particular, we examine two models: calibrated random forests and deep neural networks with weight uncertainty, also referred to as Bayes by backprop (BBB) ([Blundell et al, 2015](#)). Further details of the configurations of both models are given in [Appendix B](#). The UCI diabetes dataset was also used by [Breiman \(2001\)](#) as one of the first applications of PFI to explain model behaviour; for this example, we first review the findings of [Breiman \(2001\)](#), before examining what additional insights might be gleaned from our new approach.

In [Breiman \(2001\)](#), it is observed that the second feature (*plas*) is the most important, followed by *age* (feature 8) and *mass* (feature 6). Through additional experiments, Breiman have also showed that while feature 8 contains useful information about the target label, the predictive information that it contains is redundant with the information contained in feature 2; hence, training a model either with or without this feature has little effect on the model’s predictive power.

Where [Breiman \(2001\)](#) measured the percentage increase in classification accuracy, we measure the difference in likelihood as defined in [Equations 5 and 6](#). Doing so, we observe in [Figure 9](#) the same phenomena occur for the likelihood in our random forest model as occurred for classification accuracy in Breiman’s: *plas* (feature 2) is the most important, with *age* (feature 8) and *mass* (feature 6) also having significant effects on the model’s predictive power. We observe similar results for an MLP using BBB.

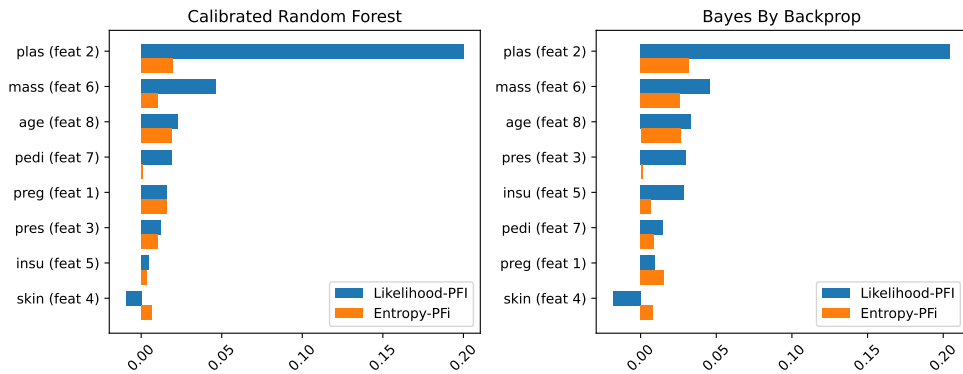


Fig. 9 Entropy-PFI and Likelihood-PFI values for calibrated random forests and Bayes by backprop neural networks fitted to the UCI diabetes dataset.

In [Figure 10](#), we display the Entropy-PDP and Entropy-ICE plots for *plas* (feature 2). We see that for lower values of the feature, there is relatively low entropy for both models. We also observe a significant increase in uncertainty for higher values, peaking between 150-175. Examining the distribution of the feature in the training set, separated by class, we see that at extreme values, the feature is strongly informative

of class label, but at values in the region 100-170, both classes are present with high frequency, leading the feature to be less informative, and therefore rendering models less confident in their predictions based on information from this feature for values in this range.

We can also see the effect of this change in the confidence level of the models on the negative log likelihood in Figure 11. For examples in the positive class, the model becomes more confident in its correct prediction (given the other features) as the *plas* value increases, leading to a decrease in the loss for those examples. For the negative class, the opposite is true: The model becomes more confident in its prediction as the feature value decreases. Again, we can interpret this in terms of the confidence increasing as the feature of interest adds evidence to support the conclusion inferred from the other features.

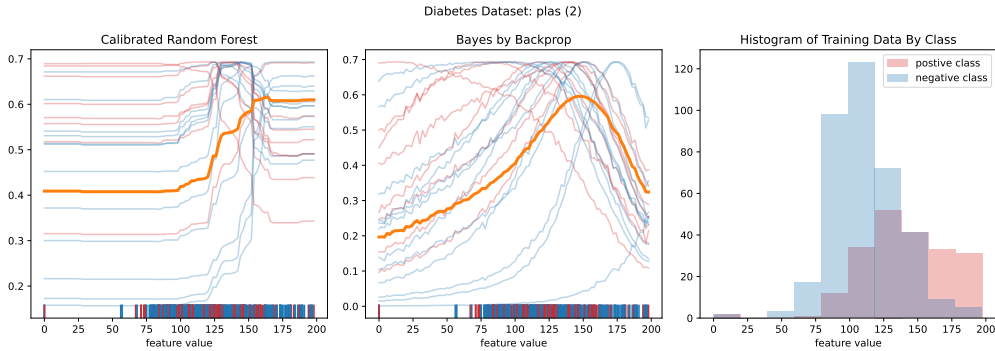


Fig. 10 Entropy-PDP and Entropy-ICE curves for the UCI diabetes dataset for *plas* (feature 2). Left: curves for a calibrated random forest. Middle: curves for Bayes by backprop. Right: distribution of feature value in training set for both classes. In the left and middle figures, the bars along the values of the feature for examples in the training set of the positive (red) and negative (blue) classes. The red and blue lines are ICE curves for test examples of the two classes, and the orange curves are the Entropy-PDP curves.

We observe a similar phenomenon for *mass* (feature 6) in Figure 12. In particular, we see that, for members of the positive class, ICE curves show a great amount of uncertainty for low feature values. This could be due to the fact that examples with low values for this feature while other features having characteristics of the positive class are not found in the UCI diabetes dataset, and therefore the examples constructed for the ICE curve are out-of-distribution, and exhibit high epistemic uncertainty. Note that this is not picked up in the Entropy-PDP, and can only be observed using Entropy-ICE.

In Figure 13, we again see a difference in behaviour of the Likelihood-ICE curves for the different classes, with the loss being lower for the positive class at higher values for the feature, and for the negative class at lower values.

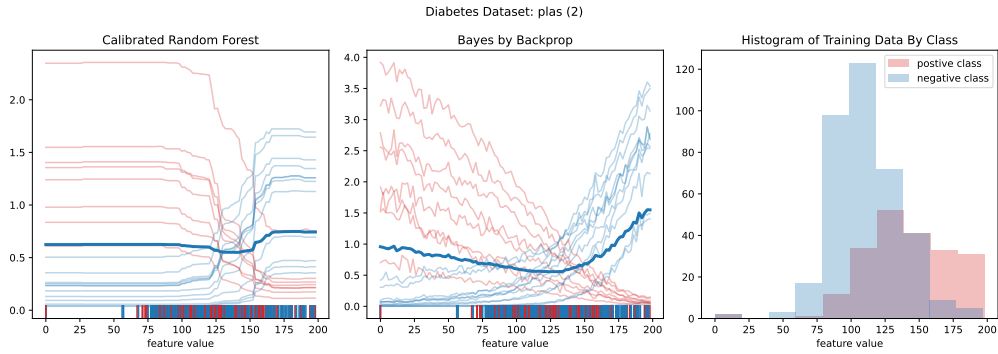


Fig. 11 Likelihood-PDP and Likelihood-ICE curves for the UCI diabetes dataset for *plas* (feature 2). Left: curves for calibrated random forest. Middle: curves for Bayes by backprop. Right: distribution of feature value in training set for both classes.

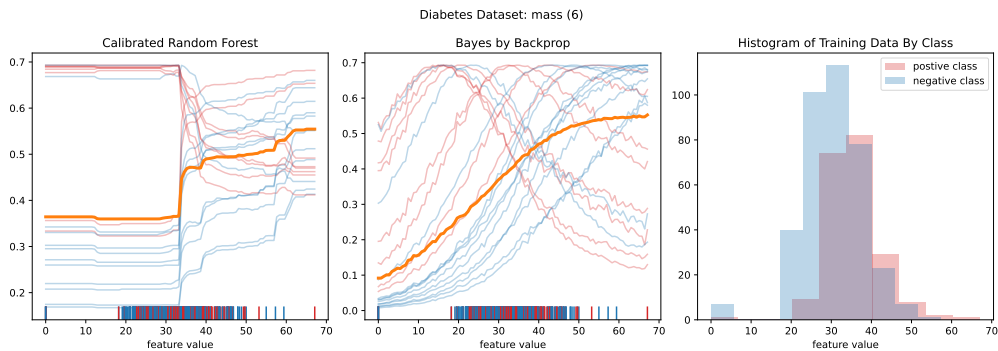


Fig. 12 Entropy-PDP and Entropy-ICE curves for the UCI diabetes dataset for *mass* (feature 6). Left: curves for a calibrated random forest. Middle: curves for Bayes by backprop. Right: distribution of feature value in training set for both positive and negative classes.

5 Conclusions

In this paper, we have proposed modifications of PFI, PDP and ICE that can be used to gain insights into the importance of features in uncertainty-aware models, both in terms of likelihood and uncertainty (as measured by the entropy of the predictive distribution).

Permutation feature importance, amongst other methods, has come under criticism for their shortcomings in forcing the model to extrapolate to unexplored regions in developing explanations. While the suggested solution is to avoid PFI in favour of methods which explicitly address this issue, the simplicity of PFI and related approaches mean that they nonetheless remain popular. Using Entropy-PFI can mitigate some of these issues, giving additional information about the level of uncertainty in a model that is attributable to a given feature. In particular, Entropy-PFI can be

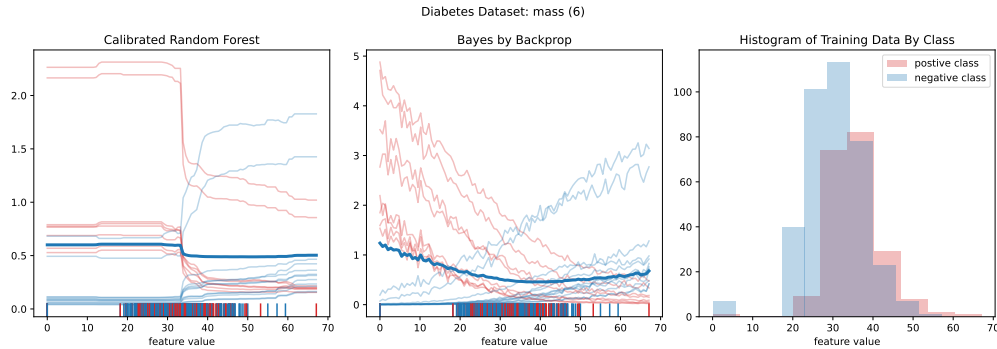


Fig. 13 Likelihood-PDP and Likelihood-ICE curves for the UCI diabetes dataset for *mass* (feature 6). Left: curves for a calibrated random forest. Middle: curves for Bayes by backprop. Right: distribution of feature value in training set for both positive and negative classes.

used to identify when a feature is likely to be independent from other informative features, and therefore its feature importance can be trusted.

We note that Entropy-PFI does not completely mitigate the issues raised in (Hooker et al, 2021), and careful interpretation and understanding of the strengths and weaknesses of each method are required. However, given the prevalence of permutation and extrapolation-based importance methods, even in light of recent criticism, having these additional tools serves to mitigate some of the shortcomings of these approaches.

With regard to PDP, we have shown examples of how aleatoric and epistemic uncertainty affect these values. However, we did not attempt to separate them. The utility of explaining these two sources of uncertainty, as well as the best methodology for doing so, remain questions for future research. Similarly, there are a wealth of other methods that could be used for explainable uncertainty, such as LIME (Ribeiro et al, 2016), and adaptations of the ones used in this paper (such as showing derivatives using ICE rather than the original values). We point out these issues as future work.

Funding. The authors acknowledge financial support from The University of Manchester’s Centre for Digital Trust and Society and Simon Industrial Fund.

Code availability. The code will be released upon acceptance.

Appendix A A note on notation

In this appendix, we further clarify the notation used in defining expectations in Likelihood-PFI and Entropy-PFI. In conventional PFI with loss function \mathcal{C} , and model output $f(X) = f(X_j, X_{-j})$, we would write

$$PFI(j) = \mathbb{E}_{Y, X, \tilde{X}}[\mathcal{C}(Y, f(\tilde{X}_j, X_{-j})) - \mathcal{C}(Y, f(X))].$$

However, for the models we are interested in, the output of the model is not a scalar (or vector)-valued function, but a density $q(Y|X)$. For a given test point, we can

write $q(Y = y|X = x)$ as the probability assigned to the label y given the features x . Additionally, we can define the Likelihood-PFI for a single point (x, y) as

$$\text{PFI}_{\mathcal{H}}(\mathbf{x}; j) = - \int \log q(Y = y|X_j = \tilde{x}_j, X_{-j} = \mathbf{x}_{-j}) P_{\tilde{X}_j}(\tilde{x}_j) d\tilde{x}_j + \log p(Y = y|X = \mathbf{x}),$$

remembering that $X = (X_j, X_{-j})$ and that \tilde{X}_j is an independent random variable whose distribution is the same as the marginal distribution of X_j .

This quantity depends on the particular test point (x, y) and on the distribution $P_{\tilde{X}_j}$. Now say that we want to find the expectation of this quantity over the true data distribution, whose density is denoted as $P_{X,Y}$. For the second term, we may simply write

$$\mathbb{E}_{X,Y}[\log q(Y|X)] = \int \log q(Y = y|X = \mathbf{x}) P_{X,Y}(\mathbf{x}, y) d\mathbf{x}.$$

The notation is natural because the density function used is the true density of X . However, note that in the above definition for PFI at a single point, we have integrated out X_j already, but rather than using its true distribution, we have integrated it out acting as though its distribution is that of \tilde{X}_j . This is what we reflect in our notation with $X_j = \tilde{X}_j$, which serves as shorthand for integrals of the following form

$$\begin{aligned} & \mathbb{E}_{X_{-j}, \tilde{X}_j, Y}[-\log q(Y|X_{-j}, X_j = \tilde{X}_j)] \\ & \stackrel{\text{def}}{=} - \int \int \int \log q(Y = y|X_j = \tilde{x}_j, X_{-j} = \mathbf{x}_{-j}) P_{\tilde{X}_j, X_{-j}, Y}(\tilde{x}_j, \mathbf{x}_{-j}, y) d\tilde{x}_j d\mathbf{x}_{-j} dy \\ & = - \int \int \int \log q(Y = y|X_j = \tilde{x}_j, X_{-j} = \mathbf{x}_{-j}) P_{\tilde{X}_j}(\tilde{x}_j) P_{X_{-j}, Y}(\mathbf{x}_{-j}, y) d\tilde{x}_j d\mathbf{x}_{-j} dy. \end{aligned}$$

Appendix B Model and hyperparameter setup

B.1 Gaussian processes

We use an exact Gaussian process (using GPyTorch [Gardner et al \(2018\)](#)) with a constant mean function and covariance kernel of an RBF kernel composed with a scale kernel. For hyperparameter optimisation, we use Adam on the hyperparameters with a learning rate of 0.1. For the experiment in Section 3 we train for 1000 epochs, for the experiment in Section 4, we train for 200 epochs.

B.2 Calibrated random forests

We use the scikit-learn ([Pedregosa et al, 2011](#)) implementation of random forests with 500 trees with maximum depth 8. 80% of training data is used to train the forest itself, with the remaining 20% used for calibration based on scikit-learn’s `CalibratedClassifierCV` with the “sigmoid” method.

B.3 MLP with Bayes by backprop

We use an MLP with two hidden layers with 100 units each. Weights are given a Gaussian prior with mean zero and standard deviation 0.5. The MLP is trained for 100 epochs with a learning rate of 0.01.

B.4 MLP with MC dropout

We use an MLP with two hidden layers with 100 units each and a dropout rate of 0.1. All features and targets are scaled so that the training set has zero mean and unit variance, and outputs are re-scaled to their original values for model evaluation metrics. The MLP is trained using SGD with learning rate 0.1 for 5000 epochs with a batch size of 200. Furthermore, the reserved 10% of the training data is used to tune the precision parameter τ by gradient descent.

B.5 Random forest regression model

The random forest regression model used in Section 4.1 is a random forest of 100 trees with no maximum depth. The scikit-learn `RandomForestRegressor` model is used with default arguments.

References

- Antoran J, Bhatt U, Adel T, et al (2021) Getting a {clue}: A method for explaining uncertainty estimates. In: International Conference on Learning Representations, URL <https://openreview.net/forum?id=XSLF1XFq5h>
- Blundell C, Cornebise J, Kavukcuoglu K, et al (2015) Weight uncertainty in neural network. In: International conference on machine learning, PMLR, pp 1613–1622
- Breiman L (2001) Random forests. *Machine learning* 45:5–32
- Casalicchio G, Molnar C, Bischl B (2019) Visualizing the feature importance for black box models. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18, Springer, pp 655–670
- Chai LR (2018) Uncertainty estimation in bayesian neural networks and links to interpretability. Master’s thesis, University of Cambridge
- Chen H, Covert IC, Lundberg SM, et al (2023) Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence* pp 1–12
- Covert IC, Lundberg S, Lee SI (2021) Explaining by removing: A unified framework for model explanation. *The Journal of Machine Learning Research* 22(1):9477–9566
- Depeweg S, Hernández-Lobato JM, Udfluft S, et al (2017) Sensitivity analysis for predictive uncertainty in bayesian neural networks. arXiv preprint arXiv:171203605

- Depeweg S, Hernandez-Lobato JM, Doshi-Velez F, et al (2018) Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In: International Conference on Machine Learning, PMLR, pp 1184–1193
- Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics* pp 1189–1232
- Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning, PMLR, pp 1050–1059
- Gardner JR, Pleiss G, Bindel D, et al (2018) Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In: *Advances in Neural Information Processing Systems*
- Goldstein A, Kapelner A, Bleich J, et al (2015) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics* 24(1):44–65
- Guo C, Pleiss G, Sun Y, et al (2017) On calibration of modern neural networks. In: International conference on machine learning, PMLR, pp 1321–1330
- Hooker G, Mentch L, Zhou S (2021) Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing* 31:1–16
- Kelly L, Sachan S, Ni L, et al (2020) Explainable artificial intelligence for digital forensics: opportunities, challenges and a drug testing case study. *Digital Forensic Science*
- Liu J, Paisley J, Kioumourtzoglou MA, et al (2019) Accurate uncertainty estimation and decomposition in ensemble learning. *Advances in neural information processing systems* 32
- Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, et al (eds) *Advances in Neural Information Processing Systems* 30. Curran Associates, Inc., p 4765–4774, URL <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>
- Lundberg SM, Erion G, Chen H, et al (2020) From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence* 2(1):56–67
- Mease D, Wyner A (2008) Evidence contrary to the statistical view of boosting. *Journal of Machine Learning Research* 9(2)
- Molnar C (2022) *Interpretable Machine Learning*, 2nd edn. Independently Published, URL <https://christophm.github.io/interpretable-ml-book>

- Molnar C, König G, Bischl B, et al (2023) Model-agnostic feature importance and effects with dependent features: a conditional subgroup approach. *Data Mining and Knowledge Discovery* pp 1–39
- Moosbauer J, Herbinger J, Casalicchio G, et al (2021) Explaining hyperparameter optimization via partial dependence plots. *Advances in Neural Information Processing Systems* 34:2280–2291
- Mukhoti J, Kirsch A, van Amersfoort J, et al (2023) Deep deterministic uncertainty: A new simple baseline. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 24384–24394
- Neal RM (2012) *Bayesian learning for neural networks*, vol 118. Springer Science & Business Media
- Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830
- Ribeiro MT, Singh S, Guestrin C (2016) ” 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
- Shaker MH, Hüllermeier E (2020) Aleatoric and epistemic uncertainty with random forests. In: *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18*, Springer, pp 444–456
- Slack D, Hilgard S, Jia E, et al (2020) Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp 180–186
- Smith JW, Everhart JE, Dickson W, et al (1988) Using the adap learning algorithm to forecast the onset of diabetes mellitus. In: *Proceedings of the annual symposium on computer application in medical care*, American Medical Informatics Association, p 261
- Strobl C, Boulesteix AL, Kneib T, et al (2008) Conditional variable importance for random forests. *BMC bioinformatics* 9:1–11
- Watson DS, O’Hara J, Tax N, et al (2023) Explaining predictive uncertainty with information theoretic shapley values. *arXiv preprint arXiv:230605724*
- Williams CK, Rasmussen CE (2006) *Gaussian processes for machine learning*, vol 2. MIT press Cambridge, MA

Wimmer L, Sale Y, Hofman P, et al (2023) Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In: Uncertainty in Artificial Intelligence, PMLR, pp 2282-2292

Yeh IC (2007) Concrete Compressive Strength. UCI Machine Learning Repository, DOI: <https://doi.org/10.24432/C5PK67>

Zhang X, Chan FT, Mahadevan S (2022) Explainable machine learning in image classification models: An uncertainty quantification perspective. Knowledge-Based Systems 243:108418