Explanation of Deep Learning Models via Logic Rules Enhanced by Embeddings Analysis, and Probabilistic Models

Victor Contreras[1 [0000−0002−6189−0217]*], Michael Schumacher[1 [0000−0002−5123−5075]], and Davide Calvaresi[1 [0000−0001−9816−7439]]

University of Applied Sciences Western Switzerland (HES-SO), Switzerland
victor.contrerasordonez@hevs.ch, davide.calvaresi@hevs.ch

Abstract. Deep Learning (DL) models are increasingly dealing with heterogeneous data (i.e., a mix of structured and unstructured data), calling for adequate eXplainable Artificial Intelligence (XAI) methods. Nevertheless, only some of the existing techniques consider the uncertainty inherent to the data. To this end, this study proposes a pipeline to explain heterogeneous data-based DL models by combining embedding analysis, rule extraction methods, and probabilistic models. The proposed pipeline has been tested using synthetic data (multi-individual food items tracking). This study has achieved (i) inference enhancement through probabilistic and evidential reasoning, (ii) generation of logical explanations based on extracted rules and predictions, and (iii) integration of textual data into the explanation pipeline through embedding analysis.

Keywords: XAI · Deep learning · Uncertainty reasoning · Rule extraction · Preference modeling · Heterogeneous data processing.

1 Introduction

Explainable Artificial Intelligence (XAI) is a research and application domain that arose to foster transparency and understanding of decision processes performed by artificial intelligence (AI) algorithms [33]. XAI enables trust between users and AI-based systems, which is crucial in user-centric applications such as virtual coaches, decision support systems, recommender systems, assistive systems, and safety-critical domains (i.e., healthcare, automotive, aviation, and nuclear energy) [18,35]. Furthermore, XAI techniques offer numerous advantages to researchers and machine learning (ML) engineers. These include bias detection, error diagnosis and debugging, accountability and responsibility, regulatory compliance, and human-AI interaction [1].

One of XAI's most prolific research areas is extracting explanations from DL models [10]. DL models learn efficiently with an excellent generalization capacity due to their ability to model nonlinear relationships between the input features and the expected output [72]. However, DL models are black boxes due

to their nonlinear, distributed, and redundant structure that encodes the knowledge learned from data into connections, weights, and nonlinear activations [20].

The dependency on DL-based applications keeps growing, entailing several XAI contributions. Among these approaches, it is worth mentioning those based on gradient analysis [61], rule extraction [81,19], perturbation analysis [31], and the use of surrogate models [7]. Most of these approaches present different sets of advantages and drawbacks that make them suitable for given architectures (e.g., models with only one hidden layer, hidden layers with linear activations), types of explanation to be generated (e.g., heat maps, logic models, etc.), and tasks (e.g., classification, regression, reinforcement learning). Notwithstanding the significant progress in the field of XAI and the diversity of methods developed, most of them entail at least one of the following challenges (C):

C1: Uncertainty is inherent in the data and affects the models, conclusions, and explanations derived from them [24]. However, uncertainty is rarely taken into account as a fundamental factor in deep learning and XAI applications. Therefore, uncertainty must be acknowledged, measurable, and bound in order to obtain flexible, confident predictions and explanations.

C2: Data heterogeneity commitment. Most of the XAI techniques for DL are designed to work with homogeneous input data, such as text, images, or tabular data. However, very few methods are capable of producing explanations with heterogeneous/multimodal data, which refers to a combination of structured (tabular data), semi-structured (data contained in JSON, XML), and unstructured (images, text, audio, video) data.

Elaborating on those challenges and the contextual limitations, this study focuses on the following main research question (MRQ):

MRQ *Can heterogeneous data DL models be explained by combining embedding analysis, reasoning under uncertainty, and logical rules?*

Such an MRQ encompasses the following two research topics:

RT1: Application of cluster analysis to integrate embeddings representing unstructured data (i.e., text) into the explanation pipeline.

RT2: Modelization of the decision processes carried out within a DL model as probabilistic decision processes, considering the uncertainty inherent in the data and the models employed.

Considering the MRQ and research topics, we formulate the hypothesis.

Hypothesis 1 (H1): *Unstructured data can be integrated into a rule-based explanation pipeline through embedding analysis (RT1). The extracted rule set can be enhanced through probabilistic modeling (RT2), which considers uncertainty in data and models.*

To test hypothesis H1, we have designed and implemented an explanation pipeline for explaining DL predictors, combining rule extraction methods, embedding analysis, and probabilistic graphical models. The proposed pipeline intends to overcome challenges C1 (considering the uncertainty of data and models) and C2 (generating explanations based on heterogeneous data – tabular and

textual data). The proposed pipeline has been tested on a DL-based food recommender model. Such a model takes in input recipe features, a user profile, and contextual information and predicts their food preferences.

The remainder of the paper is organized as follows: Section 2 presents the state-of-the-art methods and algorithms on rule extraction methods, embedding analysis, and XAI applied to DL predictors. Section 3 describes the proposed methodology and pipeline. Section 4 presents results and analysis. Section 6 discusses the overall study. Finally, Section 7 concludes the paper.

2 State of the Art

This section provides an overview and analysis of the relevant works. In particular, it covers XAI methods and dives into rule extraction algorithms, embedding representation, and probabilistic reasoning applied to DL predictors.

2.1 XAI methods in a nutshell

XAI is a research field within AI whose main objective is to explain machine learning (ML) models in human terms [5]. XAI methods can be classified into two main categories: Explainable-by-design and Post-hoc explanations.

On the one hand, explainable-by-design methods are based on transparent ML models whose structure and parameters directly explain their behavior [55] (examples are decision trees [64], rule-based systems [74], and linear models [36]). On the other hand, Post-hoc explanation methods aim to extract explanations from trained ML models whose parameters and structure cannot explain their behavior [71]. Post-hoc methods can generate global and local explanations. Local explanations describe the model's behavior in one particular example (i.e., LIME [28], CIU [32]). Global explanations refer to the overall behavior of the model [4]. Common local explanation approaches include Local feature importance [69], feature attribution [49], sensitivity analysis [77], and local surrogate models [82]. Prevalent global explanation methods include global surrogate models [34], global feature importance [65], and rule extraction approaches [6,18].

Rule extraction methods DL models hold their knowledge in a distributed and redundant manner. Rule extraction methods applied to DL models aim to explain their behavior through rules sets [19]. Rule sets can be extracted by applying one of the following approaches:

- Pedagogical: This approach considers the DL model as a black box and replaces it with a surrogate model that is explainable by design and trains it with the input features and with the DL model's predictions, thus extracting a global rule set [9]. Example of pedagogical methods are: PSyKE [62], TREPAN [16], and RxREN [11].

- Decompositional: This approach extracts rules sets from a DL predictor by analyzing its hidden layers, weights, and activations. Usually, those methods induce rule sets layer-by-layer and neuron-by-neuron and then merge them to produce a global explanation of the DL model's behavior. Common decompositional methods are FERNN [63], Eclaire [81], and DEXiRE [19].
- Eclectic: This approach iteratively combines pedagogical and decompositional methods to produce global rule sets that explain the behavior of the DL model [2]. Examples of this approach are RX [37] and DeepRED [83].

Rule sets can have different representations, such as first-order logic [62] (i.e., $\forall x\ man(x) \implies drink(x, water)$) and fuzzy logic [39] (i.e., $IF\ BMI\ is\ obese\ AND\ activity\ is\ low\ THEN\ calories\ intake\ is\ low$).

2.2 Embedding representation

An embedding is a numerical representation of a real-world object [44]. Embeddings are employed to generate numerical representations of unstructured data such as text [59] and images [21]. Embedding representation can be learned from data using different algorithms such as dimensionality reduction [66], manifold learning [41], and DL models [46]. Embedding algorithms aim to learn representations that capture semantic characteristics and spatial or temporal dependencies in the object [22].

Embedding algorithms have been successfully employed in natural language processing (NLP) as a compact, semantic, and efficient way to represent words, sentences, paragraphs, and documents [59]. The most recognized embedding algorithms for NLP include those based on machine learning, such as ones that capture the syntactic and semantic relationship between words in a text through methods like a continuous-bag-of-words (CBOW) or skip-gram (i.e., word2vec [15], Doc2vec [45]) models, and transformer-based (i.e., Universal Sentence Encoder [12] and BERT [27]). Embeddings have also been used to project different objects into a common vector space where they can be compared and processed, with applications in information retrieval [54], recommender systems [25] and multimodal machine learning [47] (i.e., image captioning [80] and multimodal question answering [68]).

2.3 Probabilistic graphical models

Probabilistic graphical models (PGM) learn a structural association between random variables, modeling complex probabilistic relationships [48]. PGM can be classified into two main categories:

- Undirected Graphical Models (UGD) model bidirectional dependencies between random variables, considering their context, can contain cycles and are widely employed in computer vision [43] and natural language processing [38]. Conditional random fields (CRF) [67], Boltzmann machines [3], and Hidden Markov Models (HMM) [30] are common examples of UGD models.

– Directed Acyclic Graph Models (DAG) model causal relationships between random variables. Bayesian networks (BN) are DAG models (each node represents a feature or measure of interest, and the edges represent causal informative dependency) [42]. BN is used in probabilistic and evidential reasoning [70,79], and to measure causal and intervention effects (Do-calculus) [60].

PGM have been employed as explainers in XAI. Dikopoulou et al. [29] introduced the gLIME method that combines LIME and graphical least absolute shrinkage and selection operator (GLASSO) to produce model agnostic PGM-based explanations describing the feature importance and their uncertainty values. Vu and Thai [73] developed PGM-explainer, a method to generate explanations on graph neural networks (GNN) via structure learning and surrogate PGM models. Chen et al. [13] proposed the Breast Cancer Causal XAI Diagnostic Model to generate causal explanations from mammography reports, employing a GNN model and causal tabular learning method (Causal-TabNet) to learn a BN from feature relationships and aggregate node information. The learned BN enables causal reasoning and feature attribution explanation.

According to Derks et al. [26], BN can model and reason causally, making them suitable for generating causal explanations.

3 Methodology

This section presents the proposed pipeline's rationale, description, and experimental protocol to test hypothesis H1.

3.1 The rationale behind the proposed pipeline

On the one hand, rule extraction methods, such as DEXiRE [19] and Eclaire [81], use frequentist and entropy measures to identify the most relevant neurons in each layer and the most probable decision path. However, DEXiRE is not a probabilistic model; consequently, it does not consider the uncertainty in the data and the DL model to be explained. For this reason, DEXiRE's explanations are limited to the domain of deterministic logic.

On the other hand, probabilistic graphical models (PGM) can reason under uncertainty, based on partial or noisy evidence, and employ frequentist or Bayesian approaches to model beliefs and causal hypotheses [48]. Despite these advantages, the computational complexity of PGM grows exponentially in the number of random variables to be considered, which limits their ability to explain complex models with multiple input features (causes), such as DL models.

Logic and probabilistic approaches cover different XAI aspects. We have merged these approaches into an explanation generation pipeline to maximize their benefits and mitigate their drawbacks. The proposed pipeline uses rule-based explanations as input to build probabilistic models, reducing the number of factors to be considered and thus limiting its computational complexity.

Rule-based and probabilistic explanations commonly operate on structured data (e.g., numerical and categorical data). Nonetheless, structured data is combined with semi-structured and unstructured data in numerous applications. DL models frequently combine different data modalities (i.e., text, image, tabular), employing embeddings as a common numerical representation for different data modalities [52,8,76]. Embedding vectors are not directly interpretable because they are generated through nonlinear mappings [23,51,75]. We employ cluster and latent factor analysis to tackle this challenge and to identify and describe the relationships between the embeddings, the objects they represent, and the predictions.

3.2 Pipeline from data generation to explanation

Figure 1 shows the five phases composing the pipeline.

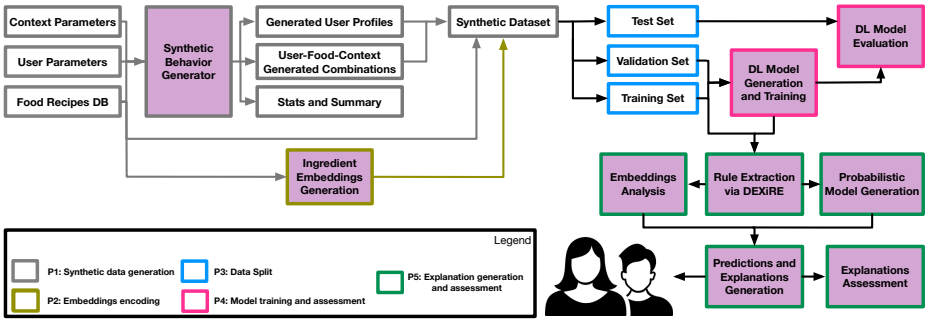

Fig. 1: Proposed pipeline. Input parameters, databases, and generated outcomes (Unfilled boxes); phases' processes (Solid-filled boxes).

P1: Synthetic data generation:

We have developed a Synthetic Behavior Generator (SBG) to preserve a controlled environment for experimentation and generate a large amount of data to train the DL models in a relatively short time[1]. The SBG's inputs are:

– **Context parameters** define the probability that a meal is consumed in a context characterized by the time of meal consumption, days to generate, place of meal consumption, and social situation of meal consumption. Table 1 describes the context parameters and the values employed to generate the experimental data used in this article.

[1] For example, the SBG can generate 50 users with a tracking of 180 days (36.5k meal tracked) in 190 seconds and 100 users with a tracking of 365 days (146k meals tracked) in 285 seconds.

Table 1: Context parameters for configuring synthetic data generator tool SBG.

Parameter	**Description**	**Selected Value**
Days to generate	Number of interaction days to simulate.	730
Meal time consumption	Probability distribution that determines the time when a user consumes a meal (0-24h).	Normal distribution for each meal based on CET, 1h of standard deviation.
Meal consumption	Probability that a user consumes a particular meal type (e.g., breakfast, snacks, lunch, dinner).	Breakfast 70%, morning snack 20%, lunch 90%, afternoon snack 40%, dinner 80%.
Place of meal consumption	Probability of a user consuming a meal in one of three locations: a restaurant, home, or outdoors.	restaurant 20%, home 70%, outdoor 10%.
social situation of meal consumption	Probability of a user consuming a meal in one of four social contexts: alone, with family/friends/colleagues.	alone 50%, with family 20%, with friends 20%, with colleagues 10%.
User's appreciation feedback (δ)	Probability determining if a given user has liked the recommended recipe.	Normal with a mean of 0.6 and standard deviation of 0.2.

- **User parameters** define the probability distributions employed to generate different user profiles. User parameters include total users, age probability, initial body mass index (BMI) probability, allergies probability, food restriction probabilities, and meal probabilities. Table 2 summarizes the user parameters and the values used to generate user profiles.
- **Food Recipes DB** is a recipe database that comprises 7000 recipes worldwide, obtained by querying large language models GPT-3.5-turbo-16k [57] and GPT-4-preview [58] from OpenAI. The query process has been performed in batches, each time targeting different recipe features to produce a diverse database that can meet different user preferences. Each recipe is defined by the following features: recipe ID, name, ingredients, preparation steps, calories, fat, fiber, proteins, carbohydrates, allergens, price, taste, and cultural factors. We are aware that such recipes might differ from reality. Nevertheless, they fulfill the purpose of training the initial model (harmless at this stage). Further studies will employ real-world datasets.

The SBG generates the following outcomes:

- **Generated user profiles:** This outcome contains the user profiles generated based on the user parameters. The user profile comprises the following features: age range, gender, nutritional goal, lifestyle, weight, height, initial BMI (BMI), final BMI, recommended daily calories, intake of daily calories, ethnicity, working status, and marital status.
- **User-Food-Context generated combinations:** This outcome consists of the combinations of user profile (U), recommended food recipe (F), and context of meal composition (C) data accompanied by the value of appreciation and user's feedback value δ continuous value between 0 and 1 describing the user's preference for the recommended recipe given his profile and food consumption context, where zero denotes strong dislike, and one indicates strong like.
- **Stats and Summary:** This outcome presents a visual overview of the transition probability employed in the simulation and a tabular summary of the simulation (HTML). Figure 2 is an example of a Markov chain that defines

Table 2: User parameters for configuring synthetic data generator tool, SBG.

Parameter	Description	Selected Value
Total users	Number of users to simulate	500
Age probability	Age distribution is segmented (18-100)	European preset for age distribution.
Sex	The probability of assigning a male or female sex.	Uniform distribution.
Initial user's BMI	BMI is the initial user's Body Mass Index categorized into four values according to CDC [78]: Underweight (< 18.5), Healthy weight (18.5–24.9), overweight (25.0–29.9), Obese (>= 30.0).	European preset extracted from Chooi et al. [14].
Allergies	The probability that a user will suffer from one or more of the ten most common allergic conditions.	European allergy prevalence distribution [56].
Cultural restrictions	The probability that a user follows cultural (e.g., vegetarian, vegan) or religious (e.g., halal, kosher) restrictions, flexible observant (e.g., flexi vegetarian, flexi halal), not restriction.	European preset.
Flexi observant	Flexible preferences indicate a strong preference for some food types. For example, a flexible vegan prefers vegan food but will eat other types of food (e.g., vegetarian) if the context requires it. This variable specifies the probability distributions for each user in the following categories: flexible vegan, flexible vegetarian, flexible halal, and flexible kosher.	Flexi-vegan (60% vegan, 20% vegetarian, 10% halal, Kosher 10%), Flexi-vegetarian (Vegetarian 60%, kosher 10%, halal 10%, No restriction 10%), Flexi-halal (vegetarian 30%, halal 60%, kosher 10%) and Flexi-kosher (vegetarian 20% halal 10%, kosher 60%).
BMI transition	The probability matrix determines the user's probability of changing from one BMI state to another during the simulation period.	Empirically defined probability matrix based on the worldwide BMI distributions. The probability matrix can be visualized in Figure 2.

the transition probabilities between different BMI states. This probabilistic model is used to simulate the users' progress toward their nutrition goals.

P2: Embeddings encoding:

The ingredients in recipes are presented in natural language (unstructured data) and need to be transformed into a numerical vector for integration into the processing pipeline. The following text embedding techniques have been selected to encode ingredients in dense numeric vectors (embedding vectors):

- **Word2vec:** Transform each word by considering its context (e.g., the surrounding words). Word2vec can be applied in two ways: a continuous bag of words (CBOW) or skip-gram [53]. The experiments conducted in this paper have employed both approaches. Each ingredient has its embedding. They are averaged to obtain the embedding per recipe.
- **Doc2vec:** This algorithm is an extension of Word2vec to produce one embedding vector per document (e.g., recipe) [50].
- **Universal Sentence Encoder (USE):** This transformer-based model has been trained in a large amount of textual data, and it encodes documents or words (e.g., recipes) into a semantic embedding [12].
- **Bidirectional Encoder Representations from Transformers (BERT):** It is a transformer-based language model (LM) trained in numerous text samples. BERT produces semantic embeddings per document (e.g., recipe) [27].

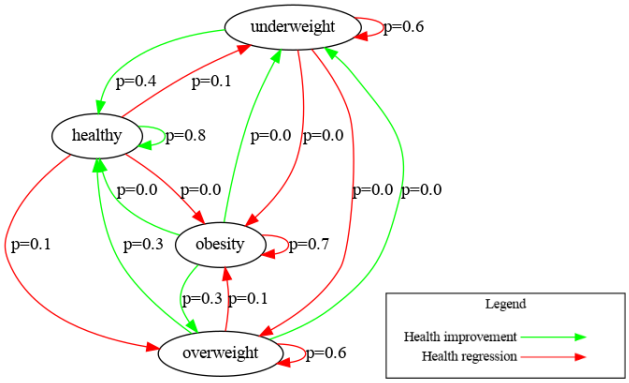

Fig. 2: A Markov chain illustrating the transition probability between different BMI states. The green arrows indicate a transition to a healthy status, while the red arrows indicate a deterioration in health status. The probability values are displayed over the edges. A zero probability value indicates that there is not a direct transition between these two states.

Before transforming into embedding, the recipes' ingredients have been pre-processed by applying the following NLP techniques: i) Text normalization, ii) Stop-word removal, ii) tokenization, and iv) Stemming and lemmatization.

Once the ingredients have been encoded in embeddings, they are merged with the **Generated user profiles** and **User-Food context combination** data (generated in phase P1) to produce the full **synthetic dataset**.

P3: Data split:

In this phase, the dataset is divided into three different groups: *Training set (60%)*, *Validation set (20%)*, and *Test set (20%)*.

P4: Model training and assessment:

This phase involves the following two processes:

- **DL model generation and training:** In this process, the DL model's architecture is defined, the model is compiled and trained on the training set, and evaluated on the validation set in each step to avoid overfitting. Two DL models (described below) have been developed for this study:
  - **Baseline model:** Its inputs are user profile (U) and food data (F) without considering the context and predicts the appreciation feedback. Its objective is to learn the user-food-appreciation relationship.
  - **Complete model:** Its inputs include user profile (U), food data (F), and food consumption context (C) and predicts appreciation feedback. Its objective is to learn the user-food-context-appreciation relationship.

The models mentioned above have been trained employing a 10-fold cross-validation. The evaluation metrics reported in the results section have been calculated on the test set.
- **DL model evaluation:** This process evaluates the DL model's performance on the test set according to the following set of metrics: i) *accuracy*, ii) *precision*, iii) *recall*, and iv) *F1-score*.

P5: Explanation generation and assessment:

In this phase, explanations are generated from the DL model trained in phase P4 employing the training set and the following explanation generation methods:

- **Embedding Analysis:** This process segments the different embeddings according to their similarity. Then, each segment's most probable ingredient distribution is found by employing cluster analysis techniques. The rule generation procedure uses the ingredient distribution and cluster membership to replace the ingredient's embedding.
- **Rule extraction via DEXiRE:** A rule set is extracted from the trained DL model using DEXiRE [17], which extracts rule sets from each hidden layer and combines into a final rule set employing the embedding analysis.
- **Probabilistic Model Generation:** The generated Rule and training set are combined to create a Bayesian Network (BN), which encodes the probabilistic and causal relationship between input features, rules, and predictions.

Finally, the predictions obtained from the set of rules and the probabilistic model are evaluated using the same metrics used in phase P4 and adding the following metrics:

- **Fidelity:** defined as the degree of similarity between the predictions generated by the DL model and those generated by its explainers.
- **Number of terms (Rule length):** Counts the number of atomic logic terms in each rule. It serves as a measure of the complexity of the rule set.

4 Results

This section presents the results of applying the explanation pipeline described in Section 3 to a set of DL models with different setups. First, the results from rule-based and PGM explanations are presented. In turn, performance metrics on baseline and complete models are introduced to contextualize the explanation results.

4.1 Explanation results

The pipeline described in Section 3 generates a rule-based explanation and a complementary Bayesian Network (BN) explainer. Both explainers integrate unstructured data through embedding transformation and cluster analysis. These

explanation methods are complementary and mutually related. The BN employs rule predictions and the training data set to build a probabilistic model that quantifies and bounds the uncertainty in data and rule-based explanations. Rule-based explanations are extracted from a trained DL predictor. Rule sets employ different features to describe the user's profile, the recommended recipe, and the context of food consumption. These features are generated by the SBG synthetic generator tool employing the configuration parameters described in Tables 1 and 2. The rule sets are not only used as an explanation method. They can also be used for inference and prediction, allowing an objective evaluation using performance metrics. To exemplify the predictive capacity of rule sets, a set of 150 users has been randomly selected alongside random recommendations. For the sake of space, only two of those are presented[2].

The rules that support their decision are shown in equations 1 and 2. Equation 1 shows the rule explaining a user acceptance. In particular, this is due to their vegan culture/diet, lack of allergen factors like gluten and wheat (the user's allergens), the recipe's taste profile is different from umami and sour, and the recipe apport less than 558 kcal (kilocalories), maximum 14 grams of protein, more than 6.9 grams of carbohydrates and 1.95 grams of fat.

$$\begin{aligned} &IF((cultural\ diet = vegan)\ AND\ (meal\ type \neq Not\ Information) \\ &AND\ (taste \neq sour)\ AND\ (fiber \leq 34.4)\ AND\ (allergens \neq gluten) \\ &AND\ (taste \neq umami)\ AND\ (carbohydrates > 6.9) \\ &AND\ (allergens \neq wheat)\ AND\ (fat > 1.95) \\ &AND\ (calories \leq 558)\ AND\ (protein \leq 14))\ THEN\ Like \end{aligned} \tag{1}$$

On the other hand, Equation 2 illustrates the rule supporting a rejection. In particular, it is because they follow a vegan culture/diet, the meal is categorized, the recipe does not contain gluten, the recipe's taste profile is umami or not sour, and the recipe apports less than 1033 kcal and less than 34.4 grams of fiber.

$$\begin{aligned} &IF\ ((cultural\ diet = vegan)\ AND\ (meal\ type \neq Not\ Information) \\ &AND\ (taste \neq sour)\ AND\ (fiber \leq 34.4) \\ &AND\ (calories \leq 1033)\ AND\ (allergens \neq gluten) \\ &AND\ (taste = umami))\ THEN\ Dislike \end{aligned} \tag{2}$$

The rules 1 and 2 explain the model's decision process based on the most informative features based on entropy metrics. However, these rules do not provide any information about the structural relationships between the different features (e.g., in the Rules 1 and 2), the calorie restrictions and nutritional values are strongly linked with the user's physical conditions (gender, age range, current BMI, lifestyle), user nutritional goals, daily nutritional requirements and

[2] Additional examples can be found in `https://github.com/aislab-hevs/pro_DEXiRE`

time of meal consumption. In contrast, features like taste strongly relate to the recipe's ingredients (encoded in embeddings and integrated into the explanation pipeline through cluster analysis). The ingredients are connected to variables like price, presence of allergens, and cultural diet. In addition, rule sets do not provide information about the data uncertainty or the probability of observing a particular combination of features in the data that support a decision. The BN-based explanation complements the rule-based explanation and quantifies the uncertainty in data, allowing probabilistic and evidential reasoning.

Figures 3 and 4 display the probabilistic explanation graph generated from the BN explainer. The probabilistic explanation graph presents the evidence (current features' values) in their peripheral nodes and the decision (like or dislike the recommendation) in their central node, middle nodes User, Context, and Recipe nodes group features according to the object they represent. Next to the edges are shown the likelihood values of observing that feature X_i take the value v_i, given the evidence and the decision. The Equation 3 employs the BN explainer's conditional probability distributions (CPD) to calculate the likelihood of features given the evidence and decision in an evidential reasoning process. One great advantage of the evidential reasoning process is its ability to be performed with incomplete and noisy evidence. Therefore, it does not require precise knowledge of all feature values.

$$P(X_i = v_i | evidence, decision) \tag{3}$$

Figure 3 presents the probabilistic explanation graph that supports the decision in Rule 1. This graph extends our understanding of the decision process, highlighting the link between the recipe's nutritional values (e.g., carbohydrates, protein, fat, and calories) and the user's physical condition, particularly BMI, nutritional goal, and gender. Finally, we observe a strong connection between the cluster membership of the recipe and the price because the cluster represents the recipe's ingredients, and the combination of the ingredients determines the recipe's price.

Figure 4 illustrates the probabilistic explanation graph that supports the decision in rule 2. Similar relationships to those found in Figure3, between the user's physical state and the recipe's nutritional values, are found in this graph, but with different interpretations given the evidence and decision. In this case, the recipe's nutritional values and the umami taste profile drive the decision. It is worth mentioning that even though the umami taste profile has a low likelihood ($\approx 2\%$), we know that it is a determining factor in the decision thanks to Rule 1. This example demonstrates the complementarity between these two explanation methods.

In addition, Table 3 summarizes the performance metrics obtained by the rule sets extracted from the completed models (i.e., DL models that consider user's features, food features, and context features) using DEXiRE. All the rule sets delivered strong performance and high fidelity, except for Doc2vec's rule set, which exhibited relatively lower performance. The following sections will analyze and discuss the reasons for this behavior.

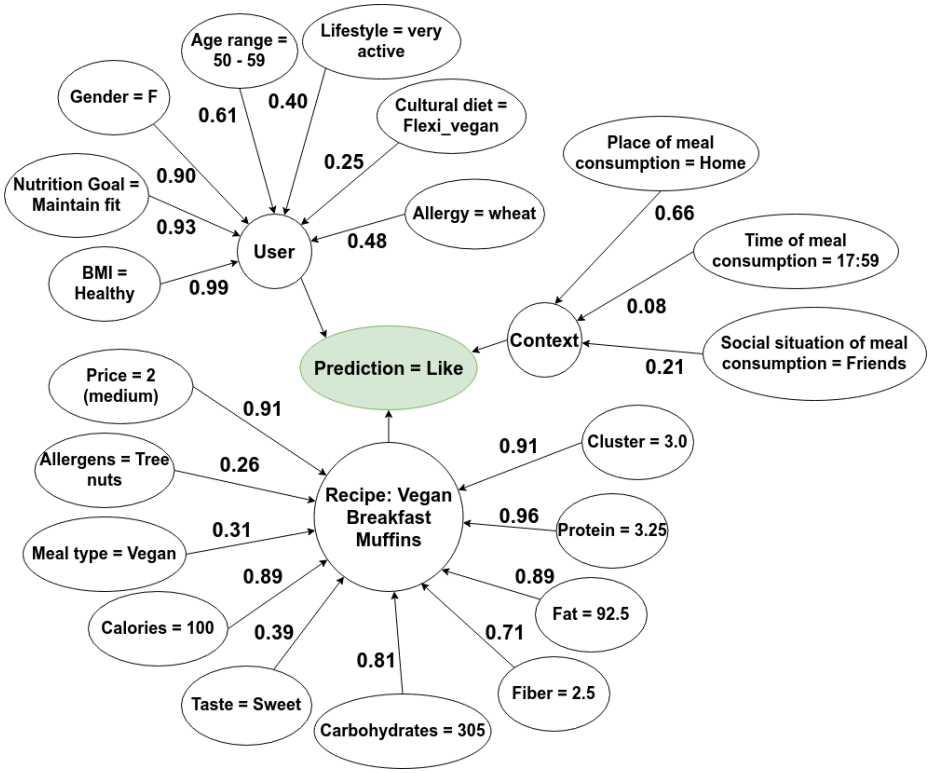

Fig. 3: The probabilistic explanation graph illustrates the evidential reasoning process, which complements Rule 1. The nodes show the features' current values. The central node in green displays the decision: *the user accepts the recommendation*. The numeric values in the edges are the likelihood of observing that feature value given evidence and the decision.

4.2 Baseline model performance metrics

This subsection presents the baseline model performance metrics to contextualize the explanation results.

Table 4 summarizes the results obtained by the baseline model from experimentation with different embedding encoding methods. According to the results reported in Table 4, the BERT outperforms the other methods in all metrics except recall, which is slightly surpassed by the universal sentence encoder (USE). It is worth noticing that the best-performing techniques are those based on transformers. This difference in performance arises because Transformer-based models can capture variable dependencies between tokens (e.g., words). In contrast, the dependencies learned by Word2Vec and Doc2Vec models are fixed and limited to the context window size (the number of adjacent words considered).

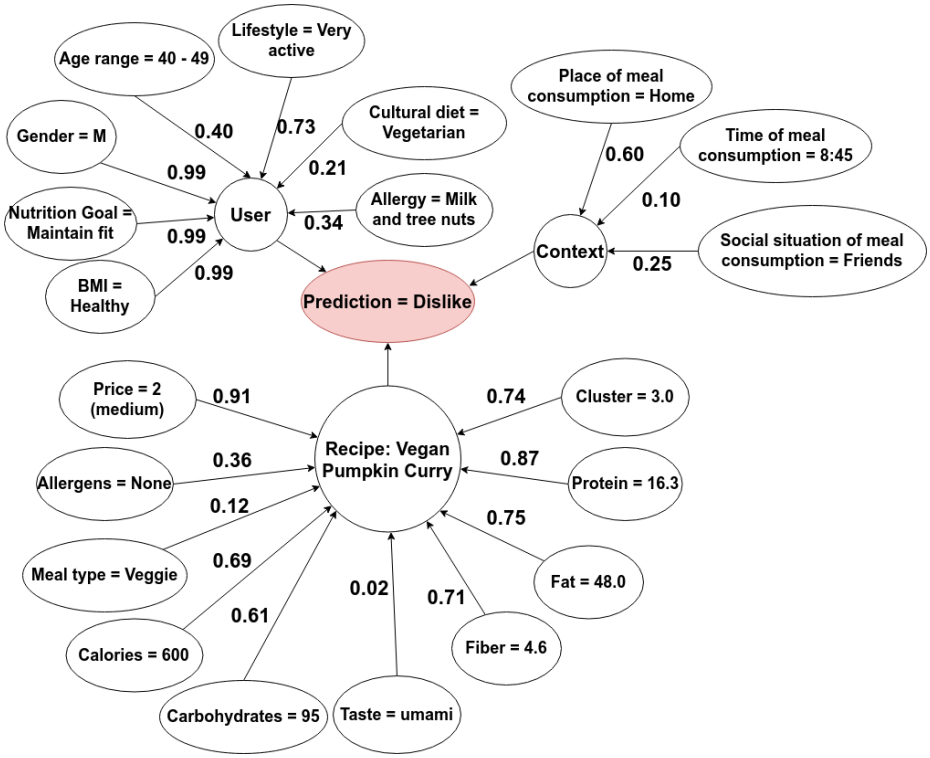

Fig. 4: The probabilistic explanation graph illustrates the evidential reasoning process, which complements Rule 2. The nodes show the features' current values. The central node in red displays the decision: *the user rejects the recommendation*. The numeric values in the edges are the likelihood of observing that feature value given evidence and the decision.

4.3 Complete model performance metrics

This subsection presents the complete model performance metrics to contextualize the explanation results. The completed model considers user profile (U), food-related features (F), and the food composition context (C).
Results reported in Table 5 indicate that adding the context features has slightly improved the performance of complete models compared to the baseline one. Results suggest a significant effect of food consumption context in modeling user preferences and behaviors in the nutritional domain.

5 Analysis and performance

This section examines the results and performance metrics discussed above. First, it addresses a statistical significance analysis of the impact of various

Table 3: The rule set performance outcomes for explaining DL models combining tabular data and textual data encoded with Word2vec (CBOW), Word2vec (Skip-gram), Doc2vec, USE, and BERT.

Rule set (RS)	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Rule length**	**Fidelity**
RS Model word2vec (CBOW)	0.986 ± 0.001	0.988 ± 0.006	0.990 ± 0.004	0.989 ± 0.001	50.7 ± 2.934	0.983 ± 0.001
RS Model word2vec (skip-gram)	0.978 ± 0.007	0.965 ± 0.010	1.0 ± 0.000	0.984 ± 0.005	41.7 ± 3.689	0.979 ± 0.001
RS Model doc2vec	0.873 ± 0.308	0.867 ± 0.306	0.8888 ± 0.314	0.877 ± 0.310	46.0 ± 16.438	0.874 ± 0.309
RS Model USE	0.9763 ± 0.007	0.9666 ± 0.009	1.0 ± 0.000	0.983 ± 0.005	40.9 ± 4.060	0.980 ± 0.001
RS Model BERT	0.985 ± 0.001	0.984 ± 0.007	0.994 ± 0.005	0.989 ± 0.001	52.7 ± 3.661	0.984 ± 0.001

Table 4: Baseline model results, combining users' profiles (U) with the recipe ingredients embedding algorithms (F), without food consumption context.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model word2vec (CBOW)	0.976 ± 0.005	0.969 ± 0.006	0.996 ± 0.005	0.982 ± 0.004
Model word2vec (Skip-gram)	0.975 ± 0.002	0.967 ± 0.005	0.997 ± 0.002	0.982 ± 0.002
Model doc2vec	0.973 ± 0.006	0.962 ± 0.008	0.998 ± 0.003	0.980 ± 0.004
Model USE	0.974 ± 0.003	0.963 ± 0.004	1.0 ± 0.000	0.981 ± 0.002
Model BERT	0.981 ± 0.003	0.974 ± 0.004	0.999 ± 0.001	0.986 ± 0.002

model configurations on their performance. Then, it examines embedding encoding methods and the influence of cluster membership on the explanation. Finally, this section concludes by analyzing the rule sets obtained as explanations.

5.1 Statistical significance analysis

A statistical significance test has been applied to measure the impact of employing different embedding methods and the inclusion or exclusion of context on the models' performance.

Figure 5 shows the p-values obtained by comparing different embedding encoding methods in the baseline model. The null hypothesis is defined as: *there are no significant differences in the baseline models' performances when different embedding encode methods are used.* The significant test results demonstrate that only BERT embedding has a statistically significant difference from the rest of the embedding techniques in the baseline model.

Figure 6 illustrates the p-values obtained by comparing different embedding encoding methods on the complete models' performance. The null hypothesis is as follows: *There are no significant differences in the complete models' performance*

Table 5: Complete model results, combining users' profiles (U), food consumption context (C), and different recipe ingredients embedding algorithms (F).

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model word2vec (CBOW)	0.990 ± 0.002	0.986 ± 0.003	1.0 ± 0.000	0.993 ± 0.001
Model word2vec (Skip-gram)	0.977 ± 0.004	0.968 ± 0.006	1.0 ± 0.000	0.983 ± 0.003
Model doc2vec	0.989 ± 0.004	0.983 ± 0.006	1.0 ± 0.000	0.991 ± 0.003
Model USE	0.976 ± 0.003	0.966 ± 0.005	1.0 ± 0.000	0.983 ± 0.002
Model BERT	0.992 ± 0.002	0.989 ± 0.003	1.0 ± 0.000	0.994 ± 0.001

when different embedding methods are applied. The tests indicate substantial differences between the USE, Doc2vec, CBOW, and BERT. However, there is no significant difference between the USE and Skip-gram embedding methods. Similarly, the BERT embedding method has shown significant differences with the other embedding methods, except CBOW. Finally, Doc2vec embedding has demonstrated significant differences with other methods, except for CBOW. The influence of context features addition can explain these differences.
Figure 7 presents the significance test results comparing the performance differences between the baseline and complete models. The null hypothesis in this test states that: There is no significant difference between baseline and complete models' performance. These results are crucial in understanding the impact of the context features on model performance. Significant differences have been found in BERT, CBOW, and Doc2vec models. For the case of USE embedding, there is only a significant difference in performance if we compare it with BERT on the baseline model. Finally, in the case of Skip-gram embedding, there is no significant difference between baseline and complete models. Statistically significant differences between embedding encode methods and between the baseline and the complete models support the hypothesis that the integration of unstructured data through embeddings and the integration of context features have a significant effect on the performance of the model and the explanations derived from them.

5.2 Embedding analysis

A cluster analysis has been performed to integrate text embeddings into the explanation pipeline. We have employed the elbow method (distortion score) and the silhouette score to determine the number of clusters centroids (K) using the KMeans clustering method. Figures 8 and 11 show the elbow and silhouette scores diagrams for USE and Doc2vec embeddings methods.
One point to note is the marked difference between Doc2vec and the rest of the embedding encoding algorithms. While the rest of the embedding encoding methods have shown an optimal cluster number between 4 and 6 clusters coinciding with taste profiles (see Figure 8), Doc2vec does not converge to an optimal K value (see Figures 10 and 11). Non-converging results indicate a mostly uniform distribution of objects in the embedding space, which implies that the embeddings are not accurately segmented and lack discriminative power and semantic meaning.

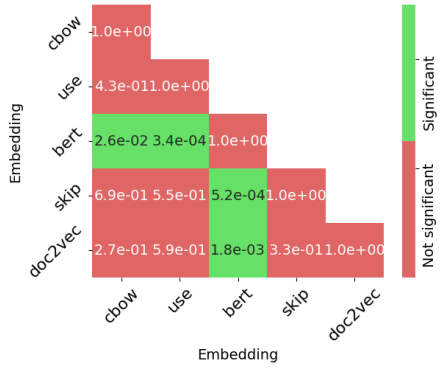

Fig. 5: Results of significance tests applied to measure the effect of different embedding on the baseline model's performance. In green are shown the cases where there is a significant difference in the results of the experiments. In red are shown the cases where there is no significant difference between the results of the experiments.

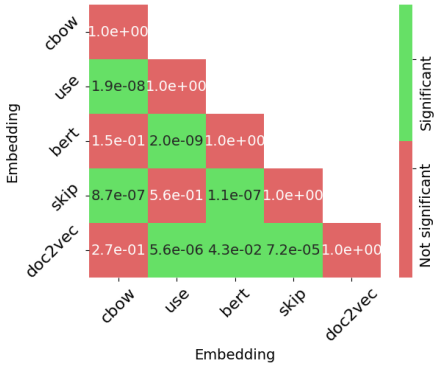

Fig. 6: Results of significance tests applied to measure the effect of different embedding on the complete model's performance. In green are shown the cases where there is a significant difference in the results of the experiments. In red are shown the cases where there is no significant difference between the results of the experiments.

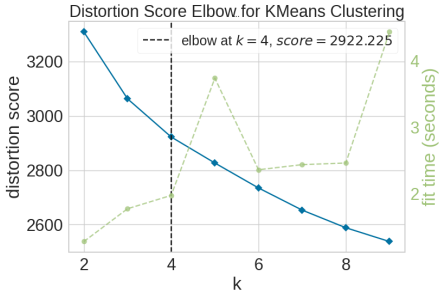

Fig. 8: Elbow method determinating the optimal number of clusters applied to USE embeddings. # of clusters (blue), execution time (green), optimal # of clusters (vertical dotted line).

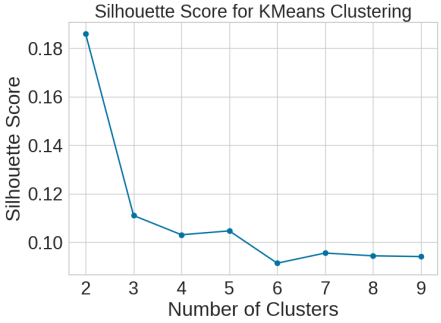

Fig. 9: Silhouette coefficient metric to determine the optimal # of clusters applied to USE embeddings.

5.3 Rule-based explanation analysis

For each embedding encoding method (e.g., word2vec (CBOW), word2vec (Skip-gram), Doc2vec, USE, BERT), different models have been trained and evaluated employing 10-fold cross-validation. A cluster analysis has been performed for

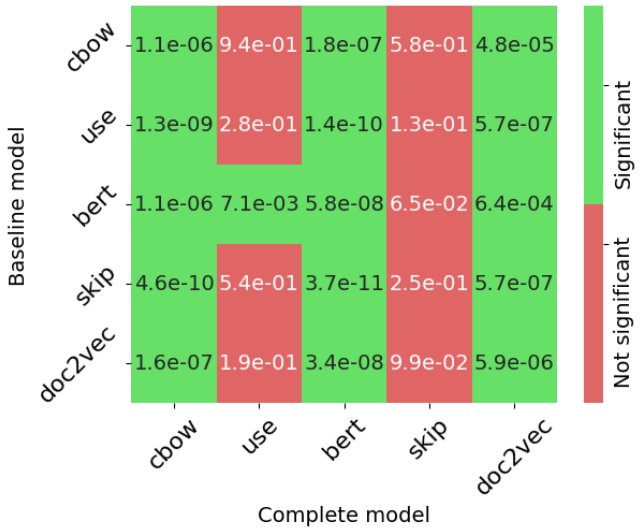

Fig. 7: Results of significance tests applied to measure the effect of context addition, comparing the baseline and complete models. In green are shown the cases where there is a significant difference in the results of the experiments. In red are shown the cases where there is no significant difference between the results of the experiments.

each embedding encode method. Then, a rule set has been extracted employing DEXiRE and cluster membership instead of the embedding vector.

Figures 12–16 illustrate the feature occurrence in rule sets for different embedding encode methods per target class. The most common feature among the rule sets is the user's allergies, a fundamental factor that cannot be neglected since it can endanger the user's life. The following most common features are the taste profile and the time of meal consumption, the recipe's calories, and the user's weight. In the middle of the top appear the recipe's nutritional values and the user's features (e.g., age range and cultural diet).

It is also worth highlighting the differences between classes (e.g., *dislike* and *like*). Based on these results, we can mark that even the rules for classes *like* and *dislike* share the most common features with different frequencies in all the embedding methods. The class *like* includes additional features in their analysis, particularly the day number, BMI, and recipe's cultural factor.
The top feature distribution is similar for all the embedding methods, with minor variations, particularly at the end of the top. The distribution of the top features for all the embedding encoding methods closely matches the generation process of the synthetic data tool SBG. The employ of synthetic data allows the verification of the rule extraction process by comparing the learned rules with the known and controlled data generation process. This verification confirms

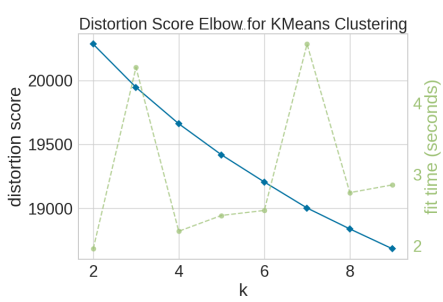

Fig. 10: Elbow method to determine the optimal number of clusters applied to Doc2vec embeddings. # of clusters (blue), execution time (green), optimal # of clusters (vertical dotted line).

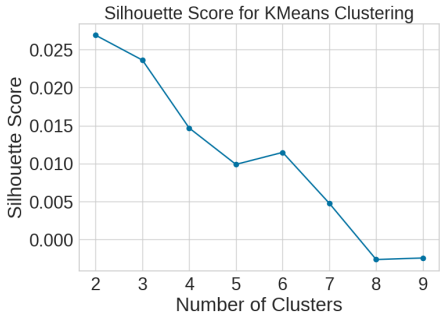

Fig. 11: Silhouette coefficient metric to determine the optimal number of clusters applied to Doc2vec embeddings. In blue silhouette score for different numbers of clusters.

that the rule sets can correctly capture the most significant variables that drive the decision.

6 Discussion

This section discusses the role of embeddings in the explanation pipeline. Then, we examine the role of Bayesian Networks (BN) as auxiliary explainers. We explore the limits of rule-based explanations and conclude the section by revising the advantages and limitations of using synthetic data in this study.

6.1 Embedding integration in the explanation pipeline

Table 4 reports the baseline model's performance results for embedding encoding methods Word2vec (CBOW and Skip-gram), Doc2vec, USE, and BERT. According to the reported results, baseline models, including BERT's embeddings, outperform the others with statistically significant differences, as is shown in Figure 5. The performance difference between these two types of models can be attributed to the following reasons: (i) the number and variety of training samples and (ii) model architecture. Concerning reason (i), the BERT model has been trained in a large amount of data extracted from the internet covering different domains. In contrast, Word2vec and Doc2vec have been trained on the recipe database employed to generate data in phase P1. Concerning reason (ii), BERT is a transformer-based model that considers a broader dynamic context (i.e., surrounding text). At the same time, Word2vec and Doc2vec are based on shallow architectures with a fixed context window (set to 4 for this paper).

Table 5 reports the performance measure of complete DL models employing different embedding encoding methods and the context of meal consumption

data. Unlike the baseline model results, the performance gap between transformer-based and shallow embedding encode methods has been reduced. Because the only difference between experiments on the baseline model and the complete one is the addition of the context, we can infer that this change is due to the influence of context and not to embedding representations since those ones remain consistent across both experiments.

Regarding the effect of embedding representation on DL models' performance based on the results reported in Section 4, we can infer that the impact of different embedding representations is moderate and reduced as more features and information are aggregated to the model. Extending this analysis, we can imply that DL models employing heterogeneous data (i.e., multimodal DL models) are more robust because the dependence on a single set of features is reduced.

Embedding vectors are integrated into the explanation pipeline through cluster analysis, replacing the embedding vector with a cluster membership. Rule sets and BN generated in phase P5 integrate the recipe's ingredients (text data) into the explanation by replacing the embedding vector with a cluster membership. The results obtained indicate that cluster memberships are strongly related to the recipe's price, taste profile, and allergens. These results can be explained because the cluster analysis captures the semantic relationship between ingredients and those strongly influencing the price, taste profile, and allergens. To conclude this discussion, it is necessary to mention that cluster analysis cannot always capture the semantic relationships encoded into embeddings. This limitation could be avoided if the embedding vectors could be automatically translated into domain concepts. However, this is still an open research topic.

6.2 Probabilistic explanations

Uncertainty is inherent to data. Thus, any data-driven explanation method must consider the data uncertainty as an essential factor. The proposed pipeline in phase P5 combines a Bayesian Network (BN) with the extracted rule set to quantify the uncertainty in data and produce probabilistic explanations. BN learns the relationship between the rule set, predictions, and input features. In particular, the BN models causal relationships and conditional probability distributions (CPDs).

Figures 3 and 4 have been generated employing evidential reasoning, calculating the conditional probability to observe a particular feature's value given evidence (i.e., other features' values) and the decision (i.e., the user accepts or rejects the recommendation). Evidential reasoning is a complementary and flexible method that supports a given decision based on data observation and Bayesian statistics.

Despite its clear advantages, BN explainers present some limitations related to scalability, particularly memory scalability. CPDs are tables stored in memory and grow exponentially with the nodes' indegree, the number of possible values in a discrete distribution, and the network structure's complexity. This condition restrings its application. The BN in this pipeline is employed as an additional explainer to complement the rule-based explanations by quantifying the uncertainty in data and enabling reasoning under noise and missing data.

One fascinating aspect of BNs is their ability to model causal relationships, resulting in causal explanations. However, it is crucial to note that this does not necessarily imply that the explained model considers causality. Instead, causal modeling is specific to the explanation constructed based on the hypothesis of causality between the input features, rule set, and prediction. However, the actual causal relationships are domain-dependent, and only domain experts' knowledge can define them. Although BN models can use structuring learning to learn their own structure, there is no guarantee that structure makes sense in the application domain and reflects a real causal phenomenon.

6.3 Limits on rule-based explanations

Based on the results obtained from the Doc2vec rule set, as presented in Table 5 and figures 11 and 10, we can infer that Doc2vec embeddings for this case are almost uniformly distributed in the embedding space, for this reason, it is not feasible to extract distinctive patterns and integrate them with the rule sets. Although rule sets can still be extracted, those exhibit worse performance and less reliability. One of the limitations of rule-based explanations is their difficulty in integrating unstructured data, such as text or images, because of the semantic gap and their wide range of variation.

Rule extraction methods like DEXiRE [19] and Eclaire [81] induce rules through distinctive pattern identification in the DL model's hidden layers and input features. In this case, DEXiRE could identify distinct patterns inside the hidden layers but could not obtain a distinctive pattern from Doc2vec embedding, producing degraded rule sets with biased predictions.

An additional limitation in explaining DL models through rule sets involves the rules sharply partitioning the decision space, in contrast to DL models, whose decision function exhibit smooth and curved surfaces. This difference limits the rules that can be extracted from a DL model, especially in cases where the decision surface is too curved, and the relationship with the input features is nonlinear. In this scenario, the extracted rule sets could have various failures, such as empty or contradictory rules, low quality, biased rule sets, or over-specificity (low coverage) rules.

In this work, we have proposed employing an ensemble of explainers (e.g., cluster membership, rule-set, and PGM) to reduce the impact of the abovementioned limitations. In this manner, if some explainers fail to cover all cases, approximate explanations can still be obtained through majority voting, first-hit, or high-performance strategies to break ties between explainers. The proposed pipeline and hypothesis H1 were formulated to overcome this limitation.

6.4 Synthetic data advantages and limitations in XAI applications

Real-world data is not always available due to time, cost, technical limitations, privacy preservation requirements, infrequent events, and ethical or legal restrictions. For these reasons, synthetic datasets are becoming increasingly common in machine learning and XAI experimentation and research [40].

On the one hand, using synthetic data for experimentation on ML and XAI offers several advantages, including a controlled environment where the underlying data generation process is known, which enables the possibility of evaluating the model performance and explanations generated qualitatively, allowing the understanding of whether the explanations are based on the actual underlying factors or not.

On the other hand, synthetic data is generated from a simplified model of the real-world scenario. The model proposed in this paper captures the user's main biological and socio-cultural features. However, the model does not capture other variables that affect users' decision processes, such as stress, political positions, personal preferences, and psychological states. These factors, among others, could not be modeled due to their complexity, unobservability, and the unavailability of data that permits their accurate estimation and integration in simulation models.

Since each user has hidden preferences and psychological states, we have employed synthetic data to overcome the cold-start problem. When feedback from the user has been collected, this information will be used to update and adapt the model preferences to the particular user.

7 Conclusions and Future Work

This paper presented a pipeline for explaining DL models employing embedding analysis, rule sets, and probabilistic graphical models. The proposed pipeline has been developed with the following hypothesis: *"Unstructured data can be integrated into a rule-based explanation pipeline through embedding analysis (RT1). The extracted rule set can be enhanced through probabilistic modeling (RT2), which considers uncertainty in data and models"*. The proposed pipeline has been evaluated using a synthetic dataset generated from software developed in-house. Elaborating on the obtained results and analysis, we can summarize the following statements:

- The proposed pipeline combines embedding analysis, rule extraction methods, and probabilistic graphical models. To produce robust, flexible explanations that consider uncertainty and combine structured and unstructured data, satisfying challenges C1 and C2.
- The embedding representation selected and the posterior cluster analysis significantly impact the DL model and explanation's performance and quality, affecting the model, rule-set generation, and posterior probabilistic models.
- Limitations of rule-based explanations (see Section 6.3) can be mitigated by combining different explainer models, such as probabilistic graphical models, in an ensemble of complementary explainers.
- Rule-based and probabilistic-based explanations are complementary XAI methods that together improve overall performance and reduce their individual limitations. Ensemble XAI methods are more flexible and robust.
- To integrate structured and unstructured data into the explanation pipeline, it is necessary to translate the embedding vector into a cluster membership.

Then, the cluster membership is integrated into the explanation pipeline, producing a set of rules and a Bayesian network that generates explanations for heterogeneous data models (e.g., text and tabular data).
– The use of synthetic datasets for XAI presents several advantages, particularly related to evaluating the explanations qualitatively. Synthetic data is generated based on known patterns that can be verified in the resulting explanations. However, synthetic datasets may not account for all the concealed factors interacting in real-world scenarios. In these cases, the uncertainty modeling takes special significance, making the explanation pipeline robust, flexible, and able to perform well in new contexts.

Finally, we envision the following future works:

– Generate natural language explanations from embedding analysis, probabilistic graphical models, and rule sets.
– Personalize and update the DL model for the user based on their feedback. Generate explanations from the updated model and validate it with the user using reinforcement learning from human feedback (RLHF) and lifelong learning techniques.

Acknowledgments

This work is supported by the Chist-Era grant CHIST-ERA19-XAI-005, and by the Swiss National Science Foundation (G.A. 20CH21_195530).

References

1. Abid, A., Yuksekgonul, M., Zou, J.: Meaningfully debugging model mistakes using conceptual counterfactual explanations. In: International Conference on Machine Learning. pp. 66–88. PMLR (2022)
2. Ables, J., Childers, N., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I., Seale, M.: Eclectic rule extraction for explainability of deep neural network based intrusion detection systems. arXiv preprint arXiv:2401.10207 (2024)
3. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for boltzmann machines. Cognitive science **9**(1), 147–169 (1985)
4. Al-Najjar, H.A., Pradhan, B., Beydoun, G., Sarkar, R., Park, H.J., Alamri, A.: A novel method using explainable artificial intelligence (xai)-based shapley additive explanations for spatial landslide prediction using time-series sar dataset. Gondwana Research **123**, 107–124 (2023)
5. Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J.M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., Herrera, F.: Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence. Information fusion **99**, 101805 (2023)
6. Barbado, A., Corcho, Ó., Benjamins, R.: Rule extraction in unsupervised anomaly detection for model explainability: Application to oneclass svm. Expert Systems with Applications **189**, 116100 (2022)

7. Blanco-Justicia, A., Domingo-Ferrer, J.: Machine learning explainability through comprehensible decision trees. In: Machine Learning and Knowledge Extraction: Third IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2019, Canterbury, UK, August 26–29, 2019, Proceedings 3. pp. 15–26. Springer (2019)
8. Boehm, K.M., Khosravi, P., Vanguri, R., Gao, J., Shah, S.P.: Harnessing multimodal data integration to advance precision oncology. Nature Reviews Cancer **22**(2), 114–126 (2022)
9. Bueff, A., Papantonis, I., Simkute, A., Belle, V.: Explainability in machine learning: a pedagogical perspective. arXiv preprint arXiv:2202.10335 (2022)
10. Burkart, N., Huber, M.F.: A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research **70**, 245–317 (2021)
11. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and sub-symbolic techniques for xai: A survey. Intelligenza Artificiale **14**(1), 7–32 (2020)
12. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
13. Chen, D., Zhao, H., He, J., Pan, Q., Zhao, W.: An causal xai diagnostic model for breast cancer based on mammography reports. In: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 3341–3349. IEEE (2021)
14. Chooi, Y.C., Ding, C., Magkos, F.: The epidemiology of obesity. Metabolism **92**, 6–10 (2019)
15. Church, K.W.: Word2vec. Natural Language Engineering **23**(1), 155–162 (2017)
16. Confalonieri, R., Weyde, T., Besold, T.R., Moscoso del Prado Martín, F.: Trepan reloaded: A knowledge-driven approach to explaining artificial neural networks (2020)
17. Contreras, V., Aydogan, R., Najjar, A., Calvaresi, D.: On explainable negotiations via argumentation. In: Proceedings of BNAIC/BeneLearn 2021: 33rd Benelux Conference on Artificial Intelligence and 30th Belgian-Dutch Conference on Machine Learning (2021)
18. Contreras, V., Bagante, A., Marini, N., Schumacher, M., Andrearczyk, V., Calvaresi, D.: Explanation generation via decompositional rules extraction for head and neck cancer classification. In: Explainable and Transparent AI and Multi-Agent Systems: 5th International Workshop, EXTRAAMAS 2023, London, UK, May 29, 2023, Revised Selected Papers. vol. 14127, p. 187. Springer Nature (2023)
19. Contreras, V., Marini, N., Fanda, L., Manzo, G., Mualla, Y., Calbimonte, J.P., Schumacher, M., Calvaresi, D.: A dexire for extracting propositional rules from neural networks via binarization. Electronics **11**(24) (2022). https://doi.org/10.3390/electronics11244171, `https://www.mdpi.com/2079-9292/11/24/4171`
20. Crnomarkovic, I., Ilakovac, M., Kercmar, R.: How much implicit knowledge is there in deep learning models? Text Analysis and Retrieval 2019 Course Project Reports p. 14
21. Cui, P., Liu, S., Zhu, W.: General knowledge embedded image representation learning. IEEE Transactions on Multimedia **20**(1), 198–207 (2017)
22. Dai, B., Shen, X., Wang, J.: Embedding learning. Journal of the American Statistical Association **117**(537), 307–319 (2022)
23. Dallago, C., Schütze, K., Heinzinger, M., Olenyi, T., Littmann, M., Lu, A.X., Yang, K.K., Min, S., Yoon, S., Morton, J.T., et al.: Learned embeddings from deep learning to visualize and predict protein sets. Current Protocols **1**(5), e113 (2021)

24. De Raedt, L., Dries, A., Thon, I., Van den Broeck, G., Verbeke, M.: Inducing probabilistic relational rules from probabilistic examples. In: Proceedings of 24th international joint conference on artificial intelligence (IJCAI). vol. 2015, pp. 1835–1842. IJCAI-INT JOINT CONF ARTIF INTELL (2015)
25. Deng, Y.: Recommender systems based on graph embedding techniques: A review. IEEE Access **10**, 51587–51633 (2022)
26. Derks, I.P., De Waal, A.: A taxonomy of explainable bayesian networks. In: Artificial Intelligence Research: First Southern African Conference for AI Research, SACAIR 2020, Muldersdrift, South Africa, February 22-26, 2021, Proceedings 1. pp. 220–235. Springer (2020)
27. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
28. Dieber, J., Kirrane, S.: Why model why? assessing the strengths and limitations of lime. arXiv preprint arXiv:2012.00093 (2020)
29. Dikopoulou, Z., Moustakidis, S., Karlsson, P.: Glime: A new graphical methodology for interpretable model-agnostic explanations. arXiv:2107.09927 (2021)
30. Eddy, S.R.: Hidden markov models. Current opinion in structural biology **6**(3), 361–365 (1996)
31. Fel, T., Ducoffe, M., Vigouroux, D., Cadène, R., Capelle, M., Nicodème, C., Serre, T.: Don't lie to me! robust and efficient explainability with verified perturbation analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16153–16163 (2023)
32. Främling, K.: Decision theory meets explainable ai. In: International workshop on explainable, transparent autonomous agents and multi-agent systems. pp. 57–74. Springer (2020)
33. Graziani, M., Dutkiewicz, L., Calvaresi, D., Amorim, J.P., Yordanova, K., Vered, M., Nair, R., Abreu, P.H., Blanke, T., Pulignano, V., et al.: A global taxonomy of interpretable ai: unifying the terminology for the technical and social sciences. Artificial intelligence review **56**(4), 3473–3504 (2023)
34. Hakkoum, H., Idri, A., Abnane, I.: Global and local interpretability techniques of supervised machine learning black box models for numerical medical data. Engineering Applications of Artificial Intelligence **131**, 107829 (2024)
35. Hassan, A., Sulaiman, R., Abdulgabber, M., Kahtan, H.: Towards user-centric explanations for explainable models: A review. Journal of Information System and Technology Management **6**(22), 36–50 (2021)
36. Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., Hussain, A.: Interpreting black-box models: a review on explainable artificial intelligence. Cognitive Computation (2024)
37. Hruschka, E.R., Ebecken, N.F.F.: Rule extraction from neural networks: modified rx algorithm. In: IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339). vol. 4, pp. 2504–2508. IEEE (1999)
38. Huang, M., Haralick, R.M.: A probabilistic graphical model for recognizing np chunks in texts. In: Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy: ICCPOL 2009, Hong Kong, March 26-27, 2009. Proceedings 22. pp. 23–33. Springer (2009)
39. Huang, S.H., Xing, H.: Extract intelligible and concise fuzzy rules from neural networks. Fuzzy Sets and Systems **132**(2), 233–243 (2002)
40. IQBAL, A., Sikdar, B.: Are classifiers trained on synthetic data reliable? an xai study. Authorea Preprints (2023)
41. Izenman, A.J.: Introduction to manifold learning. Wiley Interdisciplinary Reviews: Computational Statistics **4**(5), 439–446 (2012)

42. Jensen, F.V., Jensen, F.V.: Causal and bayesian networks. Bayesian networks and decision graphs pp. 3–34 (2001)
43. Ji, Q.: Probabilistic Graphical Models for Computer Vision. Academic Press (2019)
44. Judd, K., Mees, A.: Embedding as a modeling problem. Physica D: Nonlinear Phenomena **120**(3-4), 273–286 (1998)
45. Karvelis, P., Gavrilis, D., Georgoulas, G., Stylios, C.: Topic recommendation using doc2vec. In: 2018 International Joint Conference on Neural Networks (IJCNN). pp. 1–6. IEEE (2018)
46. Kenter, T., Borisov, A., De Rijke, M.: Siamese cbow: Optimizing word embeddings for sentence representations. arXiv preprint arXiv:1606.04640 (2016)
47. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)
48. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
49. Krakowczyk, D., Reich, D.R., Prasse, P., Lapuschkin, S., Jäger, L.A., Scheffer, T.: Selection of xai methods matters: Evaluation of feature attribution methods for oculomotoric biometric identification. In: Annual Conference on Neural Information Processing Systems. pp. 66–97. PMLR (2023)
50. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. arXiv preprint arXiv:1405.4053 (2014)
51. Lebret, R., Legrand, J., Collobert, R.: Is deep learning really necessary for word embeddings? (2013)
52. Lipkova, J., Chen, R.J., Chen, B., Lu, M.Y., Barbieri, M., Shao, D., Vaidya, A.J., Chen, C., Zhuang, L., Williamson, D.F., et al.: Artificial intelligence for multimodal data integration in oncology. Cancer cell **40**(10), 1095–1110 (2022)
53. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations (ICLR) (2013)
54. Mitra, B., Craswell, N.: Neural text embeddings for information retrieval. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 813–814 (2017)
55. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
56. Nwaru, B., Hickstein, L., Panesar, S., Roberts, G., Muraro, A., Sheikh, A., Allergy, E.F., Group, A.G.: Prevalence of common food allergies in europe: a systematic review and meta-analysis. Allergy **69**(8), 992–1007 (2014)
57. OpenAI: GPT-3.5 Language Model. `https://openai.com/gpt-3.5` (2022), accessed: Month Day, Year
58. OpenAI: GPT-4 Language Model. `https://openai.com/gpt-4` (2023), accessed: Month Day, Year
59. Patil, R., Boit, S., Gudivada, V., Nandigam, J.: A survey of text representation and embedding techniques in nlp. IEEE Access (2023)
60. Pearl, J.: The do-calculus revisited. arXiv preprint arXiv:1210.4852 (2012)
61. Qi, Z., Khorram, S., Li, F.: Visualizing deep networks by optimizing with integrated gradients. In: CVPR Workshops. vol. 2, pp. 1–4 (2019)
62. Sabbatini, F., Ciatto, G., Calegari, R., Omicini, A.: On the design of psyke: A platform for symbolic knowledge extraction. In: WOA. pp. 29–48 (2021)
63. Setiono, R., Leow, W.K.: Fernn: An algorithm for fast extraction of rules from neural networks. Applied Intelligence **12**, 15–25 (2000)
64. Sivaprasad, A., Reiter, E., Tintarev, N., Oren, N.: Evaluation of human-understandability of global model explanations using decision tree. In: European Conference on Artificial Intelligence. pp. 43–65. Springer (2023)

65. So, C.: Understanding the prediction mechanism of sentiments by xai visualization. In: Proceedings of the 4th international conference on natural language processing and information retrieval. pp. 75–80 (2020)
66. Sorzano, C.O.S., Vargas, J., Montano, A.P.: A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877 (2014)
67. Sutton, C., McCallum, A., et al.: An introduction to conditional random fields. Foundations and Trends® in Machine Learning **4**(4), 267–373 (2012)
68. Talmor, A., Yoran, O., Catav, A., Lahav, D., Wang, Y., Asai, A., Ilharco, G., Hajishirzi, H., Berant, J.: Multimodalqa: Complex question answering over text, tables and images. arXiv preprint arXiv:2104.06039 (2021)
69. Tritscher, J., Krause, A., Hotho, A.: Feature relevance xai in anomaly detection: Reviewing approaches and challenges. Frontiers in Artificial Intelligence **6**, 1099521 (2023)
70. Tversky, A., Kahneman, D.: Probabilistic reasoning. Readings in philosophy and cognitive science pp. 43–68 (1993)
71. Vale, D., El-Sharif, A., Ali, M.: Explainable artificial intelligence (xai) post-hoc explainability methods: Risks and limitations in non-discrimination law. AI and Ethics **2**(4), 815–826 (2022)
72. Valle-Perez, G., Camargo, C.Q., Louis, A.A.: Deep learning generalizes because the parameter-function map is biased towards simple functions. arXiv preprint arXiv:1805.08522 (2018)
73. Vu, M., Thai, M.T.: Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems **33**, 12225–12235 (2020)
74. van der Waa, J., Nieuwburg, E., Cremers, A., Neerincx, M.: Evaluating xai: A comparison of rule-based and example-based explanations. Artificial Intelligence **291**, 103404 (2021)
75. Wang, S., Zhou, W., Jiang, C.: A survey of word embeddings based on deep learning. Computing **102**(3), 717–740 (2020)
76. Waqas, A., Tripathi, A., Ramachandran, R.P., Stewart, P., Rasool, G.: Multimodal data integration for oncology in the era of deep neural networks: a review. arXiv preprint arXiv:2303.06471 (2023)
77. Weber, L., Lapuschkin, S., Binder, A., Samek, W.: Beyond explaining: Opportunities and challenges of xai-based model improvement. Information Fusion **92**, 154–176 (2023)
78. Weight, H.: About adult bmi. Centre for Disease Control and Prevention. Last updated on (2015)
79. Yang, J.B., Xu, D.L.: Evidential reasoning rule for evidence combination. Artificial Intelligence **205**, 1–29 (2013)
80. Yu, N., Hu, X., Song, B., Yang, J., Zhang, J.: Topic-oriented image captioning based on order-embedding. IEEE Transactions on Image Processing **28**(6), 2743–2754 (2018)
81. Zarlenga, M.E., Shams, Z., Jamnik, M.: Efficient decompositional rule extraction for deep neural networks. arXiv preprint arXiv:2111.12628 (2021)
82. Zhu, X., Wang, D., Pedrycz, W., Li, Z.: Fuzzy rule-based local surrogate models for black-box model explanation. IEEE Transactions on Fuzzy Systems (2022)
83. Zilke, J.R., Loza Mencía, E., Janssen, F.: Deepred–rule extraction from deep neural networks. In: Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016, Proceedings 19. pp. 457–473. Springer (2016)

A Most Recurring Features in Rule Sets.

Figures 12-16 summarize the most common features in the rule sets.

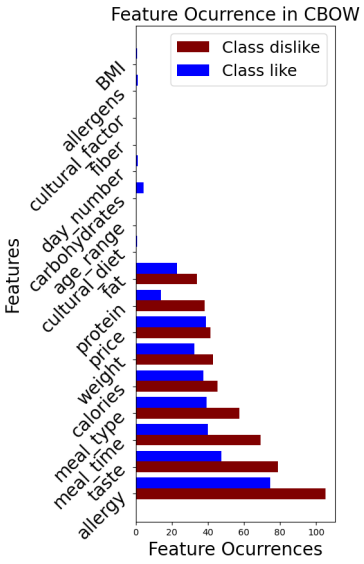

Fig. 12: Top features (CBOW).

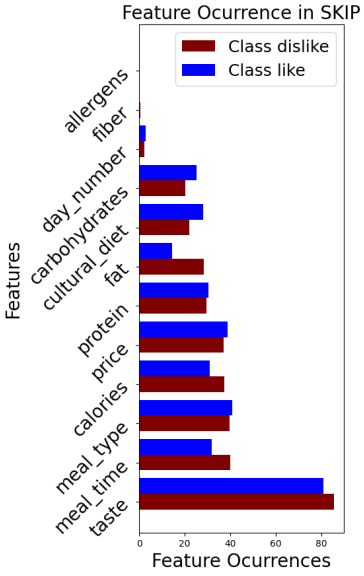

Fig. 13: Top features (Skip-gram).

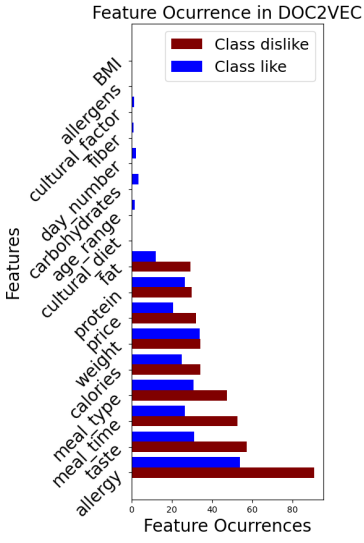

Fig. 14: Top features (Doc2vec).

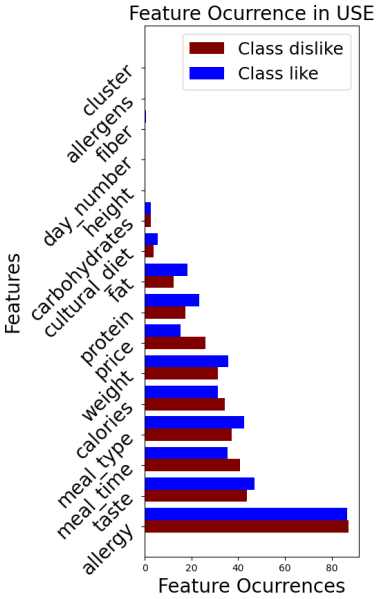

Fig. 15: Top features (USE).

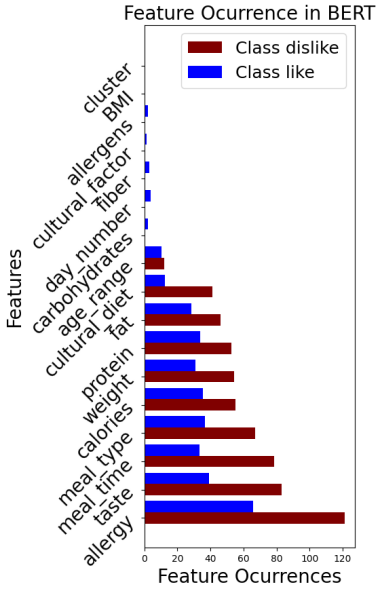

Fig. 16: Top features (BERT).