# An Efficient Shapley Value Computation for the Naive Bayes Classifier

Vincent Lemaire, Fabrice Clérot, and Marc Boullé

Orange Innovation, Lannion, France

**Abstract.** Variable selection or importance measurement of input variables to a machine learning model has become the focus of much research. It is no longer enough to have a good model, one also must explain its decisions. This is why there are so many intelligibility algorithms available today. Among them, Shapley value estimation algorithms are intelligibility methods based on cooperative game theory. In the case of the naive Bayes classifier, and to our knowledge, there is no "analytical" formulation of Shapley values. This article proposes an exact analytic expression of Shapley values in the special case of the naive Bayes Classifier. We analytically compare this Shapley proposal, to another frequently used indicator, the Weight of Evidence (WoE) and provide an empirical comparison of our proposal with (i) the WoE and (ii) KernelShap results on real world datasets, discussing similar and dissimilar results. The results show that our Shapley proposal for the naive Bayes classifier provides informative results with low algorithmic complexity so that it can be used on very large datasets with extremely low computation time.

**Keywords:** Interpretability · Explainability · Shapley value · naive Bayes

## 1 Introduction

There are many intelligibility algorithms based on the computation of variable's contribution to classifier results, often empirical and sometimes without theoretical justifications. This is one of the main reasons why the Python SHAP library was created in 2017 by Scott Lundberg following his publication [16], to provide algorithms for estimating Shapley values, an intelligibility method based on cooperative game theory. Since its inception, this library has enjoyed increasing success, including better theoretical justifications and qualitative visualizations. It provides local explanation like other methods such as LIME [17].

In the case of the naive Bayes classifier, we show in this paper that Shapley values can be computed accurately and efficiently. The key contributions are:

- an analytical formula for the Shapley values in the case of the naive Bayes classifier,
- an efficient algorithm for calculating these values, with algorithmic complexity linear with respect to the number of variables.

The remainder of this paper is organized into three contributions : (i) in the next section 2 we give our proposal for local Shapley values in the case of the naive Bayes (NB) classifier, with further discussion in the section 3; (ii) the following section 4 compares, in an analytic analysis, our Shapley proposal to another frequently used indicator in the case of the NB classifier: the Weight of Evidence (WoE); (iii) we then provide, in section 5 an empirical comparison of the results our Shapley formulation to the results of (i) the WoE and (ii) KernelShap on real world datasets and discuss similar similar and dissimilar results. The last section concludes the paper.

## 2 Shapley for naive Bayes Classifier

To our knowledge, there is no "analytical" formula of Shapley values for the naive Bayes classifier in the literature[1]. This first section is therefore devoted to a proposal for calculating these these values, exploiting the conditional variable independence assumption that characterizes this classifier .

### 2.1 Reminders on the naive Bayes classifier

The naive Bayes classifier (NB) is a widely used tool in supervised classification problems. It has the advantage of being efficient for many real data sets [9]. However, the naive assumption of conditional independence of the variables can, in some cases, degrade the classifier's performance. This is why variable selection methods have been developed [11]. They mainly consist of variable addition and deletion heuristics to select the best subset of variables maximizing a classifier performance criterion, using a wrapper-type approach [8]. It has been shown in [4] that averaging a large number of selective naive Bayes classifiers[2], performed with different subsets of variables, amounts to considering only one model with a weighting on the variables. Bayes' formula under the assumption of independence of the input variables conditionally to the class variable becomes:

$$P(Y_k|X) = \frac{P(Y_k)\prod_i P(X_i|Y_k)^{w_i}}{\sum_{j=1}^{K}(P(Y_j)\prod_i P(X_i|Y_j)^{w_i})} \tag{1}$$

where $w_i \in [0,1]$ is the weight of variable $i$. The predicted class is the one that maximizes the conditional probability $P(Y_k|X)$. The probabilities $P(X_i|Y_i)$ can be estimated by interval using discretization for numerical variables. Gaussian naive Bayes could be also considered. For categorical variables, this estimation can be done directly if the variable takes few different modalities, or after grouping (of values) in the opposite case.

Note 1: in accordance with the naive Bayes model definition, our Shapley value proposal assumes that the variables of the model are independent conditionally to the class. In practice, we expect a variable selection method to result

[1] See the introduction of the Section 4 for a very brief literature overview

[2] In this case, it is an assembly of models providing better results than a single classifier

in a classifier relying on variables which are uncorrelated or only weakly correlated conditionally to the class. A posthoc analysis of our results shows that this is indeed the case in the experiments of this article with the parsimonious classifier used (see Section 5.1).

Note 2: Even if in equation 1 the NB have transparent weights for all feature variables it is interesting to explain NB models in order to have local interpretations.

## 2.2 Definition and notations

The following notations are used:

* the classifier uses $d$ variables: $[d] = \{1, 2, ..., d\}$
* for a subset $u$ of $[d]$, we note $|u|$ the cardinality of $u$
* for two disjoint sets $u$ and $r$ of $[d]$, let $u + r$ be $u \cup r$
* for a subset $u$ of $[d]$, we denote by $-u = [d] \backslash u$, the complement of $u$ in $d$

We define a "value function" $v(.)$ indicating for each subset $u$ of variables the maximum "contribution" they can obtain together, i.e. $v(u)$, to the output of the classifier. The maximum value (or total gain) of the value function is reached when all the variables are taken into account, $v([d])$. The Shapley value for variable $j$ is denoted $\phi_j$. Shapley's theorem [19] tells us that there is a unique distribution of Shapley values satisfying the following four properties:

- Efficiency: $v([d]) = \sum_j \phi_j$; i.e. the total gain is distributed over all the variables
- Symmetry: if $\forall u \subset -\{i, j\}$, $v(u + j) = v(u + i)$, then $\phi_j = \phi_i$; i.e. if the variables $i$ and $j$ bring the same gain to any subset of variables, then they have the same Shapley value
- Null player: if $\forall u \subset -\{i\}$, $v(u + i) = v(u)$, then $\phi_i = 0$; i.e. if the variable $i$ contributes nothing to any subset of variables, then its Shapley value is zero
- Additivity: if the $d$ variables are used for two independent classification problems $A$ and $B$ associated with $v_A, v_B$, then the Shapley values for the set of two problems are the sum of the Shapley values for each problem

## 2.3 Shapley Values for the naive Bayes Classifier

**2.3.1 'Value Function':** In the case of the NB we propose to take as 'Value Function' (case of a two-class classification problem) the log ratio (LR) of probabilities:

$$\begin{aligned}
LR &= log\left(\frac{P(Y_1|X)}{P(Y_0|X)}\right) \\
&= log\left(\frac{P(Y_1)\prod_{i=1}^{d} P(X_i|Y_1)^{w_i}}{\sum_{j=1}^{K}(P(Y_j)\prod_{i=1}^{d} P(X_i|Y_j)^{w_i})} \frac{\sum_{j=1}^{K}(P(Y_j)\prod_{i=1}^{d} P(X_i|Y_j)^{w_i})}{P(Y_0)\prod_{i=1}^{d} P(X_i|Y_1)^{w_i}}\right) \\
&= log\left(\frac{P(Y_1)\prod_{i=1}^{d} P(X_i|Y_1)^{w_i}}{P(Y_0)\prod_{i=1}^{d} P(X_i|Y_1)^{w_i}}\right) \\
&= log\left(\frac{P(Y_1)}{P(Y_0)}\right) + \sum_{i=1}^{d} w_i log\left(\frac{P(X_i|Y_1)}{P(X_i|Y_0)}\right) \qquad (2)
\end{aligned}$$

The choice of the logarithm of the odd ratio as the "value function" is motivated by two reasons (i) the logarithm of the odd ratio is in bijection with the score produced by the classifier according to a monotonic transformation (ii) the logarithm of the odd ratio has a linear form that allows the derivation of an analytical formula. This value function differs from the usual value function, $f(X) = P(Y|X)$, as mentioned and analyzed later in this document when comparing with KernelShap (see section 5.3).

We stress here that the derivation above is only valid in the case of independent variables conditionally to the class variable, which is the standard assumption for the naive Bayes classifier.

For a subset, $u$, of the variables [3] given $X_u = x_u$:

$$v(u) = \mathbb{E}_{X_{-u}|X_u=x_u}\left[LR(X_u = x_u^*, X_{-u})\right] \tag{3}$$

which we write in a "simplified" way afterwards

$$v(u) = \mathbb{E}\left[(LR(X)|X_u = x_u^*)\right] \tag{4}$$

This is a proxy of the target information provided by $u$ at the point $X = x^*$. Thus, for a point (an example) of interest $x^*$ we have:

- $v([d]) = LR(X = x^*)$, everything is conditional on $x^*$ so we have the log odd ratio for $X = x^*$
- $v(\emptyset) = \mathbb{E}_X\left[LR(X)\right] = \mathbb{E}_X\left[log(\frac{P(Y_1|X)}{P(Y_0|X)})\right]$, nothing is conditioned so we have the expectation of the log odd ratio

**2.3.2 Shapley Values:** By definition of the Shapley values [19], we have for a variable $m$:

$$\phi_m = \frac{1}{d}\sum_{u \in -m}\frac{v(u+m) - v(u)}{\binom{d-1}{|u|}} \tag{5}$$

To obtain $\phi_m$, we therefore need to calculate, for a subset of variables in which the variable $m$ does not appear, the difference in gain $v(u + m) - v(u)$. This makes it possible to compare the gain obtained by the subset of variables with and without the $m$ variable, in order to measure its impact when it "collaborates" with the others.

We therefore need to calculate $v(u+m) - v(u)$ in the case of the naive Bayes classifier. If this difference is positive, it means that the variable contributes positively. Conversely, if the difference is negative, the variable is penalizing the gain. Finally, if the difference is zero, it indicates that the variable makes no contribution. Following the example of [16] and Corollary1 with a linear model whose covariates are the log odd ratio as a 'value function', one can decompose the subsets of variables into 3 groups $\{u\}, \{m\}, -\{u + m\}$.

[3] on the covariates in $u$, we average over the conditional distribution of $X_{-u}$

**Calculation** of $v(u)$ : On $\{u\}$, we condition on $X_u = x_u$ while on $\{m\}$, $\{u+m\}$, we do an averaging. By consequent:

$$\begin{aligned}
v(u) &= \mathbb{E}\left[LR(X)|X_u = x_u^*)\right] && (6)\\
&= log(P(Y_1)/P(Y_0)) \\
&+ \sum_{k\in u} w_k log\left(\frac{P(X_k = x_k{}^*|Y_1)}{P(X_k = x_k{}^*|Y_0)}\right) \\
&+ w_m \sum_{X_m}\left[P(X_m = x_m) log\left(\frac{P(X_m = x_m|Y_1)}{P(X_m = x_m|Y_0)}\right)\right] \\
&+ \sum_{k\in -\{u+m\}} w_k \sum_{X_k}\left[P(X_k = x_k) log\left(\frac{P(X_k = x_k|Y_1)}{P(X_k = x_k|Y_0)}\right)\right]
\end{aligned}$$

(7)

**Calculation of** $v(u+m)$ : The only difference is that we also condition on $X_m$

$$\begin{aligned}
v(u+m) &= \mathbb{E}\left[LR(X)|X_{u+m} = x_{u+m}^*)\right] && (8)\\
&= log(P(Y_1)/P(Y_0)) \\
&+ \sum_{k\in u} w_k log\left(\frac{P(X_k = x_k{}^*|Y_1)}{P(X_k = x_k{}^*|Y_0)}\right) \\
&+ w_m \left[log\left(\frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)}\right)\right] \\
&+ \sum_{k\in -\{u+m\}} w_k \sum_{X_k}\left[P(X_k = x_k) log\left(\frac{P(X_k = x_k{}^*|Y_1)}{P(X_k = x_k{}^*|Y_0)}\right)\right]
\end{aligned}$$

(9)

The difference $v(u+m) - v(u)$ is independent on $u$ and therefore the combinatorial sum averaging over all $u \in -m$ in equation 5 simply vanishes and finally $\phi_m = v(u+m) - v(u)$

$$\begin{aligned}
&= w_m \left(log\left(\frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)}\right) - \sum_{X_m}\left[P(X_m = x_m) log\left(\frac{P(X_m = x_m|Y_1)}{P(X_m = x_m|Y_0)}\right)\right]\right) \\
&= w_m \left(log\left(\frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)}\right) - \mathbb{E}\left(log\left(\frac{P(X_m = x_m|Y_1)}{P(X_m = x_m|Y_0)}\right)\right)\right) && (10)
\end{aligned}$$

Equation 10 provides the exact analytical expression of the Shapley value for our choice of the log odd ratio as the value function of the weighted naive Bayes.

## 3 Interpretation and Discussion

We give here some interpretation details and discussion about the Shapley formulation which are interesting arguments for its use.

• The equation 10 is the difference between the information content of $X_m$ conditionally on $X_m = x_m^*$ and the expectation of this information. In other words, it is the information contribution of the variable $X_m$ for the value $X_m = x_m^*$ of the considered instance, contrasted by the average contribution on the entire database.

• The Equation 10 can be rewritten (we just omit the product by $w_m$) in the form:

$$
\begin{aligned}
&-\left[log\left(\frac{1}{P(X_m = x_m^*|Y_1)}\right) - \sum_{X_m}\left(P(X_m = x_m) log\left(\frac{1}{P(X_m = x_m|Y_1)}\right)\right)\right] \\
&+\left[log\left(\frac{1}{P(X_m = x_m^*|Y_0)}\right) - \sum_{X_m}\left(P(X_m = x_m) log\left(\frac{1}{P(X_m = x_m|Y_0)}\right)\right)\right]
\end{aligned}
\tag{11}
$$

The terms in brackets $[\dots]$ in equation 11 are the difference between the information content related to the conditioning $X_m = x_m^*$ and the entropy of the variable $X_m$ for each class ($Y_0$ and $Y_1$). This term measures how much conditioning on $X_m = x_m^*$ brings information about the target classes.

• For a given variable, the expectation of our Shapley proposal is equal to zero, due to the conditional independence of the variables. The consequence is that high Shapley values in some parts of the data space must be exactly compensated by low values in other parts of the data space.

• For a given example if we return to our choice of value function (equation 2) and using the sum of equation 10 over the $d$ variables we have:

$$
\begin{aligned}
LR &= log\left(\frac{P(Y_1)}{P(Y_0)}\right) + \sum_{m=1}^{d} w_m log\left(\frac{P(X_m|Y_1)}{P(X_m|Y_0)}\right) \\
&= log\left(\frac{P(Y_1)}{P(Y_0)}\right) + \sum_{m=1}^{d} \phi_m + \sum_{m=1}^{d} \mathbb{E}\left(log\left(\frac{P(X_m = x_m|Y_1)}{P(X_m = x_m|Y_0)}\right)\right) \\
&= \sum_{m=1}^{d} \phi_m + \text{cste}
\end{aligned}
\tag{12}
$$

We obtain a result consistent with the notion of a value function for the Shapley's formulation. Our value function consists of a constant plus the individual contribution of the $d$ variables. The constant is the log ratio of class prior plus the sum of the average contribution of all variables.

• If we inverse the role of $Y_0$ and $Y_1$ in equation 10, we observe that the Shapley value is symmetric; i.e the positive contribution of the variable for $Y_0$ is negative for $Y_1$ (with the same absolute value).

• When the numerical (resp. categorical) variables have been previously discretized into intervals (resp. groups of values), the complexity of the equation 10 is linear in the number of discretized parts. For an input vector made up of $d$ variables, this complexity is $O(\sum_{i=1}^{d} P_i)$ where $P_i$ is the number of discretized parts of variable $i$.

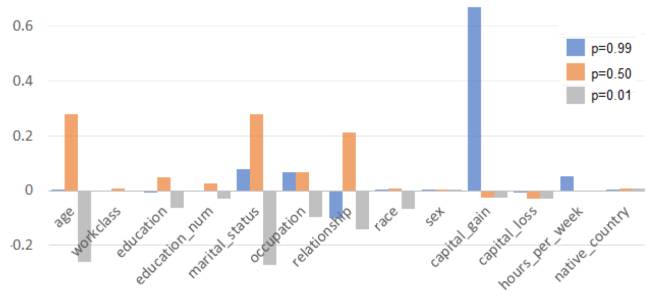

**Fig. 1.** Normalized Shapley values - Illustrative example ($\frac{\phi_m}{\sum_{i=1}^{d} \phi_i}$)

• In term of explainability, if the discretization method used for numerical attributes (resp. grouping method for categorical attributes) provides a reasonable number of intervals (resp. groups of values), then the number of potential "behaviors" of the individuals in the classification problem is small and therefore easy to understand.

• Extension to multiclass: We simply define the Shapley Value of an input variable as the sum of the absolute $C$ Shapley values when choosing in equation 10 one of the $C$ class of the problem as the "positive class" ($Y_1$) and all the others $C-1$ class as the "negative class" ($Y_0$). For example in a 3 class problems where the class are 'red', 'green', and 'yellow':

$$\begin{aligned}\phi_m &= |\phi_m(Y_1 = \{red\}, Y_0 = \{green, yellow\})| \\ &+ |\phi_m(Y_1 = \{green\}, Y_0 = \{red, yellow\})| \\ &+ |\phi_m(Y_1 = \{yellow\}, Y_0 = \{green, red\})|\end{aligned}$$

In this way, we can find out which feature has the greatest impact on all classes. Note that there are other ways of measuring the impact of features in multi-classification problems (see, for example, the discussion in [1] on using the SHAP package for multi-classification problems).

• To conclude this discussion and prior to the experiments presented in Section 5, we give here an illustrative example on the Adult dataset (the experimental conditions are the same as those presented in Section 5). Figure 1 shows the Shapley values obtained for 3 examples which are respectively predicted as belonging to the class 'more' with probabilities 0.99, 0.50 and 0.01. On this well-known data set, we find the usual results on the role of input variables for examples with high to low probabilities when considering the class 'more'.

## 4 Analytic comparison with the Weight of Evidence

In the case of the naive Bayes classifier, there are a number of "usual" methods for calculating the importance of input variables. We do not go into detail on all of them, but the reader can find a wide range of these indicators in [18,12] for a brief literature overview but nonetheless quite exhaustive. This section focuses on presenting the “Weight of evidence” (WoE) [7] and its comparison with the Shapley values proposed in the previous section, since this indicator is (i) close to the equation presented above (equation 10) and (ii) among the most widely used indicators for the naive Bayes classifier.

We give below the definition of the WoE (in the case with two classes) which is a log odds ratio calculated between the probability of the output of the model and the latter deprived of the variable $X_m$:

$$(WoE)_m = log\left(\frac{\frac{p}{1-p}}{\frac{q}{1-q}}\right) = w_m\left(log\left(\frac{\frac{P(Y_1|X)}{P(Y_0|X)}}{\frac{P(Y_1|X\backslash X_m)}{P(Y_0|X\backslash X_m)}}\right)\right) = w_m\left(log\left(\frac{P(Y_1|X)P(Y_0|X\backslash X_m)}{P(Y_0|X)(Y_1|X\backslash X_m)}\right)\right) \tag{13}$$

$$(WoE)_m = w_m\left(log\left(\frac{P(Y_1)\left[\prod_{i=1}^{d} P(X_i|Y_1)\right] P(Y_0)\left[\prod_{i=1,i\neq m}^{d} P(X_i|Y_0)\right]}{P(Y_0)\left[\prod_{i=1}^{d} P(X_i|Y_0)\right] P(Y_1)\left[\prod_{i=1,i\neq m}^{d} P(X_i|Y_1)\right]}\right)\right) \tag{14}$$

by simplifying the numerator and denominator:

$$(WoE)_m = w_m\left(log\left(\frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)}\right)\right) \tag{15}$$

**Link between $(WoE)_m$ and $\phi_m$:** If we compare the equations 15 and 10, we can see that it is the reference that changes. For the Shapley value ( equation 10), the second term takes the whole population as a reference whereas for the WoE (equation 15) the reference is zero. The averaging is not at the same place between the two indicators, as we will demonstrate just below. We can also observe that the expectation of our Shapley proposal is equal to zero, whereas the expectation of WoE is the second term of our Shapley proposal (second part, the expectation term, of equation 10).

In case of the naive Bayes classifier, “depriving” the classifier of a variable is equivalent to performing a “saliency” calculation (as proposed in [13]) which takes into account the probability distribution of the variable $X_m$. Indeed, to deprive the classifier of the variable $X_m$, it is sufficient to recalculate the average of the classifier's predictions for all the possible values of the variable $X_m$ as demonstrated in [18]. Indeed, if we assume that the variable $X_m$ has $k$ distinct values, Robnik et al. [18] have shown that the saliency calculation of [13] is exact in the naive Bayes case and amounts to “erasing” the variable $X_m$. Denoting either $Y = Y_0$ or $Y = Y_1$ by $Y_.$, we have

$$P(Y_.|X\backslash X_m) = \sum_{q=1}^{k} P(X_m = X_q)\frac{P(Y_.|X, X_m = X_q)}{P(X, X_m = X_q)} \tag{16}$$

$$P(Y_.|X\backslash X_m) = \sum_{q=1}^{k} P(X_m = X_q)\left(P(Y_.)\left(\prod_{i=1,i\neq m}^{d}\frac{P(X_i|Y_.)}{P(X_i)}\right)\frac{P(X_m = X_q|Y_.)}{P(X_m = X_q)}\right)$$

$$P(Y_{\cdot}|X\backslash X_m) = P(Y_{\cdot}) \prod_{i=1, i\neq m}^{d} P(X_i|Y_{\cdot}) \left( \sum_{q=1}^{k} \frac{P(X_m = X_q)P(X_m = X_q|Y_{\cdot})}{P(X_m = X_q)} \right) \quad (17)$$

$$P(Y_{\cdot}|X\backslash X_m) = P(Y_{\cdot}) \prod_{i=1, i\neq m}^{d} P(X_i|Y_{\cdot}) \quad (18)$$

with $P(Y_{\cdot}|X, X_m = X_q)$ being $P(Y_{\cdot}|X)$ but where the value of the variable $X_m$ has been replaced by another value of its distribution $X_q$. This last result is interesting because with the help of the equation 17 we can rewrite the equation 13 in :

$$(WoE)_m = w_m \left( log \left( \frac{P(Y_1|X)P(Y_0|X\backslash X_m)}{P(Y_0|X)(Y_1|X\backslash X_m)} \right) \right)$$

$$(WoE)_m = w_m log \frac{\left( P(Y_1) \prod_{i=1}^{d} P(X_i|Y_1) \right) \left( P(Y_0) \prod_{i=1, i\neq m}^{d} P(X_i|Y_0) \sum_{q=1}^{k} P(X_m = X_q|Y_0) \right)}{\left( P(Y_0) \prod_{i=1}^{d} P(X_i|Y_0) \right) \left( P(Y_1) \prod_{i=1, i\neq m}^{d} P(X_i|Y_1) \sum_{q=1}^{k} P(X_m = X_q|Y_1) \right)}$$

$$(WoE)_m = w_m \left( log \left( \frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)} \frac{\sum_{q=1}^{k} P(X_m = X_q|Y_0)}{\sum_{q=1}^{k} P(X_m = X_q|Y_1)} \right) \right)$$

$$(WoE)_m = w_m \left( log \left( \frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)} \right) + log \left( \frac{\sum_{q=1}^{k} P(X_m = X_q|Y_0)}{\sum_{q=1}^{k} P(X_m = X_q|Y_1)} \right) \right) \quad (19)$$

$$(WoE)_m = w_m \left( log \left( \frac{P(X_m = x_m^*|Y_1)}{P(X_m = x_m^*|Y_0)} \right) + log \left( \frac{1}{1} \right) \right) \quad (20)$$

This result allows to better understand why the WoE is referenced in zero. The comparison of the equation 10 and the equation 19 exhibits the difference in the localization of the averaging resulting in a reference in zero for the WoE. In the first case an expectation is computed on the variation of the log ratio $log(P(Y_1|X)/P(Y_0|X))$ while in the second case this expectation is computed only on the variations of $f(X) = P(Y_1|X)$ (or reciprocally $P(Y_0|X)$).

This comparison shows the effect of choosing either the odds (our Shapley proposal) or the output of the classifier (WoE) as the 'value function'. Since both results are very consistent, and WoE does not suffer from calculation exhaustion, the two methods are very close.

## 5 Experiments

The experiments carried out in this section allow us to compare our Shapley proposal with the Weight of Evidence and KernelShap to highlight similar or dissimilar behaviors. We focus below on two classes problems.

The code and data used in this section are available in the GitHub repository at `https://tinyurl.com/ycxzkffk`.

### 5.1 Datasets and Classifier

**Classifier** : The naive Bayes classifier used in the experiments exploits two main steps. A first step in which (i) the numerical variables are discretized, using the method described in [6], (ii) the modalities of the categorical variables are grouped using the method described in [5]. Then, variable weights are calculated using the method described in [4]. In the first and second steps, uninformative variables are eliminated from the learning process. In this paper, we have used the free Khiops software [3] in which the whole process is implemented. This software produces a preparation report containing a table of the values of $P(X_m = x_m|Y_.)$ for all classes and all variables, enabling us to easily implement the two methods described earlier in the article.

Note: below, the same classifier and preprocessing are used for comparing the different methods used to calculate the variable importance, so that the differences in the results will be only due to those different methods.

**Dataset** : Ten datasets have been selected in this paper and are described in the Table 1. They are all available on the UCI website [14] or on the Kaggle website[2]. They were chosen to be representative datasets in terms of variety of number of numerical attributes (#Cont), number of categorical attributes (#Cat), number of instances (#Inst) and imbalance between classes[4] (Maj. class.). They are widely used in the "machine learning" community as well as in the analysis of recently published Shapley value results. In this table, we give in the last columns the performances, for information purposes, obtained by the naive Bayes used (an averaged naive Bayes, see Section 2.1); i.e the accuracy and the Area Under the ROC curve (AUC), as well as the number of variables retained by this classifier (#Var) since uninformative variables are eliminated from the learning process. As the aim of this article is not to compare classification results, we decide simply to use 100 % of the examples to train the model[5] and to compute later the importance indicators (WoE and Shapley) .

| Name | #Cont | #Cat | #Inst ($N$) | Maj. class. | Accuracy | AUC | #Var |
|---|---|---|---|---|---|---|---|
| Twonorm | 20 | 0 | 7400 | 0.5004 | 0.9766 | 0.9969 | 20 |
| Crx | 6 | 9 | 690 | 0.5550 | 0.8112 | 0.9149 | 7 |
| Ionosphere | 34 | 0 | 351 | 0.6410 | 0.9619 | 0.9621 | 9 |
| Spam | 57 | 0 | 4307 | 0.6473 | 0.9328 | 0.9791 | 29 |
| Tictactoe | 0 | 9 | 958 | 0.6534 | 0.6713 | 0.7383 | 5 |
| German | 24 | 0 | 1000 | 0.7 | 0.7090 | 0.7112 | 9 |
| Telco | 3 | 18 | 7043 | 0.7346 | 0.8047 | 0.8476 | 10 |
| Adult | 7 | 8 | 48842 | 0.7607 | 0.8657 | 0.9216 | 13 |
| KRFCC | 28 | 7 | 858 | 0.9358 | 0.9471 | 0.8702 | 3 |
| Breast | 10 | 0 | 699 | 0.9421 | 0.975 | 0.9915 | 8 |

**Table 1.** Description of the datasets used in the experiments (KRFCC = KagRiskFactorsCervicalCancer dataset)

[4] Here we give the percentage of the majority class.

[5] To facilitate reproducibility. Nevertheless, the test performances of the models (Table 1) are very close with a 10-fold cross-validation process.

### 5.2 Comparison with the WoE

In this first part of the experiments, the comparison is made with the Weight of Evidence. and we present the observed correlation between the Shapley values (Eq. 10) and the WoE values (Eq. 15).

We compute the Shapley and WoE values per class ($C$), per variable ($J$) and per instance ($N$) then, we compute the Kendall correlation[6] line per line; that is, for each example, we compute the $d$ values of WoE or of our Shapley values and then the Kendall coefficient for that example. Finally we compute the average and the standard deviation of these $N$ values which are reported in the Table 2.

The Kendall correlation is a measure of rank correlation, therefore, it measures whether the two indicators, WoE and our Shapley values, give the same ordering in the importance of the variables.

| Name | Kendall |
|---|---|
| Twonorm | 0.9919 ±8.71e-05 |
| Crx | 0.9919 ±4.28e-04 |
| Ionosphere | 0.8213 ±1.76e-02 |
| Spam | 0.9011 ±2.66e-04 |
| Tictactoe | 1.0000 ±2.60e-04 |
| German | 0.9515 ±1.01e-03 |
| Telco | 0.9210 ±3.70e-03 |
| Adult | 0.8589 ±6.57e-03 |
| KRFCC | 0.9931 ±1.77e-03 |
| Breast | 0.9222 ±2.73e-03 |

**Table 2.** Two Class problems

In Table 2, we observe only Kendall values above 0.82. Kendall's coefficient values can range from 0 to 1. The higher the Kendall's coefficient value, the stronger the association. Usually, Kendall's coefficients of 0.9 or more are usually considered very good. Kendall's coefficient means also that the appraisers apply essentially the same standard when assessing the samples. With the values shown in this Table, we observe [10] a minimum of fair agreement to a near perfect agreement between our Shapley proposition and WoE in terms of ranking of the variable importances[7].

This good agreement can be understood from two non exclusive perspectives. First, using an averaged naive Bayes model introduces a weight $w_m$ which has a strong influence on the variable importance (the higher the weight, the stronger the influence, for both methods): the variable importance would be mainly in-

[6] We used the scipy.stats.kendalltau with the default parameter, i.e $\tau$-b.

[7] It would also be interesting to see the correlations of only the most important variables (e.g. the top five), since usually only a few of the most important features are perceptible to humans. However, for lack of space, we do not present this result. We do, however, provide the code for doing so.

fluenced by the weights ordering and therefore the same for both methods. Second, it could point out to the fact that the variable-dependent reference terms $-w_m\mathbb{E}\left(log\left(\frac{P(X_m=x_m|Y_1)}{P(X_m=x_m|Y_0)}\right)\right)$ which make the difference between the Shapley value and the WoE are either small or roughly constant in our datasets. How those two perspectives are combined to lead to the good agreement experimentally observed is left for future work.

### 5.3 Comparison with Kernel Shap

Among the libraries able to compute Shapley values, one may find 'model oriented' proposals that can only be used on particular model as for example with tree-based algorithms like random forests and XGBoost (TreeShap [15], FastTreeShap [20]), or model agnostic which can be used with any machine learning algorithm as KernelShap [16]. Here since we did not find a library dedicated to naive Bayes, we compare our results to the popular Kernel Shap. In this section we attempt to compare the results obtained, for the Shapley values, with our analytic expression and the results obtained with the KernelShap library. For a fair comparison, the first point to raise is that the two processes do not use the same 'value function'. Indeed, in our case we use a log odds ratio whereas in KernelShap, when providing the classifier to the library, the value function used is the output of the classifier.

**On the use of Kernelshap [16]:** The computation time of the library can be very long, even prohibitive. To use the library, the user has to define two datasets: (i) a first dataset, as a knowledge source, which is used to perform the permutation of variable values (ii) a second dataset on which one would like to obtain the Shapley values. The first database is used to compute the Shapley value of the variables for a given example. Given this table and a variable of interest, an example $X_i$, is modified thanks to the permutation of the others variables. This allows the KernelShap library to create a "modified table" which contains all the modified versions of this example.

To give more intuition about the size of 'the modified-example-table' we plot, in Figure 2, for the "CRX" dataset, the size of this table as a function of the number of examples in the 'knowledge table', showing the linear increase that results from a very large table. Then the classifier have to predict its output value for the considered example $X_i$ to compute the Shapley values. For this "CRX" dataset, which contains 15 inputs variables, the time taken to compute the Kernelshap values for a single example and using all the 690 examples as 'knowledge table' is 12.13 seconds[8], so 8370 seconds for the entire dataset (around 2.5 hours for a small dataset). To summarize, the algorithmic complexity of KernelShap is $O(N_k 2^d)$ where $N_k$ is the number of examples used in the 'knowledge table'.

As a consequence, we were not able to obtain a complete result on most datasets (even with a half-day credit) when using the entire dataset. As sug-

[8] The characteristics of the used computer are: Intel(R) Core(TM) i7-10875H (No. of threads. 16; 5.10 GHz) RAM:32.0 Go, Windows 10, Python 3.8.6

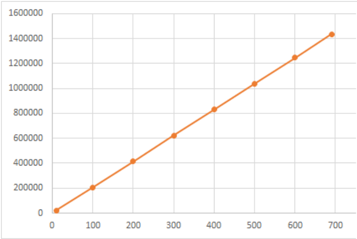

**Fig. 2.** CRX dataset: size of the "modified table" versus the number of examples in the "knowledge" data table.

gested[9] by the KernelShap library, in the results below we limit the computation time to a maximum of 2 hours per dataset: (i) the Shapley values are computed only on 1000 (randomly chosen) examples[10] and (ii) the number of examples in the 'knowledge table', $N_k$ [11], has been set to the values indicated in the Table 3 (where the number of examples of the entire dataset is given as a reminder in the brackets).

**On the use of our Shapley proposal -** In contrast, for the analytic Shapley proposed in this paper, the time required to compute the Shapley values is very low (see the discussion in Section 3). Indeed, the algorithmic complexity, for an input variable, is linear in number of parts, intervals or groups of values (see Equation 10). On the largest dataset used in this paper, the Adult dataset which contains 48842 examples, the time used to compute all the Shapley values for all the variables, all the classes and all the examples is lower than 10 seconds. This computation time could be further reduced if the $log(P(X|C)$ per variable and per interval (or group values) are precomputed as well as the expectation term of the equation 10, which is not the case in our experiments.

**Results:** The Table 3 gives the correlation between the global Shapley values, defined for each variable as the average on all samples of the absolute values of the local Shapley values. We observe good correlations for both coefficients. We also give an example of comparison on the TwoNorm dataset in Figure 3 (where we have drawn the normalized global Shapley values), for which the correlations are lowest in the Table 3. For this data set, the lower Kendall coefficient value is due to the fact that many variables have close Shapley values, resulting in differences in their value ranks. Based on all the results we may conclude that there is a nice agreement between our Shapley proposal and KernelShap on the ten datasets used in this paper.

[9] The variance in the results observed in recent publications is due to this constraint.

[10] It is obvious that for large datasets such as the "adult" the chosen sample of 1000 is statistically insignificant and, as a result, the calculated importance values, computed by KernelShap may not be reliable.

[11] We start with 50 examples (as a minimum budget) and we increment this number by step of 50 until the credit is reached.

| Name | $N_k$ | Pearson | Kendall |
|---|---|---|---|
| Twonorm | 200 (7400) | 0.9027 | 0.7052 |
| Crx | 690 (690) | 0.9953 | 0.9047 |
| Ionosphere | 351 (351) | 0.9974 | 0.8888 |
| Spam | 200 (4307) | 0.8829 | 0.7684 |
| Tictactoe | 958 (958) | 1.0000 | 1.00 |
| German | 1000 (1000) | 0.9974 | 0.9047 |
| Telco | 1000 (7043) | 0.9633 | 0.7333 |
| Adult | 1000 (48842) | 0.8373 | 0.7692 |
| KRFCC | 858 (858) | 0.9993 | 1.00 |
| Breast | 699 (699) | 0.9908 | 0.8571 |

**Table 3.** Correlation between our analytic Shapley and Kernelshap

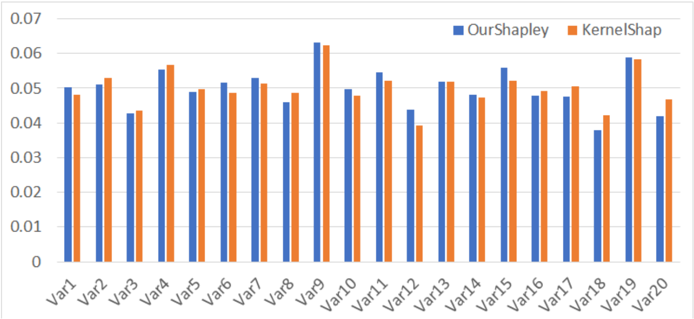

**Fig. 3.** Two Norm dataset: Comparison of our Shapley proposal and KernelShap.

## 6 Conclusion

In this paper, we have proposed a method for analytically calculating Shapley values in the case of the naive Bayes classifier. This method leverages a new definition of the value function and relies on the independence assumption of the variables conditional on the target to obtain the exact value of the Shapley values, with a linear algorithmic complexity linear with respect to the number of variables. Unlike alternative evaluation/approximation methods, we rely on assumptions that are consistent with the underlying classifier and avoid approximation methods, which are particularly costly in terms of computation time. We also presented a discussion on the key elements that help to understand the proposal and its behavior.

We compared this Shapley formulation, in an analytic analysis, to another frequently used indicator, the Weight of Evidence (WoE). We also carried out experiments on ten datasets to compare this proposal with the Weight of Evidence and the KernelShap to highlight similar or dissimilar behaviors. The results show that our Sphaley proposal for the naive Bayes classifier is in fair agreement with the WoE and with KernelShap's Shapley values, but with a much lower algorithmic complexity, enabling it to be used for very large datasets with extremely reduced computation times.

## References

1. https://github.com/slundberg/shap/issues/367
2. Kaggle. https://www.kaggle.com
3. Khiops: supervised machine learning tool for mining large multi-table databases. In: Extraction et Gestion des Connaissances. pp. 505–510 (2016), http://www.khiops.com, [available at http://www.khiops.com]
4. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research **8**, 1659–1685 (2007)
5. Boullé, M.: A Bayes optimal approach for partitioning the values of categorical attributes. Journal of Machine Learning Research **6**, 1431–1452 (2005)
6. Boullé, M.: MODL: a Bayes optimal discretization method for continuous attributes. Machine Learning **65**(1), 131–165 (2006)
7. Good, I.J.: Probability and the weighing of evidence. C. Griffin & Company Limited (1950)
8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
9. Hand, D.J., Yu, K.: Idiot's bayes-not so stupid after all? International Statistical Review **69**(3), 385–398 (2001)
10. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**(1), 159–174 (1977)
11. Langley, P., Sage, S.: Induction of selective bayesian classifiers. In: Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence. pp. 399–406. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994)
12. Lemaire, V., Boullé, M., Clérot, F., Gouzien, P.: A method to build a representation using a classifier and its use in a k nearest neighbors-based deployment. In: Proceedings of International Joint Conference on Neural Networks (2010)
13. Lemaire, V., Clérot, F.: An Input Variable Importance Definition based on Empirical Data Probability Distribution, pp. 509–516. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
14. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
15. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888 (2018)
16. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
17. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. p. 1135–1144. KDD '16, Association for Computing Machinery (2016)
18. Robnik-Sikonja, M., Kononenko, I.: Explaining classifications for individual instances **20**, 589 – 600 (06 2008)
19. Shapley, L.S., Shubik, M.: A method for evaluating the distribution of power in a committee system. American Political Science Review **48**(3), 787–792 (1954)
20. Yang, J.: Fast treeshap: Accelerating shap value computation for trees. In: Workshop on eXplainable AI approaches for debugging and diagnosis (XAI4Debugging@NeurIPS2021). (2021)