UNIVERSIDADE FEDERAL DE MINAS GERAIS

Instituto de Ciências Exatas

Programa de Pós-graduação em Ciência da Computação

Anderson Bessa da Costa

ENSEMBLE LEARNING BY DIVERSIFYING EXPLANATIONS: predicting the evolution of pain relief

Belo Horizonte

2021

Anderson Bessa da Costa

ENSEMBLE LEARNING BY DIVERSIFYING EXPLANATIONS: predicting the evolution of pain relief

Versão final

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Adriano Alonso Veloso

Coorientador: Prof. Dr. Nivio Ziviani

Belo Horizonte
2021

Anderson Bessa da Costa

ENSEMBLE LEARNING BY DIVERSIFYING EXPLANATIONS: predicting the evolution of pain relief

Final version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Adriano Alonso Veloso

Coadvisor: Nivio Ziviani

Belo Horizonte
2021

Costa, Anderson Bessa da.

C837e Ensemble learning by diversifying explanations: predicting the evolution of pain relief [manuscrito] / Anderson Bessa da Costa — 2021.
125 f.

Orientador: Adriano Alonso Veloso.
Coorientador: Nivio Ziviani
Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f.108-115.

1. Computação – Teses. 2 Controle preditivo – Teses. 3. Aprendizado de múltiplas instâncias – Teses. 4. Aprendizado conjunto (aprendizado de máquina) – Teses.I. Veloso, Adriano Alonso. II. Ziviani, Nívio. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas,Departamento de Ciência da Computação. III Título.

DU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6/1510 – Insituto de Ciências Exatas da UFMG.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Ensemble Learning by Diversifying Explanations: Predicting the Evolution of Pain Relief

ANDERSON BESSA DA COSTA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Adriano Alonso Veloso
Prof. Adriano Alonso Veloso - Orientador
Departamento de Ciência da Computação - UFMG

Nivio Ziviani
Prof. Nivio Ziviani - Coorientador
Departamento de Ciência da Computação - UFMG

Wagner meira junior
Prof. Wagner Meira Júnior
Departamento de Ciência da Computação - UFMG

Leandro Balby Marinho
Prof. Leandro Balby Marinho
Departamento de Sistemas e Computação - UFCG

Prof. Marco Antônio Pinheiro de Cristo
Instituto de Computação - UFAM

Prof. Daniel Ciampi Araújo de Andrade
Faculdade de Medicina - USP

Belo Horizonte, 20 de Dezembro de 2021.

Acknowledgments

I am deeply grateful to my advisor Professor Adriano Alonso Veloso, who gave me the golden opportunity to be part of this project. I would like to thank for your invaluable advice, continuous support, and patient during the course of this thesis. I am also grateful to my co-advisor Professor Nivio Ziviani.

I would like to thank my friends, lab mates, and colleagues for a cherished time spent together in the lab and in social settings. It is their kind help and support that have made my study and life at the UFMG a wonderful time. Special thanks to my friend Alberto Ueda for the warm welcome to LATIN. I also wish to thank my friend Amir Jalilifard for the great conversations. Thanks also to Silvana Morita, who provided me with encouragement throughout my studies. My fear of forgetting someone prevents me from listing them all. I very much appreciate all of you.

Lastly, I also appreciate all the support I received from my family.

“Beauty in things exists in the mind which contemplates them.”

(David Hume)

Resumo

A modelagem a partir de dados geralmente tem duas facetas distintas: construir modelos explicativos sólidos ou criar modelos preditivos poderosos para um sistema ou fenômeno. Embora exista um senso instintivo de que prever e explicar são tarefas distintas, muitas vezes se assume que modelos com alto poder explicativo são inerentemente de alto poder preditivo. Apesar desta relação, os mais recentes trabalhos de modelagem de dados se encaixam na metodologia de aprendizagem de máquina *tudo-em-um*, com a suposição básica de que todos os fatores explicativos importantes podem ser combinados em um único modelo preditivo. Embora altamente adotada e estabelecida, a metodologia *tudo-em-um* negligencia que muitos fenômenos são realmente definidos por várias subpopulações ou estruturas locais e, portanto, há muitos modelos de previsão possíveis que fornecem interpretações contrastantes ou explicações concorrentes para o mesmo fenômeno. Neste trabalho, apresentamos o ED-Ensemble (*Explanation-Diversifying Ensemble*), uma alternativa à metodologia *tudo-em-um*. Nossa principal intuição é que os modelos que têm suas decisões explicadas pelos mesmos fatores provavelmente farão melhores previsões dentro das mesmas estruturas locais. O ED-Ensemble obtido a partir de nossos experimentos superou consistentemente as abordagens *tudo-em-um*, mesmo empregando os algoritmos de ensemble de última geração XGBoost e Random Forest. Nossa abordagem proposta, considerando apenas primeira consulta, alcançou um AUC de 0,78 usando XGBoost como algoritmo de aprendizado, representando um ganho de desempenho relativo de até 20,37% comparado ao XGBoost *tudo-em-um*, e AUC de 0,75 quando usando Random Forest como algoritmo de aprendizado, com ganho de desempenho relativo de até 15,03% comparado ao Random Forest *tudo-em-um*. Além disso, o número de features é significativamente reduzido, fazendo uso de tão pouco quanto 15% das features. Ao considerar as consultas sequenciais, os experimentos mostraram consistentemente que quanto mais consultas consideradas, maior será o desempenho alcançado. Nossa abordagem EXP-MF combinada com o ED-Ensemble alcançou uma AUC de 0,945 (aumento de 23,37%) utilizando cinco consultas. Uma tendência de aumento semelhante na AUC também foi observada para os algoritmos

EXP-MF combinado com XGBoost e EXP-MF combinado com Random Forest, alcançando 0,843 (aumento de 50%) e 0,810 (aumento de 62,98%), respectivamente. Finalmente, o ensemble proposto baseada em diversidade de explicações se apresentou como uma alternativa superior à abordagem *tudo-em-um* em problemas de fenômenos de múltiplas estruturas tanto nos dados de corte transversal quanto dados longitudinais.

Palavras-chave: Aprendizado de Máquina, Modelagem Exploratória, Modelagem Preditiva, Estrutura de backbone, Combinação de Classificadores, Métrica de Diversidade, Estabilidade Predição-Explicação, dados longitudinais.

Abstract

Modeling from data usually has two distinct facets: building sound explanatory models or creating powerful predictive models for a system or phenomenon. While there is an instinctive sense that predicting and explaining are distinct tasks, it is often assumed that models with high explanatory power are inherently of high predictive power. In spite of this relationship, most recent data-modeling work fits into the *all-in-one* machine learning methodology, with the basic assumption that all important explanatory factors can be combined into a single predictive model. Although highly adopted and established, the *all-in-one* methodology neglects that many phenomena are actually defined by several subpopulations or local structures and therefore there are many possible predictive models that provide contrasting interpretations or competing explanations for the same phenomenon. In this work, we present ED-Ensemble (Explanation-Diversifying Ensemble), an alternative to the *all-in-one* methodology. Our main intuition is that models that have their decisions explained by the same factors will probably perform better predictions within the same local structures. We design and conduct an experimental evaluation as a case study to evaluate the performance of our methodology to model the evolution of pain relief in patients suffering from chronic pain under usual guideline-based treatment. Six hundred thirty-one participants self-completed the McGill Pain Questionnaire and the Visual Analogue Scale. Chronic pain can be primary or secondary to diseases. Its symptomatology can be classified as nociceptive, nociplastic or neuropathic, and is generally associated with many different causal structures, challenging the typical *all-in-one* methodology. We show that we can effectively combine models with competing explanations, promoting diversity in ensemble, leading to significant gains in accuracy by enforcing a stable scenario in which models that are similar in terms of their predictions are also similar in terms of explanatory factors. Further, we present EXP-MF (model-EXPlanations as Meta-Features). We follow the explanation-diversity feature selection proposed and extend it to use model-explanations as meta-features in longitudinal data, as the standard protocol for a patient typically comprises many subsequent appointments. This approach prevents us from neglecting a considerable amount of information. The ED-Ensemble obtained from our experiments consistently outperformed the *all-in-one* approaches, notwithstanding employing state-of-art ensemble algorithms XGBoost and Random Forest. Our proposed approach considering the first consultation only achieved an AUC of 0.78 using XGBoost as learning algorithm, relative performance gain up to 20.37% compared to the XGBoost *all-in-one*, and AUC of 0.75 when using Random Forest as learning algorithm, relative performance gain up to 15.03% compared to the Random Forest *all-in-one* approach. Also, the number of features is remarkably reduced, using as low as 15% of features. When considering sequential consultations, the experiments consistently showed that the more consultations granted, the higher the performance achieved. Our approach EXP-MF with an ED-Ensemble could achieve an AUC of 0.945 (increase of 23.37%) using five consultations. A similar uptrend in AUC was also observed for the XGBoost and Random Forest algorithms, achieving 0.843 (increase of 50%) and 0.810 (increase of 62.98%) respectively. Finally, our novel ensemble based on diversified explanations presented as a superior alternative to the *all-in-one* approach in multiple-structure phenomena problems with cross-sectional and longitudinal data.

Palavras-chave: Machine Learning, Explanatory Modeling, Predictive Modeling, Backbone Structures, Ensemble Learning, Diversity Metric, Prediction-Explanation Stability, Longitudinal Data.

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1

Introduction

We are interested in modeling complex phenomena defined by different sub-populations and thus present diverse local structures. Local structures consist of subsets in data space highly correlated with the output we want to predict. The underlying distribution in the data we want to model may vary in different parts of the data space. Take, for instance, a scenario where wants to predict if a particular treatment will be effective for pain relief in a patient suffering from an unknown chronic pain condition. In this case, the data consists of attributes extracted from patients' self-reports obtained at multiple appointment or consultation with the doctor. In more detail, the doctor uses a formalized pain questionnaire, asking the patients to choose the characteristics that best describe their pain (i.e., burning, tingling, sharp, or dull). The patient is also asked how long the pain lasts, what makes it worse, and what relieves it (i.e., activities, medications, and weather). Predicting the evolution of pain relief is hard because chronic pain can arise from many different conditions, like fibromyalgia, cancer, arthritis, violent traumas, and many other possibilities [Pombo et al., 2014]. It may be hard to detect these conditions at the first appointment. The data presents many local structures so that the factors contributing to the correct treatment decisions depend on a complex structure that emerges from specific characteristics reported by patients. Further, it is crucial to also make use of sequential consultations in order to increase confidence.

1.1 The Problem

Our case study consists of six hundred and thirty-one patients who sought medical help to treat chronic pain. Two thousand and five consultations are distributed non-uniformly among the patients. The data consist of information collected mainly through the McGill questionnaire and the Visual Analog Scale (VAS). The data obtained includes age, gender, pain intensity (discrete scale from 0 to 10), the McGill score, Neuropathic Pain Scale, and pain perception dimension scores. In addition, it is also obtained the medications that the doctor prescribed during the treatment. All this information makes up three hundred and thirty-two features per consultation.

The goal is to predict the effectiveness of standard treatment for chronic pain. Successful treatment occurs whether the patient achieves a significant reduction in pain sensation at the end of the treatment, which in our case is the last visit. Specifically, an overall reduction of pain intensity by 30% (aka, VAS 30) is assessed, formally considered a successful treatment outcome [Dworkin et al., 2008b]. The ground truth labels are obtained by calculating the difference in pain intensities reported in the first and last consultation.

Given the specificities of the problem, the main computational challenges are data high dimensionality and the presence of many sub-populations. Chronic pain can arise from multiple causes. Back pain, for example, it may be caused by a combination of factors that interact with each other: years of poor posture, carrying of heavy objects, overweight, or even no apparent physical cause. Each sub-population is expected to represent one cause of chronic pain. Modeling all local structures into a single model may not be optimal. Breaking it down into smaller problems would benefit from constructing simpler models with better performance since they would be easier to optimize. If we knew in advance which points belong to which local structure, we could simply create a model for each sub-population. Ideally, a model would be related to a cause. However, the cause is unknown.

1.2 Motivation

Intuitively, if different data points (i.e., patients) are associated with different local structures, we would expect each structure to be better described by a different model. Then we can get a model for the entire data by combining (potentially simpler) models for all the local structures. In this case, a simple solution is to divide the original data space into biclusters, enabling concurrent feature and data point selection. Each bicluster may approximate a local structure from which a model is built [Pansombut et al., 2011]. Another widespread solution is to estimate local structures in the data using the Expectation-Maximization (EM) algorithm to get maximum likelihood estimates [McLachlan and Peel, 2000]. More often, however, these many-structure phenomena are modeled using the simple all-in-one approach, which fits all the available factors (or features) into a single model. The all-in-one approach is clearly sub-optimal in multiple phenomena context since factors that are important for modeling one structure may become lurking or confounding variables influencing other structures in the data. A branch of a decision tree, for instance, may mix different local structures, or the same local structure may be fragmented into different branches of the tree. Also, parametric models that require combinatorial non-convex optimization, such as gradient descent, would benefit by decomposing the phenomenon into local structures and exponentially reducing the size of the search space [Friesen and Domingos, 2015].

Additionally, when considering sequential consultations from patients, the data is organized as longitudinal data. Longitudinal data involve repeated observations for the same subject at different points in time. In general, machine learning methods assume that the random variables are independent and identically distributed. However, longitudinal data from patient reports may violate this assumption, thus naively applying widely used machine learning algorithms such as Random Forest (RF) [Breiman, 2001], Support Vector Machine (SVM) [Cortes and Vapnik, 1995], or Artificial Neural Networks (ANN) [Rosenblatt, 1958] to longitudinal data without any adaptation, while possible, ultimately generates an inefficient model.

1.3 Our Solution

There are well-known solutions in the literature for estimating local structures in the data. These solutions look at the data to identify local structures. We propose a different approach. Since we have a phenomenon with a multiple-cause structure and aim to create a model for each cause, the main challenge is inferring the cause as it is unknown. The solution proposed uses explanations as a proxy for causation. We will use the concepts of explanation and reuse these concepts for prediction.

We will start constructing the model space by systematically sampling many models from randomly selected subsets of features. We will denote the explanation of a model as a vector with the average contribution of each feature. Thus, the model space will be composed of competing and, at the same time, contrasting explanations for the same phenomenon. Next, by clustering the model space based on the explanation criteria, we expect each resulting cluster to describe a similar pattern of explaining the phenomenon (similar causation). Finally, we will select one model from each group to be the representative prototype of the cluster. The combination of the prototypes allows us to build an ensemble based on a diversity of explanations (as a proxy for causation). The algorithm described we called ED-Ensemble (Explanation-Diversifying Ensemble), and it is our solution when we only consider the first consultation.

However, the standard treatment protocol for a patient typically comprises many sequential appointments. Thus, our second solution aims to work with longitudinal data. Using a human-centric approach, we propose to model the physician's behavior over sequential consultations. For instance, we aim to forget everything irrelevant, carry only a small set of information between visits, and diagnose based on the current consultation plus selected prior knowledge. Our approach extends the ED-Ensemble algorithm to use previous models' explanations as a temporal memory on longitudinal data. Hence, the second solution uses the longitudinal structure present in sequential consultations to increase the prediction confidence.

In summary, we evaluate two scenarios:

- The best we can achieve from the first consultation only (maximum anticipation and suboptimal assertiveness).
- Study the trade-off between assertiveness and anticipation (if we decrease anticipation, i.e., consider more consultations, can we increase assertiveness?).

1.4 Main Contributions

This thesis aims to reach two main contributions: 1) a new ensemble algorithm (ED-Ensemble) in data with diverse local structures and 2) a novel method EXP-MF (model-EXPlanations as Meta-Features) that enhances the usage of traditional machine learning approaches in longitudinal data. These two contributions are presented in more detail in the following sections.

1.4.1 A New Ensemble Learning Approach for Modeling Backbone-Structure Phenomena

The backbone structure is a particular type of local structure. There is a set of "backbone features" that, once set, causes the remainder of the features to decompose into independent subsets in the data space. Unlike previous attempts [Agrawal et al., 2005], we propose to cluster the explanation space[1] instead of (bi)clustering the data space. By analyzing the sampled model space, we found a strong link between model predictions and model explanations. We show evidence that models having their predictions explained by the same reasons (or factors) are likely suitable for modeling the same

[1] We assume that explanations are given in terms of the central factors which unveil systemic pattern(s) within the model predictions.

local structures in the data space. In summary, the step-by-step of this contribution are:

- We evaluate our proposed approach in a real case study that predicts whether a treatment will be effective in reducing pain relief in patients suffering from unknown chronic pain conditions. It is a fascinating case study because it is defined by phenomena that exhibit the backbone structure. Thus, by learning simpler models likely associated with different local structures, we can achieve feature decompositions that algorithms like feature elimination cannot. Many other problems seem to exhibit the backbone structure (e.g., protein folding [Friesen and Domingos, 2015], Alzheimer's diagnosis [Jha and Kwon, 2017]).
- We present ED-Ensemble, a method to learning ensembles from local models (or base models) that present diversity in their explanatory factors. While diversity is recognized as a central element for significant performance improvements with the ensemble, measuring diversity is not straightforward because there is no generally accepted formal definition [Kuncheva and Whitaker, 2003]. In order to promote diversity while learning the ensemble, we select local models associated with different explanatory factors. Thus our ensemble strategy is fundamentally a combination of competing explanations for the same phenomenon.
- We show that there is a multiplicity of performant models with diverse explanations. Learning the ensemble by forcing prediction-explanation stability in the sense that models that are similar in terms of their predictions should have similar explanations leads to gains in accuracy up to 10%.
- We demonstrate that we can generate ensembles with combined features drawn upon around 15% of the total features by inducing simpler local models. Hence, along with a performance improvement, we can markedly reduce the number of features used. The shortened features also improve interpretability, diminishing the gap between ethics and clinical decision support systems (CDSS).

1.4.2 Model-Explanation as Meta-Features in Longitudinal Data

The analysis of longitudinal data is traditionally performed using statistical methods [Verbeke et al., 2014; Perveen et al., 2020]. These methods, however, require many assumptions about the data in order to work correctly and machine learning methods, on the other hand, require considerably fewer assumptions about the data. One of the few assumptions is that the random variables are independent and identically distributed. However, longitudinal data from patient reports may violate this assumption as observations are correlated for the same patient but independent across different patients [Sela and Simonoff, 2012; Hu, 2021; Ngufor et al., 2019].

We introduce a novel machine learning method to longitudinal data to predict the evolution of pain relief. Our approach uses previous models' explanations (i.e., feature importances) to function as a temporal memory on longitudinal data. Precisely, to predict the output at consultation c, we extract feature importances [Lundberg and Lee, 2017] from a model trained on the data up to consultation $c - 1$ and use these explanations as memory meta-features about previous iterations. The intuition is that our approach improves the current model by remembering important information from previous consultations. In summary, the main contributions of this study are:

- We investigate the evolution of pain relief in patients suffering from unknown chronic pain conditions. Specifically, given data from the patient's sequence of consultations, the model predicts the likelihood that the patient will significantly reduce pain at the end of the treatment.

- Our novel modeling approach considers the temporal relationship existing in longitudinal data through a data-wise adaptation, which improves prediction performance considerably.

- We propose an explanation-diversity feature selection approach which indicates a preference for choosing feature importances carried over from previous consultations instead of the raw feature information. This preference is a strong indication that feature importances contain more decisive information than features from previous consultations.

- Finally, and most importantly, experimental results show that our method reaches an area under the ROC (AUC) curve of 0.766 using data from consultation 1 alone which follows an explanation-diversity feature selection procedure. In the second consultation, the AUC value increases to 0.818 using model-explanations from consultation 1 as meta-features and combining it with fresh data from consultation 2. Similarly, the prediction performance increases to 0.945 within five consultations (an increase of 23.37% from consultation 1 only), drastically reducing the treatment planning period.

1.5 Thesis Statement

In many situations, the data is inherently composed of several local structures and sub-populations. The traditional all-in-one approach considers the use of all data at once to induce a single model. By assuming different local structures as different views of the same data, it is harder for an algorithm to minimize the error by considering the information of all views, in many cases contrasting. This thesis aims to show, based on evidence, that in these situations, it is advantageous to take knowledge and make use of the concept of these local structures for the induction of models that are more robust and consistent with the data. Moreover, using only the information from the first consultation alone (baseline) implies ignoring a substantial amount of data. Based on this premise, we use collected data at the time point n and transfer the knowledge acquired to $n+1$ through model explanations meta-features. We repeat the process so that all accumulated knowledge is taken to the next consultation, and so on.

1.6 Thesis Structure

The remainder of the text is organized as follows. Chapter 2 provides a discussion of relevant related work along with the essential foundations. Chapter 3 describes our proposed approach for modeling multi-structure phenomena. Chapter 4 discusses the case study results on modeling the evolution of pain relief in patients suffering from an unknown chronic pain condition. Chapter 5 is dedicated to the bias and variance decomposition of the error in the ED-Ensemble. Chapter 6 examines the use of EXP-MF in longitudinal data. Finally, Chapter 7 presents the conclusions and future work.

Chapter 2

Related Work

Learning models from high-dimensional data is a well-studied problem in various fields such as feature selection, feature decomposition, and ensemble learning. Our work builds upon a wealth of previous research at the intersection of these fields. The following sections will present related works that contributed to the development of our approach.

2.1 Feature Selection

The curse of dimensionality is known as the set of phenomena that emerge when working with data containing a large number of features, compared to the number of instances [Bellman, 1966]. In this context, due to the large size, the volume of data grows so rapidly that the available data turns out to be scattered, directly impacting algorithms that explicitly use searching, such as the k-nearest neighbors (k-NN). This impact is also extended to other algorithms such as Artificial Neural Networks (ANN) and Support Vector Machine (SVM), albeit indirectly. It becomes harder to find patterns once data dimensionality has grown.

Over the years, several approaches to solving the problem have been proposed. Principal Component Analysis (PCA) is one of the most well-known approaches to dealing with high-dimensional data. PCA is a statistical method that uses an orthogonal transformation to find the most significant possible variance, converting the data into a new dimensional space, where each dimension is orthogonal [Jolliffe, 2011]. However, PCA has some limitations that make its use unfeasible in our context. First, PCA cannot be automatically used for feature selection. PCA rotates the data from one coordinate system to another, such that the dimensions in the new coordinate system are arranged in descending order concerning variance. Features with the largest variance are not necessarily the most important ones. Second, although PCA is possible to apply to non-continuous variables, its use in these cases is debatable. We desire to work with data that is not limited to continuous types only. Third, the interpretation of models supplied with data in the new coordinate system generated by PCA is more complex. New features created with PCA are not easily related to existing features, making it difficult to explain how much each original feature contributed. Finally, our data contains many different local structures. The main limitation that prevents us from using PCA is the lack of knowledge in associating the points with their respective local structure (e.g., fibromyalgia in the context of chronic pain). Decomposing data in local structures is still a problem to be addressed but not yet solved.

An alternative method to address high-dimensional data is to reduce the number of features, selecting just a critical subset. A typical case is testing all features combinations and choosing those resulting in the slightest error. However, this alternative is restricted to datasets with only a few features due to the exponential increase in combinations.

In general, methods for feature selection can be grouped into three classes: filter, wrapper, and embedded. Filter-based methods select features regardless of the model, while wrapper methods rely on the specific model used. Although filter-based methods are expected to yield poorer results than wrapper methods, it has the benefit that fewer computational resources are expended. Examples of filter-based methods are RELIEF [Kira and Rendell, 1992] (and the family of derived algorithms), Correlation-based Feature Selection (CFS) [Hall, 1999] and Consistency Measure [Hall, 2000]. One major drawback of this approach is that a subset of highly correlated features can dominate the selection.

Wrapper methods use the inductor algorithm to evaluate the quality of the features subset, leading the search in the feature space. Heuristics applied to the problem are forward selection, backward elimination [Maldonado and Weber, 2009] and its modifications as Recursive Feature Elimination (RFE) employed in SVM [Guyon et al., 1992]. The main limitation of the wrapper methods is that it requires testing a high number of combinations, being thus very expensive, clearly when the feature set is large.

Lastly, embedded methods do feature selection interleaved with learning. For instance, in tree-based ensemble algorithms [Chen and Guestrin, 2016; Breiman, 2001], each feature is evaluated as a potential splitting variable, which makes them robust to unimportant/irrelevant features. Features that are not discriminative will not be selected as the splitting variable and hence will be associated with a low importance value. Nonetheless, feature selection methods are not suitable for modeling phenomena defined by multiple local structures [Maimon and Rokach, 2002]. The model does not include features not influencing the dependent variable, and correlations between features and the dependent variable are likely to vary in different local structures that exist in the data. The dependent variable is influenced by most of the features, but the observed correlation strength may vary depending on the local structures in the data space. Removing features may cause a significant loss of relevant information.

2.2 Feature Decomposition

Whereas feature selection aims to identify a representative set of features from which to build a model, feature decomposition aims to decompose the original set of features into several subsets. The feature decomposition changes the representation of a learning problem depending on the local structures in the data space. Instead of learning a single complex model, several sub-problems with different and smaller feature sets are defined.

Co-training is a semi-supervised learning approach developed by Blum and Mitchell [1998] that also performs feature decomposition. It assumes that data naturally has two views, and each view would be a set of mutually exclusive features. A view would potentially have a different confidence in predicting a set of unlabelled instances than another view. At each step, unlabelled instances with high trust in prediction are added as "probably labeled" instances. This process continues iteratively until a stop criterion is reached. Co-training has been successfully applied to problems where the amount of unlabeled data available is enormous compared to labeled [Wan, 2009; Kiritchenko and Matwin, 2011].

Later, Chen et al. [2011] extended the original co-training work in order to perform feature partitioning during learning by generating two sets of mutually exclusive features that satisfy the required co-training hypotheses. While co-training is deemed as a robust approach, its success is highly dependent on the subset of features built, being subject to the validity of the independence hypothesis in the set of features [Nigam and Ghani, 2000].

An alternative feature decomposition approach is biclustering [Cheng and Church, 2000], which is a class of clustering algorithms that concurrently group features and data points. Formally, the goal is to find local structures in the data space defined as subsets of data points in which a specific subset of features is highly correlated. Thus, a subset of highly correlated features within a local structure may be a set of independent features within other regions of the same data space. Oliveira and Madeira

[2004] presents an excellent survey on the subject.

2.3 Ensemble Learning

Ensemble methods are learning algorithms that construct a set of models and then classify new data points by taking a (weighted) vote of their predictions. It is based on the assumption that multiple hypotheses tend to produce more robust and stable solutions than a single hypothesis [Topchy et al., 2004].

Often the objects to be clustered have multiple aspects or views, and base models may be built on distinct views that involve nonidentical sets of features or subsets of data points [Ghosh and Acharya, 2011]. This is known as multiview clustering. Different sets can be formed by different feature sets, known as Feature Distributed Clustering (FDC), where each set has a subset of different features, but all data is used. An alternative would be to use all features but in distinct subsets of data. This is known as Object Distributed Clustering (ODC) [Strehl and Ghosh, 2002].

In Pansombut et al. [2011], the authors presented Biclustering-driven ENsemble of Classifiers (BENCH), a method to create an ensemble using biclusters. Their approach first splits the original data space into biclusters, and each bicluster becomes a base model candidate. Despite biclustering being an unsupervised technique, BENCH partly takes into account labels. It considers labels and their correlation with the cluster to form candidate datasets capable of adequately distinguishing them.

Additionally, Wang et al. [2011] proposed the Nonparametric Bayesian Clustering Ensemble (NBCE), a method that can discover clusters in the consensus clustering. A benefit provided by NBCE is that it is not required to define the a priori number of clusters. Also, there is no need to maintain the desired properties of the Bayesian Clustering Ensemble.

2.4 Machine Learning on Longitudinal Data

In cross-sectional data, it is assumed that all data samples were collected at the same point in time. In longitudinal data, on the other hand, multiple observations are made for the same subject, and these observations are at different points in time. Also, the samples between different subjects may be at further points in time. In this hierarchical arrangement of longitudinal data, observations across subjects are independent, and observations for the same subject are dependent. For the accurate modeling of longitudinal data, it is crucial to consider the correlation between observations of the same subject [Sela and Simonoff, 2012; Hu, 2021; Ngufor et al., 2019].

Traditional machine learning approaches, e.g., Support Vector Machine (SVM) and Random Forest (RF), are primarily designed to work with cross-sectional data. Conversely, some approaches, such as the Hidden Markov Model (HMM), are naturally suited for longitudinal data though commonly with some limitations. For instance, HMM requires regularly sampled longitudinal data, i.e., an equal number of repeated observations across subjects [Perveen et al., 2020]. Another approach naturally tailored for longitudinal data is the recurrent Long Term Support (LSTM) network architecture. However, like the vast majority of deep learning approaches, it requires a large amount of training data [Jiang et al., 2020].

In the literature, for prediction on longitudinal data, we can find several adaptations of machine learning algorithms that naturally work on cross-sectional data to work on longitudinal data. Trees and their variations, particularly RF, are the approaches with many adaptations [Segal, 1992; Capitaine et al., 2021; Sela and Simonoff, 2012]. RF is an algorithm that works better than trees the vast majority of the time [Breiman, 2001], and especially for high-dimensional data [Capitaine et al., 2021; Verikas et al., 2011].

Fard et al. [2016] attempts to predict an event at the end of the longitudinal study using the information at the early stage of the study. Two main aspects motivate the search for prediction with the minimum number of observations. There is a high cost to obtain labeled data. And in many situations, measurements can be obtained just by waiting for such an event. The proposed frameworks are based on Naive Bayes (ESP-NB), Tree-Augmented Naive Bayes (ESP-TAN), and Bayesian Network (ESP-BN). Experiments with synthetic and real data showed they are more effective in predicting future events than RF and Linear Regression (LR) approaches. However, for datasets with the most significant number of features (77 and 54, respectively), the performance AUC metric estimated was similar to the RF. Notably, for the Kickstarter dataset with 77 features, the RF performance (0.845) outperforms ESP-NB (0.822) and ESP-TAN (0.827) and nearly equals to ESP-BN (0.847).

The work of Capitaine et al. [2021] proposes a general approach for RF on high-dimensional longitudinal data where the number of predictors p is much larger than the number of observations n, i.e., $p \gg n$. Unlike previous approaches using tree-based models within a semi-parametric mixed-effects model, the authors propose a flexible stochastic model allowing the covariance structure to vary over time. Experiments showed that the proposed approach achieved better performance when the data had high dimensionality than other state-of-the-art approaches. Again, the robustness of RF stands out. The real dataset experimented showed similar performance to algorithms specifically adapted for longitudinal data: Mixed Effects Random Forests (MERF) [Hajjem et al., 2014] and REEMforest [Capitaine et al., 2021]. However, RF showed inferior performance compared to the SMERF and SREEMforest variations, evidencing that taking into account the longitudinal aspect of the data leads to decreased prediction error. Unlike our proposal, the approach of Capitaine et al. [2021] addresses extreme high-dimensional cases, where the amount of predictors is orders of magnitude larger than the number of instances. For instance, the real data considered comprises 17 instances and 32,979 predictors. The study's main weakness is that feature selection is done after model convergence, allowing confounding situations to occur.

The authors in Ngufor et al. [2019] propose a general framework for prediction in longitudinal data, combining the advantages of the random-effects structure of the Generalized Linear Mixed-effect Model (GLMM) with the benefits provided by machine learning models. GLMM is the standard statistic approach for longitudinal data. Still, parametric linear models make assumptions about the data that are often difficult to verify in actual, complex data. Ngufor et al. [2019] proposes the Mixed-Effect machine learning (MEml) framework, where GLMM estimates the random-effects and the fixed-effects are calculated by machine learning algorithms. From the results, we can highlight: (1) the RF algorithm is robust, showing performance on longitudinal data often similar to algorithms explicitly tailored for this data organization; (2) the greater the number of observations in the samples, the more significant the performance gap between models that take into account the random-effects with those that do not. Our work, in turn, manipulates a reduced number of observations. Finally, RF presented itself as a good benchmark for our proposal, given its good performance when few observations are available.

Hence, adaptation is required to employ traditional machine learning approaches on longitudinal data properly. On the data, through manipulation to fit the specification expected by the algorithm. Or on the algorithm, making it capable of handling longitudinal data correctly. We are particularly interested in data-level adaptations to allow (enhance) the employment of traditional machine learning algorithms on longitudinal data.

2.5 Methods for Modeling Pain

Data availability and quality are of fundamental value in chronic pain modeling. In the 1970s and 1980s, few hospitals collected structured computer data. In addition, those hospitals collecting data usually had their own nomenclature and definition, hindering any attempt to algorithmically model pain [Navani and Li, 2016]. Nowadays, the amount of data available is vast, along with the advancement in the standardization of medical nomenclatures.

2.5.1 Machine Learning

Recent works seek to model chronic pain in a computer system to predict the evolution of the disease in the patient. Navani and Li [2016] build a machine learning system for calculating dynamic changes to the weight-based chronic pain risk score on various aspects of health behavior. However, only three sources of information are used: depression, nutrition, and physical activity.

Machine learning approaches to predictive analysis of pain models have some well-known limitations. Pieterse et al. [2019] provide us examples where the generated models are no better than human-analyzed regression models and, in some cases, are doomed to overfitting. Also, Goldstein et al. [2016] points out that clinicians are aware that in machine learning algorithms, it is not possible to see or understand what exactly influences model prediction directly. Besides, the prediction of an event without the ability to change the output is questionable.

A systematic review in clinical decision support systems for pain management is presented by Pombo et al. [2014]. The authors observed the great diversity of algorithms applied in the Clinical Decision Support Systems (CDSSs): rule-based algorithms such as C4.5, CART, PRISM, artificial neural networks, and statistical learning algorithms are well-known choices. Furthermore, as reported by Abad-Grau et al. [2008], it appears to be hard for medical experts to build valid models when too many variables influence the dependent variable. Pombo et al. [2014] observe that this limitation leads to the design of low-accuracy systems. Moreover, black-box approaches seem a misleading option considering the system should assist clinicians to reach a decision.

2.6 Our Work

Intuitively, for the local models to form an improved ensemble, they must make correct predictions on diverse subsets of the data space. We exploit a distinct notion of diversity in terms of the explanatory factors within each local model. By learning local models composed of different feature sets, we can achieve feature decompositions that feature selection algorithms cannot. We show that our proposed approach can model phenomena that exhibit “backbone structures”, a type of local structure induced by a specific feature that, once set, causes the remainder of the features to decompose into independent ones subsets in the data space.

Chapter 3

Ensemble by Diversifying Explanations

This chapter presents ED-Ensemble, a novel proposed approach for modeling phenomena defined by multiple local structures in the data space known as backbone structures. Formally, we want to learn a model from data that is a mixture of subpopulations where each subpopulation is associated with a particular subset of features. The corresponding optimization problem has a non-convex error surface with no obvious global minimum, thus implying a multiplicity of performant models. Each of them provides a different explanation for the phenomenon. Therefore, there may be many contrasting interpretations or competing explanations for the same phenomenon. The modeling approach we will describe in this chapter is based on finding an explanation for the phenomenon coherently with all competing explanations. Specifically, we propose to decompose the original set of features into several subsets so that a particular model is built for each subset of features. Then, the generated models are clustered according to their explanatory factors, promoting the diversity in possible explanations while learning the ensemble. We expect the final ensemble model to correspond to a more global explanation for the phenomenon, improving prediction accuracy.

3.1 Local Structures

A data space is defined as a set of n data points of the form $(\mathbf{x}, y)^n$, such that $\mathbf{x} \in \mathcal{R}^d$ is given as a feature vector $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d\}$ and y is the ground-truth output for $\mathbf{x}$. Often, in high-dimensional data spaces, regions show complex correlations among a specific set of features and the target label, and the same correlations are not necessarily so strongly observed in other regions of the data space. Thus, the theory$-$data relationship varies

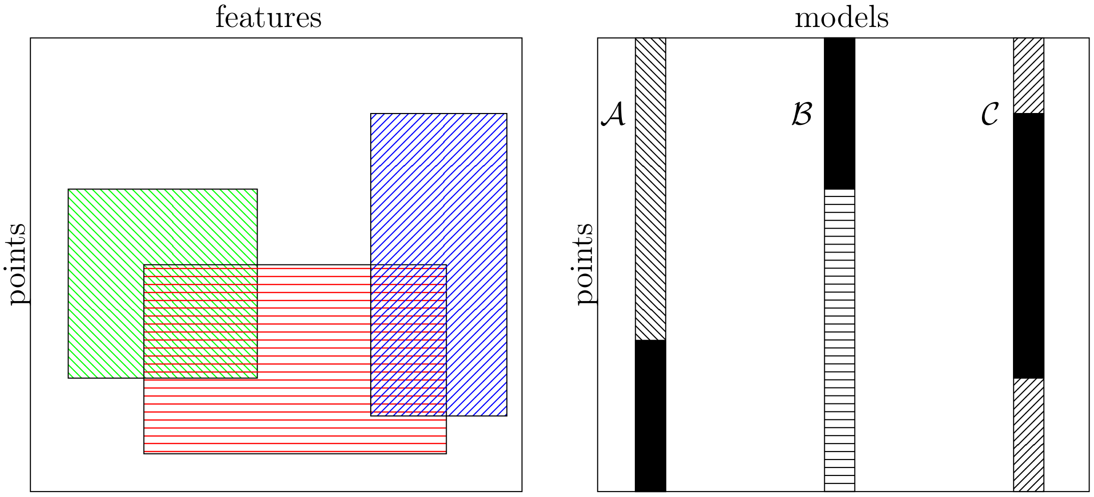


Figure 3.1: (Color online) Left – An illustrative example of a data space with three local structures. For simplicity and to avoid clutter, local structures are shown as contiguous regions in the data space, but local structures may be non-contiguous in both axes. Right – An illustrative example of model preferences for three hypothetical models $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ in $\mathcal{H}'$. Probabilities that a model assigns to points in the lighter region are closer to true label than the probabilities that the model assigns to points in the darker regions. Model $\mathcal{A}$ shows preference for points in the green structure. Model $\mathcal{B}$ shows preference for points in the red structure. Model $\mathcal{C}$ shows preference for points in the blue structure. A model may also show preference for points within multiple structures simultaneously.

in different regions of the data space, forming local structures defined as subspaces spanned by a set of data points and a set of features [Tanay et al., 2004, 2005], as illustrated in Figure 3.1 (Left). Local structures can overlap and are often the result of mixing different sub-populations or distributions into the same data space. Hence one cannot easily separate them into multiple sub-spaces. A particular type of local structure resembles a backbone in the sense that there is a set of features (aka backbone features) that show a strong correlation with a specific set of target labels. Thus, forcing a backbone feature to appear in the same model with non-related features may incur confounding situations.

3.2 Sampling of Model Space

Learning a model from the data space requires the minimization of an objective function $f(\mathbf{x})$. Instead of simply mixing multiple different structures into a single model $\mathbf{x}$ and minimize $f(\mathbf{x})$, we sample the model space by minimizing different functions $f(\mathbf{x}')$, such that $\mathbf{x}' \subseteq \mathbf{x}$ and $|\mathbf{x}'| \ll |\mathbf{x}|$. Features that compose each model $\mathbf{x}'$ are randomly selected, and we used gradient boosted trees [Chen and Guestrin, 2016] and Random Forests [Breiman, 2001] as learning algorithms (but other algorithms can be applied as well). After sampling the model space, each model $\mathbf{x}'$ is evaluated with respect to an error measure $\ell(x')$ on a validation set, so that only minimally performant models for which $\ell(\mathbf{x}') \leq \epsilon$ are included in the final model space $\mathcal{H}'$. At this point, we expect that $\mathcal{H}'$ contains performant models corresponding to possible explanations for the phenomenon.

The Algorithm 1 presents the pseudocode to generate the sampled model space. Given a learning algorithm l, a dataset d, an upper limit on the number of sampled features t, the algorithm generates an external file with each line describing a trained model supplied with a randomly selected subset of features.

Algorithm 1 Model Space Sampling

Input Learning algorithm l, dataset d and maximum number of features t.

Output External file where each line describes a trained model supplied with a random subset of features.

```
for $n \leftarrow 1, t$ do
    if $n > 1$ then
        $k \leftarrow 10000$
    else
        $k \leftarrow 500$
    end if
    combs ← Randomly generate $k$ features combinations of length $n$
    for each x_prime in combs do
        Learn a model $m$ using the learning algorithm $l$, and the training dataset $d$ taking into account the feature subset selected in x_prime only (5-fold cross validation)
        Estimate AUC-ROC
        Obtain probability values from model predictions
        Write to an external file with the following information <n, mean(AUC-ROC), variance(AUC-ROC), x_prime, probability values>
    end for
end for
```

3.3 Representing Model Preferences

We represent the model preference as a n-dimensional vector $\mathcal{P}(\mathbf{x}') = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, where $\mathbf{p}_i$ corresponds to the probability that model $\mathbf{x}'$ has assigned to data point i. We expect that models in $\mathcal{H}'$ are representative of the diverse local structures that exist in the data space, as illustrated in Figure 3.1 (Right). By filtering performant models for which $\ell(\mathbf{x}') \leq \epsilon$ we expect that the corresponding local structure is properly explained by the corresponding model $\mathbf{x}'$.

3.4 Representing Model Explanations

We represent how model $\mathbf{x}'$ explains the phenomenon as a d-dimensional vector $E(\mathbf{x}') = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_d\}$ showing which features are driving the model's prediction. Specifically, $\mathbf{e}_i$ takes a value that corresponds to the influence that the respective feature $\mathbf{x}_i$ had on the model decision. Since we do not assume feature independence while minimizing $f(\mathbf{x}')$, then correlated features within model $\mathbf{x}'$ should share credit or importance. For this reason, we employ the average SHapley Additive exPlanations (SHAP) values for accessing feature importance.

3.4.1 SHapley Additive exPlanations

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. Given an instance x, SHAP provides us with the weighted contribution of each feature in the outcome. As with LIME [Ribeiro et al., 2016], SHAP focuses on *local methods*. It aims to learn how the model behaves in the vicinity of an instance x. This behavior, however, may not truly represent the original model f in the entire data space. After all, only the model itself reliably describes the behavior globally. SHAP uses an *explanation model* g that is capable of locally represent f. The explanation model is described as:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \tag{3.1}$$

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

There is a difference in features and interpretable data representations. Along with features contributions, a model to be interpretable needs to make use of a representation that is human-understandable [Ribeiro et al., 2016]. For instance, word embedding is a well-known representation used in texts. Even if the approximate model g provides each feature's contribution, the interpretability is limited. The features themselves are not interpretable. Instead, a bag of words can be used as an interpretable alternative representation of the same information. There is a function h such that $x = h_x(x')$, mapping the *simplified input* x' into the original input x.

Lundberg and Lee (2017) describe SHAP as a permutation-based approach for feature importance attribution, which defines a model as a cooperation of features. It assigns a value for each feature in the cooperation based on its contribution to the model decisions. There are many other feature attribution methods [Ribeiro et al., 2016, 2018], but SHAP is the only method with the three desirable properties:

- **Local accuracy**: The explanations are truthfully explaining the model.
- **Missingness**: Missing features have no attributed impact on the model decisions.
- **Consistency**: If a model changes so that some feature's contribution increases or stays the same regardless of the other features, that feature's attribution should not decrease.

Options to calculate the explanation model g include LinearSHAP, KernelSHAP, DeepSHAP and TreeSHAP [Lundberg et al., 2020]. In this work, we used TreeSHAP since our learning algorithms are based on gradient boosted trees or random forests. TreeSHAP takes only "allowed" paths within the tree, meaning it does not include non-realistic combinations of features as in other permutation-based methods. Instead, it takes the weighted average of all the final nodes reachable by a certain coalition of features. Differently from the other options, TreeSHAP scales linearly with the number of data points and grows at a polynomial rate with the number of features.

3.5 Ensemble Learning

As $\mathcal{H}'$ may contain models with competing explanations, we want to build a synthetic model from $\mathcal{H}'$ by exploiting two concepts:

- The diversity between individual models. Diversity is recognized as a fundamental factor to achieve significant performance improvements with the ensemble [Kuncheva and Whitaker, 2003] by allowing the group to compensate for individual errors. However, measuring diversity is not straightforward because there is no generally accepted formal definition. In order to promote diversity while learning the ensemble, we cluster models in $\mathcal{H}'$ based on the distance between their explanation vectors (i.e., SHAP values). Ideally, this creates many groups of internally dense and separated from the rest of the models in terms of their explanatory factors, i.e., within each cluster, the explanatory factors are similar. In contrast, factors within disjoint clusters are dissimilar.

- The stability between model explanation and empirical predictions [Shmueli, 2010]. We define a configuration of clusters as stable if models within the same cluster are associated with the same explanatory factors and perform similar predictions. Achieving cluster stability is challenging, as models with similar predictions can be associated with different explanatory factors. In order to promote stability while learning the ensemble, we cluster the model space based on the distance between the explanation vector (i.e., SHAP values) associated with each model. However, we maximize cohesion and separation of the clusters based on the distance in terms of model preference. This enforces a stable configuration of clusters containing similar models in terms of their predictions and explanatory factors.

Once clusters are discovered, we select a prototype model within each cluster to have as many prototype models as clusters. In particular, we select the most performant model within each cluster to maximize the ensemble's performance. The phenomenon modeled may have many explanations, and each prototype model is a potential explanation. In order to create the ensemble, we adopted the most straightforward combination in which each prototype model is given a weighted vote (i.e., validation error), and the label with the most votes is the prediction of the ensemble.

Figure 3.2 presents an overview of the framework proposed. The novelty in ED-Ensemble is that it promotes diversity in creating an ensemble based on competing explanations. The clusters formed proved to be coherent and concise, generating stability in prediction-explanation.

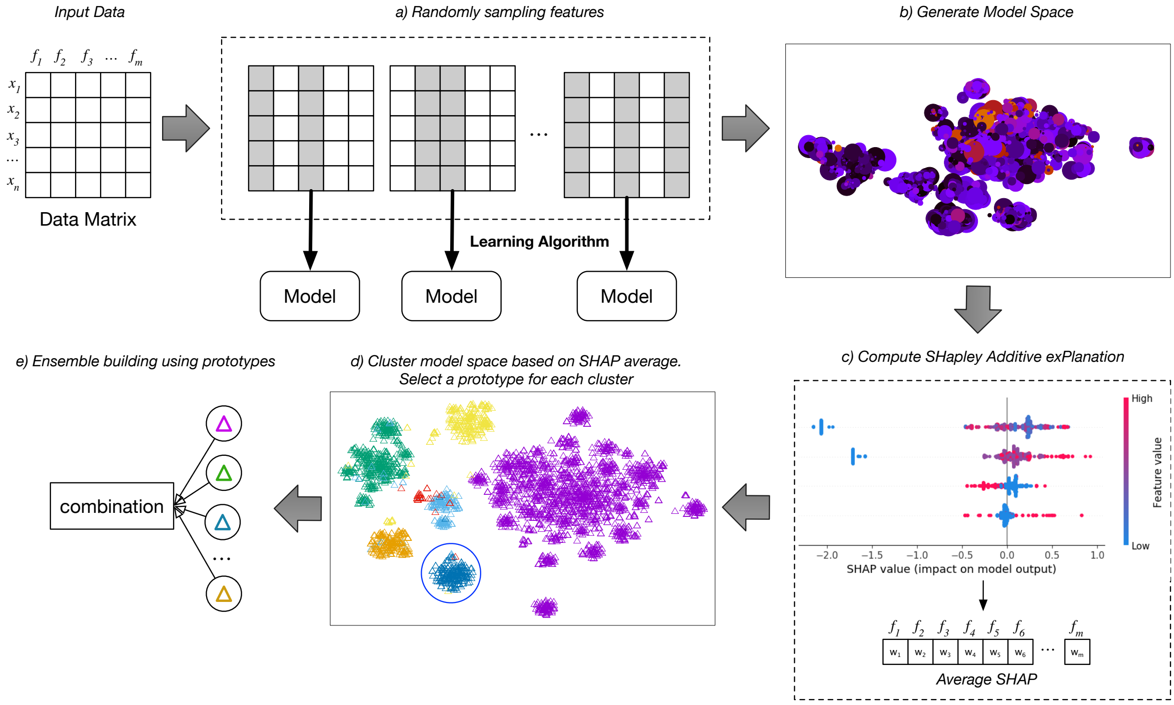

Input Data
f_1 f_2 f_3 … f_m
x_1
x_2
x_3
…
x_n
Data Matrix
a) Randomly sampling features
…
Learning Algorithm
Model
Model
Model
b) Generate Model Space
c) Compute SHapley Additive exPlanation
High
Feature value
Low
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
SHAP value (impact on model output)
f_1 f_2 f_3 f_4 f_5 f_6 f_m
w_1 w_2 w_3 w_4 w_5 w_6 … w_m
Average SHAP
d) Cluster model space based on SHAP average.
Select a prototype for each cluster
e) Ensemble building using prototypes
combination
…

Figure 3.2: (Color online) An overview of the ED-Ensemble. It starts with an input tabular matrix $n \times m$, being n the number of instances and m the number of features. a) Randomly sample features sets with sizes from 1 to f. For each set sampled, use the learning algorithm to induce a model; b) The set of all generated models will compose the model space; c) Compute the average of SHapley Additive exPlanations (SHAP) values for each model in the model space; d) Cluster the model space based on the SHAP average. Select a prototype to represent each cluster. We will choose the model with the highest AUC; e) Build an ensemble using the prototypes previously selected as base models.

Chapter 4

Case Study: Predicting the Evolution of Pain Relief

In this chapter, we discuss our case study in which it is particularly interesting to evaluate our proposed modeling approach, as it corresponds to phenomena with many complex local structures. Then we discuss our evaluation procedure and analyze the results obtained. In particular, our study aims to answer the following research questions:

RQ1: Is there a relationship between model explanation and model preferences?

RQ2: Are prototype models diverse in terms of explanatory factors?

RQ3: Can we build effective ensembles by combining models that are associated with diverse explanatory factors?

RQ4: Is our ED-Ensemble approach superior than the biclustering ensemble approach?

Pain makes us aware that something is wrong with our bodies. The International Association for the Study of Pain (IASP[1]) defines pain as *an unpleasant sensory and emotional experience associated with actual or potential tissue damage, or described in terms of such damage.* If the pain lasts beyond the time expected for healing following surgery, trauma, or other condition, it might be characterized as chronic. There is no accepted universal paradigm for chronic pain prevention and management [Navani and Li, 2016].

Chronic pain is a public health concern that affects 20−30% of the population of Western countries. It ranks among the most common diseases affecting hu-

[1] http://www.iasp-pain.org

mans [Williams and Craig, 2016] and is the most usual cause of years lived with disability worldwide [Dworkin et al., 2008a].

Despite the existence of several guidelines and recommendations for its treatment, up to 40% of chronic pain patients may remain symptomatic despite the best medical treatment. This is in part due to the heterogeneity of chronic pain mechanisms and individual variables. They are not linearly related to the etiology of pain but to the interplay of its pathophysiology, personal variables, and social context [Ferreira et al., 2016]. Chronic pain usually falls into one of the following categories:

- **Neuropathic pain:** occurs when there is actual nerve damage. Nerves connect the spinal cord to the rest of the body and allow the brain to communicate with the skin, muscles, and internal organs. Nutritional imbalance, alcoholism, toxins, infections, or autoimmunity can cause all damage to this pathway and cause pain. Neuropathic pain can also be caused by a cancer tumor pressing on a nerve or a group of nerves. People often describe this pain as a burning or heavy sensation or numbness along the affected nerve path.

- **Nociceptive pain:** is caused by damage to body tissue and is usually described as a sharp, aching, or throbbing pain. This kind of pain can be due to benign pathology, tumors, or cancer cells growing larger and crowding other body parts near the cancer site. Nociceptive pain may also be caused by cancer spreading to the bones, muscles, or joints or that causes the blockage of an organ or blood vessels.

- **Nociplastic pain:** arises from altered nociceptive function despite no clear evidence of actual or threatened tissue damage causing the activation of peripheral nociceptors or evidence for disease lesion of the somatosensory system causing the pain.

Although there have been many scientific advances in understanding the neurophysiology of pain, precisely defining the best therapy for a patient is still a challenge. Chronic pain is an individualized experience with multifactorial etiology, and understanding the biological, social, physical, and psychological contexts is vital to successful treatment. Standardized self-reported instruments and questionnaires to evaluate the patient's pain intensity, functional abilities, beliefs and expectations, and emotional distress are available. They can be used to assist in treatment planning.

4.1 Pain Data

Pain is often assessed on a scale from "no pain" to "worst pain imaginable". The Visual Analogue Scale (VAS) [Hill et al., 2008] is a 10-cm line without markings from no pain to worst pain. Patients mark their pain score, and measurement in centimeters defines their level of pain. Six hundred thirty-one participants self-completed the McGill Pain Questionnaire [Melzack, 1975] and the VAS. The McGill Questionnaire assesses both the quality and intensity of the pain. In summary, the questionnaire is composed of 78 words, of which respondents choose those that best describe their experience of pain (multiple markings are allowed). The words are organized in three dimensions:

- **Sensory dimension:** encompasses both the quality and severity of pain in terms of its temporal, spatial, pressure and thermal properties.
- **Affective dimension:** refers to feelings and sentiments in the presence of pain, that is, how the patient feels emotionally due to pain.
- **Evaluative dimension:** refers to the global evaluation of the situation experienced by the patient and is strongly influenced by previous painful experiences. It is a subjective assessment of overall pain intensity.

As a result, our pain data includes variables regarding pain severity, change in pain relief over time, pain radiation, among others. Data was also collected via self-report on socioeconomic status, global rating of overall health, known risk factors (i.e., age, smoking, alcohol intake), and concomitant illnesses. Finally, our pain data also includes the therapies prescribed by the doctor. In all, our pain data is composed of 332 variables about pain relief, socioeconomic status, and prescribed treatments.

4.1.1 Predicting the Evolution of Pain Relief

Satisfactory treatment can only come from a comprehensive assessment of the biological etiology of the pain in conjunction with the patient's specific psychosocial and behavioral presentation. A first consultation is potentially a pivotal event in a patient's pain history, affecting treatment adherence and engagement with longer-term self-management [White et al., 2016]. The objective of our retrospective study is to predict, using data obtained at the first consultation only, if a particular treatment or therapy will be effective in reducing the patient's pain relief. We evaluated three different measures to identify success in patient's pain relief:

- An overall reduction of pain intensity by 30% (aka, VAS 30), which is formally considered to be a successful treatment outcome [Dworkin et al., 2008b]. The ground truth labels are obtained by calculating the difference in pain intensities reported in the first and last consultation.
- An overall reduction of pain intensity by 50% (aka, VAS 50). Again, the ground truth labels are obtained by calculating the difference in pain intensities reported in the first and last consultation.
- The Global Impact Change (aka, GIC) as a discrete variance scale from -3 to 3 provided by the doctor indicating the degree of improvement in pain relief in the doctor's view. Success is given as a value of at least 2 in the last visit.

4.1.2 Characterization based on VAS 30

As a successful treatment outcome can be formally given by VAS 30, the analysis presented throughout this section refers explicitly to the VAS 30 label. Considering this label, we divided the 631 patients in our study into two populations:

- Population A: 277 patients for whom the treatment resulted in a significant reduction in pain relief. These patients reported a significant +30% reduction in pain relief after the treatment was completed.
- Population B: 354 patients for whom the treatment was not effective.

Table 4.1 shows the characteristics of the patients in our dataset. Pain is more prevalent in women, and it is harder to achieve significant pain reduction in patients that report low initial pain intensities. The table also shows three dimensions of pain perception. The affective dimension refers to feelings and sentiments in the presence of pain, that is, how the patient feels emotionally due to pain. The sensitive dimension encompasses both the quality and severity of pain. The evaluative dimension refers to the global evaluation of the situation experienced by the patient and is strongly influenced by previous painful experiences. Pain perceptions may overlap within the same dimension, and a total score for each dimension is given by summing up all types of pain perceptions. Table 4.1 also shows the neuropathic pain scale, which is used for assessing neuropathy pain and may be particularly useful for assessing response to therapies. The total neuropathy score is calculated as the sum of the possibilities, and the cut-off value for the diagnosis of neuropathic pain is a total score of 4. The table also shows information about pain outbreaks and the time in which pain gets worse. There are other variables that we omitted from the table due to lack of space.

Table 4.1: Patient data obtained at the first consultation. Mean, first and third quartiles within age, McGill score, initial pain intensity, and pain perception dimension scores.

	Population A	Population B
N	277 (43.89%)	354 (56.11%)
Sex (male)	110 (39.71%)	151 (42.65%)
Age, y	54.86 (46−64)	56.66 (45−60)
0−15 McGill score	7.21 (4−10)	5.75 (3−9)
0−10 intial pain intensity	6.66 (5−8)	4.80 (2−8)
Sensitive dimension	3.31 (1−5)	2.63 (1−4)
Burning	170 (61.4%)	188 (53.1%)
Painfull	131 (47.3%)	139 (39.3%)
Slapped	113 (40.8%)	115 (32.5%)
Throbbing	111 (40.1%)	104 (29.4%)
Stabbings	104 (37.5%)	96 (27.1%)
Electric shocks	100 (36.1%)	99 (27.9%)
Sharp	95 (34.3%)	102 (28.8%)
Spreads	87 (31.4%)	86 (24.3%)
Affective dimension	2.59 (1−3)	2.13 (1−3)
Tiring	209 (75.4%)	227 (64.1%)
Nauseous	186 (67.1%)	191 (53.9%)
Annoying	157 (56.7%)	166 (46.9%)
Stifling	89 (32.1%)	91 (25.7%)
Scary	74 (26.7%)	79 (22.3%)
Evaluative dimension	1.30 (1−2)	0.99 (1−1)
Unconfortable	260 (93.9%)	252 (71.2%)
Unberable	100 (36.1%)	100 (28.2%)
Neuropathic pain scale		
Burning	193 (70.4%)	220 (62.7%)
Hypoesthesia to touch	143 (48.2%)	143 (40.7%)
Numbness	109 (39.8%)	101 (28.8%)
Pins and needles	107 (39.0%)	117 (33.3%)
Tingling	89 (32.5%)	97 (27.6%)
Electric shocks	85 (31.0%)	81 (23.1%)
Painful cold	46 (16.7%)	49 (14.0%)
Brushing	40 (14.6%)	37 (10.5%)
Itchiness	28 (10.2%)	26 (7.4%)
dn4-quant ≥ 4	123 (44.9%)	120 (34.2%)
Duration of pain outbreaks		
Seconds	8 (2.9%)	3 (0.9%)
Minutes	10 (3.6%)	17 (4.8%)
Hours	19 (6.9%)	16 (4.5%)
Days	2 (0.8%)	5 (1.4%)
Weeks	1 (0.4%)	3 (0.8%)
Months	6 (2.2%)	8 (2.3%)
Not specific duration	229 (83.6%)	299 (85.2%)
Prevalence of pain sensation		
Daytime	9 (3.3%)	6 (1.7%)
Morning	4 (1.5%)	7 (2.0%)
Nocturnal	14 (5.1%)	17 (4.8%)
Afternoon	3 (1.1%)	6 (1.7%)
Not specific time	247 (89.2%)	318 (89.8%)

Figure 4.1 shows how often pain is reported in different areas of the human body. Interestingly, areas on the right side of the body are more frequently reported by patients in population B. The considered features enable a myriad of possibilities of

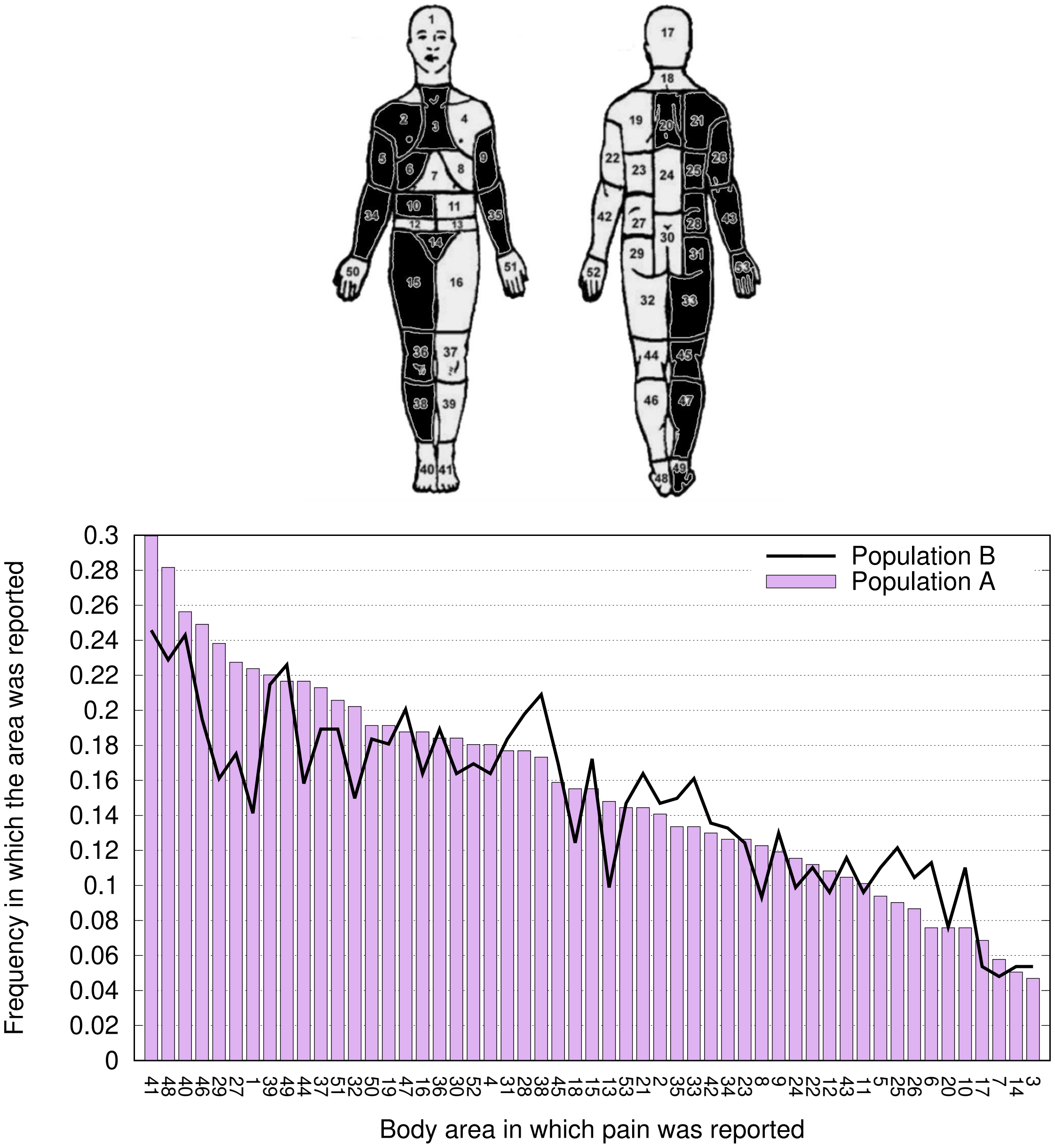


Figure 4.1: (Color online) Areas associated with pain. Top – Areas in black are more frequently reported by patients in population B than by patients in population A. Bottom – Frequency in which each area is reported by patients in populations A and B.

combining diverse aspects of pain relief while learning predictive models.

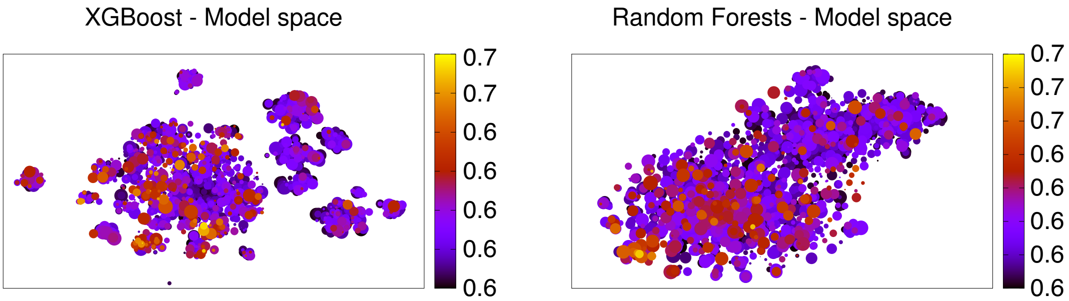


Figure 4.2: (Color online) T-SNE visualization [van der Maaten, 2009] of the sampled model space $\mathcal{H}'$. Each point represents a model $\mathbf{x}'$. Models are placed according to the probabilities assigned to patients so that models that assign similar probabilities to the same patients are placed next to each other in the space (see Section 3.3). The color indicates the average AUC value, and smaller points indicate less variance in the corresponding model.

4.2 Setup

While sampling the model space, we randomly set the number of features that compose each model. However, we assure no model has more than 15 features as a good compromise between interpretability and performance. Using this upper limit, we can test the validity and feasibility of our work. Further, it is intended to build a questionnaire to be applied by the doctor in the patient's first consultation. Restricting the number of features is also one way to limit the number of questions. Models are built using SciKit-Learn implementations of XGBoost or Random Forests algorithms [Pedregosa et al., 2011]. We sampled 150 000 models using the XGBoost algorithm and another 150 000 models using the Random Forests algorithm. To evaluate the performance of the models was used the standard AUC (area under the ROC curve) measure [Hanley and McNeil, 1982; Fawcett, 2006]. Five-fold cross-validation was conducted, and at each run, four folds are used as the training set, and the remaining fold is used as the test set. This case study also employed a separated validation set used to select the best models. Lastly, we report the average AUC value over the five runs.

The average AUC values obtained by the all-in-one model were 0.648 for XGBoost and 0.652 for Random Forests. Therefore, we consider a model as minimally performant if its average AUC value is at least 0.650. It is necessary here to clarify exactly what is meant by the all-in-one approach; we refer to XGBoost and Random Forests as all-in-one approaches when we feed the model with all features. For instance, although it is widely known that Random Forests implements some feature selection, yet all features are supplied to the model. Conversely, a non-all-in-one approach would be the case where just a subset of features is supplied.

The parameters used by the all-in-one model using learning algorithms XGBoost and Random Forests were the same used to sample the model space, and were selected empirically. Specifically, for XGBoost we set `<n_estimators=50, subsample=0.6, learning_rate=0.1 and max_depth=10>` and for Random Forest we set `<n_estimators=10 and max_depth=10>`. The ommited parameters were set with the default values.

While this performance threshold seems low, it greatly exceeds the estimated physician performance at the first consultation, which is no higher than 0.438. We consider the physician's performance as being the known outcome of the latest consultation. The close to the random performance of physicians at the first consultation reveals how difficult this predictive task is. The 0.650 performance threshold resulted in a sampled model space $\mathcal{H}'$ for XGBoost and another model space for Random Forests. The XGBoost model space comprises 2 830 models out of the original 150 000 models (1.9% of the models perform better than the all-in-one model). In contrast, the Random Forests model space comprises 2 507 models (1.7% of the models perform better than the all-in-one model). Figure 4.2 shows XGBoost and Random Forests model spaces. Each point matches a model, and the size of the point indicates the variance of the validation error. In addition, the color scale is associated with the performance achieved by the model, with light colors performing best. Thus, in the figure, the best models are shown as clearer and smaller points. The figure shows that the best models are well scattered through the model space, indicating models with different preferences but equally performant.

Additionally, we also carried out experiments with Tree-based Pipeline Optimization Tool[2] (TPOT). TPOT is a tool that optimizes the machine learning pipeline using genetic programming. We set up the time limit for optimization as 24 hours, as this is the approximate amount of time in our worst-case to run our approach. The average AUC obtained using VAS 30 as the label was 0.632. In this case, it was selected the XGBoost classifier. Interestingly, for VAS 30, the best-selected machine learning approach is a single model, even though TPOT can generate stacking setups.

4.2.1 Limitations

Currently, the main framework limitation is on the computing SHAP values. The two tree-based learning algorithms used in this work (Random Forests and Gradi-

[2] http://epistasislab.github.io/tpot/

ent Boosted Trees) rely upon TreeSHAP. Whereas KernelSHAP is a model-agnostic method to compute Shapley values, it can be slow and suffer from sampling variability. By focusing specifically on trees, TreeSHAP computes local explanations based on exact Shapley values in polynomial time [Lundberg et al., 2020]. Working with models that are KernelSHAP reliant on computing local explanations is not feasible yet, as KernelSHAP is two orders of magnitude slower than TreeSHAP. In addition, computational costs are also inherent to the framework since approximately 150 000 models are generated.

Another limitation in our approach is the various parameter settings and their interplay. Examples of parameters are the maximum number of features, the number of models generated for each feature size, minimum AUC threshold, XGBoost model parameters, Random Forest model parameters, and parameters to perform clustering. We seek to adjust the parameters through empirical evaluations in our case study. At the same time, we tried to sufficiently substantiate the choice of the approach's specific parameters, aiming to apply it to diverse problems.

4.2.2 A Note on Time Requirements

Although the time spent generating the ensembles does not influence the benefits shown throughout the work, such as the increase of the AUC and reduced feature set, it can be a critical factor in applying the technique in miscellaneous use cases. Almost the entire runtime contribution comes from generating the model space with 150 000 sampled models. As we increase the maximum number of features allowed, we can obtain ensembles with improved performance. Conversely, it is also expected to increase the computational cost. As interpretability is a crucial aspect of our work, we set the upper limit to 15 features. The total time spent sampling the entire model space with an upper limit of 15 features was 1353.78 minutes with XGBoost and only 249.01 minutes with Random Forests. Both cases use VAS 30 label. Generating the ensemble from the sampled model space takes less than 60 minutes for all configurations and learning algorithms. The specifications related to the hardware are: Intel® Core$^{\text{TM}}$ i3-6100 CPU @ 3.70GHz, 16GB DDR3 1600 MT/s, and 256GB SSD. As the algorithms used did not make use of the graphics card, we will omit the information. A more updated and detailed description of computational costs along with hardware and software specifications can be found in Appendix A.

4.3 Explanatory Factors

We are interested in evaluating four criteria for diversity: predictions, probabilities, features, and SHAP (explanatory factors). As the first step, we would like to measure how such criteria are correlated to each other. **Pearson correlation coefficient**, also known as **Pearson's r**, is a standard measure of linear correlation between two sets of data. However, it can not be used directly, as our data sets are of different dimensions.

Table 4.2 shows the dimensions for each criterion, considering an individual instance. An instance has a dimension of 271 for both the probabilities and predictions criteria. Comparatively, features and SHAP criteria result in a dimension of 332. Interestingly, when using the XGBoost learning algorithm, there are numerous SHAP explanatory factors with zeroes, meaning these features did not influence the outcome. Out of 332 features, 109 did not had any influence. Hence, the real dimension is 223. Also, note that 271 is different from the total number of instances 632, as we considered only predictions and probabilities from the instances with the true label 1.

Table 4.2: Criterion dimension for an individual instance.

Probabilities	Predictions	Features	SHAP
271	271	332	332 (223)[3]

In order to estimate the correlation, we built an adjacency matrix from the model space generated in Figure 4.2, filling in the matrix with the euclidean distance between the models. With this approach, we can quantify whether the distance between two models' probabilities vector increases as the distance considering other criteria as predictions, features, and SHAP increases likewise. As a result, Figure 4.3 shows the correlation matrix obtained from the model space using XGBoost and Random Forests learning algorithms.

Interestingly, for the XGBoost learning algorithm, the SHAP explanatory factors correlate more with predictions (0.75) than the probabilities with the prediction (0.66). Indeed, the two most correlated criteria are SHAP explanatory factors and predictions. It is also surprising that SHAP explanatory factors are more correlated with predictions (0.75) and probabilities (0.44) than with features (0.23).

A similar trend can be observed considering the Random Forest learning algorithm. SHAP explanatory factors are more similar with predictions (0.4) and prob-

[3] For the XGBoost learning algorithm, some features do not influence the outcome. So, 223 is the actual dimension if considering features influencing the outcome only.

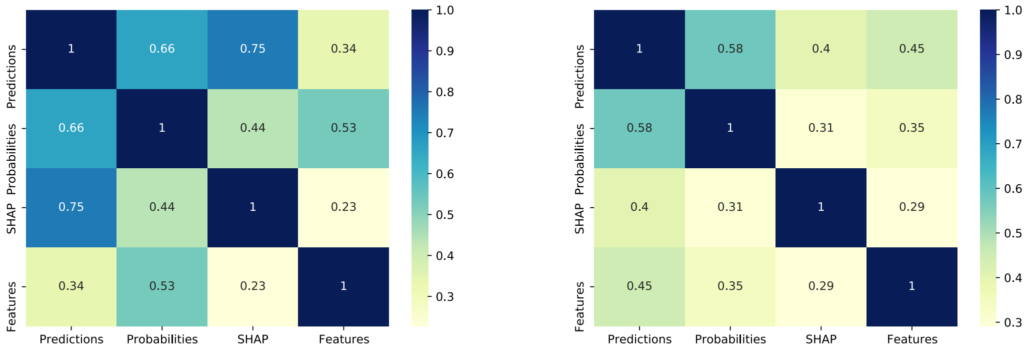


Figure 4.3: (Color online) Correlation matrix built using the adjacency matrix based on different criteria. The left image is built from model space generated using the XGBoost learning algorithm. The right image is built from model space generated using Random Forests learning algorithm.

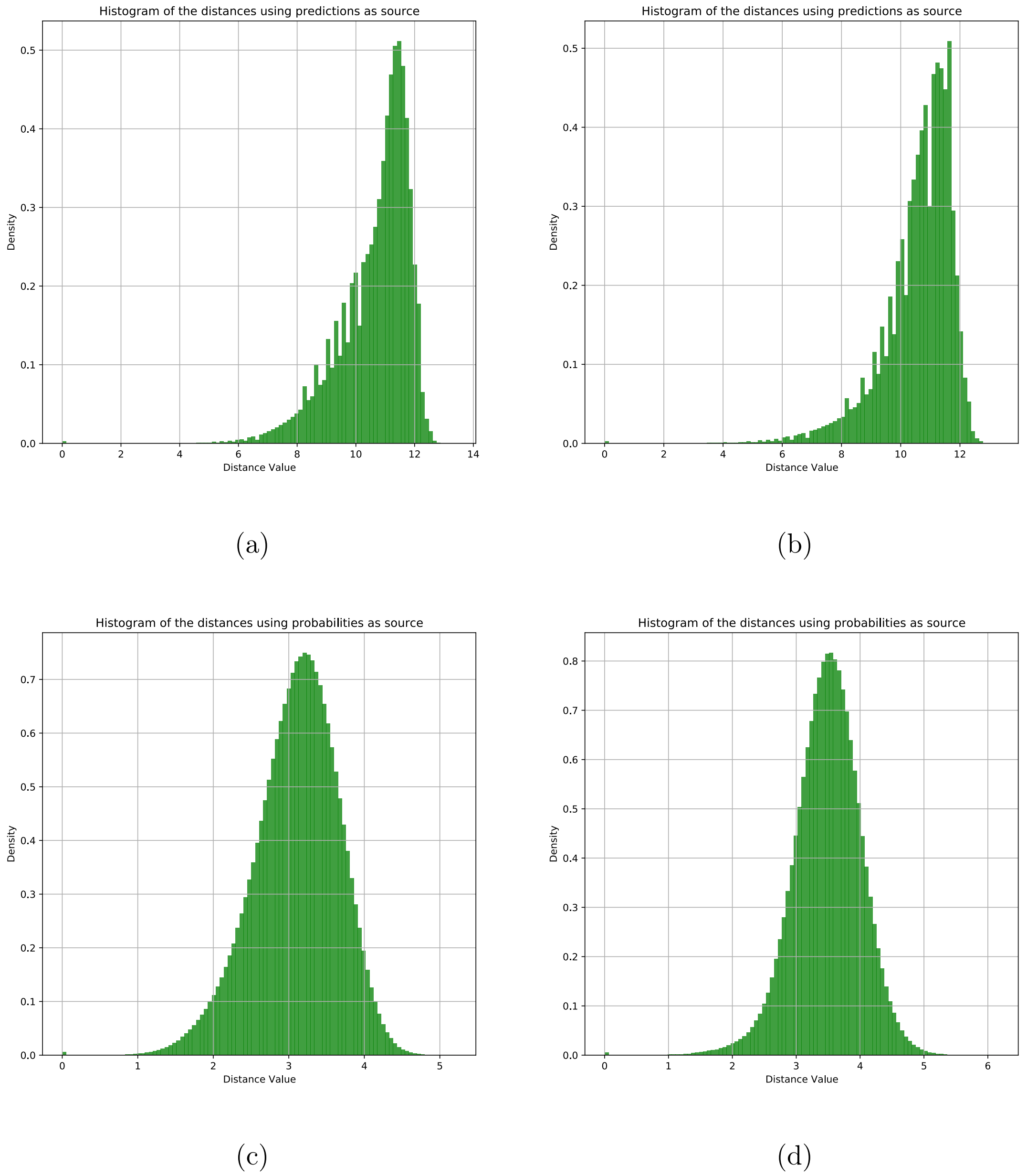
abilities (0.35) than with features (0.29). However, the most correlated criteria are probabilities and predictions in this case, with a value of 0.58.

Figures 4.4 and 4.5 presents the histogram charts of different criteria and learning algorithms. In particular, Figure 4.4 shows the histogram plots when using predictions (Figures 4.4a and 4.4b) and probabilities (Figures 4.4c and 4.4d) criteria. Images on the left use the XGBoost learning algorithm, while the right uses the Random Forest learning algorithm. Regarding the learning algorithm, the generated plots are very similar. Using probabilities criterion clearly results in a normal distribution.

Figure 4.5 presents the histogram plots employing features (Figures 4.5a and 4.5b) and SHAP explanatory factors (Figures 4.5c and 4.5d) criteria. Once again, the histogram plot using features criterion is very similar regardless of the learning algorithm. Interestingly, the SHAP explanatory factors are the only ones resulting in a very different histogram plot. For the XGBoost learning algorithm, the generated histogram plot is a binomial distribution. In the histogram plot using the Random Forest learning algorithm, the resulting plot is normal distribution, albeit not symmetric.

In summary, SHAP explanatory factors present the following benefits:

- It is more correlated to probabilities and predictions than features;
- It has a dimension size less than or equals the features criterion dimension, while probabilities and predictions dimensions equal the number of instances. In general, the number of instances surpasses the features by a large margin;
- It provides more knowledge than features criterion. Additionally to the information on whether a feature is present, it estimates how much each feature con-



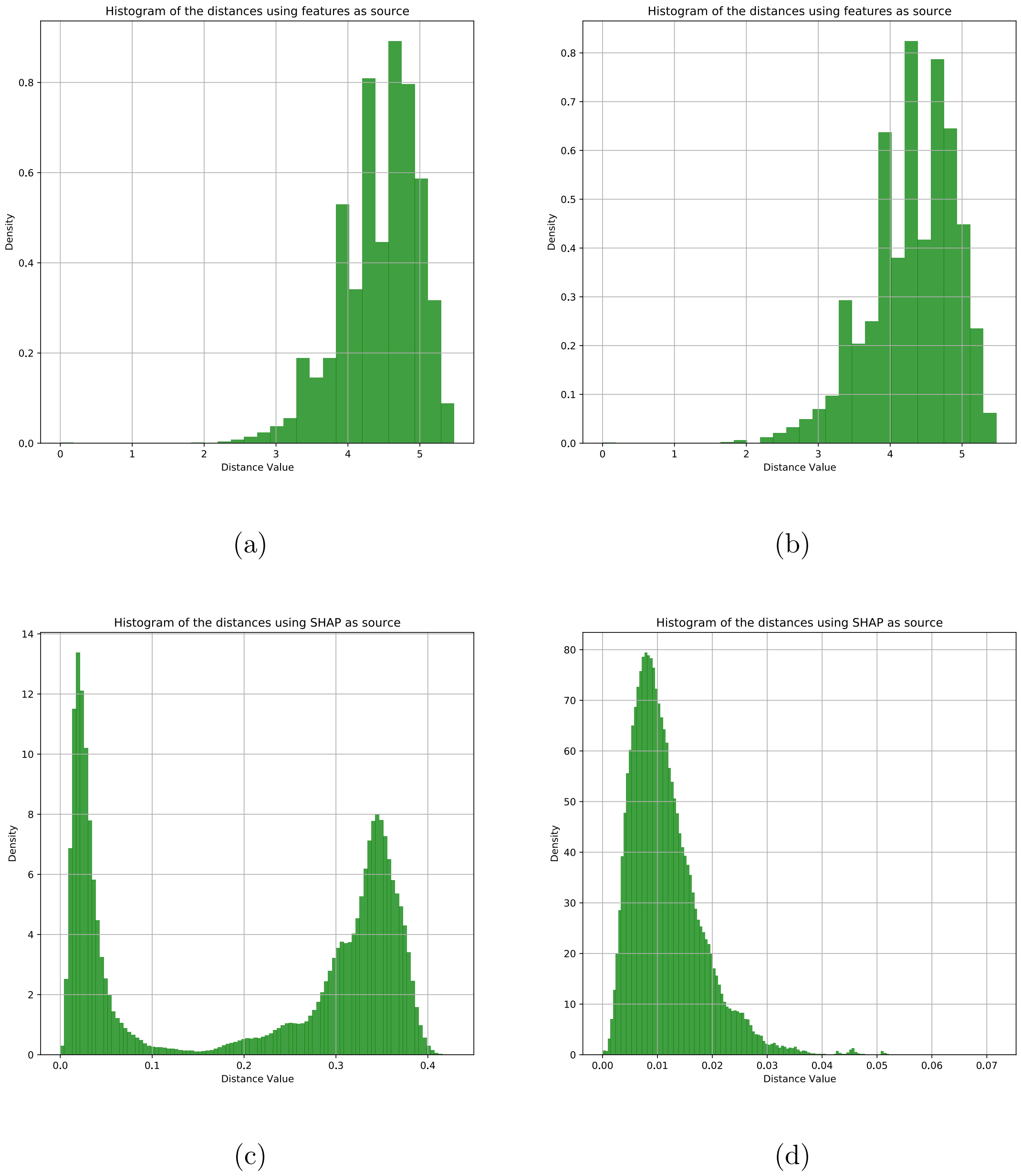
(a) (b)

(c) (d)

Figure 4.4: (Color online) Histogram plotting using criteria probabilities and predictions. Plots on the left, (a) and (c), make use of the XGBoost learning algorithm. Plots on the right, (b) and (d), make use of the Random Forests learning algorithm.

tributed to the model outcome;

- It enables identifying features that, while being supplied to the models, are simply ignored. These features can be disregarded, thus reducing the dimension.



(a)

(b)

(c)

(d)

Figure 4.5: (Color online) Histogram plotting using criteria features and SHAP (explanatory factors). Plots on the left, (a) and (c), make use of the XGBoost learning algorithm. Plots on the right, (b) and (d), make use of the Random Forests learning algorithm.

4.4 Relating Model Preferences and Explanatory Factors

In order to answer RQ1, we embedded XGBoost and Random Forests models according to their model preferences (i.e., probabilities they assign to the data points). Thus, models assigning similar probabilities to the same data points are placed next to each other in the model space (as in Figure 4.2). Next, we clustered the model space using different criteria and clustering algorithms. We employed Hierarchical clustering [Ward, 1963] and DBScan [Ester et al., 1996] algorithms, and all hyper-parameters were set by maximizing the silhouette value considering the model preference space. These two clustering algorithms represent two distinct ways to cluster data. The dendrogram is a tree-based clustering algorithm that partitions the given data rather than the entire instance space. On the other hand, DBScan is a density-based clustering connecting points within certain distances thresholds only when satisfying a density criterion.

In this paragraph is presented the silhouette value formal definition obtained from the book *Machine Learning* from Flach [2012]. For any data point x_i, let $d(x_i, D_j)$ denote the average distance of x_i to the data points in cluster D_j, and let $j(i)$ denote the index of the cluster that x_i belongs to. Furthermore, let $a(x_i) = d(x_i, D_{j(i)})$ be the average distance of x_i to the points in its own cluster $D_{j(i)}$, and let $b(x_i) = \min_{k \neq j(i)} d(x_i, D_k)$ be the average distance to the points in its neighbouring cluster. We would expect $a(x_i)$ to be considerably smaller than $b(x_i)$, but this cannot be guaranteed. So we can take the difference $b(x_i) - a(x_i)$ as an indication of how 'well-clustered' x_i is, and divide this by $b(x_i)$ to obtain a number less than or equal to 1. This leads to the following definition:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\left(a(x_i), b(x_i)\right)} \tag{4.1}$$

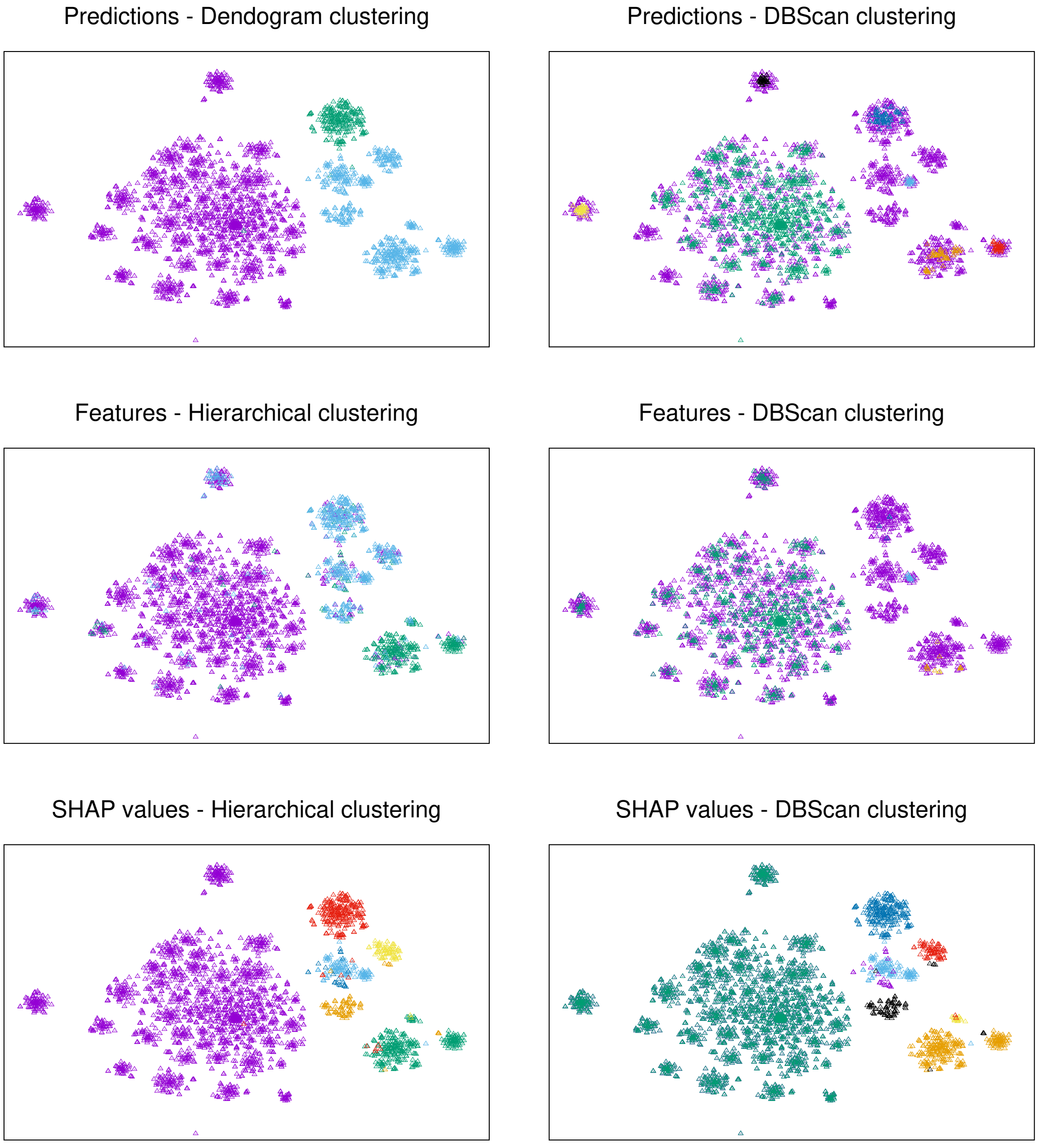
The silhouette value measures how similar a data point is to its cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to $+1$, where a high value indicates that the data point is well matched to its cluster and poorly matched to neighboring clusters. If most data points have a high value, then the clustering configuration is appropriate. The silhouette value considered in this work is the mean silhouette value over all samples.

Figure 4.6 shows 2D T-SNE visualization for the XGBoost model space after being clustered using different criteria. T-SNE was only used for the sake of visualization, and all clusters were estimated in the original $n-$dimensional model space. Firstly, for the sake of comparison, we clustered the model space using the predictions performed by each model so that models that perform the same predictions for the same data points are more likely to be associated with the same cluster. In this case, explanatory factors were not used. While hierarchical clustering leads to cohesion, it lacks performance in terms of separation. The opposite trend is observed for DBScan clusters. Both Hierarchical clustering and DBScan achieved low silhouette values. Specifically, Hierarchical clustering achieved a 0.17 silhouette value, while DBScan achieved only a 0.01 silhouette value. The low silhouette values, especially for DBScan, may be due to similar probabilities associated with opposite predictions. Small differences between probabilities that are close to the threshold may lead to opposite predictions. Hence models may be evaluated as similar in terms of model preference but different in terms of their predictions.

Furthermore, for the sake of comparison, Figure 4.6 shows the XGBoost model space clustered using the indexes of the features within each model. The purpose is to evaluate to what extent a specific set of features may be associated with a particular local structure in the data space. The problem with this clustering criterion (i.e., the features within the model) is that it neglects that different features may be correlated. Models may have similar preferences even if they are entirely different in terms of their features. Again, this leads to poor clustering performance. Precisely, Hierarchical clustering achieved a 0.05 silhouette value, while DBScan achieved only a 0.03 silhouette value.

Finally, we evaluate our proposed approach of clustering the model space based on the explanatory factors associated with each model. As detailed in Section 3.4, we represent each model as a vector composed of the SHAP values associated with the factors explaining the model decisions. Interestingly, clustering based on explanatory factors results in groups with very high values of cohesion and separation, suggesting a strong link between model preferences and model explanation. Another advantage of clustering models as vectors of SHAP values is that the importance of each factor is divided if the model contains correlated features. In particular, our approach avoids a systematic instability in which similar models in terms of their preferences can have very different explanations. Consequently, silhouette values are as high as 0.83 for Hierarchical clustering and 0.95 for DBScan. The figure shows minor differences in the configuration of groups obtained by both clustering algorithms. The data analysis allows confirming the hypothesis defined at RQ1, considering the XGBoost model space, as the high silhouette values for both clustering algorithms indicate a relationship between model preferences and model explanation.

Similarly, Figure 4.7 shows 2D T-SNE visualization for the Random Forests model space after being clustered using different criteria. Similar trend was observed. Again, for the sake of comparison, we clustered the model space using the predictions performed by each model. In this case, again, both Hierarchical clustering and DBScan achieved low silhouette values. Specifically, Hierarchical clustering achieved a 0.06 silhouette value, while DBScan achieved a -0.33 silhouette value. Clustering the model space using the distance between the feature sets within each model leads to poor co-



Predictions - Dendogram clustering

Predictions - DBScan clustering

Features - Hierarchical clustering

Features - DBScan clustering

SHAP values - Hierarchical clustering

SHAP values - DBScan clustering

Figure 4.6: (Color online) T-SNE visualization of the model space after being clustered using different clustering algorithms and different criteria. Different colors mark different clusters. All clustering parameters were selected by maximizing the silhouette value in the model preference space. Models were built using Extreme Gradient Boosting (aka XGBoost). Silhouette values: Predictions = (0.17, 0.01); Features = (0.05, 0.03); SHAP values = (0.83, 0.95), respectively for Dendrogram clustering and DBScan clustering.

hesion and separation. Hierarchical clustering achieved a 0.04 silhouette value, while DBScan achieved a 0.06 silhouette value. Again, clustering based on explanatory factors results in groups with very high values of cohesion and separation. Silhouette values are as high as 0.87 for Hierarchical clustering and 0.98 for DBScan. Therefore, our answer to RQ1 is also positive considering the Random Forests model space, as the high silhouette values for both clustering algorithms indicate a relationship between model preferences and model explanation.

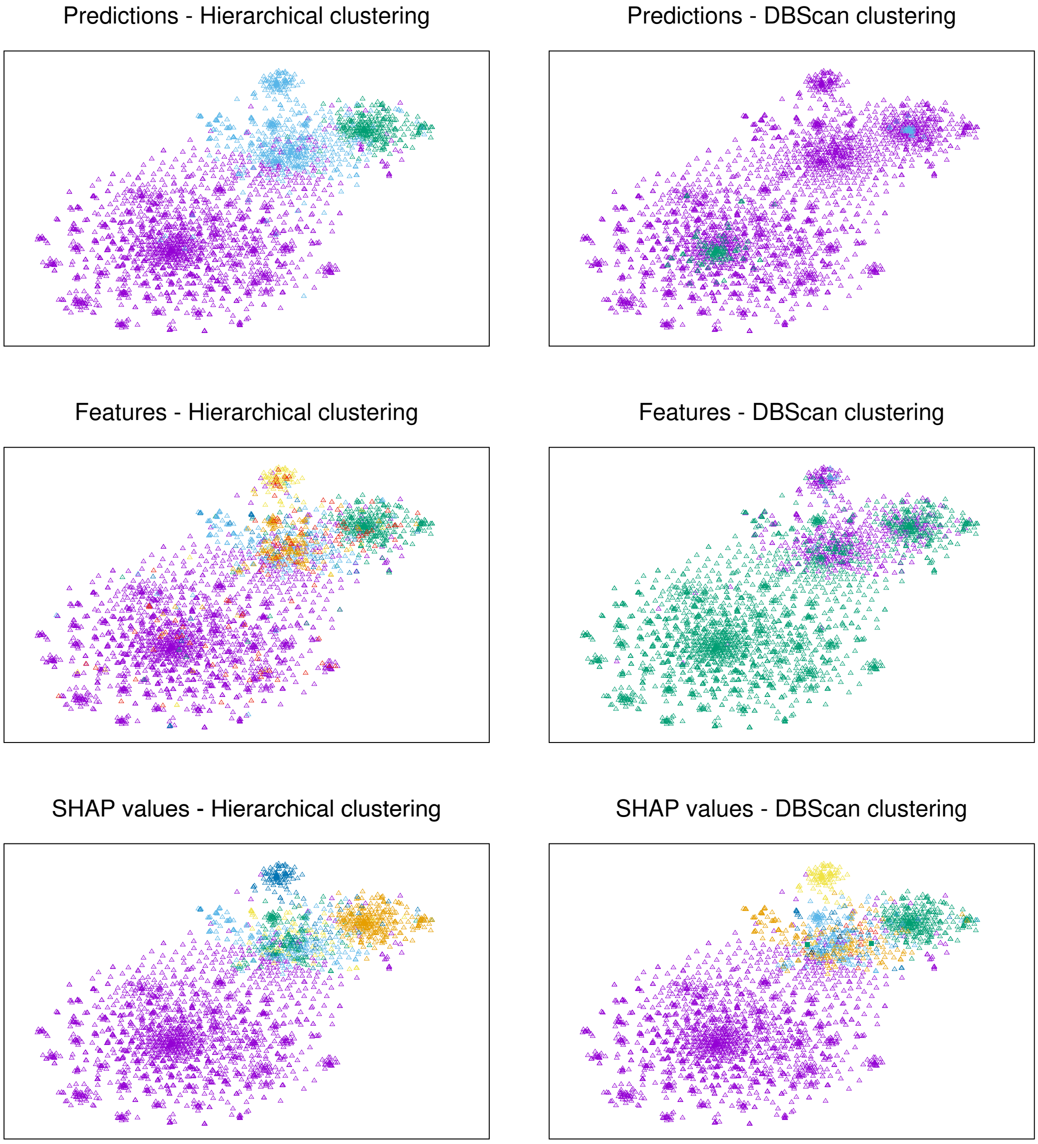


Figure 4.7: (Color online) T-SNE visualization of the model space after being clustered using different clustering algorithms and different criteria. Different colors mark different clusters. All clustering parameters were selected by maximizing the silhouette value in the model preference space. Models were built using Random Forests. Silhouette values: Predictions = (0.06, -0.33); Features = (0.04, 0.06); SHAP values = (0.87, 0.98), respectively for Dendrogram clustering and DBScan clustering.

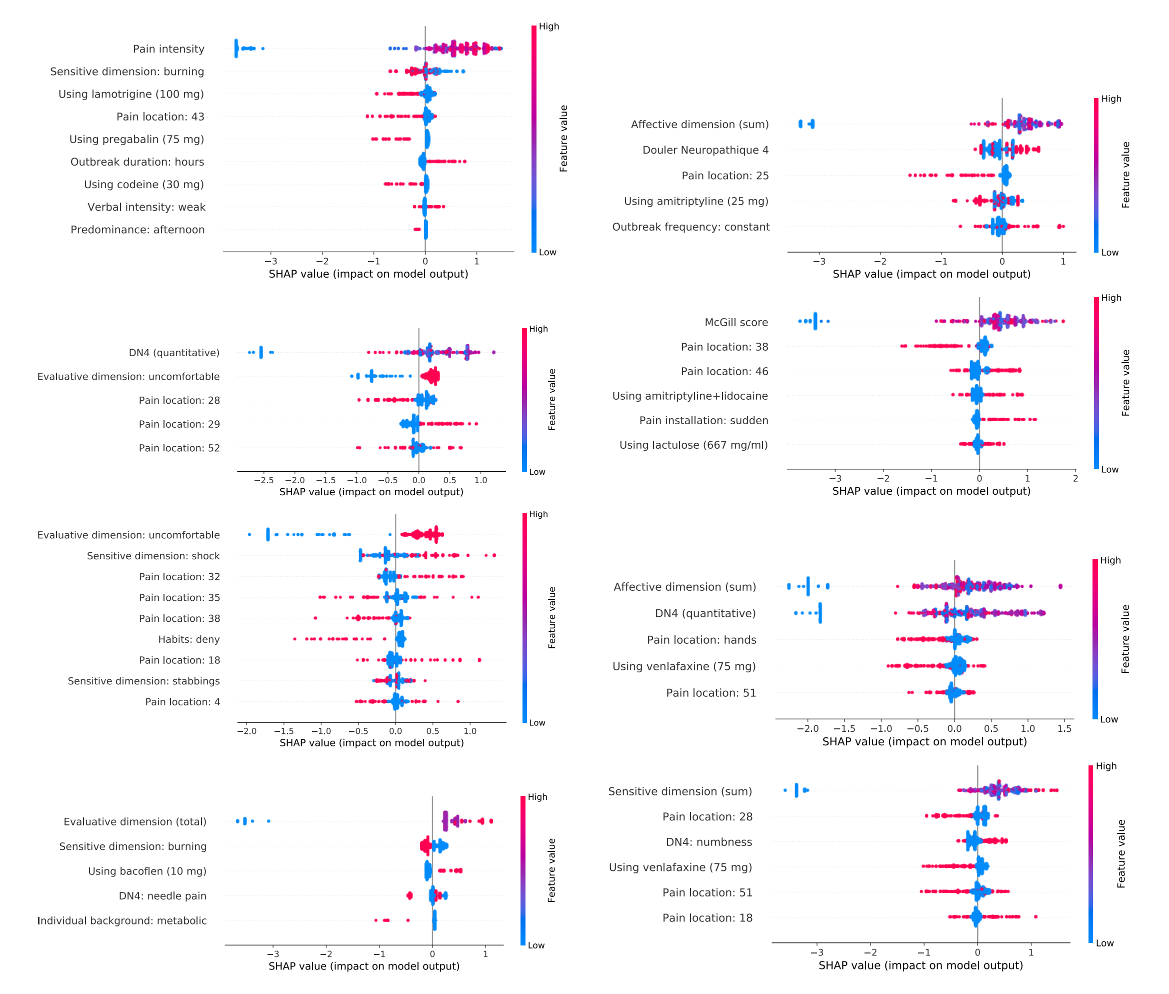


Figure 4.8: (Color online) Explanation factors (viewed as SHAP summary plots) associated with prototype models. Models were built using XGBoost.

4.5 Backbone Structure, Explanatory Factors and Diversity

In order to answer RQ2, we inspected the prototype models within each cluster in the XGBoost model space. We focus on clusters based on explanation vectors produced by DBScan. Figure 4.8 shows SHAP summary plots [4] associated with prototype models, giving an overview of which features are most important for a model. For instance, the first plot (top, leftmost) shows that a high initial pain intensity increases the chances of significant pain reduction at the end of the treatment. Typically, the most important

[4] These summary plots show the SHAP values of every feature for every data point. In each plot, features are sorted by the sum of SHAP value magnitudes over all data points. The color represents the feature value.

feature within a model is a backbone feature, such as a pain dimension or a pain scale. Then the model includes features that are somehow related to the backbone feature, such as a specific location or a particular medication. As a result, models differ significantly in terms of their explanatory factors. Diversity becomes clear as we inspect the prototype models, as each model employs a set of features that is very different from the features used by the other prototype models. Specifically, there are 41 distinct features within the eight prototype models, and only nine features are present in the two models.

The number of 41 unique features represents, precisely, 9.29% of the total number of features (332 in total). It should also be noted that the performance increased despite a notable reduction in the number of features. The reduced number of features yields many benefits, with one crucial benefit in medical area applications: it narrows the gap between ethics and its use in real-life situations. Combining with the SHAP technique allows the doctor to visualize the most critical factors contributing to the model outcome. Otherwise, the contributions would be shared with many features with the entire set of features, making it very difficult to understand how the algorithm's decision is being made. When working with a small subset of features, SHAP works much better.

Figure 4.9 shows the SHAP decision plots of two prototype models previously shown in Figure 4.8. SHAP decision plots show how complex models arrive at their predictions. Each line presents a model decision path given an input. This different view is effective mainly when many important features are involved. As can be seen, models using different features are different ways to achieve the prediction.

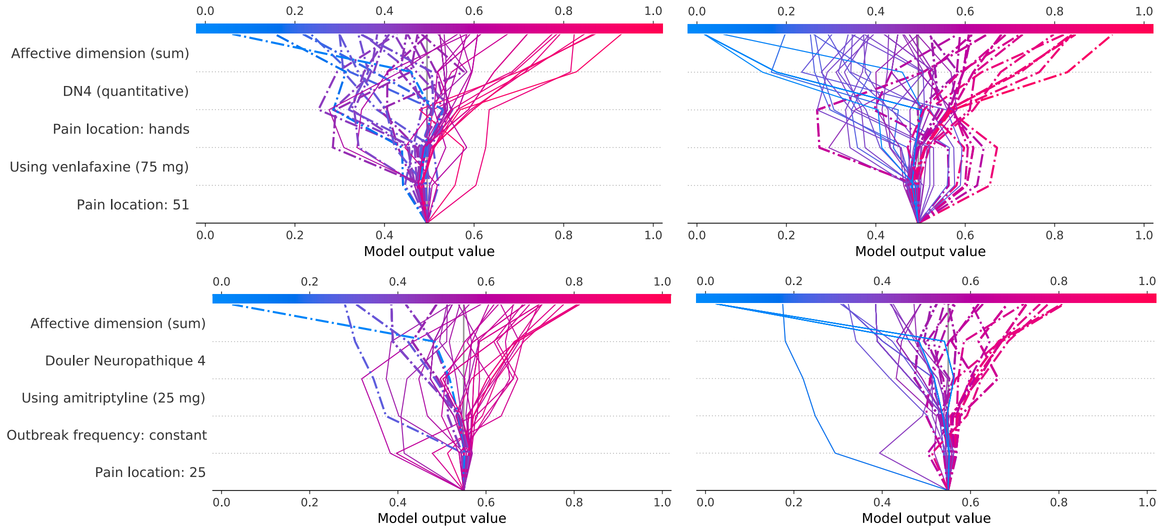


Figure 4.9: (Color online) SHAP decision plots for two models are shown in Figure 4.8. Left – True positives (line) vs False negatives (dashed line). Right – True negatives (line) vs False positives (dashed in line).

We also inspected the prototype models within each cluster in the Random Forests model space. We focus on clusters based on explanation vectors produced by DBScan. Figure 4.10 shows SHAP summary plots associated with prototype models, giving an overview of which features are most important for a model. Again, models differ greatly depending on pain dimension, location of the pain, pain duration, and medication. Diversity is also observed in these prototype models. Specifically, there are 45 distinct features within the ten prototype models, from which six features are present in two models and only two features are present in three models. Thus, our answer to RQ2 is positive considering the XGBoost and the Random Forests model spaces, as prototype models differ significantly in terms of the features being used.

As the models were selected by maximizing explanation diversity, the number of shared features is expected to be small. The most relevant features for each prototype can be obtained directly from their SHAP values (i.e., the higher the SHAP value, the more important is the feature). The most relevant features within the final model would be the combination of the most relevant features within its prototype models. Using VAS30 as the label, XGBoost as the learning algorithm, and DBScan as the clustering algorithm, the most relevant features of each prototype model are: pain intensity, evaluative dimension, affective, DN4 quantitative, evaluative dimension uncomfortable, McGill and sensitive dimension.

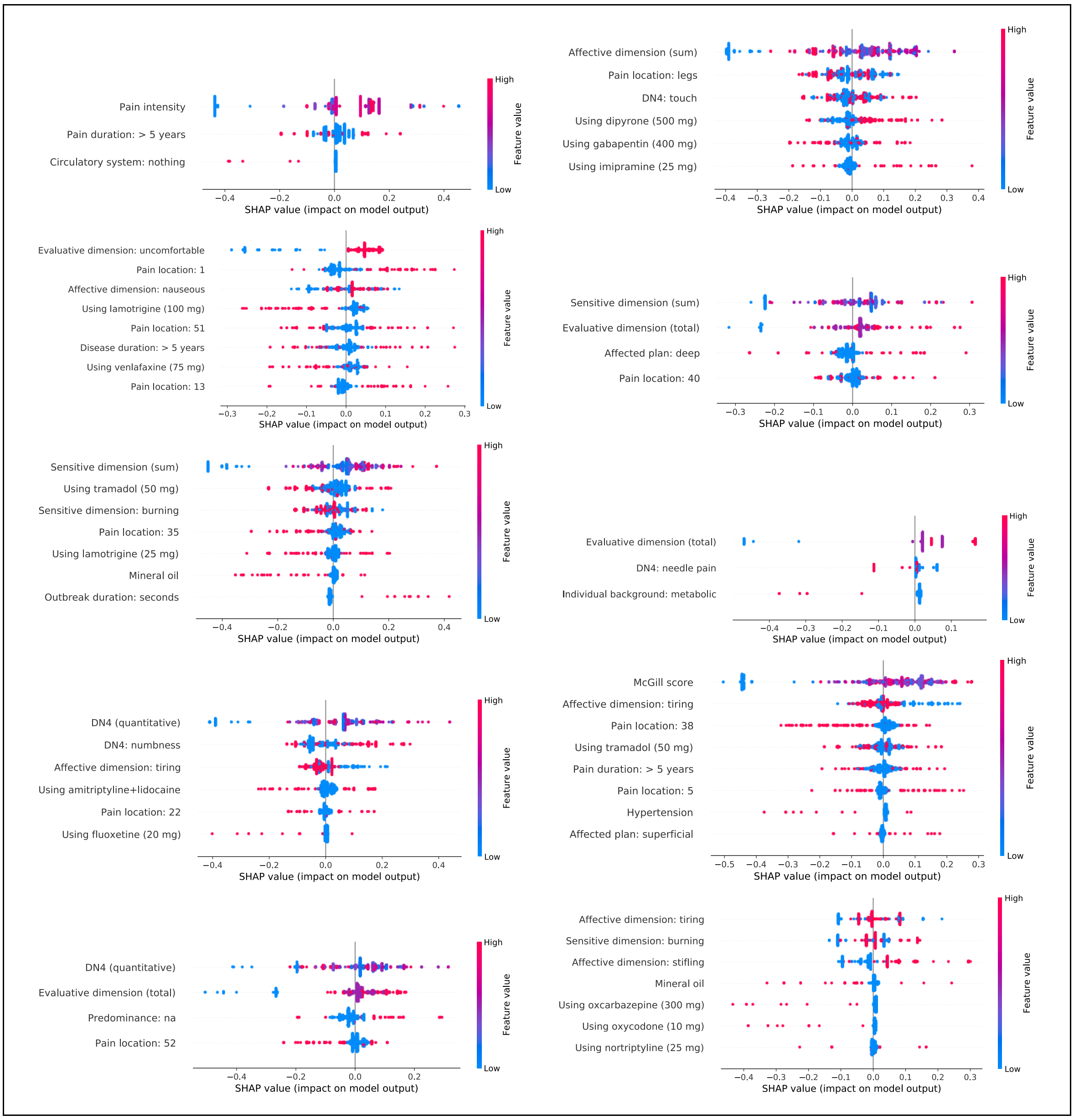


Figure 4.10: (Color online) Explanation factors (viewed as SHAP summary plots) associated with prototype models. Models were built using Random Forests.

4.6 Ensemble Performance

The next set of experiments is devoted to answering RQ3. Table 4.3 shows AUC values for different ensemble configurations using VAS 30 label. The ensemble's performance is compared to the performance of the best local model in the model space and the all-in-one approach. Different ensembles configurations achieved AUC values that range from 0.68 to 0.78. Ensembles obtained from Random Forests models provide gains up to 4.17% compared to the best local model and up to 15.03% compared to the all-in-one

Table 4.3: Ensemble performance for different clustering criteria and clustering algorithms using VAS 30 label. Baseline AUC values for the best local model for XGBoost was 0.71 and for Random Forests 0.72. Baseline AUC values for the all-in-one approach for XGBoost was 0.648 and for Random Forests 0.652.

		XGBoost			Random Forests		
Criterion	Clustering	AUC	Gain Best	Gain all-in-one	AUC	Gain Best	Gain all-in-one
Predictions	DBScan	0.73	2.82%	12.65%	0.68	-5.55%	4.29%
Predictions	Hierarchical	0.73	2.82%	12.65%	0.73	1.39%	11.96%
Feature values	DBScan	0.71	–	9.57%	0.70	-2.78%	7.36%
Feature values	Hierarchical	0.72	1.51%	11.11%	0.74	2.78%	13.50%
Explanations	DBScan	**0.78**	**9.86%**	**20.37%**	0.75	4.17%	15.03%
Explanations	Hierarchical	0.77	8.45%	18.83%	0.75	4.17%	15.03%

approach. However, for some ensemble configurations, the performance deteriorated. Ensembles obtained from XGBoost models were more effective, providing gains up to 9.86% compared to the best local model and up to 20.37% compared to the all-in-one approach.

For the VAS 50 and GIC labels, similar experiments of our proposal are provided in Appendix B. In summary, the results also showed a marked improvement in the performance of the generated ensemble for all considered labels.

Our proposed learning ensembles by clustering the model space using explanatory factors were always compelling, providing significant gains despite the ensemble configuration. Thus, our answer to RQ3 is definitely positive.

4.7 Comparison with Biclustering Performance

The set of experiments in this section is devoted to answering RQ4. For this, we consider as baseline the BENCH (Biclustering-driven ENsemble of Classifiers) method proposed in Pansombut et al. [2011], which constructs an ensemble of classifiers through concurrent feature and data point selection guided by biclustering. Figure 4.11 shows ROC curves for BENCH, XGBoost+SHAP with DBScan and Random Forests+SHAP with DBScan. ED-Ensemble outperform BENCH in all ranges of false positive and true positive rates. We performed Welch's t-tests with $p = 0.01$, and both ensemble configurations are statistically different from BENCH, and thus our answer to RQ4 is also positive.

The following chapter presents an in-depth analysis of the accuracy metric for the

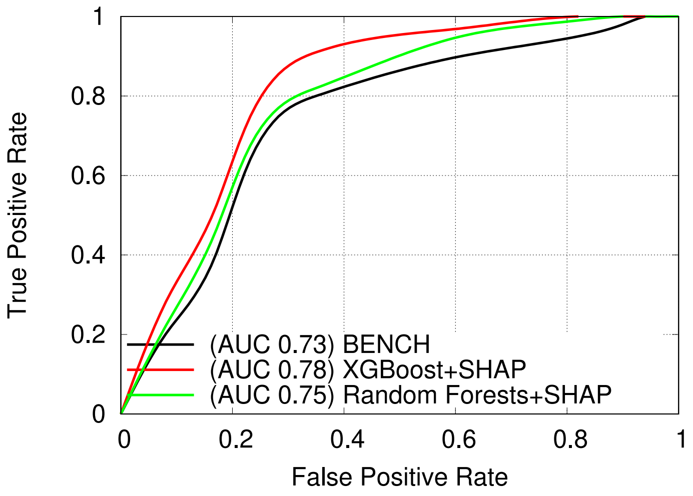


Figure 4.11: (Color online) ROC curve comparing different ensemble approaches.

proposed ensemble under different conditions. The total error is decomposed into bias plus variance, allowing to determine the ensemble accuracy and total error behavior.

Chapter 5

Bias and Variance Analysis

This chapter analyzes the bias and variance of the constructed ED-Ensemble, characterizing and describing their behavior under different conditions. We begin with the formulation of our problem in terms of bias and variance, providing a decomposition of the error using these two terms. Next, a setup is presented to define how the measurements are obtained. Finally, we discuss our evaluation procedure and report the results. In particular, our study aim to answer the following research questions:

RQ1: The generated ED-Ensemble provides consistent improvements under the evaluation metric AUC. Do the improvements extend to accuracy? How do bias and variance perform in the generated ensemble contrasting with the all-in-one alternative?

RQ2: How does the number of features impact the performance error?

RQ3: The learning algorithms used in this work are based on the techniques of bagging and boosting. It is well known that bagging reduces the total error mainly through the decrease in variance error, while boosting reduces the total error mainly through the decrease in bias error. How do bias and variance behave in the ensemble in comparison to the base models?

RQ4: Are the proposed ED-Ensembles consistently superior in performance to randomly generated ensembles?

5.1 Problem definition

Several are the factors that interact with the construction of good ensembles. Notably, the relationship between diversity and accuracy of the base learners is recognized as one key factor [Valentini and Dietterich, 2004; Kuncheva et al., 2001]. This thesis presented an approach to construct ensembles based on the selection of base models using diversity criteria in competing explanations. Now, we turn our attention to present an analysis of the accuracy of the base learner and formed ensemble. The decomposed error analysis of the base models and the generated ensemble lets us clarify how the improvement is achieved.

The bias plus variance decomposition provides a way to analyze the total error under the terms bias, variance, and an intrinsic target noise. The three terms can be defined as:

Bias: measures the difference between the average predictions made by a model ($\hat{y}$) over distinct training sets of a given size and the true predictions (y).

Variance: measures how much the predictions vary around the average for different training sets of the given size.

Intrinsic noise: is the noise independent of the learning algorithm.

Originally formulated for least-squares regression, the decomposition is a well-established analysis borrowed from statistics. Conversely, given some implications, this decomposition can not be automatically extended for the classification problem. There are a few works along the years with proposals in decomposition for 0-1 loss, notably Kohavi and Wolpert [1996] and Domingos [2000].

5.2 Bias and Variance Decomposition for Mean Squared Error

This section will address the bias plus variance decomposition considering the error function Mean Squared Error (MSE). The decomposition presented in this section is inspired by the decomposition presented by Vijayakumar [2007].

Let $f(x) = f$ be the true function we want to approximate. Next, the dataset for training is defined as $D = \{(x_1, t_1), (x_2, t_2), \dots (x_N, t_N)\}$ where $t = f + \epsilon$ and $E[\epsilon] = 0$. Given D, we train a model to approximate the function f by $y = g(x, w)$. The mean-squared error is:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (t_i - y_i)^2 \tag{5.1}$$

To assess the model's effectiveness, we want to know the expectation of the MSE if we test the model on arbitrary many test points drawn from the unknown function.

$$E[MSE] = E[\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2] = \frac{1}{N}\sum_{i=1}^{N}E[(t_i - y_i)^2] \tag{5.2}$$

Following:

$$\begin{aligned} E[(t_i - y_i)^2] &= E[(t_i - f_i + f_i - y_i)^2] \\ &= E[(t_i - f_i)^2] + E[(f_i - y_i)^2] + 2E[(t_i - f_i)(f_i - y_i)] \\ &= E[(t_i - f_i)^2] + E[(f_i - y_i)^2] + 2E[(f_i - y_i)(t_i - f_i)] \\ &= E[\epsilon^2] + E[(f_i - y_i)^2] + 2(E[f_i t_i] - E[f_i^2] - E[y_i t_i] + E[y_i f_i]) \end{aligned} \tag{5.3}$$

We start with an augmentation trick. Note $E[f_i t_i] = f_i^2$ since f is deterministic and $E[t_i] = f_i$. Next, $E[f_i^2] = f_i^2$ since f is deterministic. Finally, $E[y_i t_i] = E[y_i(f_i + \epsilon)] = E[y_i f_i + y_i \epsilon] = E[y_i f_i] + 0$. The last term is zero because the noise in the infinite test set over which we take the expectation is probabilistically independent of the model prediction. Thus the last term in the expectation above cancels to zero. Proceeding from the above-mentioned:

$$\begin{aligned} E[(t_i - y_i)^2] &= E[\epsilon^2] + E[(f_i - y_i)^2] + 2(E[f_i t_i] - E[f_i^2] - E[y_i t_i] + E[y_i f_i]) \\ &= E[\epsilon^2] + E[(f_i - y_i)^2] + 2(f_i^2 - f_i^2 - E[y_i f_i] + E[y_i f_i]) \\ &= E[\epsilon^2] + E[(f_i - y_i)^2] + 2(\cancel{f_i^2} - \cancel{f_i^2} - \cancel{E[y_i f_i]} + \cancel{E[y_i f_i])} \\ &= E[\epsilon^2] + E[(f_i - y_i)^2] \end{aligned} \tag{5.4}$$

Hence we can decompose MSE in expectation into the variance of the noise and the expectation between the true function and the predicted values. This last term can be further decomposed with the same augmentation trick as before:

$$\begin{aligned} E[(f_i - y_i)^2] &= E[(f_i - E[y_i] + E[y_i] - y_i)^2] \\ &= E[(f_i - E[y_i])^2] + E[(E[y_i] - y_i)^2] + 2E[(f_i - E[y_i])(E[y_i] - y_i)] \\ &= E[(f_i - E[y_i])^2] + E[(E[y_i] - y_i)^2] + 2E[(E[y_i] - y_i)(f_i - E[y_i])] \\ &= bias^2 + Var[y_i] + 2(E[f_i E[y_i]] - E[E[y_i]^2] - E[y_i f_i] + E[y_i E[y_i]]) \end{aligned} \tag{5.5}$$

Note $E[f_i E[y_i]] = f_i E[y_i]$ since f is deterministic and $E[E[z]] = z$. Next, $E[E[y_i]^2] = E[y_i]^2$ since $E[E[z]] = z$. Following, $E[y_i f_i] = f_i E[y_i]$ and $E[y_i E[y_i]] =$

$E[y_i]^2$. Thus, the last term in the expectation above cancels to zero.

$$\begin{aligned} E[(f_i - y_i)^2] &= bias^2 + Var[y_i] + 2(E[f_iE[y_i]] - E[E[y_i]^2] - E[y_if_i] + E[y_iE[y_i]]) \\ &= bias^2 + Var[y_i] + 2(f_iE[y_i] - E[y_i]^2 - f_iE[y_i] + E[y_i]^2) \\ &= bias^2 + Var[y_i] + 2(\cancel{f_iE[y_i]} - \cancel{E[y_i]^2} - \cancel{f_iE[y_i]} + \cancel{E[y_i]^2}) \\ &= bias^2 + Var[y_i] \end{aligned} \tag{5.6}$$

Thus the decomposition of the MSE in expectation becomes:

$$E[(t_i - y_i)^2] = Var[noise] + bias^2 + Var[y_i] \tag{5.7}$$

Note that the noise is intrinsic to the data, and it is not possible to be minimized. Thus, to minimize the total error in MSE, it is possible to minimize the bias or variance. However, this is not a trivial task. There is a tradeoff between the two terms, and they are negatively correlated. In general, when we decrease the error in one term, the error in the other term increases. Consider two extreme cases: if we create a model that ignores the input and always provides the same output, it will have zero variance error. However, the bias error will dominate the total error (*underfit*). On the other hand, if we specialize our model to achieve 100% of accuracy in the training set, this model will overfit the data and be pretty unstable once we change the input dataset. In this case, our model would not be able to generalize well (*overfit*).

As stated before, the decomposition presented in this section cannot be automatically extended to the standard classification setting, as in this context, the 0/1 loss function is usually applied, and bias and variance are not purely additive [Valentini and Dietterich, 2004]. In the next section, we present a decomposition adaptation specifically for the 0/1 loss, based on the work of Domingos [2000].

5.3 Measuring Bias and Variance with Limited Data Set

This section presents a step-by-step procedure for measuring bias and variance for the 0/1 loss when working with limited data sets. The method is inspired in the work presented by Valentini and Dietterich [2004] and Domingos [2000]. While some works also provide a decomposition approach, such as Kohavi and Wolpert [1996], we opted to Domingos [2000] because its decomposition is based on a consistent definition of bias and variance. In addition, the authors investigate loss variations as a function of bias and variance, preventing some of the common shortcomings.

Considering the difficulty of estimating noise in real-world data sets, we assume the noise-free case. Hence, given the original definition $t_i = f_i + \epsilon$ and as in the noise-free case $\epsilon = 0$, so from now on we make the assumption $t_i = f_i$ until the rest of this chapter.

We start with a data set S. First, we generate B bootstrap replicates of S (for this experiment we set $B = 200$): $S_1, \dots, S_B$. Given a learning algorithm $\mathcal{L}$, we induce a model on each of the generated S_b replicates to obtain hypothesis $f_b = \mathcal{L}(S_b)$. Let $T_b = S \setminus S_b$ be the data points that do not appear in S_b (out of bag points). We are going to use this set of instances to evaluate the bias-variance decomposition of the error.

For each data point x, we have now observed corresponding value t and several predictions $y_1, \dots, y_K$, where $K = |\{T_b \mid x \in T_b, 1 \leq b \leq B\}|, K \leq B$ and on the average $K \sim \frac{B}{3}$, because about 1/3 of the predictors is not trained on a specific input x. The value K can vary depending on the example x, as each bootstrap is randomly generated.

We are working with a binary classification problem. Let $\mathcal{C}$ be the set of classes, in this problem $\mathcal{C} = \{1, -1\}$ representing, respectively, the positive and negative classes. In order to compute the predictions for a two-class classification problem, we can define

$$p_1(k) = \frac{1}{K} \sum_{b=1}^{B} ||(x \in T_b) \text{ and } (f_b(x) = 1)||, \tag{5.8}$$

$$p_{-1}(k) = \frac{1}{K} \sum_{b=1}^{B} ||(x \in T_b) \text{ and } (f_b(x) = -1)||. \tag{5.9}$$

The main prediction $y_m(x)$ corresponds to the mode of the multiple predictions f_b:

$$y_m = \arg\max(p_1, p_{-1}) \tag{5.10}$$

The *bias* $B(x)$ is the loss of the main prediction relative to the true prediction and can be calculated as:

$$B(x) = \begin{cases} 1 & \text{if } y_m \neq t \\ 0 & \text{if } y_m = t \end{cases} \tag{5.11}$$

The *variance* $V(x)$ is the average loss of the predictions relative to the main prediction:

$$V(x) = \frac{1}{K}\sum_{b=1}^{B} \parallel (x \in T_b) \text{ and } (y_m \neq f_b(x)) \parallel . \tag{5.12}$$

The variance associated with the model prediction follows one of the two cases: a) it can be beneficial and lead to a decrease in total error, or b) it can increase the error. The difference between the two cases is whether the variance occurs in a biased prediction (prediction different from the true value) or an unbiased prediction (prediction equal to the true value). The *unbiased variance* $V_u(x)$ and the *biased variance* $V_b(x)$ can be calculated as:

$$V_u(x) = \frac{1}{K}\sum_{b=1}^{B} \parallel (x \in T_b) \text{ and } (B(x) = 0) \text{ and } (y_m \neq f_b(x)) \parallel \tag{5.13}$$

$$V_b(x) = \frac{1}{K}\sum_{b=1}^{B} \parallel (x \in T_b) \text{ and } (B(x) = 1) \text{ and } (y_m \neq f_b(x)) \parallel \tag{5.14}$$

We will denote as *net variance* the real impact of variance in the error. It is denoted as:

$$V_n(x) = V_u(x) - V_b(x) \tag{5.15}$$

With the assumption of the noise-free case, the *average loss on the example x*, the error $E_D(x)$ is calculated by a simple algebraic sum of bias, unbiased and biased variance:

$$E_D(x) = B(x) + V_u(x) - V_b(x) \tag{5.16}$$

Until this point, we introduced formulas to estimate the decomposition terms for a single instance x. Henceforth, we can calculate the *average bias, average variance, average variance unbiased, average variance biased and average net variance* averaging over the entire set of examples of the test set $\mathcal{T} = \{(x_j, t_j)\}_{j=1}^{r}$.

The average quantities are

Average bias:

$$\mathbb{E}_x[B(x)] = \frac{1}{r}\sum_{j=1}^{r} B(x_j) \tag{5.17}$$

Average variance:

$$\begin{aligned}\mathbb{E}_x[V(x)] &= \frac{1}{r}\sum_{j=1}^{r} V(x_j) \\ &= \frac{1}{rK}\sum_{j=1}^{r}\sum_{b=1}^{B} \parallel (x_j \in T_b) \text{ and } (y_m \neq f_b(x_j)) \parallel\end{aligned} \tag{5.18}$$

Average unbiased variance:

$$\begin{aligned}\mathbb{E}_x[V_u(x)] &= \frac{1}{r}\sum_{j=1}^{r} V_u(x_j) \\ &= \frac{1}{rK}\sum_{j=1}^{r}\sum_{b=1}^{B} \parallel (x_j \in T_b) \text{ and } (B(x) = 0) \text{ and } (y_m \neq f_b(x_j)) \parallel\end{aligned} \tag{5.19}$$

Average biased variance:

$$\begin{aligned}\mathbb{E}_x[V_b(x)] &= \frac{1}{r}\sum_{j=1}^{r} V_b(x_j) \\ &= \frac{1}{rK}\sum_{j=1}^{r}\sum_{b=1}^{B} \parallel (x_j \in T_b) \text{ and } (B(x) = 1) \text{ and } (y_m \neq f_b(x_j)) \parallel\end{aligned} \tag{5.20}$$

Average net variance:

$$\begin{aligned}\mathbb{E}_x[V_n(x)] &= \frac{1}{r}\sum_{j=1}^{r} V_n(x_j) \\ &= \frac{1}{r}\sum_{j=1}^{r}(V_u(x_j) - V_b(x_j))\end{aligned} \tag{5.21}$$

Lastly, *average loss on all examples*:

$$\mathbb{E}_x[L(t,y)] = \mathbb{E}_x[B(x)] + \mathbb{E}_x[V_u(x)] - \mathbb{E}_x[V_b(x)]. \tag{5.22}$$

5.4 Accuracy Performance

In order to answer RQ1, we measured the accuracy performance of our ensemble approach under different conditions. In our first experiment, presented in Table 5.1, we measured the average error (and consequently the accuracy) followed by the decomposition of the error in bias and variance. This experiment aimed to compare how the error components are summed up in the all-in-one approach compared to the ED-Ensemble approach. For both cases it was used XGBoost and Random Forests as learning algorithms.

Table 5.1: Bias plus variance decomposition estimates of the average error calculated using XGBoost and Random Forests as the learning algorithms under different approaches.

(a) Estimates of the decomposed average error using the all-in-one approach.

Learning Algorithm	Accuracy	Bias	Net variance	Var. unbiased	Var. biased
XGBoost	0.5837	0.3993	0.017	0.1289	0.1119
Random Forests	0.5811	0.3961	0.0229	0.1667	0.1438

(b) Estimates of the decomposed average error using the ED-Ensemble approach. Parameter *number of features = 15*.

Learning Algorithm	Accuracy	Bias	Net variance	Var. unbiased	Var. biased
XGBoost	0.6544	0.3077	0.0379	0.1124	0.0745
Random Forests	0.6429	0.3191	0.038	0.1306	0.0926

From the results using XGBoost learning algorithm, it was obtained the accuracy performance of 0.5837 when using the all-in-one approach and 0.6544 when using the ED-Ensemble approach. It represents a relative increase of 12.11% in the accuracy performance using the later approach. Furthermore, the average error is defined as the sum of *bias* and *net variance*. Most of the error in all-in-one and ensemble approaches came from the bias component; 0.3993 out of 0.4163 and 0.3077 out of 0.3456. The performance gain obtained in the ensemble was entirely due to the reduction of the bias error. The variance error in the ensemble has slightly increased, albeit negligible.

Similar results were obtained with Random Forests learning algorithm. The all-in-one approach obtained an accuracy performance of 0.5811 and the ensemble approach of 0.6429, representing an increase of 10.63%. The similarities extended on how this improvement was achieved. Most of the errors in the all-in-one approach come from the bias error. The ensemble approach significantly reduced this bias error (from 0.3961 to 0.3191) while maintaining the variance error low (from 0.0229 to 0.038).

Lastly, answering RQ1, we can claim that the improvements provided by the ensemble do extend to accuracy. We markedly increased the accuracy in the resulting ED-Ensemble, even when considering two different learning algorithms, XGBoost and Random Forest. Moreover, nearly the entire contribution comes from reducing the bias error. Thus, the ED-Ensemble approach can learn more complex relationships present in the data (reduced bias error) while not overfitting (stable variance error).

5.5 Measuring the Impact of the Number of Features

In the experiments carried out throughout our work, we empirically fixed in 15 the number of features as a good compromise between the ensemble performance, time spent to generate the ensemble, and explainability.

In the research question RQ2, we are interested in modeling how changing the number of features impacts the average error and its decomposed components bias and variance. To answer this question, we performed multiple experiments varying the number of features from 3 to 15. Hence, for a fixed number of features equal to 3, the ensemble was generated using the ED-Ensemble approach with the restriction that no base model could have more than 3 features. We set the minimum number of features to 3 because that was the smallest size that ensured the generation of the ED-Ensemble across all experiments. For each generated ensemble, the accuracy, average error, bias, and variance were estimated. Similar experiments were performed using XGBoost and Random Forests as learning algorithms.

The results are presented in Tables 5.2 and 5.3. We have opted to omit the accuracy metric, exhibiting the average error instead. In Table 5.2 the learning algorithm XGBoost was used. It is possible to observe that the average error decreases as we increase the number of features. In other words, as we increase the number of features, the accuracy increases. The average error has a decrease from 0.3762 to 0.3456, representing a decrease of 8.13%.

Interestingly, the net variance for the minimum and the maximum number of features goes from 0.0390 to 0.0379 and can be regarded as stable. Conversely, the bias error value decreases from 0.3372 to 0.3077 and accounts for the decrease in the final average error. As a result, based on the experiments, our answer to RQ2 when employing XGBoost as a learning algorithm is that as we increase the number of features allowed in base models, we attain a decrease in bias error while the variance error remains stable. This behavior results in an ensemble with a lower average error (therefore a superior accuracy).

Table 5.2: Bias plus variance decomposition estimates of the average error in ED-Ensembles models using XGBoost as learning algorithm.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.3762	0.3372	0.0390	0.1221	0.0831
4	0.3811	0.3633	0.0178	0.1078	0.0900
5	0.3743	0.3470	0.0273	0.1175	0.0902
6	0.3711	0.3535	0.0176	0.1131	0.0955
7	0.3661	0.3404	0.0257	0.1168	0.0911
8	0.3682	0.3257	0.0425	0.1268	0.0843
9	0.3714	0.3601	0.0113	0.1092	0.0979
10	0.3721	0.3552	0.0169	0.1106	0.0937
11	0.3718	0.3421	0.0297	0.1182	0.0885
12	0.3718	0.3584	0.0134	0.1112	0.0978
13	0.3578	0.3028	0.0550	0.1313	0.0763
14	0.3560	0.3257	0.0303	0.1132	0.0829
15	0.3456	0.3077	0.0379	0.1124	0.0745

Similarly, in Table 5.3 we present the same set of experiments although employing Random Forests as learning algorithms. The results obtained follow the same trend presented by the XGBoost learning algorithm. Bias accounts for most of the total error. Also, given the slight absolute difference in net variance error from 0.0297 to 0.038, and that it represents a small contribution to the total error, it can be said stable in the total error contribution. On the other hand, the bias error experiences a significant reduction from 0.3650 to 0.3191. The average error varies from 0.3947 to 0.3571, representing a reduction of 9.52%. As a result, our answer to RQ2 when considering Random Forests learning algorithm is that there is a clear trend in, as we increase the number of features, the bias error is reduced and consequently the average error. This improvement is possible as the net variance error remains small regardless of altering the number of features.

Figures 5.1 and 5.2 presents the experiments results in line plots as an alternative for easy viewing. It is possible to note a decreasing trend in average error and average bias, while the other components do not show a trend.

Table 5.3: Bias plus variance decomposition estimates of the average error in ED-Ensembles models using Random Forests as the learning algorithm.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.3947	0.3650	0.0297	0.1225	0.0928
4	0.3639	0.3470	0.0169	0.1115	0.0946
5	0.3656	0.3273	0.0383	0.1290	0.0907
6	0.3656	0.3339	0.0317	0.1225	0.0908
7	0.3613	0.3339	0.0274	0.1192	0.0918
8	0.3535	0.3175	0.0360	0.1249	0.0889
9	0.3565	0.3224	0.0341	0.1190	0.0849
10	0.3535	0.3159	0.0376	0.1244	0.0868
11	0.3535	0.3159	0.0376	0.1244	0.0868
12	0.3536	0.3191	0.0345	0.1291	0.0946
13	0.3536	0.3191	0.0345	0.1291	0.0946
14	0.3598	0.3241	0.0357	0.1252	0.0895
15	0.3571	0.3191	0.038	0.1306	0.0926

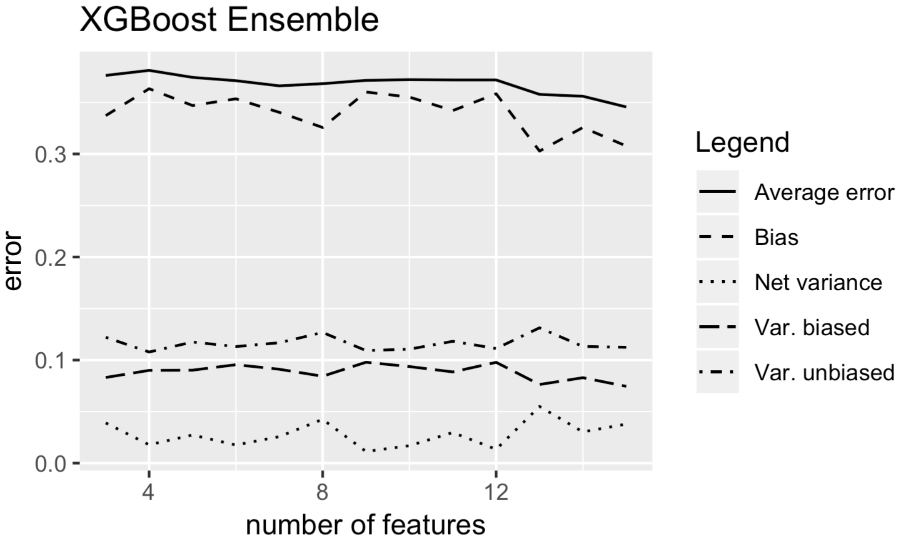


Figure 5.1: Bias plus variance decomposition of the average error in the generated ED-Ensemble when varying the number of features from 3 to 15. Using XGBoost as the learning algorithm.

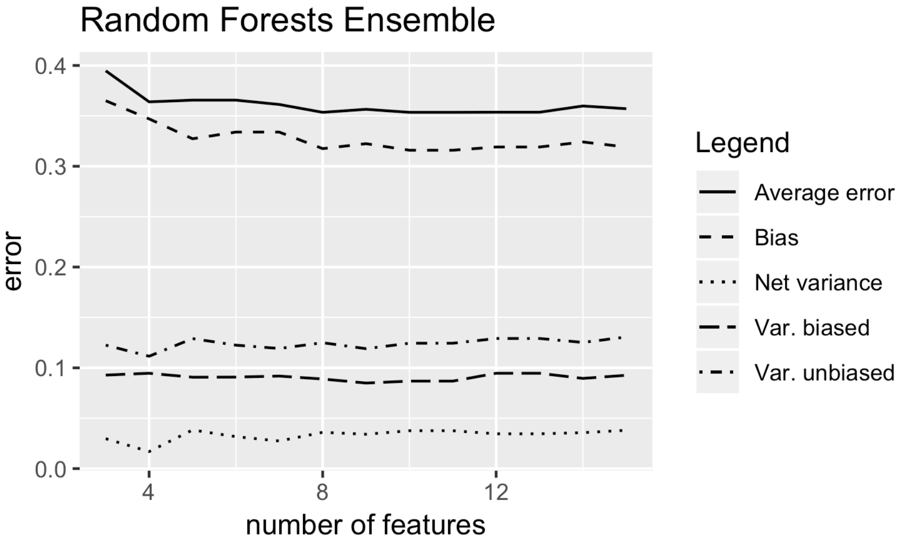


Figure 5.2: Bias plus variance decomposition of the average error in the generated ED-Ensemble when varying the number of features from 3 to 15. Using Random Forests as the learning algorithm.

5.6 Ensemble and Base Models Compared

One key aspect of ensemble evaluation is to analyze the generated ensemble and the base models that make it up. This analysis meets with the research question RQ3. In order to answer RQ3, we set up an experiment in which we measured the average error, along with its decomposition into bias plus variance, and averaged the metrics of all base models belonging to the same ensemble. The outcome is a single representative base model calculated from the base models comprising the ensemble.

Again, the metrics are calculated varying the number of features parameter from 3 to 15. The results are presented in Table 5.4 and Table 5.5 for learning algorithms XGBoost and Random Forests respectively.

For the XGBoost learning algorithm presented in Table 5.4, bias error had a slight decrease. Conversely, net variance error suffered a small increment as we increased the number of features. In percentage, the bias error had a decrease of 4.45%, while the net variance error had an increase of 31.52%, considering the first and last measurement. Although the percentage increase in net variance error represents a big leap, caution should be taken with its analysis. First, net variance error represents a small portion of the total error, and even with a significant increase, bias error still accounts for most of the error. Second, there is no apparent indication of an uptrend in the net variance error. The lack of uptrend is supported considering that the other three net variance errors are greater than the last measurement. Virtually stable, the average error varied from 0.4068 (considering 3 features) to 0.3953 (considering 15 features) representing a decrease in the total error of about 2.83% only.

Table 5.4: Bias plus variance decomposition estimates of the ED-Ensemble **base** models using XGBoost as learning algorithm.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.4068	0.3884	0.0184	0.0971	0.0787
4	0.4097	0.3993	0.0104	0.0971	0.0867
5	0.4039	0.3860	0.0179	0.0973	0.0794
6	0.4023	0.3816	0.0207	0.0998	0.0791
7	0.4016	0.3839	0.0177	0.0992	0.0815
8	0.4037	0.3744	0.0293	0.1061	0.0768
9	0.4083	0.3752	0.0331	0.1130	0.0799
10	0.4088	0.3775	0.0313	0.1113	0.0800
11	0.4107	0.3903	0.0204	0.1099	0.0895
12	0.4069	0.3888	0.0181	0.1056	0.0875
13	0.4032	0.3802	0.0230	0.1116	0.0886
14	0.3997	0.3774	0.0223	0.1121	0.0898
15	0.3953	0.3711	0.0242	0.1103	0.0861

Similar analysis is performed for the Random Forests learning algorithm, presented in Table 5.5. Again, we observe that most of the error comes from the bias error. The bias error presented an increase from 0.3863 to 0.3950, about 2.25%. On the other hand, the net variance error experienced a decrease from 0.0261 to 0.0110 (decrease of 57.87%) as we increased the number of features parameter. The total error decreased from 0.4124 to 0.4060 (1.55%). Still, the decrease was possible because the gains from the net variance error exceeded the loss in the bias error.

Finally, in Table 5.6 we present the information of how much the ensemble improves over the base models. We opted to display the accuracy performance instead of the average error to simplify visualization since a higher percentage implies a more remarkable improvement.

For the XGBoost learning algorithm, the improvement performance accuracy provided by the ensemble over the average base models goes from 5.16% to 8.22%. The gap performance steadily increases as the number of features also increases, showing an upward trending. The trend represents an appealing aspect of the obtained ensemble. As we increase the number of features allowed, we expect better base models and,

Table 5.5: Bias plus variance decomposition estimates of the ED-Ensemble **base** models using Random Forests as learning algorithm.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.4124	0.3863	0.0261	0.1059	0.0798
4	0.4133	0.3983	0.0150	0.1060	0.0091
5	0.4115	0.4096	0.0019	0.1032	0.1013
6	0.4102	0.4008	0.0094	0.1061	0.0967
7	0.4039	0.3890	0.0149	0.1002	0.0853
8	0.4008	0.3778	0.0230	0.1094	0.0864
9	0.4021	0.3819	0.0202	0.1083	0.0881
10	0.3984	0.3755	0.0229	0.1078	0.0849
11	0.3984	0.3755	0.0229	0.1078	0.0849
12	0.4024	0.3813	0.0211	0.1124	0.0913
13	0.4024	0.3813	0.0211	0.1124	0.0913
14	0.4077	0.3968	0.0109	0.1141	0.1032
15	0.4060	0.3950	0.0110	0.1123	0.1013

consequently, better ensembles. However, we also observed a surprising increase in the performance gap between the ensemble and averaged base models.

Similar measures were obtained using Random Forests learning algorithm. It is possible to observe that as the number of features increases, the improvement gap between the ensemble and base models also increases in an upward trend. When we set the number of features to 15, we reach a gap of 8.23%. Although this is not the most significant gap, it is the second-biggest gap. We observe the gap performance varying from 3.01% to 8.42%.

Thus, answering RQ3, we observe an increase in net variance error offset along with a significant decrease in bias error in the generated ensemble. Furthermore, the improvement in ensemble performance is not solely due to better-selected base models. As the number of features increases, the improvement gap between ensemble and base models also increases. Lastly, we can state that the generated ensembles can consistently learn more complex relationships in data, significantly decreasing bias error while keeping the net variance error at a low level.

5.7 ED-Ensemble compared with Randomly Generated Ensemble

Given a set of models, if we assume that these models are performant, will any combination of models generate ensembles consistently better than the base models? This issue

Table 5.6: Compared accuracy and percentage gain of the ED-Ensemble built with the average accuracy of the base models.

	XGBoost		Random Forests	
# of features	Accuracy	Gain	Accuracy	Gain
3	0.6238	5.16%	0.6053	3.01%
4	0.6189	4.84%	0.6361	8.42%
5	0.6257	4.97%	0.6344	7.80%
6	0.6289	5.22%	0.6344	7.56%
7	0.6339	5.93%	0.6387	7.15%
8	0.6318	5.95%	0.6465	7.89%
9	0.6286	6.24%	0.6435	7.63%
10	0.6279	6.21%	0.6465	7.46%
11	0.6282	6.60%	0.6465	7.46%
12	0.6282	5.92%	0.6464	8.17%
13	0.6422	7.61%	0.6464	8.17%
14	0.6440	7.28%	0.6402	8.09%
15	0.6544	8.22%	0.6429	8.23%

is in line with the research question RQ4. We are now interested in showing that the models our ED-Ensemble chooses to create the ensemble are not selected by chance. In other words, the base models selected are individually chosen by an objective criterion to increase diversity.

In order to answer RQ4, we must compare ED-Ensembles with randomly generated ensembles. Following the practices adopted in this chapter, we tested these two combinations methods to construct ensembles varying the number of features parameter from 3 to 15.

Results for XGBoost learning algorithm are presented in Table 5.7 and Table 5.8. Table 5.7 presents the average error and its decomposition in bias plus variance of randomly generated ensembles. These ensembles are constructed by randomly selecting k models with a fixed n number of features to compose the ensemble. The variable k, indicating the number of base models selected, is obtained when we generate the ED-ensemble. We opted to replicate this number in the randomly generated ensemble to provide a fairer comparison. From the results, no upward or downward trend can be observed. Table 5.8 presents the average performance of the base models that are part of the randomly generated ensemble. Again, no trend can be observed for any of the related metrics when increasing the number of features.

Similarly, we replicated the experiments for Random Forests learning algorithm. The average error and its decomposition in bias plus variance of the randomly generated

Table 5.7: Bias plus variance decomposition estimates of the average error in ensemble models using XGBoost as the learning algorithm. This ensemble is built by generating randomly base models given a size of features.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.4328	0.4326	0.0002	0.1026	0.1024
4	0.4450	0.4446	0.0004	0.1045	0.1041
5	0.4354	0.4299	0.0055	0.1204	0.1149
6	0.4347	0.4343	0.0004	0.0792	0.0788
7	0.4467	0.4441	0.0026	0.0884	0.0858
8	0.4413	0.4375	0.0038	0.1033	0.0995
9	0.4402	0.437	0.0032	0.1231	0.1199
10	0.4309	0.4212	0.0097	0.1021	0.0924
11	0.4712	0.4539	0.0173	0.1305	0.1132
12	0.4362	0.4332	0.0030	0.1139	0.1109
13	0.4272	0.4173	0.0099	0.1194	0.1095
14	0.4289	0.4288	0.0001	0.1054	0.1053
15	0.4404	0.4359	0.0045	0.1138	0.1093

Table 5.8: Bias plus variance decomposition estimates of the average error in ensemble **base** models using XGBoost as the learning algorithm. This ensemble is built by generating randomly base models given a size of features.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.4380	0.4330	0.0050	0.1027	0.0977
4	0.4559	0.4521	0.0038	0.1056	0.1018
5	0.4402	0.4308	0.0094	0.1013	0.0919
6	0.4385	0.4414	0.0029	0.0822	0.0793
7	0.4443	0.4383	0.0051	0.0912	0.0861
8	0.4439	0.4334	0.0106	0.0943	0.0837
9	0.4405	0.4283	0.0122	0.1060	0.0938
10	0.4375	0.4354	0.0021	0.0866	0.0845
11	0.4424	0.4376	0.0048	0.0962	0.0914
12	0.4397	0.4350	0.0047	0.1033	0.0986
13	0.4411	0.4383	0.0028	0.1075	0.1047
14	0.4397	0.4371	0.0026	0.0998	0.0972
15	0.4423	0.4368	0.0055	0.0997	0.0942

ensemble are shown in Table 5.9. The table shows no clear trend for any of the related metrics when varying the number of features. In Table 5.10 are presented the data for the base models that are part of the ensemble. Again, there is no clear trend for any of the related metrics.

Finally, to summarize how the randomly generated ensembles are compared with

Table 5.9: Bias plus variance decomposition estimates of the average error in ensemble models using Random Forests as learning algorithm. This ensemble is built by generating randomly base models given a size of features.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.4361	0.4354	0.0007	0.1093	0.1086
4	0.4233	0.4124	0.0109	0.1161	0.1052
5	0.4365	0.4266	0.0099	0.1166	0.1067
6	0.4385	0.4375	0.0010	0.1070	0.1060
7	0.4415	0.4386	0.0029	0.1177	0.1148
8	0.4356	0.4299	0.0057	0.1227	0.1170
9	0.4476	0.4430	0.0046	0.1198	0.1152
10	0.4297	0.4266	0.0031	0.1227	0.1196
11	0.4293	0.4212	0.0081	0.1172	0.1091
12	0.4308	0.4294	0.0014	0.1079	0.1065
13	0.4331	0.4304	0.0027	0.1097	0.1070
14	0.4311	0.4266	0.0045	0.1186	0.1141
15	0.4204	0.4148	0.0056	0.1193	0.1137

Table 5.10: Bias plus variance decomposition estimates of the average error in ensemble **base** models using Random Forests as learning algorithm. This ensemble is built by generating randomly base models given a size of features.

# of features	Avg. Error	Bias	Net variance	Var. unbiased	Var. biased
3	0.4437	0.4422	0.0015	0.0977	0.0962
4	0.4372	0.4247	0.0125	0.1092	0.0967
5	0.4419	0.4406	0.0013	0.1033	0.1020
6	0.4412	0.4394	0.0018	0.0910	0.0892
7	0.4529	0.4484	0.0045	0.1175	0.1130
8	0.4443	0.4432	0.0011	0.1141	0.1130
9	0.4423	0.4371	0.0052	0.0994	0.0942
10	0.4380	0.4373	0.0007	0.1059	0.1052
11	0.4353	0.4351	0.0002	0.0964	0.0962
12	0.4419	0.4308	0.0111	0.1055	0.0944
13	0.4383	0.4353	0.0030	0.0930	0.0900
14	0.4390	0.4333	0.0057	0.0993	0.0936
15	0.4344	0.4308	0.0036	0.1031	0.0995

Table 5.11: Compared accuracy and percentage gain of the ensemble built using randomly choosen models of a fixed feature size over the accuracy obtained in all-in-one approach.

	XGBoost		Random Forests	
# of features	Accuracy	Gain	Accuracy	Gain
3	0.5672	-2.82%	0.5639	-2.95%
4	0.5550	-4.91%	0.5767	-0.75%
5	0.5646	-3.27%	0.5635	-3.02%
6	0.5653	-3.15%	0.5615	-3.36%
7	0.5533	-5.21%	0.5585	-3.88%
8	0.5587	-4.28%	0.5644	-2.87%
9	0.5598	-4.09%	0.5524	-4.93%
10	0.5691	-2.50%	0.5703	-1.85%
11	0.5288	-9.40%	0.5707	-1.78%
12	0.5638	-3.41%	0.5692	-2.04%
13	0.5728	-1.86%	0.5669	-2.44%
14	0.5711	-2.16%	0.5689	-2.09%
15	0.5596	-4.13%	0.5796	-0.25%

their respective ensembles, we present in Table 5.11 the gain provided by the ensembles. As can be seen, for both learning algorithms, the ensembles have poorer performance. As a result, we can answer RQ4 as positive. Yes, the ED-Ensembles proved to be consistently superior to randomly generated ensembles under all conditions tested.

5.8 Discussion

In summary, one of the strategies to improve the performance of a learning algorithm consists of developing methods to reduce the variance error or bias error in the induced model. Since these two notions are contrasting, improvement in one term almost always implies worsening in the other term. Hence, the most common approach is to reduce the error in only one of the terms. Our proposal fits in reducing variance by using small models, i.e., models supplied with a small subset of features. Concurrently, we seek to reduce bias through the combination of the diversified models. The accuracy gain achieved shows that the strategy used allowed the total error to be successfully reduced. We maintained the variance error in low levels while markedly reducing the bias error despite using two substantially different learning algorithms as base models. By their nature, these learning algorithms are on opposite sides in how they seek to provide a combination of models with better performance.

Chapter 6

Model-Explanations as Meta-Features in Longitudinal Data

This chapter presents a novel approach that enhances traditional machine learning approaches in longitudinal data. Until now, we have made use of information gathered on the first consultation only. However, the standard protocol for a patient typically comprises sequential appointments. Firstly, we describe the proposed approach to use previous models' explanations to function as a temporal memory in longitudinal data. We aim to evaluate the proposed approach in the chronic pain dataset adopted throughout this work. Each patient is associated with a set of consultations. We show that using the additional information (until now disregarded) enables improving the model performance. Next, we discuss our evaluation procedure and analyze the results obtained. In particular, our study aims to answer the following research questions:

RQ1 How effective are prediction models combining fresh data from the current consultation with SHAP meta-features from the previous consultation?

RQ2 Are SHAP meta-features extracted from previous models more discriminating for pain relief than features from previous consultations?

RQ3 What are the fundamentals that allow SHAP meta-features to leverage predictive performance?

The remainder of this chapter is divided into four main sections. Section 6.1 describes the foundation for longitudinal data. Section 6.2 is devoted to the novel proposed approach to using models' explanations meta-features as memory in longitudinal data. Section 6.3 details the chronic pain data used in this chapter, being a set of related datasets with a varying number of consultations obtained from the original data. Section 6.4 discuss the main results obtained from the experiments. Finally, Section 6.5 details the findings of this research.

6.1 Longitudinal data

In longitudinal data, repeated observations are made over time for the same subject [Fitzmaurice et al., 2012; Hedeker and Gibbons, 2006]. This structure creates correlations as observations for the same subject are dependent [Speiser, 2021]. Unlike cross-sectional data, which is collected at a specific point, longitudinal data is collected for the same subject over an extended period. Unlike time series, where repeated observations are collected over time for a single subject, in longitudinal data, repeated observations are collected for multiple subjects hierarchically, and observations may be unevenly spaced in time. Longitudinal data is present in different areas such as medical field [Zhao et al., 2019; Konerman et al., 2015], econometrics [Frees et al., 1999; Heckman and Walker, 1990], and social sciences [Stenberg, 2011].

The analysis of longitudinal data is traditionally performed using statistical methods [Verbeke et al., 2014; Perveen et al., 2020]. These methods, however, require many assumptions about the data in order to work correctly and machine learning methods, on the other hand, require considerably fewer assumptions about the data. One of the few assumptions is that the random variables are independent and identically distributed. However, longitudinal data from patient reports may violate this assumption as observations are correlated for the same patient but independent across different patients [Sela and Simonoff, 2012; Hu, 2021; Ngufor et al., 2019].

Fortunately, there are some alternatives to enable the use of machine learning methods in longitudinal data modeling. A simple adaptation is to collect a set of variables in the initial period (baseline consultation) and build models using this data alone. The main limitation of this approach, however, is that a substantial amount of information is simply ignored. Another alternative consists of building aggregate features on top of the longitudinal data, such as mean, minimum, maximum, and standard deviation. The work by Zhao et al. [2019] compares the scenarios using baseline data versus incorporating the longitudinal data by aggregating features for cardiovascular disease event prediction. The experiments showed that including aggregate features performs better when compared to baseline data only. Hence, for the accurate modeling of longitudinal data, the application of machine learning methods requires some adaptation, either algorithm-wise or data-wise.

In the following section, we introduce a novel machine learning method to longitudinal data to predict the evolution of pain relief. Our approach uses previous models' explanations (i.e., feature importances) to function as a temporal memory on longitudinal data. Precisely, to predict the output at consultation c for a patient, we extract feature importances [Lundberg and Lee, 2017] from a model trained on the data up to consultation $c-1$ and use these explanations as memory meta-features about previous iterations. The intuition is that our approach improves the current model by remembering important information from previous consultations.

6.2 Feature Importances as Memory Meta-Features in Longitudinal Data

We begin by presenting the basic concepts and notations that are necessary to describe our method. Notations are shown in Table 6.1.

Table 6.1: Notations used in this chapter.

Name	Description
n	number of subjects
d	number of features
t_i	number of consultations associated with subject i
x_i	$t_i \times d$ matrix of data for subject i
y_i	true output for subject i
s^j	$n \times d$ matrix of SHAP values at consultation j
f^j	model trained on consultation j

Longitudinal data involves repeated observations for the same patient at different times (i.e., consultations with the doctor) at intervals that may not be equidistant. At each time point, a variety of information about the patient's characteristics is collected. These characteristics may represent information that will not change during treatment (e.g., gender and race) or information that is likely to change over time (e.g., pain intensity, characteristics, and medications). In general, in problems involving longitudinal data, the output is associated with each point in time. Nevertheless, in our particular case, we have a single outcome at the end of treatment. To overcome this limitation, we will obtain the label from the patient's lastest consultation with the doctor and then replicate this label for all the points belonging to the same patient. The idea is that at each time point, we will try to predict whether, at the end of the treatment, the patient will be able to experience pain relief of at least 30% (i.e., VAS-30). The prediction model outputs the probability of achieving such an improvement, which can easily be converted to a binary output.

Formally, labeled longitudinal data can be represented by a set of pairs $\{(x_i, y_i)\}_{(i=1)}^{n}$, where $x_i \in \mathbb{R}^{t_i \times d}$ (i.e., each instance x_i is a vector of real numbers of size d that is repeatedly observed at t_i times). Furthermore, $y_i \in [0,\ 1]$ is the final treatment result for the ith patient. For the same patient, repeated observations are made at different points in time $t_i = \{t_1, t_2, ..., t_k\}$. We will identify a point in time as a superscript index. Thus, for a single patient x_i, we have $x_i = \begin{bmatrix} x_i^1 & x_i^2 & \dots & x_i^k \end{bmatrix}^T$, where each x_i^j is a multidimensional feature vector. Features include the 78 questionnaire options along with demographic and socioeconomic information, resulting in a total of 332 features each visit.

6.2.1 Model-Explanations as Memory

The intuition we explore in order to learn an effective prediction model for pain relief is based on the following rationale:

- Memorize important information as meta-features from previous consultations. This information comprises key aspects of a specific patient.
- Combine it with fresh information from the current consultation.

We have used SHAP values as meta-features extracted from previous consultations in order to gather important features acting as a data-level temporal memory in longitudinal data. Consider the dataset $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^{t_i \times d}$. First, we learn an ED-Ensemble f^1 on data from the first consultation, $t = 1$, such that $f^1(x^1, y) = \hat{y}^1$. The generated ensemble predicts, with only the information from the first consultation, whether the patient will experience a significant reduction in pain at the end of treatment.

Next, we extract the SHAP values/features, s^1, from this model, where $s^1 \in \mathbb{R}^{n \times d}$. We will use the extracted SHAP values as meta-features at time $t = 2$. At the second consultation, we will consider the current questionnaire data in addition to SHAP values, s^1. Thus, we will learn another ED-Ensemble, such that $f^2(x^2 \cup s^1, y) = \hat{y}^2$. The process continues for the number of consultations performed.

Algorithm 2 presents our proposed algorithm EXP-MF (model-EXPlanations as Meta-Features) for predicting the evolution of pain relief when considering c consultations with the doctor. Figure 6.1 illustrates a step-by-step of the process considering two consecutive consultations. Our method starts with an ED-Ensemble being trained on data from consultation i. Next, SHAP values are extracted from the prediction model and the extracted SHAP values are joined with data from consultation $i+1$. Finally, another prediction model is learned from this new data.

Algorithm 2 EXP-MF in Longitudinal Data

```
Input: $(X, y)$: labeled longitudinal dataset, $c$: number of consultations to consider.
Output: A prediction model taking into account $c$ consultations.
$i \leftarrow 1$;
Learn $f^i(x^i, y)$;    ▷ Learn an ED-Ensemble using data from consultation 1
$s^i \leftarrow$ extract SHAP meta-features from $f^i(x^i, y)$;
while $i < c$ do
    Learn $f^{i+1}(x^{i+1} \cup s^i, y)$;    ▷ Learn an ED-Ensemble using fresh data from consultation $i+1$ and SHAP from previous consultation $i$
    $s^{i+1} \leftarrow$ extract SHAP meta-features from $f^{i+1}(x^{i+1} \cup s^i, y)$;
    $i \leftarrow i + 1$;    ▷ Next consultation
end while
return $f^i(x^i \cup s^{i-1}, y)$;
```

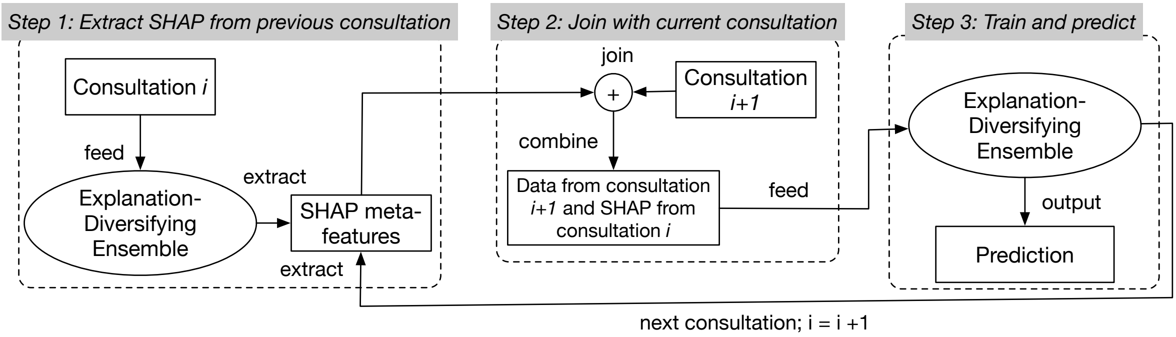

Step 1: Extract SHAP from previous consultation
Consultation *i*
feed
Explanation-Diversifying Ensemble
extract
SHAP meta-features
extract
Step 2: Join with current consultation
join
+
Consultation *i+1*
combine
Data from consultation *i+1* and SHAP from consultation *i*
feed
Step 3: Train and predict
Explanation-Diversifying Ensemble
output
Prediction
next consultation; i = i +1

Figure 6.1: Diagram with the step-by-step of the proposed approach considering two consecutive consultations i and $i+1$. In Step 1, we train an ED-Ensemble on data from consultation i. In Step 2, we extract SHAP features from the ensemble constructed and join them with data from consultation $i+1$. Finally, in Step 3, we train a new ED-Ensemble on the combined dataset. The resulting model can make predictions using the information from consultations i and $i+1$.

Figure 6.2 shows the complete process considering multiple consecutive consultations. For the same patient, each consultation can generate a prediction. Thus, it is possible to generate multiple predictions for the same patient. It is expected that confidence in the prediction will increase with each additional consultation.

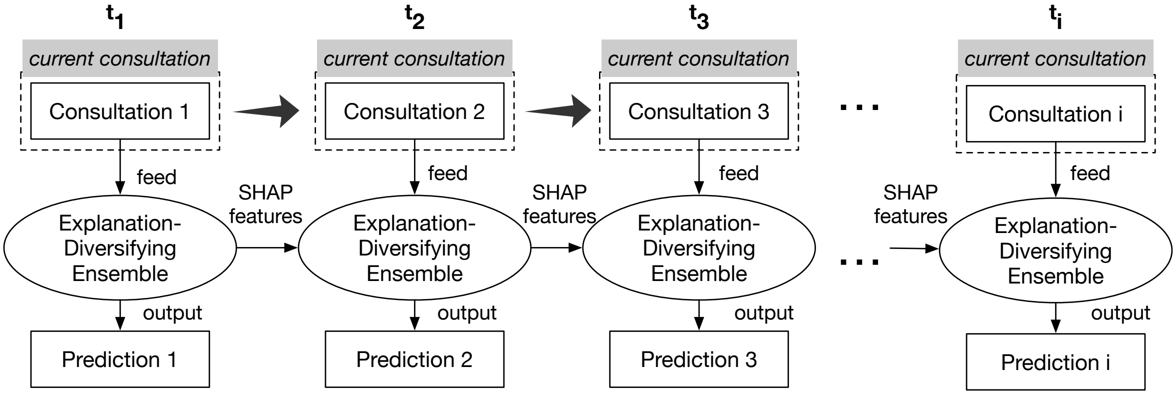


Figure 6.2: Illustration of the proposed method considering multiple sequential consultations. It starts by training an ED-Ensemble on the first consultation. From this model, it is extracted the model's feature importance (SHAP), and this data is forwarded to the next visit. A new ED-Ensemble is built at the second visit using the complete second consultation data combined with the previously extracted SHAP. This process repeats as more consultations are considered.

6.3 Chronic Pain Data

Data consists of attributes extracted from patients' self-reports gathered at multiple consultations with the doctor. Our models aim to predict whether the patient will, at the end of the treatment, experience a significant reduction in pain. Specifically, an overall reduction of pain intensity by 30% (aka, VAS 30) is assessed. The ground truth labels are obtained by calculating the difference in pain intensities reported in the first and last consultation and then replicated to all data points for the same patient.

We consider the same chronic pain data used until now in this thesis. However, the approach proposed in this chapter requires some data manipulation, as we often select patients with a minimum number of consultations. Thus, we have derived many related datasets from the original dataset.

Table 6.2 shows patients information segmented by the number of consultations. It outlines the three dimensions of pain perception. Additionally, it presents the neuropathic pain scale, which is used for assessing neuropathic pain and may be particularly useful for assessing response to therapies. The total neuropathy score is calculated as the sum of the possibilities, and the cut-off value for the diagnosis of neuropathic pain is a total score of 4.

Table 6.2: Part of the patient perception dimension scores data obtained at each consultation. Mean, first and third quartiles within age, McGill score, initial pain intensity, and pain perception dimension scores. Here we consider the VAS 30 label and pain characteristics are not mutually exclusive. Pain characteristics are not mutually exclusive.

	Consultations				
	1	2	3	4	5
			Treatment was effective		
n	112 (42.26%)	112 (42.26%)	56 (42.11%)	33 (40.74%)	25 (43.1%)
Sex (male)	42 (37.5%)	42 (37.5%)	18 (32.14%)	10 (30.3%)	7 (28%)
Age, y	54.88	54.88	53.88	50.97	50.88
	(47.75–65)	(47.75–65)	(46.75–65.25)	(42–61)	(42–62)
0–15 McGill score	7.08	5.52	7.29	7.3	9.56
	(4.0–10.25)	(3–9)	(4–11.25)	(4–11)	(7–12)
0–10 pain intensity	6.16 (5–8)	4.71 (2–8)	5.48 (3.75–8)	4.27 (2–7)	5.28 (3–8)
Sensory dimension	3.44 (1–6)	2.68 (1–4.25)	3.62 (1–6)	4 (2–6)	4.88 (3–7)
Affective dimension	2.48 (1–4)	1.81 (1–3)	2.43 (1–4)	2.24 (1–3)	3.36 (2–4)
Evaluative dimension	1.16 (1–2)	1.03 (1–1)	1.23 (1–2)	1.06 (1–1)	1.32 (1–2)
Neurophatic pain scale					
Burning	76 (67.86%)	69 (61.61%)	36 (64.29%)	21 (63.64%)	17 (68%)
Hypoesthesia to touch	37 (33.04%)	24 (42.86%)	39 (47.56%)	9 (27.27%)	11 (44%)
Numbness	49 (43.75%)	35 (31.35%)	21 (37.5%)	13 (39.39%)	16 (64%)
Pins and needles	39 (34.82%)	27 (24.11%)	17 (30.36%)	7 (21.21%)	13 (52%)
Tingling	37 (33.04%)	36 (32.14%)	23 (41.07%)	9 (27.27%)	12 (48%)
Electric shocks	35 (31.25%)	27 (24.11%)	17 (30.36%)	11 (33.33%)	12 (48%)
Painful cold	19 (16.96%)	20 (17.89%)	15 (26.79%)	8 (24.24%)	7 (28%)
Brushing	13 (11.61%)	15 (13.39%)	14 (25.0%)	8 (24.24%)	9 (36%)
			Treatment was not effective		
n	153 (57.74%)	153 (57.74%)	77 (57.89%)	48 (59.26%)	33 (56.9%)
Sex (male)	66 (43.14%)	66 (43.14%)	30 (38.95%)	17 (35.42%)	8 (24.24%)
Age, y	56.16	56.16	55.23	52.77	50.91
	(47–65)	(47–65)	(48–65)	(42.25–64.25)	(39–62)
0–15 McGill score	6.58	6.51	7.75	7.88	8.33
	(4–10)	(3–10)	(4–11)	(3.75–11.25)	(6–11)
0–10 pain intensity	5.78 (4–8)	5.95 (4–8)	5.9 (4–8)	5.31 (3.75–8)	5.85 (4–8)
Sensory dimension	2.99 (1–5)	2.94 (1–5)	3.86 (1–6)	4.04 (1.75–6)	4.48 (2–7)
Affective dimension	2.48 (1–4)	2.37 (1–4)	2.65 (1–4)	2.24 (1–3)	2.55 (1–4)
Evaluative dimension	1.16 (1–2)	1.2 (1–2)	1.25 (1–2)	1.06 (1–1)	1.3 (1–2)
Neurophatic pain scale					
Burning	116 (75.82%)	114 (74.51%)	59 (76.62%)	35 (72.92%)	21 (63.64%)
Hypoesthesia to touch	63(41.18%)	42(37.17%)	29 (37.66%)	18(37.5%)	10 (30.3%)
Numbness	51 (33.33%)	55 (35.95%)	37 (48.05%)	20 (41.67%)	16 (48.48%)
Pins and needles	47 (30.72%)	58 (37.91%)	25 (32.47%)	16 (33.33%)	9 (27.27%)
Tingling	58 (37.91%)	58 (37.91%)	37 (48.05%)	23 (47.92%)	20 (60.61%)
Electric shocks	42 (27.45%)	39 (25.49%)	26 (33.77%)	15 (31.25%)	13 (39.39%)
Painful cold	35 (22.88%)	25 (16.34%)	15 (19.48%)	10 (20.83%)	8 (24.24%)
Brushing	24 (15.69%)	21 (13.73%)	13 (16.88%)	10 (20.83%)	12 (36.36%)

We always consider the occurrence of an additional consultation, which is the one we wish to predict. Thus, the description of the third consultation refers to patients with at least four consultations with the doctor. Intuitively, making use of a minimum number of visits, we aim to predict with a high degree of confidence whether the patient will respond positively to standard treatment for chronic pain. As expected, increasing the number of consultations decreases the number of patients attending. Hence, 133 patients followed at least four consultations, 81 patients at least five consultations, and 58 patients at least six consultations. As the number of consultations increases, it also increases the percentage of patients for whom the treatment was effective. Interestingly, women became more prevalent.

6.4 Experimental Setup

TreeSHAP requires a trained model in order to compute SHAP values. As previously discussed, these SHAP values are used as features in posterior iterations/consultations. As the model is trained using labels, it is important to clarify that labels on the training set are not included on a posterior validation set via SHAP values. Instead, SHAP values are extracted from the current validation set (not from the training set) and transferred to the next training set as features. More specifically, we conducted five-fold cross-validation in longitudinal data, i.e., we arranged the patients into five folds and at each run, four folds are used as training set, and the remaining fold is used as test set. To evaluate the performance of the prediction models, we used the standard AUC (area under the ROC curve) measure [Fawcett, 2006; Hanley and McNeil, 1982]. In all experiments we report the average AUC over the five runs.

For comparison purposes, we provide the following baseline algorithms:

- Prediction models trained using features from the first consultation only. This configuration does not make use of any longitudinal information in the data.

- Prediction models trained using features from the current consultation only. This configuration does not implement memory.

- Prediction models trained using features accumulated from all previous consultations. This is the stronger baseline as these models have access to features from previous consultations.

In addition to the ED-Ensemble, we also evaluate general machine learning algorithms, namely XGBoost and Random Forests. Table 6.3 shows the prediction performance for the models used as baselines. Clearly, ED-Ensemble shows superior prediction performance in all cases.

Accumulating features from previous consultations leads to better results. In general, there is a gain in prediction performance as more features are included into the model (i.e., more consultations). The gain provided is quite significant. One major drawback of systematically accumulating consultations is that it markedly increases the total number of features. The data resulting from accumulating the first five consultations has 1,660 features. Algorithms that are not robust to handle so many features present limitations.

Table 6.3: AUC measures for prediction models used as baselines.

Data	XGBoost	Random Forests	ED-Ensemble
Consultation 1 only	0.562 (±0.005)	0.497 (±0.080)	0.766 (±0.031)
Consultation 2 only	0.523 (±0.065)	0.526 (±0.101)	0.761 (±0.095)
Consultation 3 only	0.432 (±0.212)	0.326 (±0.134)	0.805 (±0.031)
Consultation 4 only	0.516 (±0.056)	0.463 (±0.066)	0.822 (±0.082)
Consultation 5 only	0.558 (±0.188)	0.520 (±0.117)	0.872 (±0.011)
Up to consultation 2	0.539 (±0.079)	0.504 (±0.035)	0.805 (±0.080)
Up to consultation 3	0.623 (±0.092)	0.472 (±0.153)	0.813 (±0.063)
Up to consultation 4	0.715 (±0.142)	0.589 (±0.238)	0.935 (±0.048)
Up to consultation 5	0.743 (±0.197)	0.639 (±0.189)	0.917 (±0.044)

6.4.1 Answering RQ1

Table 6.4 presents our main results. It shows the estimated AUC of prediction models trained using our proposed method. For instance, the description "SHAP + Consultation 5" indicates that the prediction model was trained and evaluated using features from the current consultation (i.e., consultation 5) combined with SHAP values as meta-features extracted from a prediction model that was trained using data up to consultation 4.

Table 6.4: AUC measures for prediction models trained using features of the current consultation combined with the SHAP values extracted from the previous prediction model.

Data	XGBoost	Random Forests	ED-Ensemble
SHAP + Consultation 2	0.653 (±0.065)	0.666 (±0.090)	0.818 (±0.061)
SHAP + Consultation 3	0.822 (±0.011)	0.684 (±0.199)	0.903 (±0.056)
SHAP + Consultation 4	0.829 (±0.085)	0.733 (±0.085)	0.914 (±0.035)
SHAP + Consultation 5	0.843 (±0.045)	0.810 (±0.110)	0.945 (±0.053)

The results show that all prediction models present significant gains when increasing the number of consultations granted. Comparing the relative gains over the baselines using the first visit only with our approach using five consultations, our method increased 50% relative to XGBoost, 62.98% relative to Random Forests, and 23.37% relative to ED-Ensemble. The prediction model with the best absolute performance obtained is the ED-Ensemble, achieving an AUC of 0.945.

Table 6.5 compares the prediction performance obtained by each algorithm using the feature accumulation method and our proposed method. The gain is shown as the percentage difference between the performance of the methods being compared. We also performed Welch's t-tests with $p = 0.01$. Our approach using XGBoost and Random Forest learning algorithms is statistically different from the accumulation strategy under all configurations. For the ED-Ensemble, the methods compared are statistically different, except when considering up to 5 consultations. Ten of the twelve comparisons performed were verified positive for our method. For one configuration, our approach was poorer, and for one configuration, there was no statistical difference. For the ten positive cases, the gain achieved ranged from 1.61% to 44.91%. Thus, EXP-MF presented as a pretty effective method, taking into account XGBoost and Random Forest learning algorithms. The higher accuracy with ED-Ensemble in both strategies indicates that the algorithm by nature is capable of selecting only the essential information from previous consultations. Also, ED-Ensemble is robust to dimensionality increase.

Table 6.5: Gains in prediction performance provided by our method when compared with accumulating features.

Data	XGBoost Gain	Random Forests Gain	ED-Ensemble Gain
Consultation 2	+21.15%	+31.14%	+1.61%
Consultation 3	+31.94%	+44.91%	+11.07%
Consultation 4	+15.94%	+24.44%	−2.24%
Consultation 5	+13.45%	+26.76%	+3.05%

In addition to the observed increase in prediction performance, another benefit provided by our proposed method is to avoid a significant rise in the number of features within the prediction model. Models built using only features from the current consultation employ 332 features, regardless of the consultation. Models built using concatenated features utilize 332 features for the first consultation and 1,660 for the fifth consultation. Finally, our method on the fifth consultation uses 60 features only, 52 of which were unique. Our approach varies the number of features between visits since the SHAP importance meta-features size depends on how many features have been selected by the ED-Ensemble algorithm.

6.4.2 Answering RQ2

We aim to evaluate whether the SHAP values extracted from previous consultations are more discriminative than features from previous consultations.

We create datasets combining the strategies of accumulating consultation and SHAP values as meta-features from previous consultations (i.e., memory). These generated datasets are used to train the Random Forest and XGBoost models. Next, we estimate the SHAP summary plot from these models, thus obtaining insight into which features each model considers most important. We want to verify whether the SHAP values as meta-features of previous consultations or the previous consultations themselves are considered more discriminating by the respective models.

The summary plot presents a global overview on the importance of features and measures. The resulting plot has the following characteristics: each row corresponds to a factor and has as many dots as patients; a dot represents the value of the corresponding factor for a patient; red dots indicate that the factor assumes a high value for the corresponding patient and blue dots indicate that the factor assumes a low value; the vertical line shows whether the impact of a factor increased the prediction (i.e., the dot is on the right size) or decreased it (i.e., the dot is on the left side); and factors impacting most the model prediction appear on the top of the plot.

Figures 6.3 and 6.4 displays the summary plot of the Random Forest and XGBoost models for distinct datasets. Each plot corresponds to a specific dataset in both figures: the first plot is the dataset with the first two concatenated consultations plus the SHAP values as meta-features extracted from the first consultation. The second plot has the first three concatenated consultations plus the SHAP values as meta-features extracted from the model trained up to the second consultation, and so on. Considering the high quantity of features in the dataset, we decided to present only the five most important features for better visualization.

Each feature can fit in one category: feature with information about current consultation, feature with information about previous consultation, or meta-feature with previous explanatory factor (SHAP value). In this experiment, we added suffixes to their names in order to differentiate the features. The suffix '$\_t\{n\}$', where $1 \leq n \leq 5$, indicates the consultation in which the feature was obtained. Thus, in Figure 6.3, in the first plot, we have the feature $intensity\_t1$ specifying that the feature $intensity$ was obtained in consultation 1. Since information from the first two consultations is considered in this first plot, we have that $t1$ refers to a previous consultation. On the other hand, in Figure 6.4, also in the first plot we have the feature $intensity\_t2$. Similarly, this graph deals with the first two consultations, and thus this feature contains information about the fresh current consultation.

Another suffix used distinguishes whether the variable is a meta-feature with SHAP value extracted from previous queries. It is described by '$\_m\{k\}\_shap$', where $0 \leq k \leq b$. The variable b represents the number of base models that make up the ensemble generated by the ED-ensemble algorithm. For instance, in the first plot in Figure 6.3, the feature $intensity\_m0\_shap$ is a meta-feature with SHAP value extracted from the feature *intensity* obtained from the base model 0 of the generated ED-ensemble. When we consider two consultations, an ED-ensemble is generated with the data from the first consultation, and we extract SHAP values for each base model composing the ensemble. In particular, this ensemble is composed of 7 base models, identified from 0 to 6. Therefore, the SHAP value $intensity\_m0\_shap$ is obtained from the base model identified as 0. Similarly, in the same graph, the variable $mcgill\_m3\_shap$ is a meta-feature with the SHAP value extracted from the variable *mcgill* obtained from the base model 3 of the generated ED-ensemble. When considering more consultations, concatenation of this suffix may occur. Thus, in the second plot in Figure 6.3, we have the variable $intensity\_m3\_shap\_m2\_shap$. We must read it from right to left to correctly analyze this feature. Thus, considering that we are using the information from three consultations, the meta-feature has the SHAP values extracted from model 2 of the ED-ensemble generated under the previous consultations (consultations 1 and 2). In turn, during the processing of the first two consultations, this meta-feature was extracted from the SHAP values extracted from the base model 3 of the ED-ensemble generated under the previous consultations (consultation 1). Therefore, we can observe that a meta-feature obtained from consultation 1 data can be carried into later consultations. The number of suffixes allows us to identify when the meta-feature was originally estimated.

In Figure 6.3, of the 20 most important features for the Random Forest model, 17 are SHAP values as meta-features extracted from previous models. Three features are features from previous consultations, and no information about the current consultation is used. Hence, Random Forest does not make use of any raw data from prior consultation. In Figure 6.4, by contrast, when considering the XGBoost model, 15 SHAP values as meta-features are among the most critical. One feature is information from current consultation, and four features are from previous consultations.

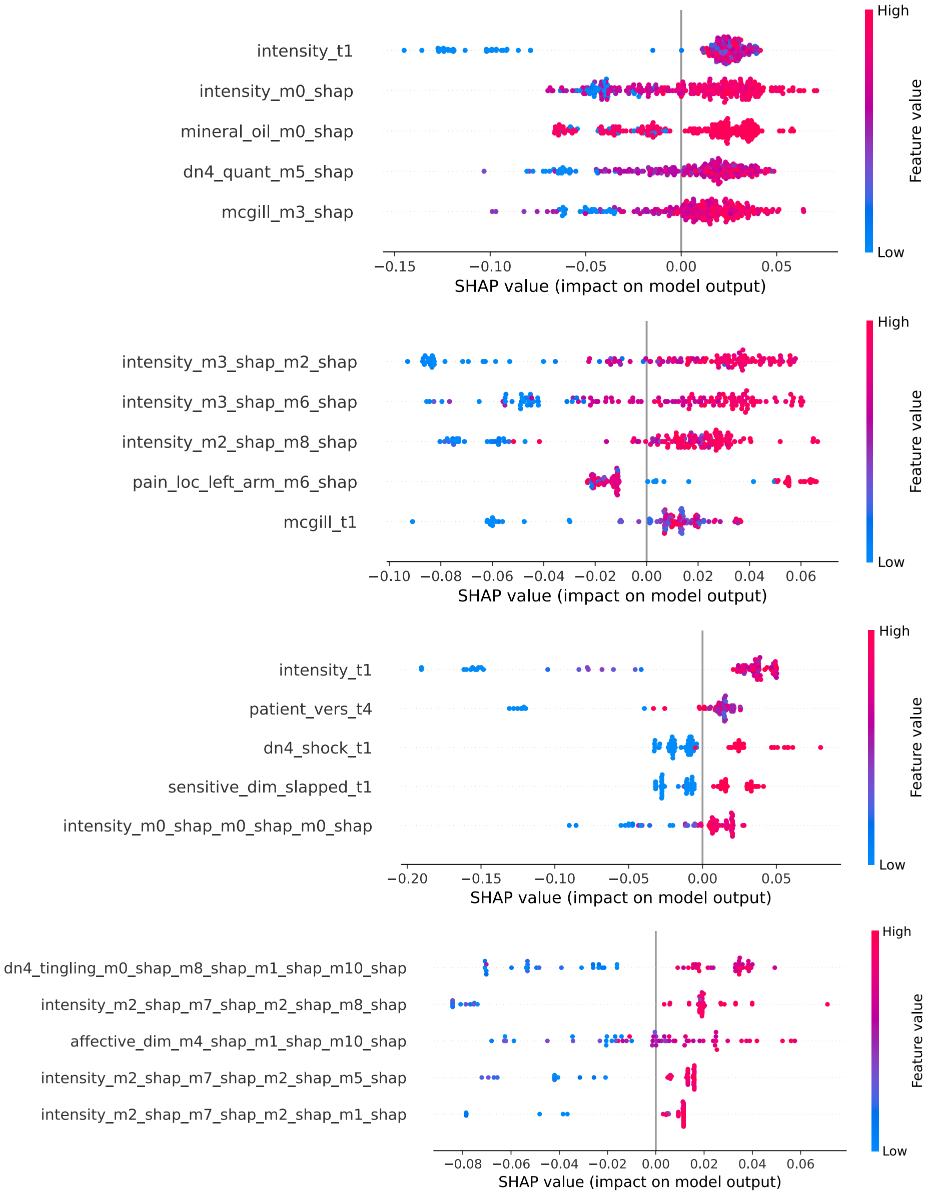


Figure 6.3: (Color online) A set of summary plots showing only the five most important features extracted from Random Forest models. Each plot represents the maximum amount of consultations used. The first plot uses up to two consultations. The dataset was constructed from the concatenation of the first two consultations, combined with SHAP values as meta-features obtained from a model trained on the first consultation only. The second plot considers up to three consultations, the fourth plot is four consultations, and the last plot is five consultations.

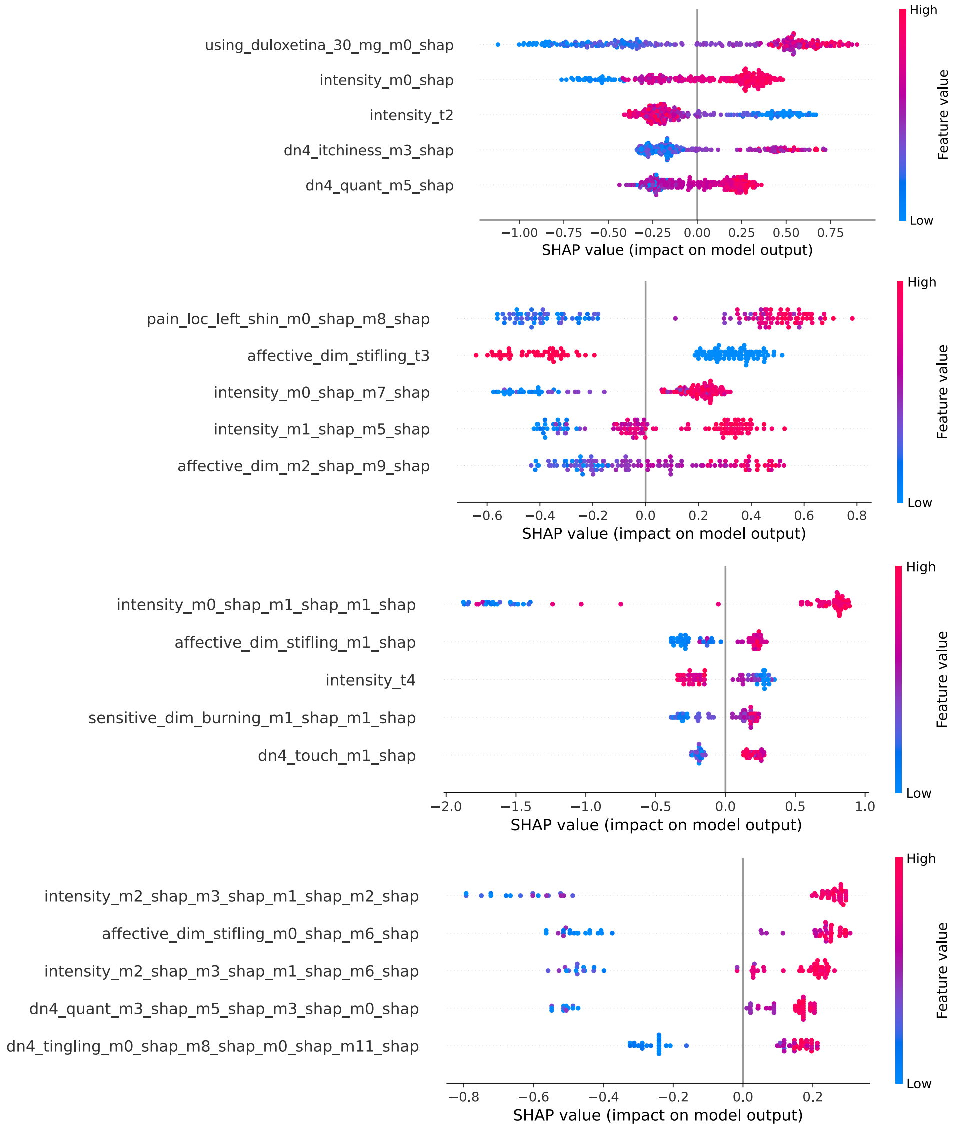


Figure 6.4: (Color online) A set of summary plots showing only the five most important features extracted from XGBoost models. Each plot represents the maximum amount of consultations used. The first plot uses up to two consultations. The dataset was constructed from the concatenation of the first two consultations, combined with SHAP values as meta-features obtained from a model trained on the first consultation only. The second plot considers up to three consultations, the fourth plot is four consultations, and the last plot is five consultations.

It is remarkable that this proportion of previous' models importance features is achieved. The more consultations considered, the more features are incorporated into the generated dataset. For each additional consultation, just over 300 features are included. For instance, when five consultations are considered, we have 1,697 features. Of this total, 1,660 features refer to previous and current consultations and only 37

SHAP values as meta-features from earlier consultations. That is, about 2.23% of the features are SHAP values as meta-features.

Finally, to answer question RQ2, we affirmatively state that SHAP meta-features have a more significant impact on models than the prior information itself. When considering more consultations, more features are added. Yet, looking at the top five features more critically, most models regarded the SHAP meta-features as most impactful. This strongly indicates that SHAP meta-features add more knowledge than the value itself. The next question concerns the knowledge incorporated.

6.4.3 Answering RQ3

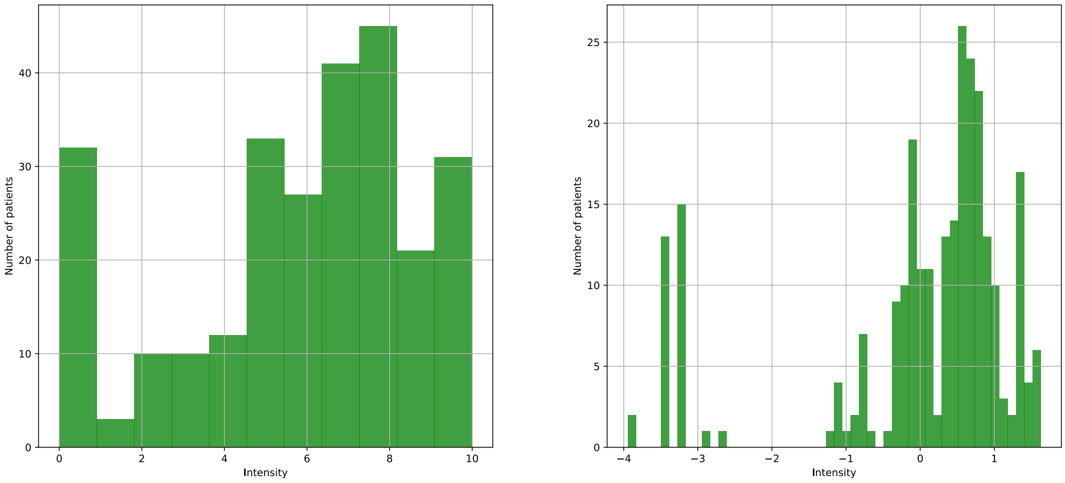
In order to answer research question RQ3, we will study the gain from replacing information from the previous visit with its SHAP.

Two hundred and sixty-five patients that make up the data set have at least three visits. Amidst the 332 variables in the first visit, one of the most important is the variable *intensity*. The variable *intensity* represents the degree of pain reported by the patient and consists of an integer that takes on values in the range 0 to 10. It will be the variable we will analyze. Figure 6.5a presents a histogram correlating the number of patients distributed over each intensity value. The mean of the variable is 5.93 (±3.05). As can be seen, the highest intensity values are 7 and 8, also the only ones to exceed the forty-patient threshold.

The SHAP values extracted from the variable *intensity*, here called by the variable *intensity_m0_shap*, have different behavior from the original variable. Figure 6.5b presents a histogram correlating the number of patients distributed over the meta-features SHAP obtained from intensity values at the first consultation. Firstly, it can be seen that the data is distributed over a much larger number than 11 possible values. Precisely, all the values generated are unique, with mean $2.51e-18$ (± 1.363).

Figure 6.6 shows the histogram plot of the SHAP values extracted from feature *intensity* with value 7 at first consultation. It is interesting to note that the same original values have even been mapped to negative values. Precisely, there are 41 patients with intensity 7 at consultation 1. Using only the information from the first consultation, the ED-ensemble trained on this data can predict 24 of these patients correctly and 17 incorrectly.

Figures 6.7, 6.8 and 6.9 comparatively present force plots using only consultation 1 (first plot), and using consultation 2 plus SHAP values as meta-features from the previous consultation (second plot). The force plot exhibits how each variable and its respective value contributes to the prediction. Visually, only the largest contributions



(a) (b)

Figure 6.5: (Color online) Histogram plot showing the distribution of intensity values and SHAP values extracted from the same intensity values. Figure 6.5a presents the intensity values at the first consultation, and Figure 6.5b presents the SHAP values extracted from intensity values at the first consultation.

are displayed. When using the information from the second consultation plus SHAP values as meta-features from the previous consultation, of the 17 incorrectly predicted at consultation 1, 8 are correctly predicted at consultation 2.

In the first plot in Figure 6.7, the variable *intensity* with value 7 has a positive contribution in the final output. The force plot exhibits how each variable and its respective value contributes to the prediction. Visually, only the largest contributions are displayed. In this case, the largest contribution to the final output emerges from the variable *Pain loc: right foot*, with value 0. Our model predicts a probability, and we aim to obtain a classification. We will assume a class 1 if output ≥ 0.5, and class 0 otherwise. The expected (correct) output for this instance is class 0, and thus the model prediction is incorrect. In the second graph in Figure 6.7, instead of using the *intensity* variable from consultation 1, we use the SHAP value, denoted here by the *intensity_ m0_ shap* variable. The value of 0.02864 assigned to the variable corresponds to its contribution in consultation 1. Here, what is interesting is that the contribution of the variable *intensity_ m0_ shap* is negative, and the model prediction is correct.

Similar behavior is observed in Figures 6.8 and 6.9. In both cases, the variable *intensity* takes on the value 7 at consultation 1, generating a positive contribution to the output. However, since the true output is 0, the direction of the contribution is

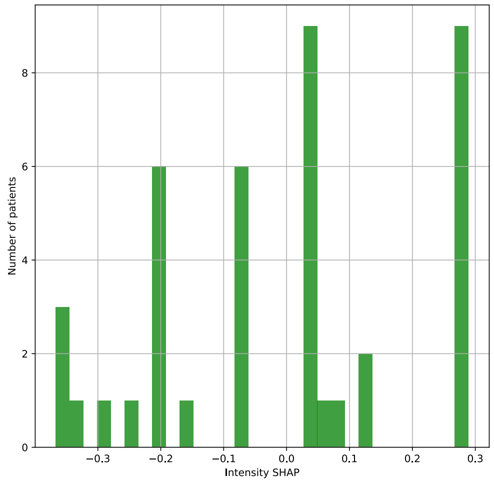
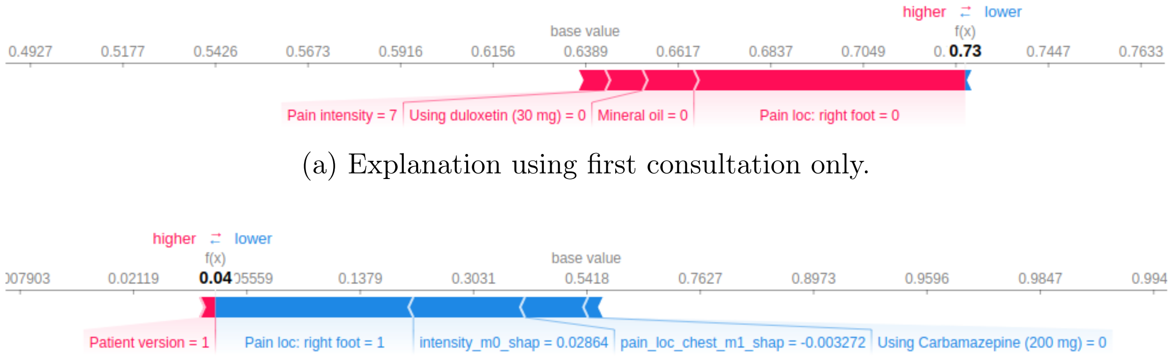


Figure 6.6: (Color online). Histogram plot showing the distribution of patients over the meta-features SHAP extracted from patients with intensity value 7 at the first consultation.



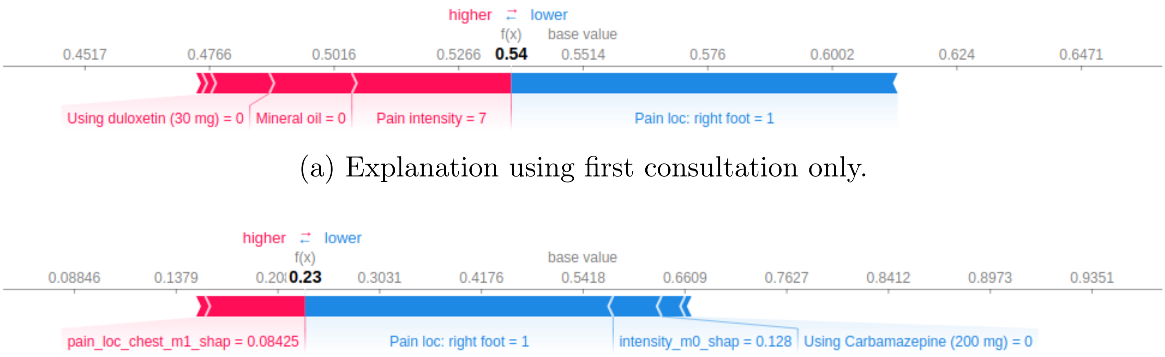
(a) Explanation using first consultation only.

(b) Explanation using the first two consultations.

Figure 6.7: (Color online) SHAP force plots explaning the same instance (83). Figure 6.7a considering the first visit only, and Figure 6.7b considering the second consultation plus the SHAP features extracted from the first consultation.

incorrect. At consultation 2, the correct output is predicted, and the SHAP variables now contributes in the opposite (and correct) direction. Finally, unlike in consultation 1, in consultation 2, the models can predict the correct output.

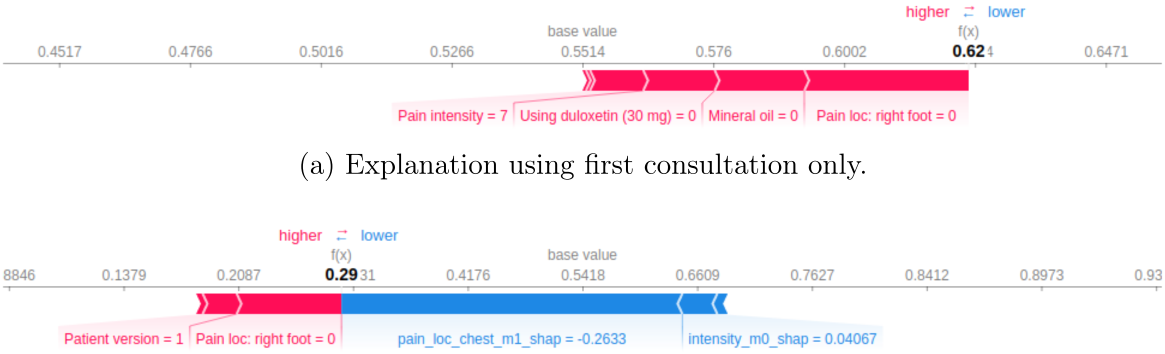
Regarding RQ3, we can observe that when moving from consultation 1 to 2, we transferred knowledge instead of carrying over raw information from previous consultations. While 41 patients have the same *intensity* value at consultation 1, SHAP computes 41 contributions of different magnitudes, including positive and negative values. The main benefit of using SHAP values as meta-features is that the computation



(a) Explanation using first consultation only.

(b) Explanation using the first two consultations.

Figure 6.8: (Color online) SHAP force plots explaning the same instance (121). Figure 6.8a considering the first visit only, and Figure 6.8b considering the second consultation plus the SHAP features extracted from the first consultation.



(a) Explanation using first consultation only.

(b) Explanation using the first two consultations.

Figure 6.9: (Color online) SHAP force plots explaning the same instance (133). Figure 6.9a considering the first visit only, and Figure 6.9b considering the second consultation plus the SHAP features extracted from the first consultation.

of a variable's contribution considers its value and its often complex relationship with other variables.

6.5 Discussion

The aim of the present research was to investigate the use of models' explanations acting as a memory in longitudinal data. One of the more significant findings to emerge from this study is that we show evidence suggesting explanatory factors from previous data point time renders more critical information than the raw data itself.

The research has also shown that this enhanced data leads to an increase in accuracy performance. The proposed approach, using the ED-Ensemble algorithm and up to five consultations, achieved an increase of 23.37% in AUC, from 0.766 to 0.945. A similar trend is also observed in the XGBoost and Random Forests algorithms. As we increase the number of consultations considered, our proposed approach increases the AUC, with XGBoost improving 50% and Random Forest 62.98%.

Conversely, by simply accumulating raw data from all sequential consultations, the performance in considering more data seems to deteriorate. Two factors might contribute to this diminish in performance. First, each additional visit accounts for more than 300 features. Also, considering more consultations, the number of instances is reduced, doubly impacting the dimensional space size. Additionally, there are correlations within consultations from the same subjects (inherent in longitudinal data). These correlations become more strong as more information is granted. Traditional machine learning approaches are usually not explicitly tailored for handling longitudinal data, thus mitigating the accuracy performance. Lastly, in most scenarios, using EXP-MF presented as statistically superior to accumulating consultations.

Finally, we presented a data-wise approach that enhances traditional machine learning approaches in longitudinal data, even if such algorithms are not specifically designed to handle this data organization.

Chapter 7

Conclusions and Future Work

This chapter presents the conclusions and further research for this thesis. It is organized as follows. Section 7.1 presents a review of our statement and contributions. Section 7.2 discusses future works. Finally, Section 7.3 outlines the publications during candidature.

7.1 Conclusions of this Thesis

This thesis aims to study an underexplored link between explanatory modeling and predictive modeling, leading to a novel ensemble learning approach. Recently, the emergence of model-agnostic explanation methods, notably SHAP [Lundberg and Lee, 2017], allowed obtaining the prediction explanation from models characterized as black-boxes. Such models as XGBoost [Chen and Guestrin, 2016] and Random Forests [Breiman, 2001] ensembles, by their nature, are not explainable. We show a nearly unknown relationship between explanation and prediction. The explanation of a model is more correlated with the model output than the feature set. Also, explanations provide more knowledge than the features information, as it sums up each feature's contribution.

The ensemble is a technique that uses a combination of models to generate predictions, achieving better performance than any of the single classifiers of the ensemble alone. Several factors are associated with the effectiveness of an ensemble: (a) relationship between accuracy and diversity of the base models [Kuncheva and Whitaker, 2003; Opitz, 1999] and (b) stability [Breiman, 1996] are two of them. Despite diversity being recognized as an essential characteristic in constructing good ensembles, there is no single, universally accepted measure for diversity.

Specifically, we are interested in the use of model explanations as a measure of diversity. Our approach exploits two concepts. First, local models that compose the ensemble should be diverse in terms of their explanatory factors. Also, candidate models should be arranged by seeking stability, i.e., models that perform similar predictions should also be similar in terms of their explanatory factors.

We evaluate our ensemble learning approach to predict the evolution of pain relief in patients with unknown chronic pain conditions. Despite the existence of several guidelines and recommendations for its treatment, up to 40% of chronic pain patients may remain symptomatic despite the best medical treatment. Precisely defining the best therapy for a patient is still a challenge. The chronic pain dataset is well suited for our purposes because it is characterized as multi-structure phenomena. Chronic pain can originate from multiple factors, and as such, highly correlated data regions and ground-truth output are expected.

In multiple phenomena problems, typically exists a particular set of "backbone features" that, once set, causes the remainder of the features to decompose into different subsets in the data space. The backbone structure suggests that the problem is defined by multiple local structures. Often, these many-structure phenomena are modeled using the simple *all-in-one* approach, which fits all the available factors (or features) into a single sub-optimal model. Instead, by learning local models composed of different feature sets, we can achieve feature decompositions that feature selection algorithms can not. Further, our approach proved superior to BENCH [Pansombut et al., 2011], significantly outperforming by 6.8%. BENCH is a well-known biclustering technique that performs feature decomposition.

Our experiments demonstrate that our novel ED-Ensemble approach using XGBoost provides relative performance gain up to 9.86% when compared with the best XGBoost local model and up to 20.37% compared to the XGBoost *all-in-one* approach. When using the Random Forests learning algorithm, the performance is 4.17% higher than the best local model and 15.03% higher than the *all-in-one* approach. Along with the improved performance, the generated ensemble makes use of a significantly reduced number of features. In particular, for the XGBoost learning algorithm and the VAS30 label, the generated ED-Ensembles uses as low as 15% of the features. This remarkably reduced subset yields side benefits as improving the predictions' explainability.

Interestingly, our ED-Ensemble approach shows a superior performance even when base models are ensemble algorithms representing distinguished ensemble construction approaches: bagging (Random Forest) and boosting (XGBoost). Although the error reduction of both models may be given as similar in the literature, how this objective is achieved is different. Bagging seeks to reduce the total error mainly by reducing the variance while boosting seeks to reduce the error mainly by reducing the bias.

We also show an analysis of the error decomposition into bias and variance to characterize the gain provided by the proposed ensemble. Our ensemble can reduce the error by two competing strategies. Learning more complex relations by selecting performant and diverse models, thus reducing the bias error. At the same time, using smaller base models keeps the variance low. This strategy allowed a performance gain to be achieved with either algorithms.

Further, motivated by the standard chronic pain treatment protocol that comprises many subsequent appointments, we extend our work to handle longitudinal data. In general, tradicional machine learning approaches are not specifically tailored to work with longitudinal data. Neglecting the assumption that these algorithms were designed for cross-section data leads to the induction of models with poor accuracy performance. We used the models' explanations as meta-features acting as memory from previous consultations. We follow the explanation-diversity feature selection procedure proposed which indicates a preference for choosing feature importances carried over earlier consultations instead of the raw information. This method is a strong indication that feature importances contain more decisive information than features from previous consultations. Our experiments consistently showed that the more consultations granted, the higher the performance achieved. Our approach EXP-MF with an ED-Ensemble could achieve an AUC of 0.945 considering five consultations. A similar uptrend in AUC was also observed for the Random Forests and XGBoost algorithms.

Finally, this thesis presents an approach to ensemble generation based on explanations diversity, aimed at multi-structure phenomena modeling. By relating local structures and model explanations, our ensemble learning approach achieved superior performance to the well-established ensemble methods XGBoost and Random Forests. Our novel ED-Ensemble approach presents as a superior alternative to the *all-in-one* approach in multiple phenomena problems with cross-sectional and also longitudinal data.

7.2 Future Work

- "Would the patient x have a decrease of blood pressure if he took the medication m_1?". "What if he takes the medication m_2?". These types of questions are often framed as *counterfactual* questions. Counterfactuals are specific missing values cases in which the missing values cannot be observable. By relating local structures and model explanations, we can generate many reduced models. Each reduced model can generate a counterfactual, identifying minimal changes to alter its outcome. Instead of constructing a single counterfactual, we would like to generate a set of minimal agreeing changes.

- When we analyze the feature impact, we estimate it through the average of the impacts in the model. Nonetheless, there are features in which the impact distribution is nearly symmetrical. This distribution produces an average close to zero. Partitioning the feature to distinguish cases resulting in positive and negative impacts could add more knowledge to the model explanatory factors.

- Cooperate to develop a Clinical Decision Support System (CDSSs) for chronic pain evolution prediction from the findings in this work. Using the explanation-based ensemble would bring two practical benefits: increased performance for predicting the evolution of chronic pain and a reduced number of features. In particular, the reduction of features would allow the creation of questionnaires with fewer questions.

7.3 Publications During Candidature

We achieved some results and conclusions for this proposal. The following publications have explicitly indicated the authors' contribution to this work.

Published

– Costa, A. B. D., Moreira, L., Andrade, D. C. D., Veloso, A., and Ziviani, N. (2021). Predicting the Evolution of Pain Relief: Ensemble Learning by Diversifying Model Explanations. *ACM Transactions on Computing for Healthcare*, 2(4), 1-28.

Submitted (Under Review)

– Costa, A. B. D., Moreira, L., Andrade, D. C. D., Veloso, A., and Ziviani, N. (2022). Predicting the Evolution of Pain Relief: Learning Treatment Effectiveness Using Model-Explanations as Meta-Features. *Artificial Intelligence in Medicine.*

Bibliography

Abad-Grau, M., Ierache, J., Cervino, C., and Sebastiani, P. (2008). Evolution and challenges in the design of computational systems for triage assistance. *Journal of biomedical informatics*, 41(3):432--441.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (2005). Automatic subspace clustering of high dimensional data. *Data Mininig and Knowledge Discovery*, 11(1):5--33.

Bellman, R. (1966). Dynamic programming. *Science*, 153(3731):34--37.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92--100. ACM.

Breiman, L. (1996). Bias, variance, and arcing classifiers. Technical report, Tech. Rep. 460, Statistics Department, University of California, Berkeley.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5--32.

Capitaine, L., Genuer, R., and Thiébaut, R. (2021). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research*, 30(1):166--184. ISSN 0962-2802, 1477-0334.

Chen, M., Weinberger, K. Q., and Chen, Y. (2011). Automatic feature decomposition for single view co-training. In *Proceedings of the 28th International Conference on Machine Learning*, pages 953--960.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Knowledge Discovery and Data Mining*, pages 785--794.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 93--103.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273--297. Publisher: Springer.

Domingos, P. (2000). A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*, pages 231--238.

Dworkin, R., Turk, D., Wyrwich, K., Beaton, D., Cleeland, C., Farrar, J., Haythornthwaite, J., Jensen, M., Kerns, R., and Ader, D. (2008a). Interpreting the clinical importance of treatment outcomes in chronic pain clinical trials: Immpact recommendations. *The journal of pain*, 9(2):105--121.

Dworkin, R., Turk, D., Wyrwich, K., Beaton, D., Cleeland, D., Farrar, J., Haythornthwaite, J., Jensen, M., Kerns, R., Ader, D., Brandenburg, N., Burke, L., Cella, D., Chandler, J., P, P. C., Dimitrova, R., Dionne, R., Hertz, S., Jadad, A., Katz, N., Kehlet, H., Kramer, L., D.Manning, McCormick, C., McDermott, M., McQuay, H., Patel, S., Porter, L., Quessy, S., Rappaport, B., Rauschkolb, C., Revicki, D., Rothman, M., Schmader, K., Stacey, B., Stauffer, J., von Stein, T., White, R., Witter, J., and Zavisic, S. (2008b). Interpreting the clinical importance of treatment outcomes in chronic pain clinical trials: IMMPACT recommendations. *The Journal of Pain*, 9(2):105--121.

Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, pages 226--231.

Fard, M. J., Wang, P., Chawla, S., and Reddy, C. K. (2016). A Bayesian Perspective on Early Stage Event Prediction in Longitudinal Data. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3126--3139. ISSN 1041-4347.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861--874.

Ferreira, K., Bastos, T., Ciampi, D., Silva, A., Appolinario, J., Jacobsen, M., and Latorre, M. (2016). Prevalence of chronic pain in a metropolitan area of a developing country: a population-based study. *Arquivos de neuro-psiquiatria*, 74(12):990--998.

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2012). *Applied longitudinal analysis*, volume 998. John Wiley & Sons.

Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data.* Cambridge University Press.

Frees, E. W., Young, V. R., and Luo, Y. (1999). A longitudinal data analysis interpretation of credibility models. *Insurance: Mathematics and Economics*, 24(3):229--247. Publisher: Elsevier.

Friesen, A. L. and Domingos, P. M. (2015). Recursive decomposition for nonconvex optimization. In *International Joint Conference on Artificial Intelligence*, pages 253--259.

Ghosh, J. and Acharya, A. (2011). Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):305--315.

Goldstein, B., Navar, A., and Carter, R. (2016). Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges. *European heart journal*, 38(23):1805--1814.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (1992). Gene Selection for Cancer Classification using Support Vector Machines. *Biopolymers*, 32(3):277--292. ISSN 10970282.

Hajjem, A., Bellavance, F., and Larocque, D. (2014). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, 84(6):1313--1328. Publisher: Taylor & Francis.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hall, M. A. (2000). Correlation-based feature selection of discrete and numeric class machine learning.

Hanley, J. and McNeil, B. (1982). The meaning and use of the area under a receiver operating characteristic ROC curve. *Radiology*, 143:29--36.

Heckman, J. J. and Walker, J. R. (1990). The relationship between wages and income and the timing and spacing of births: Evidence from Swedish longitudinal data. *Econometrica: journal of the Econometric Society*, pages 1411--1441. Publisher: JSTOR.

Hedeker, D. and Gibbons, R. D. (2006). *Longitudinal data analysis.* Wiley-Interscience.

Hill, J., Dunn, K., Lewis, M., Mullis, R., Main, C., Foster, N., and Hay, E. (2008). A primary care back pain screening tool: identifying patient subgroups for initial treatment. *Arthritis Rheum*, 5(59):632--641.

Hu, S. (2021). *Statistical modeling and machine learning in longitudinal data analysis*. PhD Thesis, Queensland University of Technology.

Jha, D. and Kwon, G.-R. (2017). Diagnosis of alzheimer's disease using a machine learning technique. *Alzheimer's & Dementia*, 13(7):1538.

Jiang, Z., Do, H. N., Choi, J., Lee, W., and Baek, S. (2020). A Deep Learning Approach to Predict Abdominal Aortic Aneurysm Expansion Using Longitudinal Data. *Frontiers in Physics*, 7:235. ISSN 2296-424X.

Jolliffe, I. (2011). *Principal Component Analysis*. Springer.

Kira, K. and Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *AAAI Conference on Artificial Intelligence*, volume 2, pages 129--134.

Kiritchenko, S. and Matwin, S. (2011). Email classification with co-training. In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, pages 301--312. IBM Corp.

Kohavi, R. and Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275--83.

Konerman, M. A., Zhang, Y., Zhu, J., Higgins, P. D., Lok, A. S., and Waljee, A. K. (2015). Improvement of predictive models of risk of disease progression in chronic hepatitis C by incorporating longitudinal data. *Hepatology*, 61(6):1832--1841. Publisher: Wiley Online Library.

Kuncheva, L. I., Roli, F., Marcialis, G. L., and Shipp, C. A. (2001). Complexity of data subsets generated by the random subspace method: an experimental investigation. In *International Workshop on Multiple Classifier Systems*, pages 349--358. Springer.

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181--207.

Lundberg, S. and Lee, S. (2017). A unified approach to interpreting model predictions. In *Annual Conference on Neural Information Processing Systems*, pages 4768--4777.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56--67. ISSN 2522-5839.

Maimon, O. and Rokach, L. (2002). Improving supervised learning by feature decomposition. In *International Symposium on the Foundations of Information and Knowledge Systems*, pages 178--196.

Maldonado, S. and Weber, R. (2009). A wrapper method for feature selection using Support Vector Machines. *Information Sciences*, 179(13):2208--2217. ISSN 00200255.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Willey Series in Probability and Statistics John Wiley & Sons, New York.

Melzack, R. (1975). The McGill pain questionnaire. major properties and scoring methods. *Pain*, 1:277--299.

Navani, A. and Li, G. (2016). Chronic Pain Challenge: A Statistical Machine-learning Method for Chronic Pain Assessment. *Journal on Recent Advances in Pain*, 2(3):82--86. ISSN 2454-6607.

Ngufor, C., Van Houten, H., Caffo, B. S., Shah, N. D., and McCoy, R. G. (2019). Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c. *Journal of biomedical informatics*, 89:56--67. Publisher: Elsevier.

Nigam, K. and Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Cikm*, volume 5, page 3.

Oliveira, A. L. and Madeira, S. C. (2004). Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24--45.

Opitz, D. W. (1999). Feature selection for ensembles. *AAAI/IAAI*, 379:384.

Pansombut, T., Hendrix, W., Gao, Z. J., Harrison, B. E., and Samatova, N. F. (2011). Biclustering-driven ensemble of bayesian belief network classifiers for underdetermined problems. In *International Joint Conference on Artificial Intelligence*, pages 1439--1445.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825--2830.

Perveen, S., Shahbaz, M., Saba, T., Keshavjee, K., Rehman, A., and Guergachi, A. (2020). Handling Irregularly Sampled Longitudinal Data and Prognostic Modeling of Diabetes Using Machine Learning Technique. *IEEE Access*, 8:21875--21885. ISSN 2169-3536.

Pieterse, A., Stiggelbout, A., and Montori, V. (2019). Shared Decision Making and the Importance of Time. *JAMA - Journal of the American Medical Association*, 322(1):25--26. ISSN 15383598.

Pombo, N., Araújo, P., and Viana, J. (2014). Knowledge discovery in clinical decision support systems for pain management: A systematic review. *Artificial Intelligence in Medicine*, 60(1):1--11.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining*, pages 1135--1144.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, pages 1527--1535.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386. Publisher: American Psychological Association.

Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, 87(418):407--418. Publisher: Taylor & Francis.

Sela, R. J. and Simonoff, J. S. (2012). RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine Learning*, 86(2):169--207. ISSN 0885-6125, 1573-0565.

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, 25(3):289--310.

Speiser, J. L. (2021). A random forest method with feature selection for developing medical prediction models with clustered and longitudinal data. *Journal of Biomedical Informatics*, 117:103763. ISSN 15320464.

Stenberg, A. (2011). Using longitudinal data to evaluate publicly provided formal education for low skilled. *Economics of Education Review*, 30(6):1262--1280. Publisher: Elsevier.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583--617.

Tanay, A., Sharan, R., Kupiec, M., and Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci*, 101(9):2981--2986.

Tanay, A., Sharan, R., and Shamir, R. (2005). Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 9:26--1.

Topchy, A., Jain, A. K., and Punch, W. (2004). A mixture model for clustering ensembles. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 379--390. SIAM.

Valentini, G. and Dietterich, T. G. (2004). Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *Journal of Machine Learning Research*, 5(Jul):725--775.

van der Maaten, L. (2009). Learning a parametric embedding by preserving local structure. In *International Conference on Artificial Intelligence and Statistics*, pages 384--391.

Verbeke, G., Fieuws, S., Molenberghs, G., and Davidian, M. (2014). The analysis of multivariate longitudinal data: a review. *Statistical methods in medical research*, 23(1):42--59. Publisher: Sage Publications Sage UK: London, England.

Verikas, A., Gelzinis, A., and Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern recognition*, 44(2):330--349. Publisher: Elsevier.

Vijayakumar, S. (2007). The Bias-Variance Tradeoff.

Wan, X. (2009). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-volume 1*, pages 235--243. Association for Computational Linguistics.

Wang, P., Laskey, K. B., Domeniconi, C., and Jordan, M. I. (2011). Nonparametric bayesian co-clustering ensembles. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 331--342. SIAM.

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236--244.

White, K., Lee, J., and de C Williams, A. (2016). Are patients' and doctors' accounts of the first specialist consultation for chronic back pain in agreement? *Journal of Pain Research*, pages 1109--1120.

Williams, A. and Craig, K. (2016). Updating the definition of pain. *PAIN*, 157:1.

Zhao, J., Feng, Q., Wu, P., Lupu, R. A., Wilke, R. A., Wells, Q. S., Denny, J. C., and Wei, W.-Q. (2019). Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction. *Scientific reports*, 9(1):1--10. Publisher: Nature Publishing Group.

Appendix A

Running Time Computational Cost

In this appendix, we will discuss the runtime required to generate the proposed ED-Ensemble. Although the time spent to generate the ensembles does not influence the benefits already shown throughout the work, such as increase of the AUC, increase in accuracy, and reduced feature set, it can be a critical factor in the feasibility of applying the technique in miscellaneous use cases.

Nearly the entire runtime contribution originates from generating the model space. Therefore, this will be our study case. Moreover, to provide a standard on measured time, we will begin describing the hardware and software specifications on which the experiments were performed.

This appendix is organized as follows. Section A.1 describes the hardware and software specifications used throughout the work. In Section A.2, we show the time spent to sample the model space under different parameter configurations.

A.1 Hardware and Software Specifications

The hardware specification is Intel(R) Core(TM) i5-10400 CPU @ 4.30GHz, 16GB DDR4 26666 MT/s, and 256GB SSD. As the algorithms used did not make use of the graphics card, we will omit the information.

Regarding the software specifications, we used the Ubuntu 20.04 LTS operating system along with Python 3.8 installed through Anaconda[1].

[1] https://anaconda.org

A.2 Sampling Model Space

As we increase the maximum number of features allowed, we can obtain improved ensembles. Conversely, it is also expected to increase the computational cost. Since interpretability is a crucial aspect of our work, we set the upper limit to 15 features.

We have shown that it is possible to use multiple configurations in sampling model space throughout the work. It should be selected the learning algorithm (XGBoost or Random Forests), the label (VAS30, VAS50, or GIC), and the maximum number of features. Furthermore, there are two possible data sets; the entire dataset (631 instances) or the dataset with at least three visits for each patient (265 instances).

First, we will use the entire dataset. As the most costly scenario, it is a valid representation of the worst case. We will use the upper limit of 15 features, the highest value used. Also, our experiments show that there is no significant difference in running time spent for different labels. Hence we will select the VAS30 label in the experiments. Finally, the only parameter that will be changed to show the computational cost employed is the learning algorithm.

Table A.1 presents the running time (in minutes) when sampling the model space with the entire dataset, XGBoost learning algorithm, VAS30 label, and setting the maximum number of features at 15. As can be seen, the time taken to sample the model space steadily rises as we increase the number of features. This trend is expected as each additional feature necessarily implies an additional data dimension, requiring more time into training. For the specific case where the number of variables is 1, the small amount of 0.17 minutes occurs because only 500 models are sampled. Conversely, for other quantities, 10 000 are sampled. When the maximum number of features is set to 2, 6.5 minutes are needed to generate the model space. On the other hand, 15 features take 16.85 minutes, an increase of 259.23%.

Figure A.1 presents the line plot for better showing the relationship between the number of features and the time spent. It is interesting to note that the time required grows at an approximately constant rate.

Table A.1: Running time spent to sample the model space using XGBoost learning algorithm and VAS30 label. Each row contains data about the maximum number of features, the total time spent to sample 10 000 models with this length, and the third column is the average time per 1 000 models.

# of features	Total time (in minutes)	Avg. time per 1 000 samplings (in minutes)
1	0.17	N/A
2	6.5	0.65
3	7.69	0.76
4	8.4	0.84
5	9.02	0.90
6	9.7	0.97
7	10.63	1.63
8	11.24	1.12
9	12.13	1.21
10	13.19	1.31
11	13.82	1.38
12	14.54	1.45
13	15.58	1.55
14	16.31	1.61
15	16.85	1.68
Total	165.79	N/A

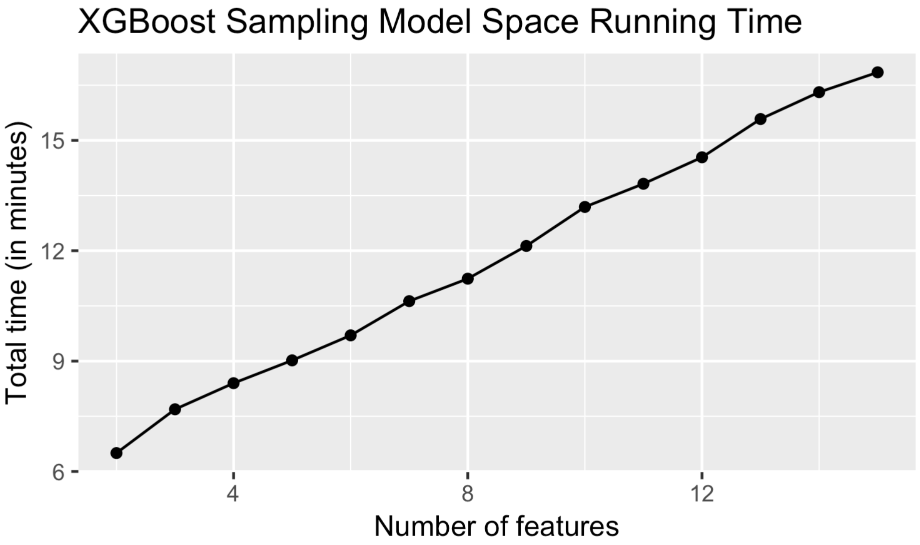


Figure A.1: Line plot showing the correlation between the number of features and the total time spent to sample the model space, using the learning algorithm XGBoost. It can be easily seen that there is a positive and steady increase in the computational cost as the number of features increase.

Following, we perform similar experiments using the Random Forests learning algorithm. Table A.2 shows that the computational cost when using the Random Forests learning algorithm is lower than XGBoost.

Table A.2: Running time spent to sample the model space using Random Forests learning algorithm and `VAS30` label. Each row contains information about the maximum number of features, the total time spent to sample 10 000 models with this length, and the third column is the average time per 1 000 models.

# of features	Total time (in minutes)	Avg. time per 1 000 samplings (in minutes)
1	0.25	N/A
2	8.59	0.85
3	9.45	0.94
4	9.5	0.95
5	9.56	0.95
6	9.61	0.96
7	9.66	0.96
8	9.71	0.97
9	9.8	0.98
10	9.85	0.98
11	9.89	0.98
12	9.87	0.98
13	9.9	0.99
14	9.93	0.99
15	9.97	0.99
Total	135.55	N/A

Figure A.2 shows similar plot using Random Forests. As can be seen, the increase in the computational cost also presents a steady growth. Nonetheless, we observe that the growth is small yet constant. The relative increase from 8.59 minutes to 9.97 minutes represents an increase of 16.06%.

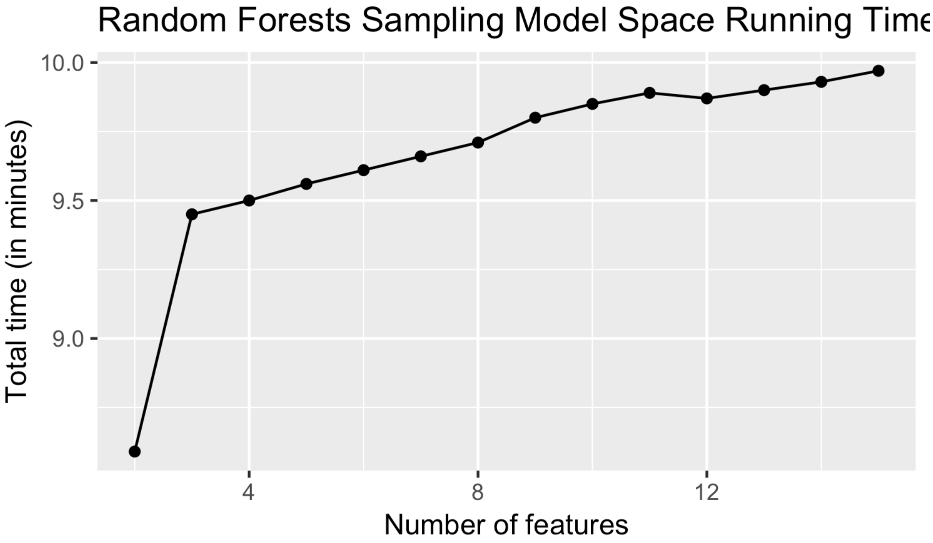


Figure A.2: Line plot showing the correlation between the number of features and the total time spent to sample the model space, using the learning algorithm Random Forests. There is a positive and steady increase in the computational cost as the number of features increases.

Finally, the total time to sample the entire model space was 165.79 minutes with XGBoost and only 135.55 minutes with Random Forests. In order to sample the model space with a 15 features limit, using the Random Forests learning algorithm is 20% faster than using the XGBoost learning algorithm.

Appendix B

Extended Experiments Case Study for labels VAS 50 and GIC

In this appendix we will perform, for VAS 50 and GIC labels, similar experiments performed in Chapter 4.

Before turning to the ED-Ensemble approach, we calculated the pair-wise correlation between GIC, VAS 30, and VAS 50. We observed that VAS 30 and VAS 50 are highly correlated, reaching a correlation value as high as 0.85. On the other hand, GIC is not related to either VAS 30 or VAS 50, obtaining correlation values of 0.1 and 0.097, respectively. This difference means that ratings from patients and assessments from clinicians may disagree. Further, when a patient achieves an overall reduction of pain intensity by 30%, most of the time, it will also reach an overall reduction of 50%.

B.1 Baseline Models

As a baseline, we averaged AUC values by the all-in-one models and also carried out Tree-based Pipeline Optimization Tool[1] (TPOT). The first scenario represents the standard approach. The second scenario employs a tool that optimizes the machine learning pipeline using genetic programming. We set up the time limit for optimization as 24 hours, once this is the approximate amount of time in our worst scenario case to run our approach.

Table B.1 presents the average values of AUC obtained using the all-in-one approach with labels VAS 50 and GIC. It is worth mentioning the difficulty of predicting the GIC label. TPOT is a machine learning pipeline optimization tool that automati-

[1] http://epistasislab.github.io/tpot/

cally selects the learning algorithm. The average AUC obtained using VAS 50 label was 0.598 by stacking up multiple estimators: Multinomial Naive Bayes, Gaussian Naive Bayes, and k-Nearest Neighbors Classifier. Finally, using the GIC label resulted in an average AUC of 0.568 through the stacking up of the following estimators: Stochastic Gradient Descent (SGD) and XGBoost.

Table B.1: Baselines, sorted by label, with average AUC values obtained by the all-in-one approach and TPOT to optimize the machine learning pipeline (time limit of 24 hours).

	XGBoost	Random Forests		TPOT
Label	AUC	AUC	Mean	AUC
VAS 50	0.634	0.597	0.615	0.598
GIC	0.564	0.575	0.569	0.568

B.2 Predicting the Evolution of Pain Relief using VAS50 and GIC labels

As the first step of our approach, we generate the model space $\mathcal{H}'$ by repeatedly sampling random subsets of features followed by filtering out only those models that meet the minimum performance criteria. The XGBoost VAS 50 model space comprises 1 408 models (0.94% of the models perform better than the all-in-one model), and 11 829 (7.89% of the models perform better than the all-in-one model) for Random Forests. Regarding GIC, 18 575 models (12.38% of the models perform better than the all-in-one models) are selected for XGBoost and 10 035 models for Random Forests (6.69% of the models perform better than the all-in-one model).

Figure B.1 shows XGBoost and Random Forests model spaces for labels VAS 50 and GIC. Each point corresponds to a model, and the size of the point indicates the variance of the validation error.

The next step is to perform clustering in the model space. The VAS 50 model space is clustered using three different criteria: using the predictions performed by each model, using the indexes of the features within each model, and using the explanatory factors associated with each model. Specifically for VAS 50 in conjunction with the XGBoost learning model, the silhouette score when clustering using feature criterion is 0.017 for hierarchical clustering and -0.021 for DBScan. For the probability criterion the values of 0.089 and -0.288 respectively are obtained with hierarchical clustering and DBScan. Finally, considering the SHAP criterion, good silhouetted values of 0.8115

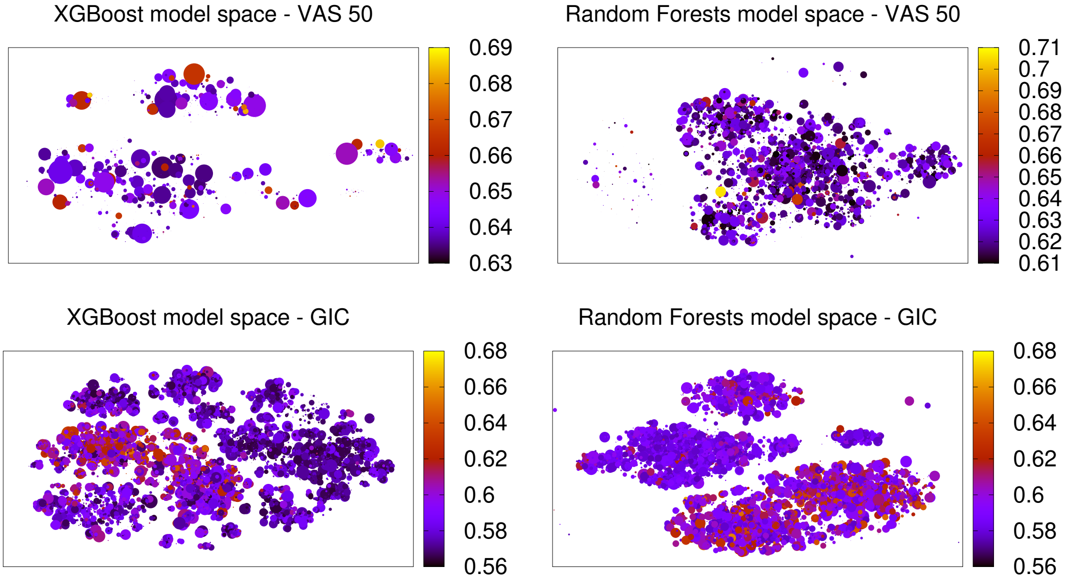


Figure B.1: (Color online) T-SNE visualization [van der Maaten, 2009] of the sampled model space $\mathcal{H}'$. Each point represents a model $\mathbf{x}'$. Models are placed according to the probabilities of significant pain relief assigned to patients. Models that assign similar probabilities to the same patients are placed next to each other in the space. The color indicates the average (cross-validation) AUC value, and smaller points indicate that the corresponding model has a smaller variance.

and 0.95 are obtained, again showing the cohesion and separation obtained when we use explanatory factors criterion. When using the Random Forests model, the silhouette values follow the same pattern. For the features criterion, we get 0.0055 and -0.0076. For the probability criterion 0.0377 and -0.3741, lastly the explanatory factor criterion we get 0.7839 and 0.8895. Always the first value refers to the hierarchical clustering algorithm and the second to the DBScan clustering algorithm.

For GIC with XGBoost model, the silhouette score when clustering with feature criterion is 0.0055 for hierarchical clustering and -0.0175 for DBScan. For the probability criterion, the values of 0.2311 and -0.4190 respectively are obtained with hierarchical clustering and DBScan. Finally, considering the SHAP criterion, silhouette values of 0.3762 and 0.0064 are obtained. When we use the Random Forests model, the silhouette values follow the same pattern. For the features criterion, we get 0.01779 and 0.-0041. For the probability criterion 0.1592 and 0.2454, lastly the explanatory factor criterion we get 0.4027 and 0.2167.

Following, we inspected the prototype models within each cluster in the XGBoost and Random Forests model spaces. In this case, the XGBoost prototypes comprise 91 features, of which 76 are unique. For Random Forests, the total number of features within the corresponding models is 155, from which 123 are unique and have occurred in only one model. Again, this is a strong indication that each prototype representing a cluster is diverse when grouped by explanatory factors, a critical strategy for building effective ensembles models.

For the GIC label, the XGBoost prototypes comprise 13 features divided into two prototypes only. Interestingly, for Random Forests, 19 prototypes were generated, resulting in 203 features used, from which 143 were unique. Again, our ensemble strategy shows to employ diverse information while building the final model.

Table B.2 shows AUC values under different ensemble configurations using VAS 50. When using the XGBoost learning algorithm, the explanations criterion yielded a relative gain of 12.41% over the best local model and 21.45% over the all-in-one approach. An identical gain for both clustering algorithms. Using the DBScan clustering algorithm in conjunction with the Random Forests learning algorithm, the gains obtained were 11.26% over the best local model and 32.33% (the largest gain) over the all-in-one approach. The explanations criterion alone yielded positive gains only, regardless of the clustering and learning algorithm used. Further, it also led to the largest gain with the combination of the DBScan clustering algorithm and Random Forests learning algorithm.

Table B.2: Ensemble performance for different clustering criteria and clustering algorithms using VAS 50 label. The baseline AUC value for the best local model for XGBoost was 0.68 and for Random Forests 0.71. Baseline AUC values for the all-in-one approach for XGBoost was 0.634 and for Random Forests 0.597.

		XGBoost			Random Forests		
Criterion	Clustering	AUC	Gain Best	Gain all-in-one	AUC	Gain Best	Gain all-in-one
Predictions	DBScan	0.69	0.73%	8.83%	0.66	-7.04%	10.55%
Predictions	Hierarchical	**0.79**	**15.33%**	**24.60%**	0.72	1.41%	20.60%
Feature values	DBScan	0.73	6.57%	15.14%	0.69	-2.82%	15.58%
Feature values	Hierarchical	0.77	12.41%	21.45%	**0.79**	**11.26%**	**32.33%**
Explanations	DBScan	0.77	12.41%	21.45%	**0.79**	**11.26%**	**32.33%**
Explanations	Hierarchical	0.77	12.41%	21.45%	0.78	9.86%	30.65%

Table B.3 shows the AUC values when using GIC label. It seems clear the advantage provided by our ensemble approach proposed. The explanations criterion led to the biggest gain in whole configurations tested. Up to 5.97% over the best local model when using XGBoost learning algorithm and 11.96% when using Random Forests learning algorithm. Following, 25.59% over the all-in-one approach when using XGBoost learning algorithm and 32.17% when using Random Forests learning algorithm. Finally, the proposed approach proved to the robust, once again was the only criterion that led to only positive gains regardless of the clustering and learning algorithms.

Table B.3: Ensemble performance for different clustering criteria and clustering algorithms using GIC label. The baseline AUC value for the best local model for XGBoost was 0.67 and for Random Forests 0.68. Baseline AUC values for the all-in-one approach for XGBoost was 0.564 and for Random Forests 0.575.

			XGBoost			Random Forests	
Criterion	Clustering	AUC	Gain Best	Gain all-in-one	AUC	Gain Best	Gain all-in-one
Predictions	DBScan	0.70	4.48%	24.11%	0.66	-2.97%	14.78%
Predictions	Hierarchical	0.70	4.48%	24.11%	0.72	5.88%	25.22%
Feature values	DBScan	0.65	-2.98%	15.24%	0.67	-1.47%	16.52%
Feature values	Hierarchical	0.68	1.49%	20.57%	0.69	1.47%	20.00%
Explanations	DBScan	0.68	1.49%	20.57%	**0.76**	**11.76%**	**32.17%**
Explanations	Hierarchical	**0.71**	**5.97%**	**25.89%**	0.74	8.82%	28.70%

The experiments using VAS 50 and GIC labels showed that the ED-Ensemble approach was consistently effective, with significant gains despite the learning algorithm and clustering algorithm used. Finally, we conclude with the statement that our proposal has proven robust with this new set of experiments.