From Human Explanation to Model Interpretability: A Framework Based on Weight of Evidence

David Alvarez-Melis,[1] **Harmanpreet Kaur,**[2]
Hal Daumé III,[1, 3] **Hanna Wallach,**[1] **Jennifer Wortman Vaughan**[1]

[1] Microsoft Research
[2] University of Michigan
[3] University of Maryland
{alvarez.melis,hal3,hanna,jenn}@microsoft.com, harmank@umich.edu

Abstract

We take inspiration from the study of human explanation to inform the design and evaluation of interpretability methods in machine learning. First, we survey the literature on human explanation in philosophy, cognitive science, and the social sciences, and propose a list of design principles for machine-generated explanations that are meaningful to humans. Using the concept of weight of evidence from information theory, we develop a method for generating explanations that adhere to these principles. We show that this method can be adapted to handle high-dimensional, multi-class settings, yielding a flexible framework for generating explanations. We demonstrate that these explanations can be estimated accurately from finite samples and are robust to small perturbations of the inputs. We also evaluate our method through a qualitative user study with machine learning practitioners, where we observe that the resulting explanations are usable despite some participants struggling with background concepts like prior class probabilities. Finally, we conclude by surfacing design implications for interpretability tools in general.

Introduction

Interpretability has long been a desirable property of machine learning (ML) models. With the success of complex models like neural networks, and their expanding reach into high-stakes and decision-critical applications, explaining ML models' predictions has become even more important. Interpretability can enable model debugging and lead to more robust ML systems, support knowledge discovery, and boost trust (Hong, Hullman, and Bertini 2020). It can also help to mitigate unfairness by surfacing undesirable model behavior (Tan et al. 2018; Dodge et al. 2019), lead to increased accountability by enabling auditing (Selbst and Barocas 2018), and enable ML practitioners to better communicate model behavior to stakeholders (Veale, Van Kleek, and Binns 2018; Hong, Hullman, and Bertini 2020).

There are two primary techniques for achieving interpretability of ML models. The first is to train transparent, or *glass-box*, models that are intended to be inherently interpretable, such as decision trees (Quinlan 1986) and sets (Lakkaraju, Bach, and Jure 2016), simple point systems (Zeng, Ustun, and Rudin 2017; Jung et al. 2020), and generalized additive models (Hastie and Tibshirani 1990; Caruana et al. 2015). Although some researchers have argued that glass-box models should always be used in high-stakes scenarios (Rudin 2019), complex *black-box* models, such as neural networks, random forests, and ensemble methods, are very widely used in practice. As a result, other ML researchers have gravitated towards interpretability methods that generate post-hoc *local* explanations for individual predictions produced by such models (e.g., Simonyan, Vedaldi, and Zisserman 2013; Selvaraju et al. 2017; Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017; Alvarez-Melis and Jaakkola 2017).

Local explanations aim to answer the question of why a model $\mathcal{M}$ predicted a particular output y for some input x. There are many ways of operationalizing this abstract question, but most methods do so by addressing the proxy question of how much the value of each input feature x_i contributed to the prediction y. Thus, in practice, the explanations generated by many such methods consist of importance scores indicating the positive or negative relevance of each input feature x_i. Although the way these scores are computed varies from method to method, most start from an axiomatic or algorithmic derivation of some notion of feature importance, and only later investigate whether the resulting explanations are useful to humans. Some methods forgo this last step altogether, relying exclusively on intrinsic evaluation of mathematical properties of explanations, such as robustness or faithfulness to the underlying model (Alvarez-Melis and Jaakkola 2018a,b; Jacovi and Goldberg 2020).

Interpretability, however, is fundamentally a human-centered concept. In light of this, we put human needs at the center of both the design and evaluation of interpretability methods. Our work builds upon and weaves together two literatures that study the relationship between humans and explanations. First, researchers in philosophy, cognitive science, and the social sciences have long studied what it means to explain, and how humans do it (e.g., Pitt 1988; Miller 2019b, and references therein). Second, a recent line of work within the human–computer interaction community has focused on how humans understand and utilize interpretability tools (i.e., software implementations of interpretability methods) (Lim and Dey 2011; Bunt, Lount, and Lauzon 2012; Bussone, Stumpf, and O'Sullivan 2015; Hohman et al. 2019; Lage et al. 2019; Abdul et al. 2020; Poursabzi-Sangdeh et al. 2021), where a common finding is that practitioners misunderstand, over-trust, and misuse these tools (e.g., Kaur et al. 2020).

Inspired by Miller (2019b), we start by surveying the literature on the nature of explanation, revealing recurring characteristics of human explanation that are often missing from interpretability methods. We distill these characteristics into design principles that we argue human-centered machine-generated explanations should satisfy. We then realize our design principles using the concept of *weight of evidence* (WoE) from information theory (Good 1985), which has recently been advocated for by Spiegelhalter (2018), but, to the best of our knowledge, has yet to be investigated in the context of interpretability. We demonstrate that WoE can be adapted to handle high-dimensional, multi-class settings, yielding a suitable theoretical foundation for interpretability. We provide a general, customizable meta-algorithm to generate explanations for black-box models. We also show experimentally that WoE can be estimated from finite samples and is robust to small perturbations of the inputs.

Evaluation of interpretability methods is notoriously difficult (Doshi-Velez and Kim 2017; Kaur et al. 2020). Although recent work has focused on abstract, intrinsic metrics such as robustness or faithfulness to the underlying model (Alvarez-Melis and Jaakkola 2018a), considerably less attention has been given to understanding how the resulting explanations are used in practice. This discrepancy between the intended use of the explanations — by a human, for a specific goal such as auditing, debugging, or building trust in a model — and their experimental evaluation — typically performed using abstract, intrinsic metrics, in generic settings — hampers understanding of the benefits and failure points of different interpretability methods.

We build on a recent thread of work (e.g., Lage et al. 2019; Nourani et al. 2019; Li et al. 2020; Kaur et al. 2020; Vaughan and Wallach 2021; Poursabzi-Sangdeh et al. 2021) — including several recent papers from the human computation community — that argues that evaluations should be grounded in concrete use cases and should put humans at the center, taking into account not only how they use interpretability tools, but how well they understand the principles behind them. We carry out an artifact-based interview study with ten ML practitioners to investigate their use of a tool implementing our meta-algorithm in the context of a practical task. Qualitative themes from this study suggest that most participants successfully used the tool to answer questions, despite struggling with background concepts like prior class probabilities. Although the study was designed to identify preferences for different tool modalities, participants often used all of them and requested the option to switch between them interactively. Our results additionally highlight the importance of providing well-designed tutorials for interpretability tools — even for experienced ML practitioners — which are often overlooked in the literature on interpretability methods, and which we argue should be an integral part of any interpretability tool.

Human-Centered Design Principles

What it means to explain and how humans do it have long been studied in philosophy, cognitive science, and the social sciences. We draw on this literature to propose human-centered design principles for interpretability methods.

Hempel and Oppenheim (1948) and van Fraassen (1988) define an explanation as consisting of two main pieces: the *explanandum*, a description of the phenomenon to be explained, and the *explanans*, the facts or propositions that explain the phenomenon, which may rely on relevant aspects of context. As is often done colloquially, we will refer to the explanans as the *explanation*. Different ways of formalizing the explanation have given rise to various theories, ranging from logical deterministic propositions (Hempel and Oppenheim 1948) to probabilistic ones (Salmon 1971; van Fraassen 1988). An excellent historical overview can be found in the surveys by Pitt (1988) and Miller (2019b).

In the context of local explanations for predictions made by ML models, the phenomenon to be explained is why a model $\mathcal{M}$ predicted output y for input x. This why-question can be operationalized in different ways. The facts used to explain this phenomenon may include information about the input features, the model parameters, the data used to train the model, or the manner in which the model was trained.

Although the nature of explanation is far from settled, recurring themes emerge across disciplines. At the core of the theories by van Fraassen (1988) and Lipton (1990) is the hypothesis that humans tend to explain in contrastive terms (e.g., "a fever is more consistent with the flu than with a cold"), with explanations that are both factual and counterfactual (e.g., "had the patient had chest pressure too, the diagnosis would instead have been bronchitis"). Yet, the explanations produced by most current interpretability methods refer only to why the input x points to a single hypothesis (i.e., the prediction y) rather than ruling out all alternatives.[1] In light of this, we propose our first two design principles:

1. **Explanations should be contrastive**, i.e., explicate why the model predicted y instead of alternative y'.
2. **Explanations should be exhaustive**, i.e., provide a justification for why every alternative y' was not predicted.

Another theme, featured prominently by Hempel (1962), is that human explanations decompose into simple components. In other words, humans usually explain using multiple simple accumulative statements, each addressing a few aspects of the evidence (e.g., "a fever rules out a cold in favor of bronchitis or pneumonia; among these, chills suggest the latter"). Each component is intended to be understood without further decomposition. Again, this contrasts with current interpretability methods that explain in one shot, for example, by providing importance scores for all features simultaneously. Our next two design principles are therefore:

3. **Explanations should be modular and compositional**, breaking up predictions into simple components.

[1] Exceptions include recent work advocating for contrastive or counterfactual explanations (Wachter, Mittelstadt, and Russell 2017; Miller 2019a; van der Waa et al. 2018), partly inspired by contrast sets (Azevedo 2010; Bay and Pazzani 1999; Webb, Butler, and Newlands 2003; Novak, Lavrač, and Webb 2009).

4. **Explanations should rely on easily-understandable quantities**, so that each component is understandable.

Another recurring theme is minimality. In a survey of over 250 papers, Miller (2019b) argued that it is important, but underappreciated in ML, that only the most relevant facts be included in explanations. Thus, our final principle is:

5. **Explanations should be parsimonious**, i.e., only the most relevant facts should be provided as components.

These design principles are not exhaustive; each could be refined or generalized, and other principles could be derived from the same literature. However, we posit that these principles provide a reasonable starting point because they capture some of the most apparent discrepancies between human and machine-generated explanations. More generally, these principles point to a broader theme of human explanations as a *process* rather than (only) a *product* (Miller 2019b; Lombrozo 2012). Therefore, these principles work to shift interpretability methods from the latter towards the former.

Explaining with the Weight of Evidence

The set of design principles proposed in the previous section outlines a framework for human-centered interpretability in ML. In this section, we show how this framework can be operationalized by means of the *weight of evidence*, a simple but powerful concept from information theory. We operationalize the question of why model $\mathcal{M}$ predicted output y for input x in terms of how much *evidence* each input feature x_i (or feature group) provides in favor of y relative to alternatives. An explanation based on this question adheres to our design principles because it is based on a familiar concept (evidence) that is grounded in common language, it naturally evokes a contrastive statement (evidence *for* or *against* something), and, as we explain below, it can be formalized using simple quantities that admit modularity.

Weight of Evidence: Foundations

The weight of evidence (WoE) is a well-studied probabilistic approach for analyzing variable importance that traces its origins back to Peirce (1878), but was popularized by Good (1950, 1968, 1985), whose definition and notation we follow here. Given a hypothesis and some evidence, the WoE seeks to answer the following question: *"How much does the evidence speak in favor of or against the hypothesis?"*

The WoE is usually defined for some evidence e, a hypothesis h, and its logical complement $\bar{h}$. For example, in a simple binary classification setting, $e = (X_1, \dots, X_n)$, $h : Y = 1$, and $\bar{h} : Y = 0$. The WoE of e in favor of h is the log-odds ratio between h conditioned on e and h marginally:

$$\text{woe}(h : e) \triangleq \log \frac{O(h \mid e)}{O(h)}, \tag{1}$$

where $O(\cdot)$ denotes the odds of a hypothesis, i.e.,

$$O(h) \triangleq \frac{P(h)}{P(\bar{h})} \qquad \text{and} \qquad O(h \mid e) \triangleq \frac{P(h \mid e)}{P(\bar{h} \mid e)}. \tag{2}$$

Using Bayes' rule, woe$(h : e)$ can also be defined as

$$\text{woe}(h : e) \triangleq \log \frac{P(e \mid h)}{P(e \mid \bar{h})}. \tag{3}$$

These two equivalent definitions provide complementary views of the WoE: the *hypothesis-odds* and *evidence-likelihood* interpretations. Using Equation (1), woe$(h : e) > 0$ indicates that the odds of h are higher under e than marginally. Equivalently, using Equation (3), it indicates that the likelihood of e is larger when conditioning on h than on its complement. In other words, the evidence *speaks in favor of* hypothesis h. Analogously, if woe$(h : e) < 0$ we would say that the evidence *speaks against* h. The quantities in Equations (1) and (3) are contrastive (cf. Principle 1) — that is, defined in terms of ratios.

As a concrete example, suppose that a doctor wants to know whether a patient's symptoms indicate the presence of a certain disease, say, the flu. Denote $e =$ "*the patient has a fever*," $h =$ "*the patient has the flu*," and $\bar{h} =$ "*the patient doesn't have the flu*." The doctor might know that for a patient, the odds of having the flu roughly double once the patient's fever is taken into account (i.e., the hypothesis-odds interpretation), which corresponds to woe$(h : e) \approx \log 2$. Alternatively, using the evidence-likelihood interpretation, the doctor could conclude that a patient is twice as likely to have a fever if they have the flu compared to when they do not. Note that neither interpretation tells us anything about the base rate of the flu.

The WoE generalizes beyond these simple scenarios. For example, it can be conditioned on additional information c:

$$\text{woe}(h : e \mid c) \triangleq \log \frac{P(e \mid h, c)}{P(e \mid \bar{h}, c)}.$$

It can also contrast h to an arbitrary alternative hypothesis h' instead of $\bar{h}$ (e.g., evidence in favor of the flu and against a cold): woe$(h/h' : e) \triangleq$ woe$(h : e \mid h \vee h')$. Thus, we can, in general, talk about the strength of evidence in favor of h and against h' provided by e (perhaps conditioned on c).

When the evidence is decomposable into multiple parts — that is, when $e = \bigcup_{i=1}^{n} e_i$ — the WoE is also decomposable:

$$\text{woe}(h/h' : e) = \sum_{i=1}^{n} \log \frac{P(e_i \mid e_{i-1}, \dots, e_1, h)}{P(e_i \mid e_{i-1}, \dots, e_1, h')}. \tag{4}$$

This is crucial to defining an extension of the WoE to high-dimensional inputs that adheres to Principle 3 (modularity).

A further appealing aspect of the WoE is its immediate connection to Bayes' rule through the following identity:

$$\underbrace{\log \frac{P(h \mid e)}{P(h' \mid e)}}_{\text{Posterior log odds}} = \underbrace{\log \frac{P(h)}{P(h')}}_{\text{Prior log odds}} + \underbrace{\log \frac{P(e \mid h)}{P(e \mid h')}}_{\text{Weight of evidence}}. \tag{5}$$

In other words, the WoE can be understood as an adjustment to the prior log odds caused by observing the evidence. In a simple binary classification setting, this amounts to

$$\log \frac{P(Y=1 \mid X)}{P(Y=0 \mid X)} = \log \frac{P(Y=1)}{P(Y=0)} + \text{woe}(Y = 1 : X),$$

which shows that a positive (respectively, negative) WoE implies that the posterior log odds of $Y = 1$ versus $Y = 0$ are higher (lower) than the prior log odds, indicating that the evidence makes $Y = 1$ more (less) likely than it was *a priori*.

Equation (5) shows that the WoE is modular (cf. Principle 3) in another important way: it disentangles prior class probabilities and input likelihoods. This is important because of the *base rate fallacy* studied in the behavioral science literature (Tversky and Kahneman 1974; Bar-Hillel 1980; Eddy 1982; Koehler 1996). This cognitive bias, prevalent even among domain experts, is characterized by a frequent misinterpretation of posterior probabilities, primarily caused by a neglect of base rates (i.e., prior probabilities). Despite this, many interpretability methods do not explicitly display prior probabilities, and even when they do, they focus on explaining posterior probabilities, which invariably entangle information about priors and the input being explained.

Additionally, the units in which the WoE is expressed (log-odds ratios) are arguably easily understandable (cf. Principle 4). There is evidence from the cognitive-neuroscience literature that log odds are a natural unit in human cognition. For example, degrees of confidence expressed by humans are proportional to log odds (Peirce and Jastrow 1885), people are less biased when responding in log odds that in linear scales (Phillips and Edwards 1966), and there exist plausible neurological hypotheses for encoding of log odds in the human brain (Gold and Shadlen 2001, 2002). We refer the reader to Zhang and Maloney (2012) for a meta-analysis of these various studies of log odds.

We provide additional properties of the WoE, along with an axiomatic derivation, in the appendix, which can be found in the longer version of this paper, available online.[2]

Composite Hypotheses and Evidence

Traditionally, the WoE has been mostly used in simple settings, such as a single binary output and only a few input features. Its use in the more complex settings typically considered in modern ML therefore poses new challenges.

The first such challenge is that in multi-class settings, there is flexibility in choosing the hypotheses h and h' to contrast. The obvious choice of letting h correspond to the predicted class y^* and h' its complement is unlikely to yield useful explanations when the number of classes is large (e.g., explaining the evidence in favor of one disease against one hundred thousand others). Following Principle 3 (modularity), and taking inspiration from Hempel's model (1962) and the view of explanation as a process (Lombrozo 2012; Miller 2019b), we address this by casting explanation as a sequential procedure, whereby a subset of the possible classes is ruled out at each step. For example, in medical diagnosis, we might first explain why bacterial diseases were ruled out in favor of viral ones, and then explain why a specific viral disease was predicted instead of the others. In general, for a classification problem over labels $\mathsf{Y} = \{1, \dots, k\}$, we will consider a (given or constructed) nested partitioning of Y into a sequence of T-many subsets U_i of classes such that $\{y^*\} \triangleq \mathsf{U}_T \subset \mathsf{U}_{T-1} \subset \cdots \subset \mathsf{U}_0 \triangleq \mathsf{Y}$. As we show in Figure 4 in the appendix, this partition implies a sequence of pairs of hypotheses $(h_t, h'_t) = (y \in \mathsf{U}_t, y \in \mathsf{U}_{t-1} \setminus \mathsf{U}_t)$.

A second challenge arises when the the number of input features is large. For very high-dimensional inputs (such as images or detailed health records), providing a WoE value for each feature will rarely be informative. Again, imagine our hypothetical doctor having to simultaneously analyze the relevance of thousands of symptoms. For such cases, we propose aggregating the input features into feature groups (e.g., super-pixels for images or groups of related symptoms for medical diagnosis). Formally, for an input X of dimension n, we partition the feature indices into m disjoint subsets, with $\mathsf{S}_1 \cup \cdots \cup \mathsf{S}_m = \{1, \dots, n\}$. Equation (4) (or, equivalently, the chain rule of probability) allows for arbitrary groupings, so for any such partition we can compute

$$\operatorname{woe}(h/h' : X) = \sum_{i=1}^{m} \underbrace{\log \frac{P(X_{\mathsf{S}_i} \mid X_{\mathsf{S}_{i-1}}, \dots, X_{\mathsf{S}_1}, h)}{P(X_{\mathsf{S}_i} \mid X_{\mathsf{S}_{i-1}}, \dots, X_{\mathsf{S}_1}, h')}}_{= \operatorname{woe}(h/h' : X_{\mathsf{S}_i} \mid X_{\mathsf{S}_{i-1}}, \dots, X_{\mathsf{S}_1})} \tag{6}$$

where $X_{\mathsf{S}_i} = \{X_j\}_{j \in \mathsf{S}_i}$ is the ith feature group, or "atom."

Algorithm 1 WoE meta-algorithm for complex models

```
 1: Input: Instance $X \in \mathbb{R}^n$, prediction $y^* \in \{1, \dots, k\}$
 2: Parameters: Features $\mathcal{A} = \{\{1\}, \dots, \{n\}\}$ or feature
    groups $\mathcal{A} = \{S_1, \dots, S_m\}$
 3: Initialize $\mathsf{U}_0 \leftarrow \{1, \dots, k\}$
 4: $t \leftarrow 0$
 5: while $|\mathsf{U}_t| > 1$ do
 6:     $t \leftarrow t + 1$
 7:     $\mathsf{U}_t \leftarrow \textsc{SelectHypothesis}(\mathsf{U}_{t-1}, y^*)$
 8:     $\overline{\mathsf{U}}_t \leftarrow \mathsf{U}_{t-1} \setminus \mathsf{U}_t$ {relative complement}
 9:     $\pi(\mathsf{U}_t) \leftarrow \log \frac{P(y \in \mathsf{U}_t)}{P(y \in \overline{\mathsf{U}}_t)}$ {prior log odds}
10:     for $i = 1, \dots, |\mathcal{A}|$ do
11:         $\omega_i^t \leftarrow \operatorname{woe}(y \in \mathsf{U}_t / y \in \overline{\mathsf{U}}_t : X_{\mathcal{A}_i} \mid X_{\mathcal{A}_{i-1}}, \dots, X_{\mathcal{A}_1})$
12:     end for
13:     $\Omega_t \leftarrow \sum_{i=1}^{|\mathcal{A}|} \omega_i^t$
14:     $\textsc{DisplayExplanation}(\mathsf{U}_t, \overline{\mathsf{U}}_t, \mathcal{A}, \pi(\mathsf{U}_t), \{\omega_i^t\}_i, \Omega_t)$
15: end while
```

A Meta-Algorithm for WoE Explanations

Using these extensions of the WoE, we propose a meta-algorithm for generating explanations for complex classifiers (Algorithm 1). Given a model, an input, and a prediction, the algorithm generates an explanation for the prediction sequentially by producing WoE values for progressively smaller nested hypotheses. Specifically, at every step t, a subset of classes $\mathsf{U}_t \subset \mathsf{U}_{t-1}$ is selected and the remaining classes $\overline{\mathsf{U}}_t = \mathsf{U}_{t-1} \setminus \mathsf{U}_t$ are ruled out. The user is shown a comparison of hypotheses $h_t : y \in \mathsf{U}_t$ and $h'_t : y \in \overline{\mathsf{U}}_t$ consisting of both their prior log odds $\pi(\mathsf{U}_t)$ (line 9) and the WoE in favor of h_t and against h'_t. WoE values are computed sequentially with each atom $\mathcal{A}_i$ (either an individual input feature or a group of features) as the evidence (line 11) and these values are summed to obtain the total WoE using the additive property (line 13). These values are presented to the user, and the process continues until all classes except the prediction y^* have been ruled out (cf. Principles 2–3).

Left unspecified in this meta-algorithm are four key choices that are application-dependent and require fur-

[2] https://arxiv.org/abs/2104.13299

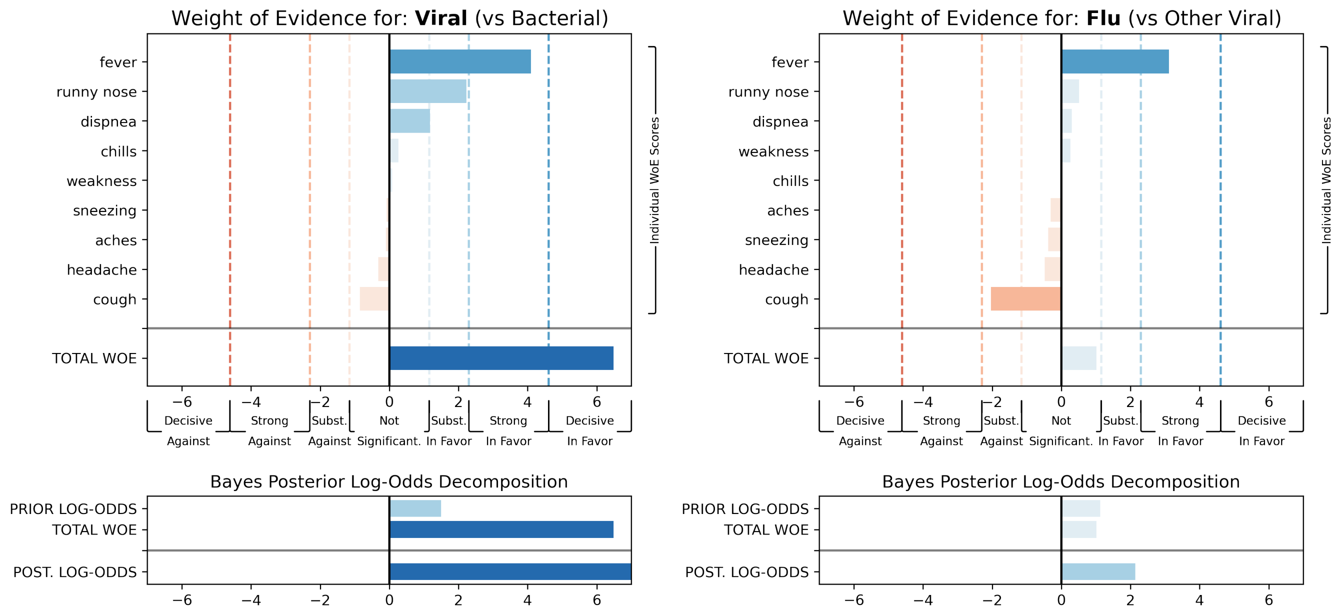


Figure 1: Example two-step explanation produced by our method for a model that predicted the flu for some input. The first step explains, using the weight of evidence, why the model favors viral diseases (instead of bacterial ones) for this input. Then, the second step explains why the model predicted the flu instead of the remaining possible classes (i.e., other viral diseases).

ther discussion. First is the question of how to define the SELECTHYPOTHESIS method to progressively partition the classes (line 7). If there is an inherent natural partitioning (e.g., "viral" versus "bacterial," as in the example discussed previously), then SELECTHYPOTHESIS simply amounts to retrieving the largest subset in the partition containing the prediction y^*. For the general case, we propose selecting the hypothesis that maximizes a WoE-based objective:

$$\mathsf{U}_t \leftarrow \operatorname*{argmax}_{\mathsf{U}\subset \mathsf{U}_{t-1};\, y^*\in\mathsf{U}} \mathrm{woe}(y\in\mathsf{U} \,/\, (y\in \mathsf{U}_{t-1}\setminus\mathsf{U}) : X) - R(\mathsf{U}),$$

where R is a cardinality-based regularizer. R should be chosen to penalize sets that are too small (which would yield granular explanations with many steps, in opposition to Principle 5) or too large (which would yield coarse explanations, to the detriment of Principle 3). Although the choice of R should ideally be informed by the application and the user, a sensible generic choice is $R(\mathsf{U}) \propto \left||\mathsf{U}| - \frac{1}{2}|\mathsf{U}_{t-1}|\right|^p$, normalized so that $R(\mathsf{U}) \in [0,1]$. Using this regularizer, Algorithm 1 approximately splits the remaining classes in half at every step, yielding roughly $O(\log k)$ steps in total.

Second, it should be noted that lines 10–12 in Algorithm 1 implicitly assume an ordering of the atoms, and that this ordering might affect the WoE values. In some applications, there might be a conditional independence structure known a priori that could inform the choice of atoms and their ordering (e.g., those simplifying the conditioning in line 11 the most). If not, the ordering can again be chosen randomly or based on the sorted per-atom conditional WoE values.

Third, computing the per-atom WoE (line 11) requires the conditional likelihoods $P(X_{\mathcal{A}_i}|X_{\mathcal{A}_{i-1}},\dots,X_{\mathcal{A}_1},Y)$. Ideally, the model would compute these likelihoods internally. If, instead, it computes only marginal feature likelihoods $P(X_{\mathcal{A}_i}|Y)$, we can use a naïve Bayes (NB) approximation — that is, use these in place of the conditional likelihoods in Equation (6). If the model is a black box or does not compute conditional likelihoods internally, then these must be estimated as we explain below.

Finally, there is the question of how to implement DISPLAYEXPLANATION. When the number of atoms is large, the WoE values for only the most salient atoms can be displayed (cf. Principle 5) — e.g., those with absolute WoE larger than a given threshold τ; Good (1985) suggests $\tau = 2$ as a rule of thumb. Otherwise, all per-atom WoE values can be displayed along with the total WoE and prior log odds.

An example two-step explanation produced by our method (on fabricated data from our user study tutorial) is shown in Figure 1. The sorting and color coding of the features by their WoE values makes it apparent which of these contribute the most evidence in favor or against the selected class (or set of classes), and the labeling along the x-axis provides guidelines for context. The visualization suggests the additive nature of these values (i.e., that stacking blue bars and subtracting red ones yields the total WoE). The bottom panel, a graphical representation of Equation (5), disentangles the model's estimated prior class odds (which *a priori* weakly favor the flu and other viral diseases), from its total WoE (very strong when contrasting viral and bacterial diseases, less so for the flu versus other viral diseases).

WoE Estimation for Black-box Models

As we noted above, computing the per-atom WoE (line 11 in Algorithm 1) requires evaluating the conditional likelihoods $P(X_{\mathcal{A}_i}|X_{\mathcal{A}_{i-1}},\dots,X_{\mathcal{A}_1},Y)$. In many practical settings, including those in which the model is a black box, these conditional likelihoods are not computed internally, so they must be estimated. In such settings, we propose fitting a conditional likelihood estimation model using the model's predictions $\hat{y} = \mathcal{M}(x)$ (not the true labels y) as a preliminary step. This conditional likelihood estimation model can then be called on demand when computing the per-atom WoE.

In some settings, it may be possible to fit a full (i.e., conditioned on *both* the class Y and all previous atoms $A_i - 1, \ldots, A_1$) conditional likelihood estimation model, for example, via kernel or spectral density estimation methods when working with low-dimensional data, or via autoregressive or recurrent neural networks when working with text or time-series data. For more complex types of data, such as images, methods based on normalizing flows and neural autoregressive models (e.g., Rezende and Mohamed 2015; Papamakarios, Pavlakou, and Murray 2017) are likely to be more appropriate. In settings where fitting a full conditional likelihood estimation model is infeasible, an NB approximation can be used to estimate class- (but not atom-) conditional likelihoods, for example, via a Gaussian NB classifier.

We emphasize that fitting a conditional likelihood estimation model — the main computational bottleneck of our method — must be done only once, potentially offline. This is in contrast to perturbation-based interpretability methods like LIME that fit a new model for every prediction.

We assess the quality of finite-sample WoE estimation experimentally in the "Quantitative Experiments" section.

Relation to LIME and SHAP

When viewed from a probabilistic perspective, most post-hoc interpretability methods revolve around a model's predictive posterior — that is, they seek explanations that deconstruct $P(Y = y^* \mid X)$ in various ways. For example, LIME (Ribeiro, Singh, and Guestrin 2016) seeks to approximate $f(x) = P(Y \mid X = x)$ in the vicinity of x_0 through a simpler, interpretable surrogate model $\tilde{f}(x)$. Similarly, SHAP (Lundberg and Lee 2017) quantifies variable importance by analyzing the effect on the posterior of "dropping" variables X_i from the input X. In contrast, the WoE focuses — directly, in the case of the evidence-likelihood interpretation and indirectly in the case of the hypothesis-odds interpretation — on the conditional likelihood $P(X = x \mid Y)$. In other words, for a given input x, a WoE explanation is based on the probability assigned by the model to x (or a subset of its features) given, e.g., $Y = y^*$.

When viewed in this way, the relationship between WoE explanations and other post-hoc interpretability methods like LIME and SHAP is akin to the relationship between generative and discriminative models. Indeed, as is the case for some pairs of generative and discriminative models (e.g., naïve Bayes and logistic regression), these different interpretability methods also turn out to be equivalent — two sides of the same coin — for some simple classifiers, as we show in the appendix for logistic regression. However, this is not generally the case. Moreover, even when the explanations generated by different interpretability methods qualitatively agree (i.e., the same features are highlighted as being important), the specific interpretations of the explanations will differ. Indeed, the WoE uses a different operationalization of the notion of feature importance, in turn entailing different units of explanation: log likelihoods and log-odds ratios in the case of the WoE and linear attribution scores for posterior probabilities in the case of LIME and SHAP.

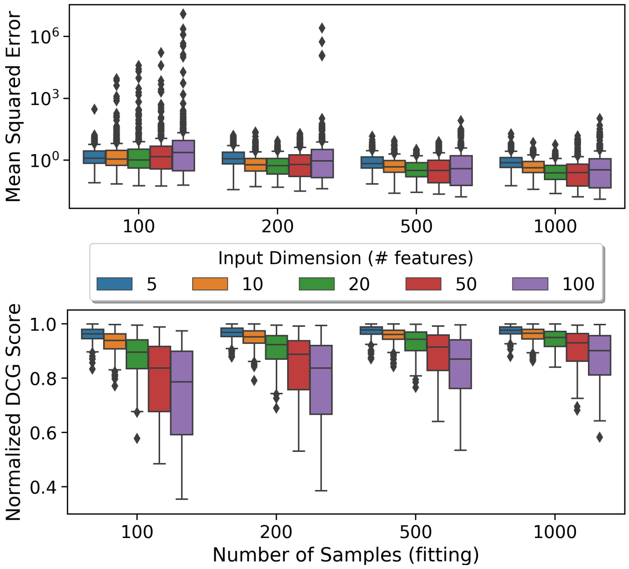


Figure 2: Quality of WoE estimation. Top: MSE. Bottom: NDCG, ranging from 0 (worst) to 1 (perfect) ranking quality.

Quantitative Experiments

Here, we assess the quality of finite-sample WoE estimation and the robustness of the WoE to perturbations of the inputs.

Quality of Finite-Sample WoE Estimation

Our first experiment evaluates the quality of WoE estimates from finite samples. As we explained above, such estimates are needed if the model is a black box or does not compute likelihoods internally. For evaluation purposes, we consider a model that, by construction, computes all the quantities required for exact WoE computation, but treat it as black box — that is, its internal WoE computation will be used only for evaluation, and is not available to our method. Instead, we separately fit a conditional likelihood estimation model by querying the model for a small number of inputs, and use this estimation model to compute WoE values at explanation time. We control for model misspecification by having both the model and our conditional likelihood estimation model use the NB assumption. Specifically, we use a smoothed Gaussian NB (GNB) classifier. This allows us to focus on the quality of WoE estimation from finite samples, but does not address model misspecification.

First, we generate a dataset of a given dimension. We train a model on a subset of this dataset of size $N_{\text{train}} = 1000$ and fit the conditional likelihood estimation model on a separate subset of size N_{fit}, which we vary. For every test input x_i ($N_{\text{test}} = 10$), we compute true WoE values for each input feature using the model's prior and posterior probabilities,

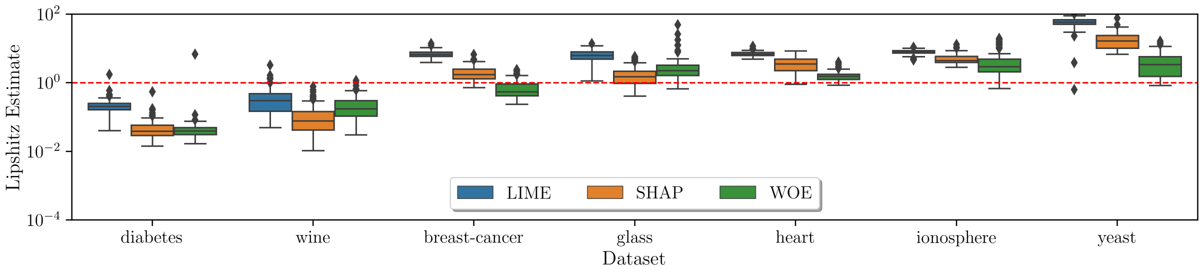


Figure 3: Explanation robustness across benchmark datasets. Extreme values away from $L = 1$ (dashed line) are undesirable.

and then compute estimated WoE values according to the estimation model in the appendix. We compare these using two metrics: mean squared error and normalized discounted cumulative gain (NDCG), a measure of ranking quality that might be relevant to practitioners, applied to the relative ranking of input features by their (true or estimated) WoE.[3]

Figure 2 shows these metrics as a function of input dimension and sample size N_{fit}. As expected, estimation quality improves with the number of samples used for fitting, and degrades gracefully as the input dimension increases. These results suggest that the WoE can be accurately estimated — even in relatively high dimensions — from finite samples, although we caution that these results may look different for other models, and we do not measure model misspecification error due to the NB approximation.

Robustness of WoE

Previous work has argued that interpretability methods should be robust in the sense that the explanations they provide should not vary dramatically when the input whose prediction is being explained changes by a small amount. To investigate the robustness of our method, we follow the setup of Alvarez-Melis and Jaakkola (2018a). Letting $\mathcal{E}(\cdot)$ be a function that maps feature vectors $x \in \mathbb{R}^n$ to explanation vectors (e.g., importance scores) $e \in \mathbb{R}^n$, we quantify its robustness around x_0 through its local Lipschitz constant:

$$L(x_0) = \max_{x_j \in \mathcal{B}_\varepsilon(x_0)} \frac{\|\mathcal{E}(x_j) - \mathcal{E}(x_0)\|}{\|x_j - x_0\|}, \qquad (7)$$

where $\mathcal{B}_\varepsilon(x_0) = \{x \mid \|x - x_0\| \leq \varepsilon\}$. Intuitively, $L(x_0)$ quantifies the largest relative change in importance scores in a small neighborhood around x_0. Extreme values are usually undesirable, as they indicate explanations that are either too sensitive (large L) or not responsive enough (very small L) to changes in the input features. In most settings, values below 1 but bounded away from 0 are preferable.

Concretely, we first train a GNB classifier. Then, for any input x we use the classifier to generate a prediction y, and input both of these to our method to generate $\mathcal{E}(x)$, a vector of WoE values for each feature x_i. Since computing the robustness metric (7) involves maximization, we estimate this quantity from finite samples using Bayesian optimization, making repeated calls to $\mathcal{E}(\cdot)$. We focus on standard benchmark classification datasets from the UCI repository (Dua and Graff 2017), and compare our method to LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017). Figure 3 shows the results for ten repetitions with different random seeds for each dataset and interpretability method pair. The red dashed line indicates the bound $L(x) = 1$. For all but one dataset, WoE explanations are, on average, as close or closer to the ideal Lipschitz robustness as explanations generated by LIME and SHAP.

User Study

Throughout this paper, we have argued that interpretability is fundamentally a human-centered concept and that the evaluation of interpretability methods should therefore focus on the needs of humans, exploring how they use interpretability tools, as well as their understanding of the concepts that underlie them. In this section, we present a user study that we carried out to assess the usefulness of our method in the types of scenarios in which it would be used in practice. Such studies are commonly used in the HCI community to distinguish between designers' intended use of a tool and users' mental models (Gibson 1977; Norman 2013).

Study Design

We conducted an artifact-based interview study with 10 ML practitioners to evaluate the use of our interpretability method, implemented in a simple tool (the artifact), in a controlled setting. In such qualitative studies, the goal is to ensure sufficient interaction time and nuanced data collection for each participant, which is typically only feasible for relatively small sample sizes (Hudson and Mankoff 2014; Olsen Jr 2007; Turner, Lewis, and Nielsen 2006). Our study followed a think-aloud protocol, in which participants were asked to verbalize their thought processes as they used the tool to perform specific tasks, in order to help us identify specific concepts and functionalities that might be confusing. We placed participants in a controlled setting (rather than observe them using the tool on their own models) because of challenges including data access, inconsistencies in the types of data analyzed by the participants, and the potential difficulty of establishing patterns across settings.

Our study consisted of two main parts: a tutorial intended to introduce the concepts and functionalities needed to use our tool, followed by the main study in which participants answered questions about a pre-trained ML model using the tool. We also conducted pre-study interviews to establish participants' backgrounds in ML and post-study interviews

[3] The NDCG is only defined for positive values, so we compute it separately for positive and negative values and average them.

in which participants reflected on their experiences with the tool and how they might use it in their ML pipelines. The study design was approved by our internal institutional review board. Excerpts (screenshots) from the Jupyter notebooks that we used in the tutorial and in the main study are in the appendix; the complete notebooks are available online.[4]

Tutorial Kaur et al. (2020) found that practitioners often use interpretability tools without fully understanding them, highlighting the importance of providing well-designed tutorials and other accompanying documentation. For our user study, we designed a tutorial to introduce the concepts and functionalities needed to use our tool, to evaluate participants' understanding of these concepts, and to check whether their responses in the main study were likely based on a sound understanding — without being too time-consuming or tedious. After several iterations and pilot studies, we converged on an approximately 40-minute-long tutorial based on a Jupyter notebook containing equations, text, and images. This tutorial covered log-odds ratios, weight of evidence, feature group for high-dimensional inputs, and sequential explanations for multi-class settings.

Main Study The goal of the main study was to assess participants' understanding of the WoE and to investigate their use of our interpretability tool in the context of a realistic ML task. Participants were given a Jupyter notebook that included a dataset, an ML model trained using the dataset, and our tool. They were then asked to answer several questions with the help of outputs (i.e., visualizations) from our tool.

The ML model was a random forest classifier trained using the Online News popularity dataset (Fernandes, Vinagre, and Cortez 2015), which consists of 39,797 news articles. Each article is represented using 59 features that capture metadata about the article, such as its length, any links, and its sentiment polarity. We trained the model to predict the category that the article was published under (e.g., "Lifestyle" or "Business"), creating a six-class classification task. We chose this dataset and this task because the domain is understandable without expert knowledge or prior experience, the number of classes is large enough to permit meaningful sequential explanations, and there are enough features to make explanations based on feature groups sufficiently different from explanations based on individual features.

The main study was itself divided into two parts, which were designed to let us observe the use of the two extensions of the WoE described in the "Composite Hypotheses and Evidence" section: feature groups and sequential explanations. In the first part, participants were given the option to view explanations based on feature groups or explanations based on individual features, and were asked questions that could be answered using either type of explanation (e.g., "What aspects of the news article contributed the most to this prediction?"). This part of the study was intended to surface participants' preferences. In the second part, participants were given the option to generate one-shot or sequential explanations, and were asked questions that could only be precisely answered using sequential explanations (e.g., "Why didn't the model predict [subset of classes]?"). This part of the study was intended to assess whether participants could successfully use sequential explanations.

Participants Potential participants were recruited via email. To be considered for the study, they were asked to complete a survey about their ML background and their experience with interpretability tools. Of 41 survey respondents, we randomly selected 10 to participate in the study. All participants were ML practitioners (e.g., data scientists) with 1–20 years of experience. On average, participants rated the role of ML in their jobs as 6.7 and their experience with interpretability tools as 3.2, both on a scale of 0 ("not at all") to 7 ("extremely"). Participants also rated their familiarity with concepts from probability relevant to the WoE on a scale of 0 to 7. Their average ratings were 2.7 for posterior class probabilities, 3.3 for log likelihoods, 3.2 for log-odds ratios, and 0.9 for the WoE. On average, participants took 1.7 hours to complete the study. Each participant was compensated with a $40 Amazon gift card.

Methods Participants' open-ended answers were scored by comparing them to an answer key prepared in advance by two of the authors. Answers that correctly identified key aspects (e.g., a feature with a large positive WoE value, pushing the model toward a particular prediction) were treated as correct even if the participants' specific language did not exactly match the language in the answer key. To examine patterns of tool use, the usability of our tool, participants' interpretability needs, and participants' general impressions, we analyzed automatically generated audio transcripts for high-level themes using inductive thematic analysis (Braun and Clarke 2012) and affinity diagramming.

Results

The results from our user study divide naturally into three categories: participants' understanding of relevant concepts, tool usability and participants' preferences, and general needs for interpretability tools. First, the pre-study interviews and answers to the checkpoint questions in the tutorial provided insight into participants' understanding of concepts relevant to the WoE. Second, participants' approaches to the questions in the main study and their patterns of tool use enabled us to examine the usability of our tool. Finally, via the main study and the post-study interviews, we were able to uncover participants' general interpretability needs and additional criteria (beyond our design principles) to consider when designing and evaluating interpretability tools.

Understanding of Relevant Concepts Our analysis showed that most participants (7/10) struggled to understand and use prior class probabilities in the tutorial. The section on this topic was time-consuming: on average, participants spent a third of their tutorial time on this section. Eventually, most participants either ignored the prior class probabilities or used them incorrectly, supporting the base rate fallacy. Nonetheless, participants were able to use the WoE to correctly answer questions in the main study for which prior class probabilities were relevant. This raises the possibility that although they struggled with the abstract concept,

[4] http://github.com/dmelis/interpretwoe

they were able to use the information indirectly (e.g., via displayed class probabilities). This finding is consistent with those of Kaur et al. (2020), who showed that data scientists struggle to explain the concepts underlying the explanations produced by generalized additive models (Hastie and Tibshirani 1990; Caruana et al. 2015) and SHAP (Lundberg and Lee 2017), even though they still find these tools useful.

Although participants generally understood the concept of WoE, some confused negative WoE values with negative values for input features, thus finding it challenging to make sense of the explanations. As a result, two participants provided incorrect answers for the questions in the main study.

Tool Usability and Participants' Preferences Participants had no overwhelming preference between explanations based on feature groups and explanations based on individual features. Indeed, they noted that the two levels of granularity provide complementary information, and switching between the two options was a clear pattern across all participants. Although feature groups provide a high-level overview, making it "easier to manage [reading the plot]...[and the] direction of analysis is a lot clearer" (P8), explanations based on individual features help participants in "looking into more details in general...to know exactly which feature it was [that was responsible for a prediction]" (P10). We observed some differences in behavior based on participants' roles and expertise, though of course these are inconclusive with our small sample size. Participants with more ML experience tended to rely on feature-level plots, while those with customer-facing jobs more often provided high-level answers based on feature groups, noting that feature groups "provide customer-friendly explanations" (P8).

Participants found sequential explanations to be a helpful breakdown of a larger explanation into parts. P3 noted these were like a "story of how the prediction was made." Sequential explanations prompted more detailed answers to our questions and most participants (7/10) accurately answered questions in the second part of the main study using sequential explanations. They explained that the type of questions — which required understanding how each of the classes were ruled out — could not be answered via one-shot explanations. P8 commented, "I find this to be quite helpful... I guess without this breaking it down to this point I wouldn't have thought twice really about this [input feature group] being a [differentiating] factor between the two [output classes]...I think that this would be a nice like final understanding [of the predicted class], this goes a lot deeper than I probably could have gone just looking at that without the tool. So I think it was very helpful in that case."

Although most participants were happy with the level of detail presented, some participants with more ML experience expressed a desire for deeper understanding of how the explanations were generated. They understood the underlying concepts, but were wary of anything that appeared automated, including the breakdown of class comparisons in sequential explanations (which was automated) and the feature groups (which were actually manually generated).

Even though participants said that the tool helped them understand the model's predictions, not all of them envisioned the tool being added to their ML pipelines. Participants with significant prior ML experience already had established ways of ensuring that model predictions are reasonable, but recognized other exciting use cases for the tool, such as communicating complex predictions to less experienced end users. Particularly for high-risk domains, visualizations from the tool could help users probe odd predictions. P7 noted, "With my focus on medical data, I do see the need in working with a customer...there this [tool] would be a must-have. My team, we are engaged with customers and we have to educate them fast... So for me model interpretability there comes very close side by side with fairness."

General Needs for Interpretability Tools Most interpretability tools, including ours, rely on tutorials and other accompanying documentation to provide an introduction to the tool's concepts and functionalities. All participants appreciated the information presented in our tutorial: "I can't imagine doing this [study] without the tutorial. I generally know a lot more about these concepts now" (P5). The tutorial seemed to impact participants' overall accuracy in answering the questions in the main study — those who spent longer on the tutorial tended to provide more accurate and more thoughtful answers. This manifested as longer time spent on exploring the tool in the tutorial and ensuring that their answers to the checkpoint questions were accurate and thorough. Even participants with less ML experience provided accurate answers when they devoted time to the tutorial. Participants appreciated the example in the tutorial and were able to generalize from this example to the questions the main study: "The tutorial...helps you start in the right place. I went back to the example in the tutorial to [determine how to] answer questions in the study" (P6).

Finally, participants expressed a desire to be able to more easily switch between different options (e.g., input features versus feature groups) rather than re-running code. Interactivity was consequently the most commonly requested functionality in the post-study interview. This is in line with prior work on human-centered design principles for ML (Amershi et al. 2019; Hohman et al. 2019; Weld and Bansal 2019).

Limitations

All interpretability methods, including ours, involve various design choices and assumptions (both implicit and explicit), many of which give rise to potential limitations. First, the concept of interpretability is notoriously ambiguous, and unlike supervised ML tasks, there is no ground truth to use for evaluation, even for proxy concepts like feature importance. As a result, different interpretability methods assume different notions of interpretability, propose different quantities to operationalize them, and (when needed) rely on different techniques to estimate them. In turn, these choices mean that no interpretability method will ever be universally ideal. Moreover, summarizing the behavior of complex models comes at a price (Rudin 2019) — that is, the explanations are partial, only hold in a small neighborhood (Ribeiro, Singh, and Guestrin 2016), or make strong assumptions about the data (Lundberg and Lee 2017). As a result, explanations generated by one interpretability method seldom strictly dominate explanations generated by another. Furthermore, different explanations might reveal information about different aspects of the underlying model's behavior.

Under this perspective, the design choices and assumptions involved in our interpretability method necessarily limit its scope and applicability. Starting from our decision to distill characteristics of human explanation into human-centered design principles, our method assumes that human characteristics are desirable for machine-generated explanations. And, although the specific characteristics that we focus on yield a coherent set of design principles, these principles are not exhaustive or universal. Some may not be necessary in all settings, and all are open to refinement.

By relying on the concept of weight of evidence (WoE) from information theory, our method inherits many of its strengths and limitations. Concretely, there are three main settings in which there is a clear case for using explanations based on the WoE: 1) when the underlying model is generative, 2) when the underlying model is log linear, and 3) when the underlying model is a multi-class classifier. We provide a detailed discussion of these three settings in the appendix. In terms of limitations, the WoE requires access to the conditional likelihoods $P(X_{\mathcal{A}_i}|X_{\mathcal{A}_{i-1}}, \dots, X_{\mathcal{A}_1}, Y)$, which limits its use to settings in which these are accessible or can be accurately estimated from finite samples. Estimating densities for more complex types of data, such as images, is an active area of research, and although it may be possible to integrate new advances into our method, its applicability to such types of data is currently limited. Other important design choices involved in our method include the technique for partitioning the classes in multi-class settings to yield sequential explanations and the technique for partitioning input features into feature groups. Although we chose generic solutions to these challenges, there are other techniques, which may be more appropriate in some settings, which will invariably lead to different explanations. We defer a thorough investigation of these choices for future work.

The main limitations of our user study are the number of participants, the type of participants, and the extent to which the study conditions mimic a realistic setting. We chose to conduct an artifact-based interview study to ensure sufficient interaction time and nuanced data collection for each participant, but this limited the number of participants that we could consider, thereby precluding statistical analyses. Our participants were also limited to ML practitioners. Following previous work (e.g., Kaur et al. 2020), we chose to focus on ML practitioners because they are frequent users of interpretability tools in the wild. Finally, although we tried to design the study so as to mimic a realistic setting, we cannot be sure that this experience was representative of participants' day-to-day experiences (e.g., working with their own datasets). Ideally, we would have run a longitudinal field study with multiple types of participants to enable us to observe participants' tool use over time as they gain expertise in using it. However, this would have required additional resources (e.g., to support multiple types of data) and was therefore infeasible. Instead, our user study serves as an initial evaluation of our interpetability method.

Discussion

In this paper, we take inspiration from the study of human explanation, drawing on the literature on human explanation in philosophy, cognitive science, and the social sciences to propose a list of design principles for machine-generated explanations that are meaningful to humans. We develop a method for generating explanations that adhere to these principles using the concept of weight of evidence from information theory. We show that this method can be adapted to meet the needs of modern ML — that is, high-dimensional, multiclass settings — and that the explanations can be estimated accurately from finite samples, are robust to small perturbations of the inputs, and are usable by ML practitioners.

This paper opens several avenues for future work. Adapting modern density estimation methods for complex types of data, such as images, might hold the key to wider applicability of our method. Regarding evaluation, an immediate next step would be to carry out a follow-up user study to investigate various design choices, such as the technique for partitioning the classes in multi-class settings. Ideally, future user studies should involve participants' own models and should rely on questions that attempt to uncover insights that are relevant to their day-to-day experiences.

The findings from our user study offer important lessons that we believe are generally applicable to other interpretability tools. Chief among these is the importance of user-friendly and engaging tutorials that provide users with the necessary understanding of the tool and its intended usage, and users' desire for flexibility in tools. These results underscore the importance of putting human needs at the center of the design and evaluation of interpretability methods. The human computation community is uniquely situated to drive this work as it requires interdisciplinary expertise in both ML and HCI, fields that are central to HCOMP.

In the spirit of developing AI responsibly, we believe that papers proposing new interpretability methods should also provide a discussion, as we have done in the previous section, of not only those settings for which the proposed method is suitable, but also those settings that fall outside its scope. In addition, we recommend that authors should explicitly describe the notion of interpretability (or explanation) that they aim to operationalize, allowing readers to situate the contributions in relation to other interpretability methods and to understand their scope and applicability.

References

Abdul, A.; von der Weth, C.; Kankanhalli, M.; and Lim, B. Y. 2020. COGAM: Measuring and Moderating Cognitive Load in Machine Learning Model Explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery.

Alvarez-Melis, D.; and Jaakkola, T. S. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 412–421.

Alvarez-Melis, D.; and Jaakkola, T. S. 2018a. On the Robustness of Interpretability Methods. In *ICML Workshop on Human Interpretability in Machine Learning*.

Alvarez-Melis, D.; and Jaakkola, T. S. 2018b. Towards Robust Interpretability with Self-explaining Neural Networks. In *Neural Information Processing Systems*, 7775–7784. Curran Associates Inc.

Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Suh, J.; Iqbal, S.; Bennett, P. N.; Inkpen, K.; et al. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. Association for Computing Machinery.

Azevedo, P. J. 2010. Rules for contrast sets. *Intelligent Data Analysis* doi:10.3233/IDA-2010-0444.

Bar-Hillel, M. 1980. The base-rate fallacy in probability judgments. *Acta Psychol.* 44(3): 211–233.

Bay, S. D.; and Pazzani, M. J. 1999. Detecting change in categorical data: Mining contrast sets. In *Conference on Knowledge discovery and data mining (KDD)*. Association for Computing Machinery. doi:10.1145/312129.312263.

Braun, V.; and Clarke, V. 2012. Thematic analysis. In *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological.*, APA handbooks in psychology., 57–71. Washington, DC, US: American Psychological Association. doi:10.1037/13620-004.

Bunt, A.; Lount, M.; and Lauzon, C. 2012. Are Explanations Always Important? A Study of Deployed, Low-Cost Intelligent Interactive Systems. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces (IUI)*, 169–178. Association for Computing Machinery.

Bussone, A.; Stumpf, S.; and O'Sullivan, D. 2015. The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems. In *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI)*, 160–169. doi:10.1109/ICHI.2015.26.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible Models for HealthCare : Predicting Pneumonia Risk and Hospital 30-day Readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1721–1730. Association for Computing Machinery. doi: 10.1145/2783258.2788613.

Dodge, J.; Liao, Q.; Zhang, Y.; Bellamy, R. K. E.; and Dugan, C. 2019. Explaining Models: An Empirical Study of How Explanations Impact Fairness Judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI)*, 275–285. Association for Computing Machinery.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* .

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Eddy, D. M. 1982. Probabilistic Reasoning in Clinical Medicine: Problems and Opportunities. In Kahneman, D.; Slovic, P.; and Tversky, A., eds., *Judgment Under Uncertainty: Heuristics and Biases*, 249–267. Cambridge University Press.

Fernandes, K.; Vinagre, P.; and Cortez, P. 2015. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. In *Progress in Artificial Intelligence*, 535–546. Springer International Publishing. doi: 10.1007/978-3-319-23485-4_53.

Gibson, J. J. 1977. The theory of affordances. R. Shaw and J. Bransford (eds.), Perceiving, Acting and Knowing.

Gold, J. I.; and Shadlen, M. N. 2001. Neural computations that underlie decisions about sensory stimuli. *Trends Cogn. Sci.* 5(1): 10–16. ISSN 1364-6613, 1879-307X. doi: 10.1016/s1364-6613(00)01567-9.

Gold, J. I.; and Shadlen, M. N. 2002. Banburismus and the brain: decoding the relationship between sensory stimuli, decisions, and reward. *Neuron* 36(2): 299–308. doi: 10.1016/s0896-6273(02)00971-6.

Good, I. J. 1950. *Probability and the Weighing of Evidence*. Charles Griffin & Company Limited: London.

Good, I. J. 1968. Corroboration, Explanation, Evolving Probability, Simplicity and a Sharpened Razor. *Br. J. Philos. Sci.* 19(2): 123–143. doi:10.1093/bjps/19.2.123.

Good, I. J. 1985. Weight of evidence: A brief survey. *Bayesian statistics* 2: 249–270.

Goodman, N. D.; Baker, C. L.; Bonawitz, E. B.; Mansinghka, V. K.; Gopnik, A.; Wellman, H.; Schulz, L.; and Tenenbaum, J. B. 2006. Intuitive theories of mind: A rational approach to false belief. In *Proceedings of the twenty-eighth Annual Meeting of the Cognitive Science Society*, volume 6. Cognitive Science Society Vancouver.

Hastie, T. J.; and Tibshirani, R. J. 1990. *Generalized Additive Models*. CRC Press.

Hempel, C. G. 1962. *Deductive-nomological vs. statistical explanation*, volume 3 of *Minnesota Studies in the Philosophy of Science*. University of Minnesota Press, Minneapolis.

Hempel, C. G.; and Oppenheim, P. 1948. Studies in the Logic of Explanation. *Philos. Sci.* 15(2): 135–175. doi: 10.1086/286983.

Hohman, F.; Head, A.; Caruana, R.; DeLine, R.; and Drucker, S. M. 2019. Gamut: A Design Probe to Understand How Data Scientists Understand Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery.

Hong, S. R.; Hullman, J.; and Bertini, E. 2020. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. Association for Computing Machinery.

Hudson, S. E.; and Mankoff, J. 2014. Concepts, values, and methods for technical human–computer interaction research. In *Ways of Knowing in HCI*, 69–93. Springer.

Jacovi, A.; and Goldberg, Y. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4198–4205. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.386.

Jung, J.; Concannon, C.; Shroff, R.; Goel, S.; and Goldstein, D. G. 2020. Simple rules to guide expert classifications. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 183(3): 771–800. doi:https://doi.org/10.1111/rssa.12576.

Kaur, H.; Nori, H.; Jenkins, S.; Caruana, R.; Wallach, H.; and Vaughan, J. W. 2020. Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery.

Koehler, J. J. 1996. The base rate fallacy reconsidered: Descriptive, normative, and methodological challenges. *Behav. Brain Sci.* 19(1): 1–17.

Lage, I.; Chen, E.; He, J.; Narayanan, M.; Kim, B.; Gershman, S. J.; and Doshi-Velez, F. 2019. Human Evaluation of Models Built for Interpretability. In *Proceedings of the 7th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 59–67.

Lakkaraju, H.; Bach, S. H.; and Jure, L. 2016. Interpretable Decision Sets: A Joint Framework for Description and Prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery. doi:10.1145/2939672.2939874.

Li, Q.; Chu, S.; Rao, N.; and Nourani, M. 2020. Understanding the Effects of Explanation Types and User Motivations on Recommender System Use. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)* 8(1): 83–91.

Lim, B. Y.; and Dey, A. K. 2011. Investigating Intelligibility for Uncertain Context-Aware Applications. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp)*, 415–424.

Lipton, P. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplements* 27: 247–266.

Lombrozo, T. 2012. Explanation and Abductive Inference. In Keith J. Holyoak and Robert G. Morrison, ed., *The Oxford Handbook of Thinking and Reasoning*. Oxford University Press. ISBN 9780199968718. doi:10.1093/oxfordhb/9780199734689.013.0014.

Lundberg, S.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, 4768–4777.

Miller, T. 2019a. Contrastive Explanation: A Structural-Model Approach. *arXiv preprint arXiv:1811. 03163* .

Miller, T. 2019b. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* 267: 1–38. ISSN 0004-3702.

Norman, D. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.

Nourani, M.; Kabir, S.; Mohseni, S.; and Ragan, E. D. 2019. The Effects of Meaningful and Meaningless Explanations on Trust and Perceived System Accuracy in Intelligent Systems. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)* 7(1): 97–105.

Novak, P. K.; Lavrač, N.; and Webb, G. I. 2009. Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. *J. Mach. Learn. Res.* 10: 377–403. ISSN 1532-4435. doi:10.1145/1577069.1577083.

Olsen Jr, D. R. 2007. Evaluating user interface systems research. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, 251–258.

Papamakarios, G.; Pavlakou, T.; and Murray, I. 2017. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2335–2344. USA: Curran Associates Inc. ISBN 9781510860964.

Peirce, C.; and Jastrow, J. 1885. On small differences in sensation. *Memoirs of the National Academy of Science* 3: 73–83.

Peirce, C. S. 1878. Illustrations of the Logic of Science: IV The Probability of Induction. *Popular Science Monthly* 12: 705–718.

Phillips, L. D.; and Edwards, W. 1966. Conservatism in a simple probability inference task. *J. Exp. Psychol.* 72(3): 346–354. ISSN 0022-1015. doi:10.1037/h0023653.

Pitt, J. C. 1988. *Theories of explanation*. Oxford University Press. ISBN 9780195049701.

Poursabzi-Sangdeh, F.; Goldstein, D. G.; Hofman, J. M.; Vaughan, J. W.; and Wallach, H. 2021. Manipulating and Measuring Model Interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Quinlan, J. R. 1986. Induction of Decision Trees. *Mach. Learn.* ISSN 0885-6125, 1573-0565. doi:10.1023/A:1022643204877.

Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, 1530–1538.

Ribeiro, M. T.; Singh, S.; and 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 - KDD '15*, 1135–1144. New York, NY, USA: ACM. ISBN 9781450342322. doi:10.1145/2939672.2939778.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5): 206–215.

Salmon, W. C. 1971. *Statistical Explanation and Statistical Relevance*. University of Pittsburgh Press.

Selbst, A.; and Barocas, S. 2018. The Intuitive Appeal of Explainable Machines. *Fordham Law Review* 87.

Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2017. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. In *ICCV*.

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

Spiegelhalter, D. 2018. Making Algorithms Trustworthy: What Can Statistical Science Contribute to Transparency, Explanation and Validation? Neural Information Processing Systems, Invited Talk (Breiman Lecture).

Tan, S.; Caruana, R.; Hooker, G.; and Lou, Y. 2018. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. In *Proceedings of the AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)*.

Turner, C. W.; Lewis, J. R.; and Nielsen, J. 2006. Determining usability test sample size. *International encyclopedia of ergonomics and human factors* 3(2): 3084–3088.

Tversky, A.; and Kahneman, D. 1974. Judgment under uncertainty: Heuristics and biases. *Science* 185(4157): 1124–1131. ISSN 0036-8075.

van der Waa, J.; Robeer, M.; van Diggelen, J.; Brinkhuis, M.; and Neerincx, M. 2018. Contrastive explanations with local foil trees. In *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*.

van Fraassen, B. 1988. The Pragmatic Theory of Explanation. In Pitt, J. C., ed., *Theories of Explanation*. Oxford University Press.

Vaughan, J. W.; and Wallach, H. 2021. A Human-Centered Agenda for Intelligible Machine Learning. In Pelillo, M.; and Scantamburlo, T., eds., *Machines We Trust: Perspectives on Dependable AI*. MIT Press.

Veale, M.; Van Kleek, M.; and Binns, R. 2018. Fairness and Accountability Design Needs for Algorithmic Support in High-Stakes Public Sector Decision-Making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology* 31(2).

Webb, G. I.; Butler, S.; and Newlands, D. 2003. On detecting differences between groups. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 256–265. Association for Computing Machinery.

Weld, D. S.; and Bansal, G. 2019. The Challenge of Crafting Intelligible Intelligence. *Communications of the ACM* 62(6): 70–79.

Zeng, J.; Ustun, B.; and Rudin, C. 2017. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 180(3): 689–722.

Zhang, H.; and Maloney, L. T. 2012. Ubiquitous log odds: a common representation of probability and frequency distortion in perception, action, and cognition. *Front. Neurosci.* 6. ISSN 1662-4548. doi:10.3389/fnins.2012.00001.

Design Principles in Further Detail

We surveyed the literature on the nature of explanation from philosophy, cognitive science, and the social sciences, leading us to propose a list of design principles for machine-generated explanations that are meaningful to humans. In this section, we further discuss each of these principles.

D1. Explanations should be contrastive. A recurring theme across disciplines is the hypothesis that humans tend to explain in contrastive terms. We therefore argue that machine-generated explanations should also be contrastive. In other words, the *explanandum* should be the question of why a model $\mathcal{M}$ predicted a particular output y for some input x *instead of predicting* y'. However, most current interpretability methods do not explicitly refer to an alternative y', implicitly assuming it to be the complement of y.

D2. Explanations should be exhaustive. The explanations produced by most current interpretability methods refer only to why the input x points to a single hypothesis (i.e., the prediction y) rather than ruling out all alternatives. However, human explanations tend to provide a justification for why every alternative y' was not predicted. This is important for (at least) two reasons: First, it is necessary for generating explanations that are counterfactual. Second, it entails explanations that are, to use Hempel's terminology, *complete*.[5] For example, an explanation for a diagnosis of pneumonia based solely on the presence of a cough is not exhaustive because a cough is a symptom of other diseases too.

D3. Explanations should be modular and compositional. Explanations are most needed for the predictions of complex black-box models like neural networks. Yet, when used in these settings, most current interpretability methods generate a single, high-dimensional static explanation for each prediction (e.g., a heatmap for a prediction made by an image classifier). Explanations like these can be difficult to understand and draw conclusions from, particularly for non-expert users. In addition, this approach differs from human explanations, which tend to decompose into simple components (Hempel 1962). We therefore argue that machine-generated explanations should also explain using multiple simple accumulative statements, each addressing a few aspects of the evidence. We note, however, that this type of modularity introduces an inherent trade-off between the number of components in an explanation (too many components might be difficult to understand simultaneously) and their relative complexity (simpler components are easier to draw conclusions from, but using simpler components may mean that more of them are needed). Breaking up predictions into simple components also helps shift interpretability methods from treating explanations as a product towards explanations as a process (Lombrozo 2012; Miller 2019b).

D4. Explanations should rely on easily-understandable quantities. Explanations are only as useful as the information that they provide to users. We therefore argue that explanations should rely on quantities that can be easily understood, thereby making it less likely that users will misunderstand them, potentially leading to misuse (Kaur et al. 2020).

D5. Explanations should be parsimonious. As noted by Miller (2019b), one of the most salient recurring themes in the literature on human explanation is minimality — i.e., only the most relevant facts should be included in explanations. We therefore argue that explanations should adhere to Occam's Razor, which states that the simpler of two equally good explanations should be preferred. Furthermore, if omitting less-relevant facts makes a machine-generated explanation easier to understand while remaining equally faithful to the underlying model, then they should be omitted.

Axiomatic Derivation of the WoE

Interpretability is fundamentally a human-centered concept. As such, human needs should therefore be at the center of both the design and evaluation of interpretability methods. We achieve this by first proposing a list of design principles that we argue human-centered machine-generated explanations should satisfy and then realizing these design principles starting from the concept of weight of evidence (WoE).

Good (1985) provides an axiomatic derivation for the WoE of e in favor of h, as defined in Equation (1). This derivation shows that the WoE is (up to a constant) the only function F of e and h that satisfies the following properties:

1. $F(e, h)$ depends only on $P(e \mid h)$ and $P(e \mid \overline{h})$.
2. $P(h \mid e)$ is a function of only $P(h)$ and $F(e, h)$.
3. When e is decomposable, $F(e, h)$ is additive in terms of e, e.g., $F(e_1 \wedge e_2, h) = F(e_1, h) + F(e_2, h)$.

We note that all three of these properties are fairly general. For example, the second property is a reasonable requirement for any probabilistic model of feature importance. Meanwhile, the first property is a sufficient (although not a necessary) condition for obtaining contrastive explanations.

Well-Known Model Families

In this section, we show that for well-known families of simple classifiers, the WoE has a closed-form expression. Consider a binary classification setting — i.e., $y \in \{0, 1\}$ — with n input features — i.e., $x = (x_1, \ldots, x_n)$ — and, for conciseness, let $p_1 = P(Y = 1)$ and $p_0 = P(Y = 0)$.

Logistic Regression

For logistic regression, the predictive posterior is

$$P(Y = 1 \mid X) = \sigma(w^\top x + w_0) \quad \in [0, 1],$$

where $\sigma(t) = \frac{e^t}{e^t+1} = \frac{1}{1+e^{-t}}$ is the sigmoid function. Logistic regression already has a natural log-odds interpretation:

$$\log \frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)} = \frac{\sigma(w^\top x + w_0)}{1 - \sigma(w^\top x + w_0)}$$
$$= \log e^{w^\top x + w_0} = w^\top x + w_0.$$

[5] Not to be confused with *complete* according to the terminology of Goodman et al. (2006), where all variables are explained.

From this, the WoE is easy to compute and unsurprising:

$$\begin{aligned}\mathrm{woe}(Y=1:X=x) &= \log \tfrac{P(Y=1|X)}{P(Y=0|X)} / \log \tfrac{P(Y=1)}{P(Y=0)} \\ &= w^\top x + w_0 - \log \tfrac{p_1}{p_0}.\end{aligned}$$

Therefore, the WoE for logistic regression can be interpreted as removing the effect of the base rate (via the log-odds ratio $\log \frac{p_1}{p_0}$) from the linear attribution model (i.e., $w^\top x + w_0$).

Unfortunately, obtaining closed-form expressions for per-feature WoE values is not as simple. Because logistic regression is not a generative model, it does not explicitly model the conditional probabilities needed to obtain these values. For example, neither $P(Y=1 \mid X_i)$ nor $P(X_i \mid Y=1)$ can be unambiguously obtained from $P(Y=1 \mid X)$. To determine these conditional probabilities, assumptions must be made. For example, the simplest coherent identification of the coefficients with per-feature log-likelihood ratios is:

$$\begin{aligned}w_0 &= \log \frac{p_1}{p_0} \\ w_i x_i + w_0 &= \log \frac{P(Y=1 \mid x_i)}{P(Y=0 \mid x_i)}.\end{aligned}$$

However, this identification makes the assumption that the coefficients do not model feature interactions. Under this identification, the WoE value for X_i is then as follows:

$$\mathrm{woe}(Y=1:X_i) = w_i x_i.$$

This is a natural notion of feature importance for log-linear models under a conditional independence assumption. The posterior log odds of $Y=1$ versus $Y=0$ is then given by:

$$\begin{aligned}&\log \tfrac{P(Y=1|X)}{P(Y=0|X)} \\ &\quad = \sum_{i=1}^{n} \mathrm{woe}(Y{=}1{:}x_i) + \log \frac{p_1}{p_0} = w^\top x + w_0.\end{aligned}$$

Other interpretability methods can be shown to recover logistic regression's coefficients too. For example, it is easy to show that without penalization, with a logistic link function, and asymptotically in the number of fitting samples, LIME (Ribeiro, Singh, and Guestrin 2016) would find a surrogate model that coincides with the true log-linear model, so the resulting explanations would coincide with the WoE values.

Naïve Bayes

The equivalence between the WoE and naïve Bayes (NB) is well known, but we revisit it here for completeness. An NB classifier already assumes conditional independence:

$$P(X_i \mid Y) \perp P(X_j \mid Y) \quad \forall i \neq j.$$

The conditional probabilities are modeled as

$$p(Y=1|X) = P(Y=1) \prod_{i=1}^{n} P(X_i \mid Y=1),$$

where $P(X_i \mid Y)$ is estimated during training. Therefore, for a NB classifier, the total WoE is defined as follows:

$$\begin{aligned}\mathrm{woe}(Y=1:X) &= \log \frac{P(X \mid Y=1)}{P(X \mid Y=0)} \\ &= \log \frac{\prod_{i=1}^{n} P(X_i \mid Y=1)}{\prod_{i=1}^{n} P(X_i \mid Y=1)} \\ &= \sum_{i=1}^{n} \log \tfrac{P(X_i|Y=1)}{P(X_i|Y=0)}.\end{aligned}$$

In addition,

$$\begin{aligned}\mathrm{woe}(Y=1:X_i) &= \mathrm{woe}(Y=1:X_i|X_0,\ldots,X_{i-1}) \\ &= \log \tfrac{P(X_i|Y=1)}{P(X_i|Y=0)}.\end{aligned}$$

A Gaussian NB (GNB) classifier would parametrize the conditional likelihoods as n-dimensional Gaussians:

$$\begin{aligned}&p(X \mid Y=k) = \\ &\quad \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} \exp\left\{-\frac{1}{2}(x-\mu_k)^\top \Sigma_i^{-1}(x-\mu_k),\right\}\end{aligned}$$

where the covariance matrix Σ_k is diagonal in order to satisfy the NB assumption. This then implies the following:

$$\begin{aligned}&p(X_i \mid Y{=}k) \\ &\quad = \frac{1}{\sqrt{(2\pi)^n [\Sigma_k]_{ii}}} \exp\left\{-\frac{1}{2[\Sigma_k]_{ii}} |x_i - \mu_{ki}|^2\right\}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathrm{woe}(Y=1:X_i) \\ &\quad = \log \left(\frac{[\Sigma_0]_{ii}}{[\Sigma_1]_{ii}}\right)^{1/2} \exp\left\{-\frac{1}{2}\left(\tfrac{1}{[\Sigma_1]_{ii}}|x_i-\mu_{1,i}|^2 \right.\right. \\ &\qquad \left.\left. - \tfrac{1}{[\Sigma_0]_{ii}}|x_i-\mu_{0,i}|^2\right)\right\} \\ &\quad = -\frac{1}{2}\left(\tfrac{1}{[\Sigma_1]_{ii}}|x_i{-}\mu_{1,i}|^2 - \tfrac{1}{[\Sigma_0]_{ii}}|x_i{-}\mu_{0,i}|^2\right) \\ &\qquad - \frac{1}{2}\log\frac{[\Sigma_1]_{ii}}{[\Sigma_0]_{ii}}\end{aligned}$$

and

$$\begin{aligned}&\mathrm{woe}(Y=1:X) \\ &\quad = -\frac{1}{2}\Bigg(x^\top(\Sigma_1^{-1} - \Sigma_0^{-1})x - 2(\mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1})x \\ &\qquad + (\mu_1^\top \Sigma_1^{-1}\mu_1 - \mu_0^\top \Sigma_0^{-1}\mu_0)\Bigg) - \frac{1}{2}\log\frac{|\Sigma_1|}{|\Sigma_0|}.\end{aligned}$$

This implies that $\mathrm{woe}(Y=1:X_i) > 0$ if and only if:

$$\frac{1}{\Sigma_{1,ii}}|x_i - \mu_{1,i}|^2 - \log \Sigma_{1,ii} < \frac{1}{\Sigma_{0,ii}}|x_i - \mu_{0,i}|^2 - \log \Sigma_{0,ii}.$$

In other words, $\mathrm{woe}(Y=1:X_i) > 0$ if and only if x_i is closer (according to a variance-normalized distance) to $\mu_{1,i}$ than to $\mu_{0,i}$. Similarly, the total WoE satisfies the following:

$$\begin{aligned}&\mathrm{woe}(Y{=}1\,|\,X) > 0 \iff \\ &\qquad d^2_{\Sigma_1}(x,\mu_1) - d^2_{\Sigma_0}(x,\mu_0) < \log \tfrac{|\Sigma_1|}{|\Sigma_0|},\end{aligned}$$

where d_Σ is the Mahalanobis distance with covariance Σ.

LDA and QDA

Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are similar to Gaussian naïve Bayes, except that they do not make a conditional independence assumption — i.e., the covariance matrix need not be diagonal. Although non-conditional per-feature WoE values can still be computed easily (because marginalizing a multivariate Gaussian boils down to taking subsets of the mean and covariance matrix), the conditional per-feature WoE values do not have as simple an expression. In the general case, where the variables are not independent, if $X \mid Y \sim \mathcal{N}(\mu, \Sigma)$, then $X_i \mid X_1, \dots, X_{i-1}, Y \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma})$, where

$$\begin{aligned} \tilde{\mu} &= \mu_i + \Sigma_{[i,1:i-1]} \Sigma^{-1}_{[1:i-1,1:i-1]} (x_{[1:i-1]} - \mu_{[1:i-1]}) \\ \tilde{\sigma} &= \Sigma_{ii} - \Sigma_{[i,1:i-1]} \Sigma^{-1}_{[1:i-1,1:i-1]} \Sigma_{[1:i-1,i]}. \end{aligned}$$

Setting off-diagonal elements to be 0, we recover

$$P(X_i \mid X_{i-1}, \dots, X_1, Y) \sim \mathcal{N}(\mu_i, \Sigma_{ii}).$$

We can use the values of $\tilde{\mu}$ and $\tilde{\sigma}$ above to derive closed-form (albeit cumbersome) expressions for woe$(Y = 1 : X_i \mid X_{i-1}, \dots, X_1)$ in terms of log-density ratios of normal distributions. The total WoE is analogous to the total WoE for GNB. For example, LDA makes a homoscedasticity assumption ($\Sigma_1 = \Sigma_0 = \Sigma$), so the total WoE satisfies

$$\text{woe}(Y = 1 \mid X) > 0 \iff d^2_\Sigma(x, \mu_1) - d^2_\Sigma(x, \mu_0) < 0.$$

Use Cases and Limitations

In this section, we discuss how best to choose between our method and other methods, such as LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017).

No Explanation is Inherently "Better"

In the "Design Principles" section, we proposed a list of design principles for machine-generated explanations, distilled from the literature on the nature of explanation from philosophy, cognitive science, and the social sciences. However, distilling universal principles for what makes one explanation better than another is much more difficult. Indeed, the answer to this question is highly context dependent. This implies that explanations generated by different interpretability methods might be useful in different situations, depending on the type of model, the data, the user, and the intended use of the explanations (Vaughan and Wallach 2021).

Further complicating matters is the fact that post-hoc interpretability methods are, by definition, imperfect. Summarizing the behavior of complex models comes at a price (Rudin 2019) — that is, the explanations are partial, only hold in a small neighborhood (Ribeiro, Singh, and Guestrin 2016), or make strong assumptions about the data (Lundberg and Lee 2017). As a result, explanations generated by one interpretability method seldom strictly dominate explanations generated by another: there are always scenarios in which one method's assumptions and approximations will be more appropriate than those of another. Furthermore, different explanations might reveal information about different aspects of the underlying model's behavior.

Based on this premise, we next discuss when WoE explanations are likely to be useful — as complements to or substitutes for other interpretability methods — and when they are not. Our goal is to provide guidelines for deciding when to use our method and when to use other methods. We focus on LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017) because they are very widely used.

When Are WoE Explanations Useful?

We first discuss three main settings in which there is a clear case for using explanations based on the weight of evidence.

The underlying model is generative. Explaining the predictions made by a generative model in generative terms (see the "Relation to LIME and SHAP" section) seems, conceptually, a better fit than doing so in discriminative terms. But there are other tangible reasons to use WoE explanations for generative models too: First, generative models already compute conditional likelihoods internally. Even when these conditional likelihoods are not accessible (e.g., as is the case with black-box models), it is reasonable to think that they will be stable, as will be their empirical estimates, even for small sample sizes. Indeed, we showed in the "Quantitative Experiments" section that when the underlying model is generative, WoE values can be very accurately estimated.

The underlying model is log linear. If the underlying model is log linear or yields simple log-odds ratios, then the WoE might have a closed-form, or at least simple, expression. For example, for logistic regression, the log odds are linear, as we explained above. This differs from post-hoc interpretability methods like LIME and SHAP, which need to be modified to accommodate non-linear link functions.

The underlying model is a multi-class classifier. In their original forms, both LIME and SHAP produce one-shot explanations in for the predicted class. When the number of classes is large, this is unlikely to yield useful explanations. The WoE, on the other hand, naturally extends to settings where the hypotheses h and h' to contrast correspond to sets of classes, as discussed in the main text. Indeed, in our user study, participants found that sequential explanations helped them to understand how each of the classes were ruled out.

Even when the underlying model is not generative or log linear, it might still be desirable to explain its predictions in generative terms, perhaps as a complement to other post-hoc interpretability methods. However, WoE explanations should only be used if the following two conditions hold:

Conditional likelihoods are accessible or can be accurately estimated from finite samples. The meta-algorithm for generating WoE explanations requires access to the conditional likelihoods $P(X_{\mathcal{A}_i} | X_{\mathcal{A}_{i-1}}, \dots, X_{\mathcal{A}_1}, Y)$. If the underlying model computes these conditional likelihoods internally, then they can be used directly; however, in other cases, they must be estimated from samples. Our quantitative experiments suggest that the WoE can be accurately estimated — even in relatively high dimensions — from finite samples.

Log-odds ratios are easily understandable by the stakeholders who will use the explanations. Following Principle 4, explanations should rely on easily understandable quantities. As we discussed in the "Weight of Evidence" section, there

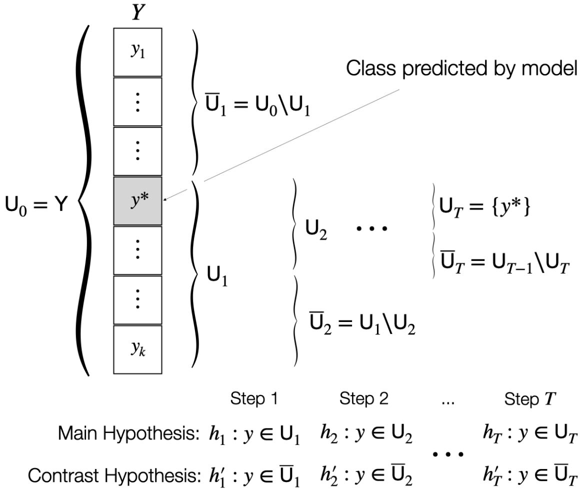

Y
y_1
y^*
y_k
Class predicted by model
$\mathsf{U}_0 = \mathsf{Y}$
$\overline{\mathsf{U}}_1 = \mathsf{U}_0 \backslash \mathsf{U}_1$
U_1
U_2
$\overline{\mathsf{U}}_2 = \mathsf{U}_1 \backslash \mathsf{U}_2$
$\mathsf{U}_T = \{y^*\}$
$\overline{\mathsf{U}}_T = \mathsf{U}_{T-1} \backslash \mathsf{U}_T$
Step 1
Step 2
...
Step T
Main Hypothesis: $h_1 : y \in \mathsf{U}_1$
$h_2 : y \in \mathsf{U}_2$
$h_T : y \in \mathsf{U}_T$
Contrast Hypothesis: $h_1' : y \in \overline{\mathsf{U}}_1$
$h_2' : y \in \overline{\mathsf{U}}_2$
$h_T' : y \in \overline{\mathsf{U}}_T$

Figure 4: Our method breaks up up explanations in multi-class settings into sequences of “smaller” explanations.

is ample evidence to suggest that the units in which the WoE is expressed (log-odds ratios) are easily understandable. However, we recommend experimentally verifying that this is indeed the case in the context in which WoE explanations will be used, for the stakeholders who will use them.

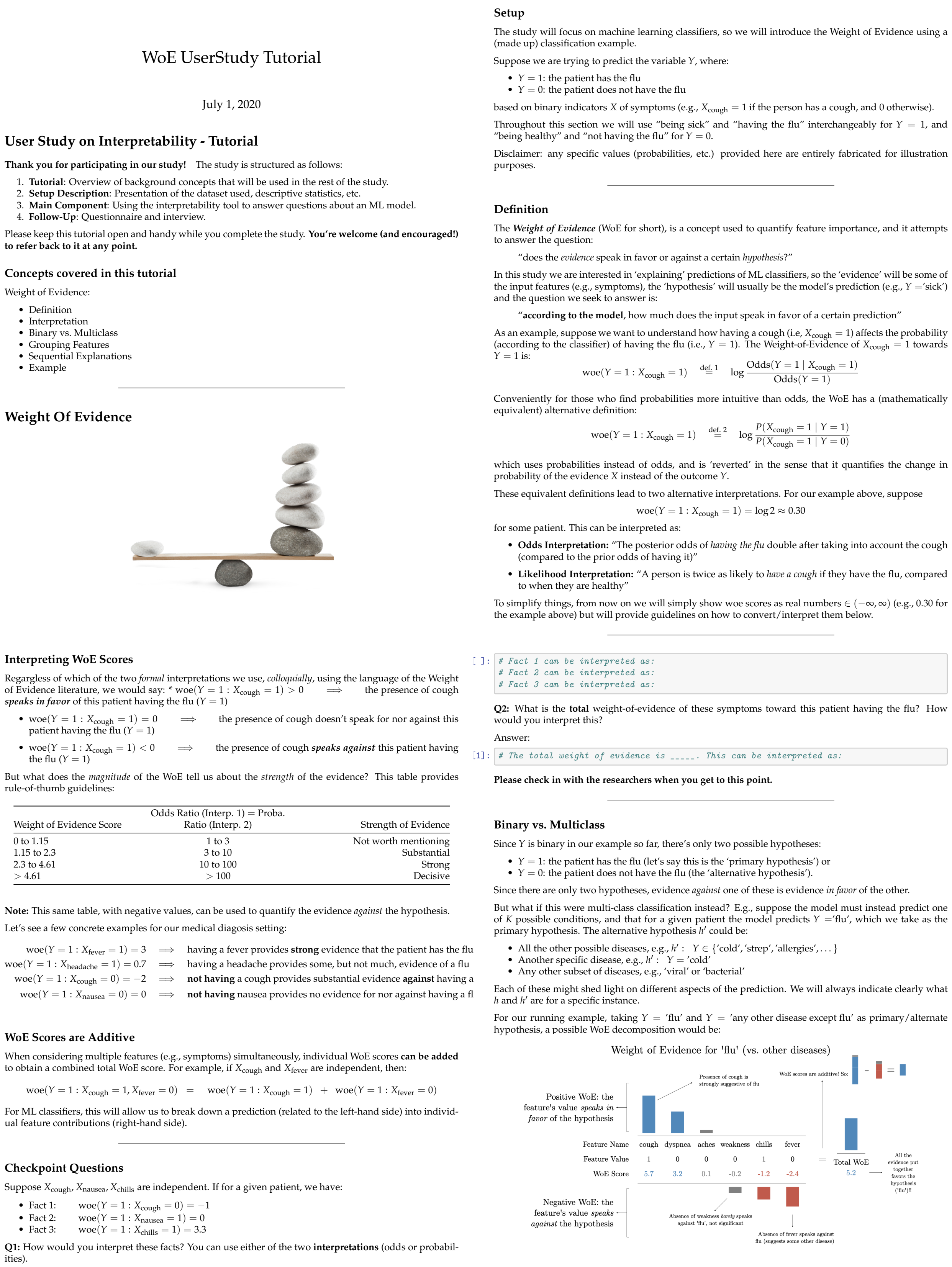
Sequential Explanations

Figure 4 provides a schematic illustration of the procedure for generating sequential explanations that our tool uses in settings where the number of classes is very large. At each step, a subset of the possible classes U_i is ruled out. By design, the subset that is retained contains the predicted class y^*. In other words, with each subsequent step, a smaller set of classes is retained until h consists of only y^*. At the end of this sequential procedure, every alternative y' will have been ruled out (c.f. Principle 2).

Jupyter Notebooks

In the rest of the appendix, we provide excerpts from the Jupyter notebooks that we used in the tutorial and in the main study. The complete notebooks are available online.[6]

[6] http://github.com/dmelis/interpretwoe



WoE UserStudy Tutorial

July 1, 2020

User Study on Interpretability - Tutorial

Thank you for participating in our study! The study is structured as follows:

1. **Tutorial**: Overview of background concepts that will be used in the rest of the study.
2. **Setup Description**: Presentation of the dataset used, descriptive statistics, etc.
3. **Main Component**: Using the interpretability tool to answer questions about an ML model.
4. **Follow-Up**: Questionnaire and interview.

Please keep this tutorial open and handy while you complete the study. **You're welcome (and encouraged!) to refer back to it at any point.**

Concepts covered in this tutorial

Weight of Evidence:

- Definition
- Interpretation
- Binary vs. Multiclass
- Grouping Features
- Sequential Explanations
- Example

Weight Of Evidence

Interpreting WoE Scores

Regargless of which of the two *formal* interpretations we use, *colloquially*, using the language of the Weight of Evidence literature, we would say: * $\text{woe}(Y = 1 : X_{\text{cough}} = 1) > 0 \implies$ the presence of cough ***speaks in favor*** of this patient having the flu ($Y = 1$)

- $\text{woe}(Y = 1 : X_{\text{cough}} = 1) = 0 \implies$ the presence of cough doesn't speak for nor against this patient having the flu ($Y = 1$)
- $\text{woe}(Y = 1 : X_{\text{cough}} = 1) < 0 \implies$ the presence of cough ***speaks against*** this patient having the flu ($Y = 1$)

But what does the *magnitude* of the WoE tell us about the *strength* of the evidence? This table provides rule-of-thumb guidelines:

Weight of Evidence Score	Odds Ratio (Interp. 1) = Proba. Ratio (Interp. 2)	Strength of Evidence
0 to 1.15	1 to 3	Not worth mentioning
1.15 to 2.3	3 to 10	Substantial
2.3 to 4.61	10 to 100	Strong
> 4.61	> 100	Decisive

Note: This same table, with negative values, can be used to quantify the evidence *against* the hypothesis.

Let's see a few concrete examples for our medical diagosis setting:

$\text{woe}(Y = 1 : X_{\text{fever}} = 1) = 3 \implies$ having a fever provides **strong** evidence that the patient has the flu

$\text{woe}(Y = 1 : X_{\text{headache}} = 1) = 0.7 \implies$ having a headache provides some, but not much, evidence of a flu

$\text{woe}(Y = 1 : X_{\text{cough}} = 0) = -2 \implies$ **not having** a cough provides substantial evidence **against** having a

$\text{woe}(Y = 1 : X_{\text{nausea}} = 0) = 0 \implies$ **not having** nausea provides no evidence for nor against having a fl

WoE Scores are Additive

When considering multiple features (e.g., symptoms) simultaneously, individual WoE scores **can be added** to obtain a combined total WoE score. For example, if X_{cough} and X_{fever} are independent, then:

$$\text{woe}(Y = 1 : X_{\text{cough}} = 1, X_{\text{fever}} = 0) \quad = \quad \text{woe}(Y = 1 : X_{\text{cough}} = 1) \;+\; \text{woe}(Y = 1 : X_{\text{fever}} = 0)$$

For ML classifiers, this will allow us to break down a prediction (related to the left-hand side) into individual feature contributions (right-hand side).

Checkpoint Questions

Suppose $X_{\text{cough}}, X_{\text{nausea}}, X_{\text{chills}}$ are independent. If for a given patient, we have:

- Fact 1: $\text{woe}(Y = 1 : X_{\text{cough}} = 0) = -1$
- Fact 2: $\text{woe}(Y = 1 : X_{\text{nausea}} = 1) = 0$
- Fact 3: $\text{woe}(Y = 1 : X_{\text{chills}} = 1) = 3.3$

Q1: How would you interpret these facts? You can use either of the two **interpretations** (odds or probabilities).

Setup

The study will focus on machine learning classifiers, so we will introduce the Weight of Evidence using a (made up) classification example.

Suppose we are trying to predict the variable Y, where:

- $Y = 1$: the patient has the flu
- $Y = 0$: the patient does not have the flu

based on binary indicators X of symptoms (e.g., $X_{\text{cough}} = 1$ if the person has a cough, and 0 otherwise).

Throughout this section we will use "being sick" and "having the flu" interchangeably for $Y = 1$, and "being healthy" and "not having the flu" for $Y = 0$.

Disclaimer: any specific values (probabilities, etc.) provided here are entirely fabricated for illustration purposes.

Definition

The ***Weight of Evidence*** (WoE for short), is a concept used to quantify feature importance, and it attempts to answer the question:

> "does the *evidence* speak in favor or against a certain *hypothesis*?"

In this study we are interested in 'explaining' predictions of ML classifiers, so the 'evidence' will be some of the input features (e.g., symptoms), the 'hypothesis' will usually be the model's prediction (e.g., $Y =$'sick') and the question we seek to answer is:

> "**according to the model**, how much does the input speak in favor of a certain prediction"

As an example, suppose we want to understand how having a cough (i.e, $X_{\text{cough}} = 1$) affects the probability (according to the classifier) of having the flu (i.e., $Y = 1$). The Weight-of-Evidence of $X_{\text{cough}} = 1$ towards $Y = 1$ is:

$$\text{woe}(Y = 1 : X_{\text{cough}} = 1) \overset{\text{def. 1}}{=} \log \frac{\text{Odds}(Y = 1 \mid X_{\text{cough}} = 1)}{\text{Odds}(Y = 1)}$$

Conveniently for those who find probabilities more intuitive than odds, the WoE has a (mathematically equivalent) alternative definition:

$$\text{woe}(Y = 1 : X_{\text{cough}} = 1) \overset{\text{def. 2}}{=} \log \frac{P(X_{\text{cough}} = 1 \mid Y = 1)}{P(X_{\text{cough}} = 1 \mid Y = 0)}$$

which uses probabilities instead of odds, and is 'reverted' in the sense that it quantifies the change in probability of the evidence X instead of the outcome Y.

These equivalent definitions lead to two alternative interpretations. For our example above, suppose

$$\text{woe}(Y = 1 : X_{\text{cough}} = 1) = \log 2 \approx 0.30$$

for some patient. This can be interpreted as:

- **Odds Interpretation:** "The posterior odds of *having the flu* double after taking into account the cough (compared to the prior odds of having it)"
- **Likelihood Interpretation:** "A person is twice as likely to *have a cough* if they have the flu, compared to when they are healthy"

To simplify things, from now on we will simply show woe scores as real numbers $\in (-\infty, \infty)$ (e.g., 0.30 for the example above) but will provide guidelines on how to convert/interpret them below.

```
[ ]: # Fact 1 can be interpreted as:
     # Fact 2 can be interpreted as:
     # Fact 3 can be interpreted as:
```

Q2: What is the **total** weight-of-evidence of these symptoms toward this patient having the flu? How would you interpret this?

Answer:

```
[1]: # The total weight of evidence is _____. This can be interpreted as:
```

Please check in with the researchers when you get to this point.

Binary vs. Multiclass

Since Y is binary in our example so far, there's only two possible hypotheses:

- $Y = 1$: the patient has the flu (let's say this is the 'primary hypothesis') or
- $Y = 0$: the patient does not have the flu (the 'alternative hypothesis').

Since there are only two hypotheses, evidence *against* one of these is evidence *in favor* of the other.

But what if this were multi-class classification instead? E.g., suppose the model must instead predict one of K possible conditions, and that for a given patient the model predicts $Y =$'flu', which we take as the primary hypothesis. The alternative hypothesis h' could be:

- All the other possible diseases, e.g., $h' : \; Y \in \{\text{'cold'}, \text{'strep'}, \text{'allergies'}, \ldots\}$
- Another specific disease, e.g., $h' : \; Y = \text{'cold'}$
- Any other subset of diseases, e.g., 'viral' or 'bacterial'

Each of these might shed light on different aspects of the prediction. We will always indicate clearly what h and h' are for a specific instance.

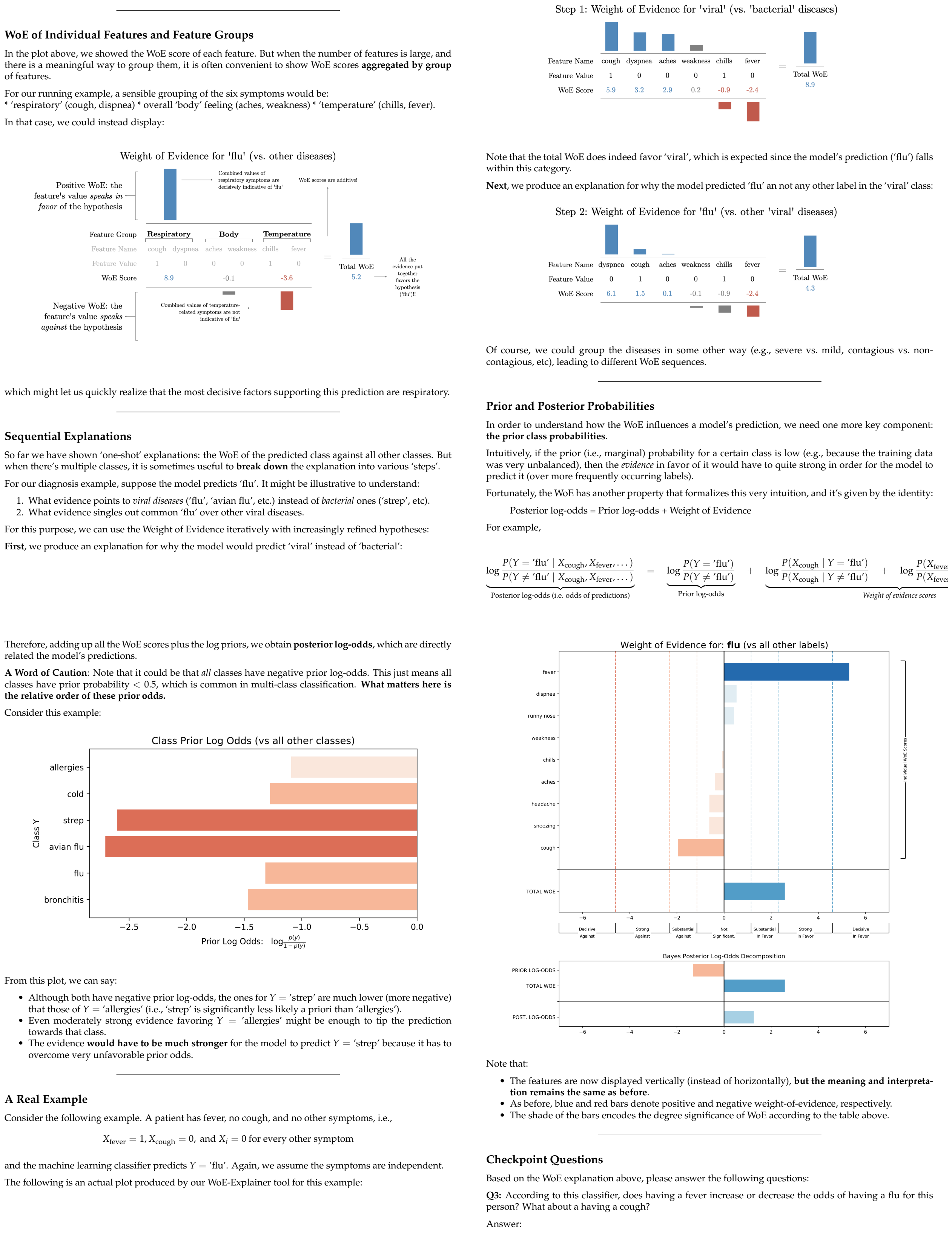
For our running example, taking Y = 'flu' and Y = 'any other disease except flu' as primary/alternate hypothesis, a possible WoE decomposition would be:

Weight of Evidence for 'flu' (vs. other diseases)

Feature Name	cough	dyspnea	aches	weakness	chills	fever
Feature Value	1	0	0	0	1	0
WoE Score	5.7	3.2	0.1	-0.2	-1.2	-2.4

Positive WoE: the feature's value *speaks in favor* of the hypothesis
Presence of cough is strongly suggestive of flu
WoE scores are additive! So:
-
=
=
Total WoE
5.2
All the evidence put together favors the hypothesis ('flu')!!
Negative WoE: the feature's value *speaks against* the hypothesis
Absence of weakness *barely* speaks against 'flu', not significant
Absence of fever speaks against flu (suggests some other disease)

Figure 5: Jupyter notebook for the tutorial.



WoE of Individual Features and Feature Groups

In the plot above, we showed the WoE score of each feature. But when the number of features is large, and there is a meaningful way to group them, it is often convenient to show WoE scores **aggregated by group** of features.

For our running example, a sensible grouping of the six symptoms would be:
* 'respiratory' (cough, dispnea) * overall 'body' feeling (aches, weakness) * 'temperature' (chills, fever).

In that case, we could instead display:

Weight of Evidence for 'flu' (vs. other diseases)

Feature Group	Respiratory		Body		Temperature	
Feature Name	cough	dyspnea	aches	weakness	chills	fever
Feature Value	1	0	0	0	1	0
WoE Score	8.9		-0.1		-3.6	

Positive WoE: the feature's value *speaks in favor* of the hypothesis
Combined values of respiratory symptoms are decisively indicative of 'flu'
WoE scores are additive!
=
Total WoE
5.2
All the evidence put together favors the hypothesis ('flu')!!
Negative WoE: the feature's value *speaks against* the hypothesis
Combined values of temperature-related symptoms are not indicative of 'flu'

which might let us quickly realize that the most decisive factors supporting this prediction are respiratory.

Sequential Explanations

So far we have shown 'one-shot' explanations: the WoE of the predicted class against all other classes. But when there's multiple classes, it is sometimes useful to **break down** the explanation into various 'steps'.

For our diagnosis example, suppose the model predicts 'flu'. It might be illustrative to understand:

1. What evidence points to *viral diseases* ('flu', 'avian flu', etc.) instead of *bacterial* ones ('strep', etc).
2. What evidence singles out common 'flu' over other viral diseases.

For this purpose, we can use the Weight of Evidence iteratively with increasingly refined hypotheses:

First, we produce an explanation for why the model would predict 'viral' instead of 'bacterial':

Step 1: Weight of Evidence for 'viral' (vs. 'bacterial' diseases)

Feature Name	cough	dyspnea	aches	weakness	chills	fever
Feature Value	1	0	0	0	1	0
WoE Score	5.9	3.2	2.9	0.2	-0.9	-2.4

Note that the total WoE does indeed favor 'viral', which is expected since the model's prediction ('flu') falls within this category.

Next, we produce an explanation for why the model predicted 'flu' an not any other label in the 'viral' class:

Step 2: Weight of Evidence for 'flu' (vs. other 'viral' diseases)

Feature Name	dyspnea	cough	aches	weakness	chills	fever
Feature Value	0	1	0	0	1	0
WoE Score	6.1	1.5	0.1	-0.1	-0.9	-2.4

Of course, we could group the diseases in some other way (e.g., severe vs. mild, contagious vs. non-contagious, etc), leading to different WoE sequences.

Prior and Posterior Probabilities

In order to understand how the WoE influences a model's prediction, we need one more key component: **the prior class probabilities**.

Intuitively, if the prior (i.e., marginal) probability for a certain class is low (e.g., because the training data was very unbalanced), then the *evidence* in favor of it would have to quite strong in order for the model to predict it (over more frequently occurring labels).

Fortunately, the WoE has another property that formalizes this very intuition, and it's given by the identity:

Posterior log-odds = Prior log-odds + Weight of Evidence

For example,

$$\underbrace{\log \frac{P(Y = \text{'flu'} \mid X_{\text{cough}}, X_{\text{fever}}, \ldots)}{P(Y \neq \text{'flu'} \mid X_{\text{cough}}, X_{\text{fever}}, \ldots)}}_{\text{Posterior log-odds (i.e. odds of predictions)}} = \underbrace{\log \frac{P(Y = \text{'flu'})}{P(Y \neq \text{'flu'})}}_{\text{Prior log-odds}} + \underbrace{\log \frac{P(X_{\text{cough}} \mid Y = \text{'flu'})}{P(X_{\text{cough}} \mid Y \neq \text{'flu'})} + \log \frac{P(X_{\text{feve}}}{P(X_{\text{feve}}}}_{\textit{Weight of evidence scores}}$$

Therefore, adding up all the WoE scores plus the log priors, we obtain **posterior log-odds**, which are directly related the model's predictions.

A Word of Caution: Note that it could be that *all* classes have negative prior log-odds. This just means all classes have prior probability < 0.5, which is common in multi-class classification. **What matters here is the relative order of these prior odds.**

Consider this example:

Class Prior Log Odds (vs all other classes)
Class Y
allergies
cold
strep
avian flu
flu
bronchitis
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
Prior Log Odds: $\log \frac{p(y)}{1-p(y)}$

From this plot, we can say:

- Although both have negative prior log-odds, the ones for $Y =$ 'strep' are much lower (more negative) that those of $Y =$ 'allergies' (i.e., 'strep' is significantly less likely a priori than 'allergies').
- Even moderately strong evidence favoring $Y =$ 'allergies' might be enough to tip the prediction towards that class.
- The evidence **would have to be much stronger** for the model to predict $Y =$ 'strep' because it has to overcome very unfavorable prior odds.

A Real Example

Consider the following example. A patient has fever, no cough, and no other symptoms, i.e.,

$$X_{\text{fever}} = 1, X_{\text{cough}} = 0, \text{ and } X_i = 0 \text{ for every other symptom}$$

and the machine learning classifier predicts $Y =$ 'flu'. Again, we assume the symptoms are independent.

The following is an actual plot produced by our WoE-Explainer tool for this example:

Note that:

- The features are now displayed vertically (instead of horizontally), **but the meaning and interpretation remains the same as before**.
- As before, blue and red bars denote positive and negative weight-of-evidence, respectively.
- The shade of the bars encodes the degree significance of WoE according to the table above.

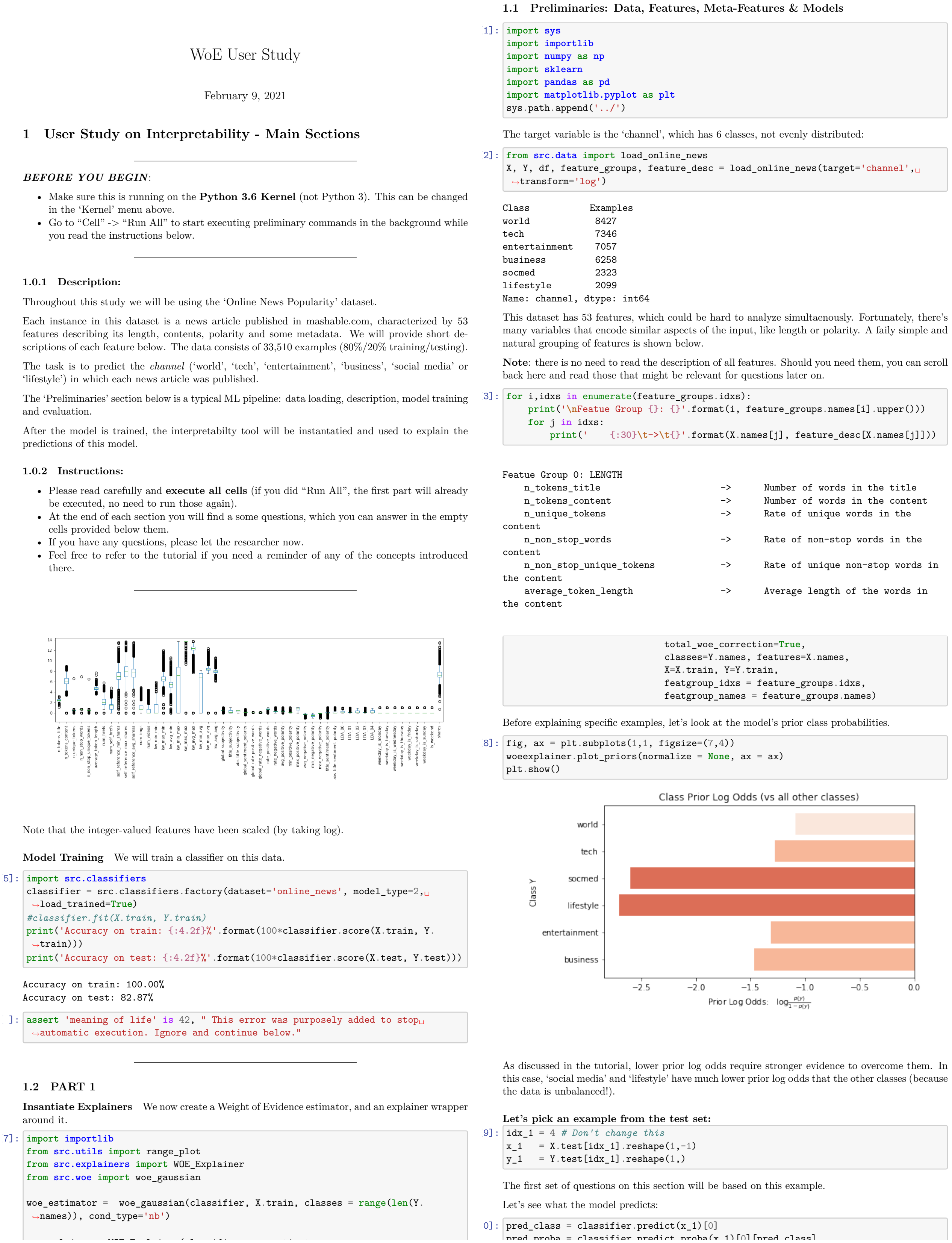
Checkpoint Questions

Based on the WoE explanation above, please answer the following questions:

Q3: According to this classifier, does having a fever increase or decrease the odds of having a flu for this person? What about a having a cough?

Answer:

Figure 6: Jupyter notebook for the tutorial (continued).



WoE User Study

February 9, 2021

1 User Study on Interpretability - Main Sections

BEFORE YOU BEGIN:

- Make sure this is running on the **Python 3.6 Kernel** (not Python 3). This can be changed in the 'Kernel' menu above.
- Go to "Cell" -> "Run All" to start executing preliminary commands in the background while you read the instructions below.

1.0.1 Description:

Throughout this study we will be using the 'Online News Popularity' dataset.

Each instance in this dataset is a news article published in mashable.com, characterized by 53 features describing its length, contents, polarity and some metadata. We will provide short descriptions of each feature below. The data consists of 33,510 examples (80%/20% training/testing).

The task is to predict the *channel* ('world', 'tech', 'entertainment', 'business', 'social media' or 'lifestyle') in which each news article was published.

The 'Preliminaries' section below is a typical ML pipeline: data loading, description, model training and evaluation.

After the model is trained, the interpretabilty tool will be instantatied and used to explain the predictions of this model.

1.0.2 Instructions:

- Please read carefully and **execute all cells** (if you did "Run All", the first part will already be executed, no need to run those again).
- At the end of each section you will find a some questions, which you can answer in the empty cells provided below them.
- If you have any questions, please let the researcher now.
- Feel free to refer to the tutorial if you need a reminder of any of the concepts introduced there.

Note that the integer-valued features have been scaled (by taking log).

Model Training We will train a classifier on this data.

```
5]: import src.classifiers
    classifier = src.classifiers.factory(dataset='online_news', model_type=2,␣
     ↪load_trained=True)
    #classifier.fit(X.train, Y.train)
    print('Accuracy on train: {:4.2f}%'.format(100*classifier.score(X.train, Y.
     ↪train)))
    print('Accuracy on test: {:4.2f}%'.format(100*classifier.score(X.test, Y.test)))
```

```
Accuracy on train: 100.00%
Accuracy on test: 82.87%
```

```
]: assert 'meaning of life' is 42, " This error was purposely added to stop␣
    ↪automatic execution. Ignore and continue below."
```

1.2 PART 1

Insantiate Explainers We now create a Weight of Evidence estimator, and an explainer wrapper around it.

```
7]: import importlib
    from src.utils import range_plot
    from src.explainers import WOE_Explainer
    from src.woe import woe_gaussian

    woe_estimator =  woe_gaussian(classifier, X.train, classes = range(len(Y.
     ↪names)), cond_type='nb')
```

1.1 Preliminaries: Data, Features, Meta-Features & Models

```
1]: import sys
    import importlib
    import numpy as np
    import sklearn
    import pandas as pd
    import matplotlib.pyplot as plt
    sys.path.append('../')
```

The target variable is the 'channel', which has 6 classes, not evenly distributed:

```
2]: from src.data import load_online_news
    X, Y, df, feature_groups, feature_desc = load_online_news(target='channel',␣
     ↪transform='log')
```

```
Class            Examples
world             8427
tech              7346
entertainment     7057
business          6258
socmed            2323
lifestyle         2099
Name: channel, dtype: int64
```

This dataset has 53 features, which could be hard to analyze simultaenously. Fortunately, there's many variables that encode similar aspects of the input, like length or polarity. A faily simple and natural grouping of features is shown below.

Note: there is no need to read the description of all features. Should you need them, you can scroll back here and read those that might be relevant for questions later on.

```
3]: for i,idxs in enumerate(feature_groups.idxs):
        print('\nFeatue Group {}: {}'.format(i, feature_groups.names[i].upper()))
        for j in idxs:
            print('    {:30}\t->\t{}'.format(X.names[j], feature_desc[X.names[j]]))
```

```
Featue Group 0: LENGTH
    n_tokens_title                          ->      Number of words in the title
    n_tokens_content                        ->      Number of words in the content
    n_unique_tokens                         ->      Rate of unique words in the
content
    n_non_stop_words                        ->      Rate of non-stop words in the
content
    n_non_stop_unique_tokens                ->      Rate of unique non-stop words in
the content
    average_token_length                    ->      Average length of the words in
the content
```

```
                              total_woe_correction=True,
                              classes=Y.names, features=X.names,
                              X=X.train, Y=Y.train,
                              featgroup_idxs = feature_groups.idxs,
                              featgroup_names = feature_groups.names)
```

Before explaining specific examples, let's look at the model's prior class probabilities.

```
8]: fig, ax = plt.subplots(1,1, figsize=(7,4))
    woeexplainer.plot_priors(normalize = None, ax = ax)
    plt.show()
```

Class Prior Log Odds (vs all other classes)
world
tech
socmed
lifestyle
entertainment
business
Class Y
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
Prior Log Odds: $\log\frac{p(y)}{1-p(y)}$

As discussed in the tutorial, lower prior log odds require stronger evidence to overcome them. In this case, 'social media' and 'lifestyle' have much lower prior log odds that the other classes (because the data is unbalanced!).

Let's pick an example from the test set:

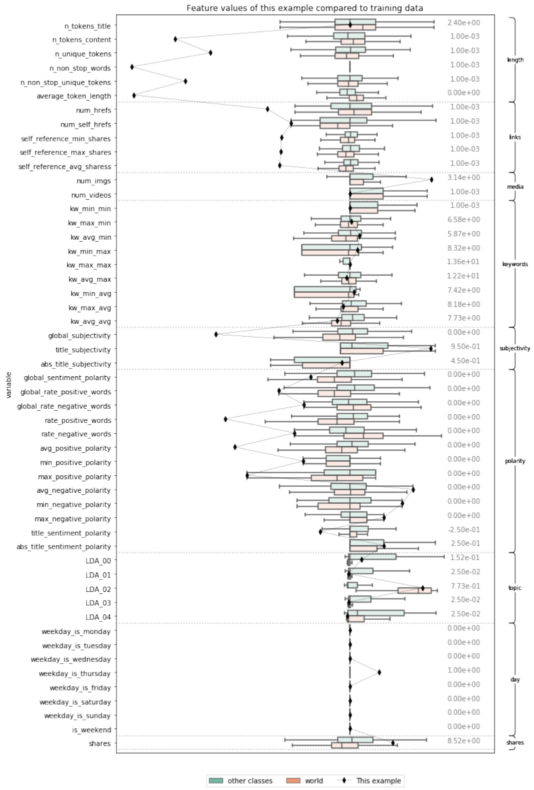
```
9]: idx_1 = 4 # Don't change this
    x_1   = X.test[idx_1].reshape(1,-1)
    y_1   = Y.test[idx_1].reshape(1,)
```

The first set of questions on this section will be based on this example.

Let's see what the model predicts:

```
0]: pred_class = classifier.predict(x_1)[0]
    pred_proba = classifier.predict_proba(x_1)[0][pred_class]
```

Figure 7: Jupyter notebook for the main study (some output pages omitted for brevity).



The boxplots have been centered and scaled in this plot to facilitate visualization.

While the actual values of the features are not too important, the position of the black dots (the example being explained) with respect to the training is useful to understand how this instance relates other examples.

Now we explain the model's prediction for this example, using the Explainer tool.

Attention: Here, **you have to choose** whether to visualize the explanation by features of by feature groups. Don't worry! You can switch as needed.

2]:

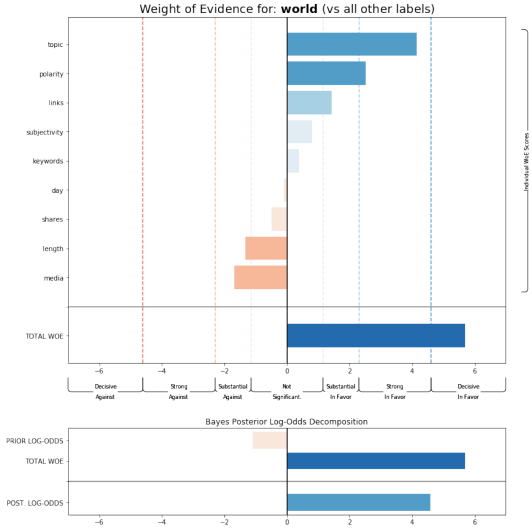
```
### Uncomment one to select TYPE of explanation unit

#explanation_units = 'features'
explanation_units = 'feature_groups'

e = woeexplainer.explain(x_1,y_1, totext=False, units=explanation_units)
```

```
Prediction: 'world' (p=0.99)
True Class: 'world'
Primary Hyp. (in favor of): 'world'
Altern. Hyp. (against):
'business','entertainment','lifestyle','socmed','tech'
Bayes Odds Decomposition:
                      4.60      =           -1.09  +  5.69
                post. log-odds  =  prior log-odds  +  total_woe

Total WoE in favor of 'world':     5.69
```



We will now take a look at a different example.

5]:

```
idx_2 = 20 # Don't change this
x_2   = X.test[idx_2].reshape(1,-1)
y_2   = Y.test[idx_2].reshape(1,)
```

If you want to look at the feature boxplots for this example, uncomment the following:

6]:

```
#woeexplainer.plot_ranges(x_2, groupby='predicted', annotate='value',␣
 ↪rescale=True)
#plt.show()
```

As before, you can **select** how to display the explanation:

7]:

```
### Uncomment one to select TYPE of explanation unit

#explanation_units = 'features'
explanation_units = 'feature_groups'

e = woeexplainer.explain(x_2, y_2, units=explanation_units, totext=False)
```

```
Prediction: 'entertainment' (p=0.59)
True Class: 'entertainment'
Primary Hyp. (in favor of): 'entertainment'
Altern. Hyp. (against): 'business','lifestyle','socmed','tech','world'
Bayes Odds Decomposition:
                      0.36      =           -1.32  +  1.68
                post. log-odds  =  prior log-odds  +  total_woe

Total WoE in favor of 'entertainment':     1.68
```

Q1: In plain English, what would you say are main characteristics of this news article that the model is relying on to make its prediction? Answer:

3]:

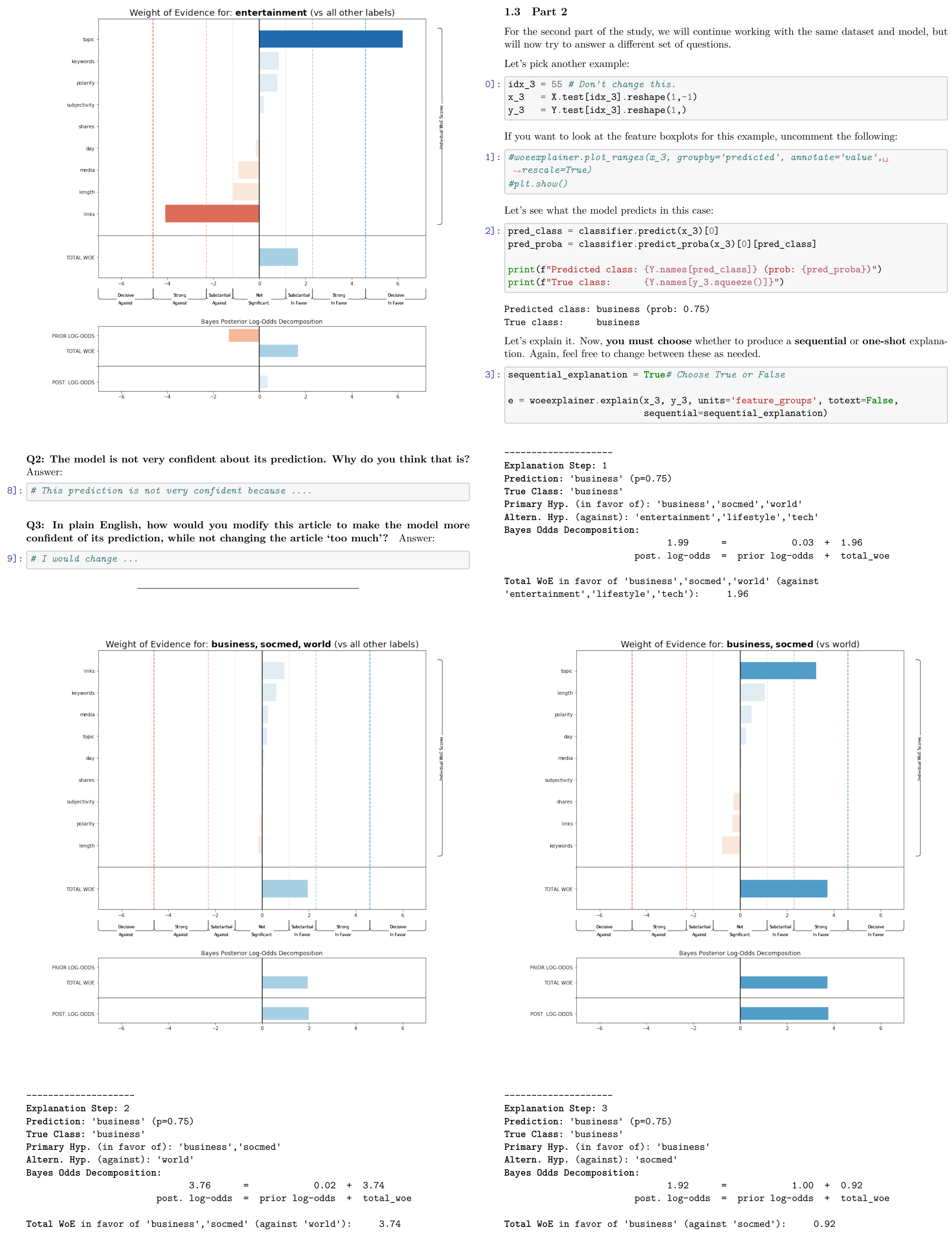
```
# The prediction is mostly based on these chacterteristics:
# 1. The article ...
# 2. The article ...
# 3. The article ...
```

Let's clear the variables before moving on:

4]:

```
if 'idx_1' in globals(): del idx_1
if 'explanation_units' in globals(): del explanation_units
```

Figure 8: Jupyter notebook for the main study (continued).



Q2: The model is not very confident about its prediction. Why do you think that is? Answer:

```
8]: # This prediction is not very confident because ....
```

Q3: In plain English, how would you modify this article to make the model more confident of its prediction, while not changing the article 'too much'? Answer:

```
9]: # I would change ...
```

```
--------------------
Explanation Step: 2
Prediction: 'business' (p=0.75)
True Class: 'business'
Primary Hyp. (in favor of): 'business','socmed'
Altern. Hyp. (against): 'world'
Bayes Odds Decomposition:
                             3.76      =           0.02  +  3.74
                        post. log-odds  =  prior log-odds  +  total_woe

Total WoE in favor of 'business','socmed' (against 'world'):     3.74
```

1.3 Part 2

For the second part of the study, we will continue working with the same dataset and model, but will now try to answer a different set of questions.

Let's pick another example:

```
0]: idx_3 = 55 # Don't change this.
    x_3   = X.test[idx_3].reshape(1,-1)
    y_3   = Y.test[idx_3].reshape(1,)
```

If you want to look at the feature boxplots for this example, uncomment the following:

```
1]: #woeexplainer.plot_ranges(x_3, groupby='predicted', annotate='value',␣
     ↪rescale=True)
    #plt.show()
```

Let's see what the model predicts in this case:

```
2]: pred_class = classifier.predict(x_3)[0]
    pred_proba = classifier.predict_proba(x_3)[0][pred_class]

    print(f"Predicted class: {Y.names[pred_class]} (prob: {pred_proba})")
    print(f"True class:      {Y.names[y_3.squeeze()]}")
```

```
Predicted class: business (prob: 0.75)
True class:      business
```

Let's explain it. Now, **you must choose** whether to produce a **sequential** or **one-shot** explanation. Again, feel free to change between these as needed.

```
3]: sequential_explanation = True# Choose True or False

    e = woeexplainer.explain(x_3, y_3, units='feature_groups', totext=False,
                             sequential=sequential_explanation)
```

```
--------------------
Explanation Step: 1
Prediction: 'business' (p=0.75)
True Class: 'business'
Primary Hyp. (in favor of): 'business','socmed','world'
Altern. Hyp. (against): 'entertainment','lifestyle','tech'
Bayes Odds Decomposition:
                             1.99      =           0.03  +  1.96
                        post. log-odds  =  prior log-odds  +  total_woe

Total WoE in favor of 'business','socmed','world' (against
'entertainment','lifestyle','tech'):     1.96
```

```
--------------------
Explanation Step: 3
Prediction: 'business' (p=0.75)
True Class: 'business'
Primary Hyp. (in favor of): 'business'
Altern. Hyp. (against): 'socmed'
Bayes Odds Decomposition:
                             1.92      =           1.00  +  0.92
                        post. log-odds  =  prior log-odds  +  total_woe

Total WoE in favor of 'business' (against 'socmed'):     0.92
```

Figure 9: Jupyter notebook for the main study (continued).