Partial Responses: Unlocking Black Box AI Models

Walters, Bradley

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

July 2025

Table of Contents

Abstract

This thesis extends the partial response methodology to a range of non-linear black box models, such as Random Forests and Multi-Layer Perceptron neural networks. The outcome is a model-agnostic interpretability framework capable of maintaining the predictive power of the original black box models whilst offering full transparency into their decision-making processes. The proposed framework demonstrates competitive performance when evaluated against established interpretability techniques, both in terms of accuracy and explainability.

The framework enables the construction of intuitive univariate and bivariate visualisations derived from the partial response functions. These visual tools effectively communicate how individual variables, or pairs of variables, influence predictions across their entire respective domains. By providing a detailed, range-wide view of the variables, these plots support more comprehensive insights into model behaviour and facilitate informed decision-making.

In addition, preliminary experimentation is shown in the area of bootstrapping, wherein repeated resampling of the data was employed to assess the stability and reliability of the derived partial responses. This approach enhances the robustness of the interpretability outputs by incorporating measures of uncertainty, such as confidence intervals, thereby increasing user trust in the resulting explanations.

All experimental analyses are conducted using a combination of synthetic datasets, designed to evaluate the methodology under controlled and interpretable conditions, and real-world datasets, which served to examine the framework's efficacy in capturing complex, non-linear interactions among variables in noisy and heterogeneous environments. The use of both types of data ensures a comprehensive assessment of the method's generalisability and practical utility.

Declaration

I, Bradley Walters, confirm that the work presented in this thesis is my own. Furthermore, I confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Bradley Walters

Word count (excluding acknowledgements, appendices, and references): 23,999 words.

Acknowledgements

I would like to thank Liverpool John Moores University for funding my work through a Vice-Chancellor PhD Scholarship. I would also like to thank Sandra Ortega-Martorell, Ivan Olier-Caparroso, and Paulo Lisboa for their supervision throughout. Finally, I would like to thank my friends and family for their unwavering support.

List of Figures

List of Tables

1 Introduction

Machine learning serves as an umbrella term encompassing a broad range of algorithms and statistical methodologies capable of autonomously learning patterns from existing data and constructing predictive models without requiring the user to implement them using programming languages. Traditional machine learning approaches, such as logistic regression, are typically grounded in linear relationships between input variables and the target outcome. These models assign weights to each variable to quantify its influence on the response, thereby rendering the model inherently interpretable.

In contrast, more recent advancements in neural networks and deep learning have significantly enhanced predictive performance across a variety of complex tasks. However, these improvements in accuracy have come at the expense of interpretability, primarily due to the vast number of parameters and the non-linear interactions within such models. Consequently, models of this nature are commonly referred to as “black box” models.

Black box models are, by design, not transparent; they obscure the internal mechanisms by which predictions are generated, raising concerns around accountability and trust (Rudin, 2019). Despite their superior performance in many domains, users are often left without insight into the specific factors driving a model’s decisions (Liang et al., 2021). In contrast, while traditional machine learning models may underperform relative to black box counterparts, their behaviour is more readily understood, enabling users to identify weaknesses and avenues for improvement. Interpretative tools such as feature importance rankings and partial dependence plots are crucial in providing actionable insights, particularly in high-stakes contexts.

This issue is especially critical in medical applications, where predictions cannot be accepted on blind faith due to potentially severe consequences. In such settings, the inability to explain a model’s reasoning may lead to distrust and non-adoption by users (Ribeiro et al., 2016).

Various definitions of transparency and interpretability are found throughout the literature (e.g., (Richard et al., 2020)). However, this work adopts five core desiderata that serve as guiding principles for achieving robust model interpretability and explainability Alvarez-Melis & Jaakkola, 2018; Lisboa, Ortega-Martorell, Jayabalan, et al., 2020):

- Intelligibility: “Are the explanations immediate and understandable?”
- Faithfulness: “Are relevance scores indicative of “true” importance?”
- Stability: “How consistent are the explanations for neighbouring examples?”
- Parsimony: “Do the explanatory variables comprise a minimal set?”
- Consistency: “How robust are the explanations to perturbations in the data?”.

The methodologies developed in this study adhere to these five desiderata. Specifically, we have introduced a novel framework for explaining predictions made by black box models through an ANOVA decomposition. This approach enables us to quantify how much each univariate or bivariate component contributes to individual predictions, thereby offering a more granular and faithful explanation than traditional feature importance metrics alone. Furthermore, we employ Lasso regularisation to reduce the number of components obtained from the ANOVA decomposition, ensuring parsimony, and we present the results visually using partial response plots. These plots

illustrate how the contribution of each feature (or pair of features) varies across its full range, thereby enhancing intelligibility and interpretive value.

The rest of this chapter is organised as follows: The scope of the thesis is presented in section 1.1, with the publications resulted from this thesis presented in section 1.2. A background into previous literature in the field of interpretability is provided in section 1.3, followed by a general description of the novelty and gaps in the literature my work fills. Finally, an overview of the rest of the thesis chapters is described in section 1.4.

1.1 Scope of thesis

The scope of this thesis is the extension of the partial response methodology to a range of non-linear, structured models, collectively referred to under the umbrella term Partial Responses in Structured Models (PRiSM). This framework enables the development of interpretable versions of widely used machine learning models, leading to the Partial Response Support Vector Machine (prSVM), Partial Response Random Forest (prRF), and Partial Response Gradient Boosting Machine (prGBM).

A key aspect of this work involves comparing two methodological approaches within the PRiSM framework: the Lebesgue method, a novel variant, and the Dirac method, which has been previously established in the literature. The comparative analysis of these approaches provides insights into their effectiveness in capturing model behaviour.

To assess the validity and applicability of the proposed methods, the thesis employs both artificial and real-world datasets, demonstrating the capability of PRiSM models to enhance interpretability while maintaining predictive performance across different domains.

1.2 Publications

The following papers have been published in international conferences and journals:

- Walters B., Ortega-Martorell, S., Olier, I., & Lisboa, P. J. G. (2021). **The Partial Response SVM**. In ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2021-October. 10.14428/esann/2021.es2021-36
- Walters, B., Ortega-Martorell, S., Olier, I., & Lisboa, P. J. G. (2022). **Towards interpretable machine learning for clinical decision support**. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2022-July. https://doi.org/10.1109/IJCNN55064.2022.9892114
- Walters, B., Ortega-Martorell, S., Olier, I., & Lisboa, P. J. G. (2023). **How to Open a Black Box Classifier for Tabular Data**. Algorithms 2023, Vol. 16, Page 181, 16(4), 181. https://doi.org/10.3390/A16040181

1.3 Literature Review

Taxonomies in the literature typically classify interpretability methods based on several key dimensions. One common distinction is between local and global explanations, where local methods provide insights for individual predictions or small subsets of data, while global methods aim to explain the overall behaviour of the entire model. Another widely used categorisation differentiates between model-agnostic and model-specific approaches, depending on whether the interpretability technique can be applied to any black box model or is restricted to a particular architecture. A third major division

is between post-hoc and intrinsic explanations, where post-hoc methods generate explanations after a model has made its predictions (e.g., feature attribution techniques), whereas intrinsic methods embed interpretability directly into the model structure itself (Barredo Arrieta et al., 2020; Linardatos et al., 2021; Marcinkevičs & Vogt, 2020).

While these classifications provide a useful foundation, they are not without limitations. In particular, the assumption that methods must fall neatly into one category or another often fails to capture the nuanced relationships between different interpretability techniques. For example, a method can be both global and model-agnostic, offering explanations that describe overarching model behaviour while being applicable across different model architectures. Similarly, some techniques blur the line between post-hoc and intrinsic approaches by integrating explainability into training while still allowing for post-hoc analysis. As a result, rigid taxonomies may oversimplify the landscape of interpretability research and fail to accommodate hybrid methods that do not fit neatly into predefined categories.

Given these limitations, in this thesis, we propose a more flexible framework for organising interpretability methods. Instead of adhering to traditional classifications, we group existing literature into four broad categories based on their methodological approach and objectives: feature attribution, activation maximisation, metric learning and interpretation by design.

This categorisation, illustrated in **Figure 1**, provides a more comprehensive and adaptable framework for understanding interpretability methods, accommodating both established techniques and emerging approaches. By focusing on methodological principles rather than rigid distinctions, we aim to offer a more intuitive and practical perspective on the diverse landscape of explainable ML.

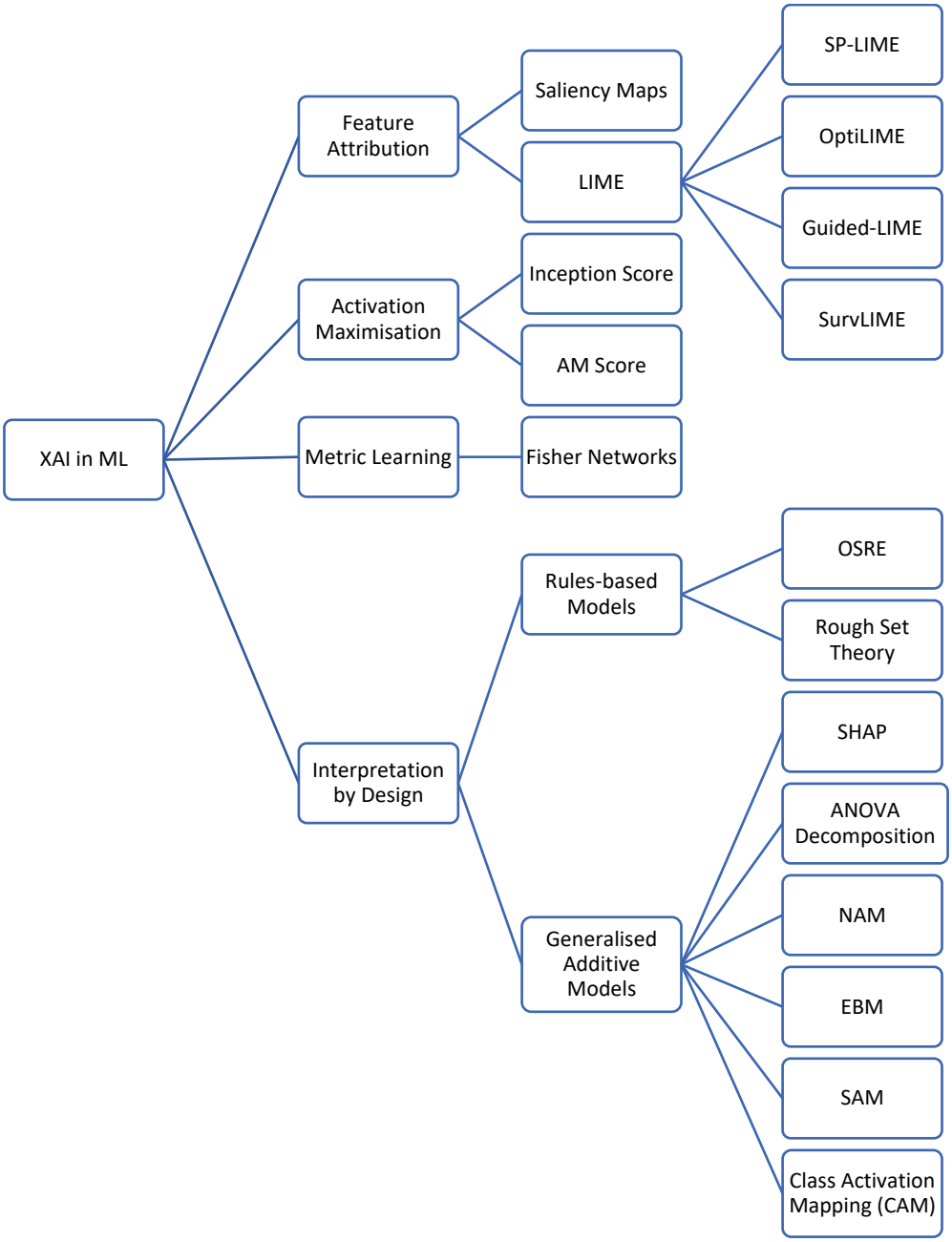


Figure 1. Taxonomy for explainability in Machine Learning.

Feature attribution refers to the process of assigning a small number of numerical or semantic features to a classification outcome, thereby identifying which components of a model contribute most significantly to its predictions. One well-known approach in this domain is the use of saliency maps, which highlight sparse components of the original model that exert the greatest influence on its decision-making process (Simonyan et al., 2014). By visualising these influential features, saliency maps provide insight into the inner workings of machine learning models, facilitating interpretability and transparency.

A notable methodology for feature attribution was introduced by Ribeiro et al. (2016) in the form of Local Interpretable Model-agnostic Explanations (LIME). This approach is designed to generate local explanations for any classification model while maintaining fidelity to the original decision boundaries. LIME functions by perturbing input data and observing changes in the model's predictions, ultimately

constructing an interpretable approximation of the classifier's behaviour within a localised decision region.

To extend LIME's applicability beyond individual predictions, Submodular Pick LIME (SP-LIME) was proposed as a mechanism for selecting a representative subset of instances whose explanations collectively provide a global understanding of the model's performance (Ribeiro et al., 2016). This approach is particularly useful in identifying both the strengths and weaknesses of a classifier. By analysing the explanations generated by SP-LIME, users can discern not only the factors contributing to a model's correct predictions but also the erroneous reasoning underlying its incorrect classifications. Consequently, SP-LIME serves as a crucial tool for mitigating overreliance on machine learning models by exposing cases where the classifier may be making unreliable or misleading decisions.

Ribeiro et al. (2016) demonstrated the utility of LIME in the context of text classification by applying it to a support vector machine (SVM) trained to distinguish between documents related to "Christianity" and "Atheism." Despite the model's high classification accuracy, LIME revealed that it disproportionately relied on arbitrary words such as "Posting," "Host," and "Re" to classify documents as related to "Atheism." This finding underscored a critical limitation of the classifier, namely its reliance on spurious correlations rather than meaningful textual features, thereby highlighting the importance of interpretability techniques in assessing model reliability.

Building on LIME's foundation, several variations have been introduced to address its limitations and extend its applicability. For instance, OptiLIME (Visani et al., 2020) aims to enhance the stability and robustness of LIME's explanations by optimising the perturbation process and feature selection. Similarly, Guided-LIME (Sangroya et al., 2020) refines the explanation generation process by incorporating domain-specific guidance to improve interpretability. Additionally, SurvLIME (Kovalev et al., 2020) extends the LIME framework to survival analysis models, providing explanations for predictions in contexts where time-to-event outcomes are of interest. These advancements collectively contribute to the broader effort of improving model interpretability, thereby enhancing trust and accountability in machine learning applications.

The second major category of interpretability techniques is activation maximisation, a method that can be applied in the context of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). In conventional neural network training, model parameters are optimised by iteratively adjusting the weights of neurons based on backpropagation and gradient descent. In contrast, activation maximisation takes a different approach by holding these weights constant and instead modifying the input data to maximise the activation of specific neurons (Erhan et al., 2009). This process provides insights into the internal representations learned by the model, helping to elucidate which features are most influential in the decision-making process of individual neurons or layers.

A significant development in the evaluation of activation maximisation techniques is the introduction of the AM Score by Zhou et al. (2017). This metric was proposed as an alternative to the widely used Inception Score (IS), which primarily assesses the diversity and realism of generated samples in generative models. While the Inception Score evaluates sample diversity based on the entropy of predicted class distributions, the AM Score offers an alternative measure by assessing how effectively a generated sample activates specific neurons. By doing so, it provides a complementary perspective on the quality of synthetic data generated by deep learning models.

Zhou et al. (2017) also addressed a fundamental limitation in activation maximisation, known as the overlaid-gradient problem. This issue arises when multiple target classes are simultaneously encouraged during the optimisation process, leading to conflicting gradient directions that may prevent convergence toward any specific class. As a result, the generated samples may fail to meaningfully represent any of the intended categories. To mitigate this problem, the authors introduced an explicit constraint that assigns a single target class to each generated sample, ensuring

that the optimisation process remains well-defined and produces interpretable outputs. This refinement enhances the robustness and effectiveness of activation maximisation techniques, making them more applicable in scenarios requiring detailed interpretability of deep learning models, such as feature visualisation and generative model evaluation.

Metric learning is a fundamental approach in machine learning that involves deriving a distance metric from a classifier to effectively capture and represent the underlying structure of a given dataset. This process is crucial for tasks such as clustering, similarity learning, and feature embedding, as it enables models to learn representations that reflect meaningful relationships between data points.

A notable example of metric learning is Fisher Networks, a framework introduced by Ruiz et al. (2013) to generate an interpretable representation of a dataset with labelled indicators in the form of a similarity network. This network leverages Fisher Information (FI), a statistical measure that quantifies the sensitivity of a probability function with respect to changes in a parameter, thereby providing insight into the significance of different features in the dataset. In addition to Fisher Information, the framework incorporates the Gaussian kernel of the Support Vector Machine classifier to dynamically adjust the degree of locality in the connections within the network. By modulating these connections, Fisher Networks enhance the interpretability of data representations, making it easier to identify underlying structures and relationships.

To validate their approach, Ruiz et al. (2013) generated graphical representations of the Fisher Network and applied Newman's algorithm for community detection to identify distinct clusters within datasets. This methodology was tested on benchmark datasets, including the Pima Indians Diabetes dataset and the Sonar dataset from the UCI Machine Learning Repository. The results demonstrated the framework's ability to reveal inherent data structures and improve interpretability by grouping data points based on learned similarity measures. The integration of metric learning with community detection techniques in this context highlights the potential of Fisher Networks as a powerful tool for exploratory data analysis and model interpretability.

The fourth category is interpretation by design, in which methods specify the contribution of each input feature to the model output. These fall into two subcategories, rules-based models and Generalised Additive Models (GAMs). In terms of rules-based models, (Rögnvaldsson et al., 2009) aimed to extract conjunctive rules for viral protease cleavage specificities using Orthogonal Search-Based Rule Extraction (OSRE). Conjunctive rules are formulated as a list of requirements that must all be true, in the same way as the logical operator AND. This methodology produced fewer rules than other previous rules-based approaches, such as rough set theory rules (Kontijevskis et al., 2007), while also having predictive power matching that of a state-of-the-art predictor, the linear Support Vector Machine (Rögnvaldsson et al., 2007).

In terms of GAMs, Van Belle et al. (2016) aimed to explain the Support Vector Machine (SVM) algorithm by defining a kernel as the addition of subkernels. They decomposed said kernels into univariate and bivariate terms with an extra term denoting all higher order terms. If this extra term is small enough to be ignored without a reduction in classification performance, then the decomposed SVM model can be visualised easily using a colour based nomogram. They found that this decomposed approximation produced comparable performance to the original SVM model, while being explainable. However, this methodology fails when the extra term is not small enough to be ignored without performance reduction and therefore the approximation is invalid, meaning that any conclusions drawn from the approximation cannot be assumed as correct.

A popular GAM method is Shapley Additive Explanations, known as SHAP (Lundberg & Lee, 2017). Based in game theory, SHAP utilises Shapley values to find the fairly distributed contribution of features. The method compares all permutations of model pairs that include and exclude a feature, which leads to a computationally expensive $\left(2^k\right) * k$ models for an entire dataset, where k is the number of features. The Shapley values can be approximated, to reduce computation time, by

replacing features individually with a set of representative data points. These Shapley values are used as coefficients in a linear model.

Recent GAM approaches include Neural Additive Models (NAMs) in which univariate responses are each modelled by a separate neural network (Agarwal et al., 2020), explainable Boosting Machines (EBMs) which includes both univariate and bivariate terms as part of the gradient boosting GAM (Nori et al., 2019), and Sparse Additive Models (SAMs) which utilises splines in their component terms (Ravikumar et al., 2009). These methods are restricted to being standalone interpretable models, rather than interpreting pre-trained black box models. NAMs do not perform any kind of feature selection, instead creating a model with univariate components for all available input variables. EBMs implement feature selection via ANOVA significance statistical tests, which is similar to how we calculate our univariate and bivariate responses. SAMs are based on Logistic Regression and utilise Lasso regularisation, again similarly to our methodology, to select sparser models.

Recently developed is the idea of using ANOVA decomposition, similarly to Van Belle et al. (2016). The premise is to take partial responses of univariate and bivariate contributions to a model's predictions and using them to create an approximation of this model that is interpretable. In the case of Paulo J.G. Lisboa et al. (2020) they used a multi-layer perceptron (MLP). The result was a method to explain the black box MLP but also keep comparable performance with other state-of-the-art black box models. The idea for the ANOVA decomposition stems from the ANOVA expansion, first suggested by Friedman (2001).

Class Activation Mapping (CAM) is a method of explaining image classifications using feature maps extracted from the final convolutional layer of a Convolutional Neural Network (CNN) (Zhou et al., 2016). The feature maps are weighted by a global average pooling layer and combined in the form of a heatmap highlighting which parts of the image activate the class the most. The method was shown to have comparable classification performance to other widely used models, while having better local interpretability when generating bounding boxes for specific parts of an image, generating smaller, more accurate bounding boxes than other methods for parts of images such as identifying animals, vehicles or clothing. In the variant Grad-CAM (Selvaraju et al., 2020), the weighting process is computed via the gradient of the output class with respect to the image channel the feature map pertains to. It also generalises CAM to be able to apply it to existing networks (the original CAM only works with CAM-compatible architectures).

This work introduces a novel methodology for computing and visualising partial responses in complex machine learning models. The main contributions are as follows:

- The thesis develops a mathematically grounded framework for partial responses, based on the explicit use of Dirac and Lebesgue measures, providing a principled foundation that extends beyond existing heuristic or model-specific approaches.
- The proposed methodology is designed to be model-agnostic, allowing for the consistent calculation of partial responses across a wide range of machine learning architectures, including SVMs, RFs and MLPs.
- The methodology is rigorously validated on controlled artificial datasets with known interaction structures. This validation provides empirical support for the accuracy and reliability of the extracted partial responses.
- The proposed framework demonstrates computational efficiency, particularly through the use of the Dirac measure, offering a fast and practical alternative to traditional marginalisation or sampling-based interpretability methods.

1.4 Overview of thesis

The thesis is structured into eight chapters, the remaining of which are organised as follows:

Chapter 2 describes the black box models that are used in the thesis, as well as other pertinent definitions.

Chapter 3 sets out the methodology of the partial responses, including relevant equations.

Chapter 4 extends the partial response methodology to the Support Vector Machine (SVM) algorithm.

Chapter 5 develops from chapter 4, showing that the methodology is model agnostic for tabular data.

Chapter 6 introduces the Lebesgue measure as an alternative to the previously used Dirac measure, and compares the two methods.

Chapter 7 gives a brief investigation into the benefit of bootstrapping the partial response method, giving a user more trust in the output.

Chapter 8 concludes the thesis with a summary of the main contributions, as well as a discussion on possible future directions for the research.

2 Black box algorithms

2.1 Introduction

While interpretable models, such as those developed using linear regression and decision trees, offer transparency in their decision-making processes, black box models are often characterised by their complexity and lack of direct interpretability. These models, despite their opaque nature, have often shown superior performance in various predictive tasks, particularly in high-dimensional and nonlinear data environments.

Black box models rely on intricate mathematical transformations and complex feature interactions to generate predictions, often making it challenging to understand their internal mechanisms. This lack of interpretability poses significant challenges in critical applications such as healthcare, finance, and law, where model decisions must be explainable and trustworthy. Nevertheless, black box models remain widely used due to their high accuracy, adaptability, and ability to uncover complex patterns within data.

This chapter focuses on four prominent black box models: Support Vector Machines (SVMs), Random Forests (RFs), Gradient Boosting Machines (GBMs), and Multi-Layer Perceptrons (MLPs). Each of these models has distinct methodological foundations and strengths, making them valuable for different types of machine learning tasks. Here we summarise the theoretical foundations of these models, their algorithmic formulations, and the ways they can be interpreted.

2.2 Support Vector Machine

Support Vector Machines (SVMs) were introduced at the 1992 Conference on Computational Learning (Boser et al., 1992) and have since become a fundamental technique in machine learning for classification tasks. SVMs construct decision boundaries known as hyperplanes that maximise the margin between different classes, thereby enhancing the model's ability to generalise to unseen data. This margin-maximisation property is particularly advantageous in high-dimensional spaces, where SVMs mitigate the risk of overfitting.

However, in real-world applications, datasets often exhibit overlapping class distributions, making it challenging to define a perfectly separating hyperplane. To address this issue, Cortes & Vapnik (1995) introduced the soft margin classifier in their seminal work, *Support-Vector Networks*. The soft margin classifier permits certain observations to reside on the incorrect side of the decision boundary, thereby accommodating datasets with non-linearly separable classes. The formulation of this optimisation problem includes the slack variable ϵ to account for misclassifications and an upper-bound parameter C, which controls the permissible total error.

The optimal decision function for the soft margin classifier takes the following mathematical form:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle, \qquad \textbf{Equation 1}$$

where S represents the set of support vectors, α_i denotes the estimated coefficients obtained during the optimisation process, and $\langle x, x_i \rangle$ corresponds to the inner (dot) product between feature vectors x_i and x'_i (Guenther & Schonlau, 2016). Given that the dot product quantifies similarity between data points, it can be replaced with a more general similarity function, $K(x, x_i)$, known as a kernel function. Kernel functions enable SVMs to project data into a higher-dimensional feature space, thereby facilitating linear separability of complex datasets (Cristianini & Schölkopf, 2002). Class label probabilities are calculated using the Platt approximation (Platt, 1999). Platt's method fits a parametric

sigmoid function to the SVM decision scores, which transforms them into probability estimates and is commonly used in practice.

2.2.1 Kernel functions

Kernel functions play a crucial role in extending the applicability of SVMs to non-linearly separable data. Several kernel functions exist, with three of the most fundamental being the linear, polynomial, and radial basis function (RBF) kernels. More complex kernels, such as sigmoid, ANOVA, and circular kernels, incorporate multiple hyperparameters, increasing computational complexity and making them less practical for many classification tasks.

For each kernel function, the hyperparameter C regulates the trade-off between maximising the margin and minimising classification errors. This is the only hyperparameter required for the linear kernel, which is mathematically formulated as follows:

$$K(x_i, x'_i) = \sum_{j=1}^{p} x_{ij} x_{i'j}. \qquad \textbf{Equation 2}$$

The linear kernel is particularly suitable for scenarios where both the number of observations and the number of variables are very large (Guenther & Schonlau, 2016). However, for datasets exhibiting nonlinear relationships, more sophisticated kernels are preferred.

The polynomial kernel extends the linear kernel by introducing the hyperparameter d, which determines the degree of the polynomial transformation:

$$K(x_i, x'_i) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d. \qquad \textbf{Equation 3}$$

Empirical findings suggest that polynomial degrees of 2 or 3 suffice for capturing complex patterns, whereas higher-degree polynomials tend to overfit and increase model complexity (Guenther & Schonlau, 2016).

The radial basis function (RBF) kernel, also known as the Gaussian kernel, introduces another hyperparameter, γ, which controls the spread of the kernel function:

$$K(x_i, x'_i) = exp\left(-\gamma \sum_{j=1}^{p} \left(x_{ij} x_{i'j} \right)^2 \right). \qquad \textbf{Equation 4}$$

The RBF kernel is widely regarded as one of the most effective kernel functions due to its capacity to model complex decision boundaries and adapt to diverse data distributions (Hsu et al., 2003). Its flexibility makes it a preferred choice over polynomial and linear kernels for many real-world applications. Furthermore, research has demonstrated that the performance of the RBF kernel is highly dependent on the careful tuning of C and γ, as these hyperparameters interact and influence the classification outcome (Mantovani et al., 2015).

2.2.2 Hyperparameter optimisation

Selecting appropriate hyperparameters is essential for achieving optimal performance in SVMs, particularly when using the RBF kernel. A common approach is grid search, where different values of

C and γ are systematically evaluated to identify the optimal configuration. This ensures that the model maintains both high accuracy and generalisability. More advanced optimisation methods, such as Bayesian optimisation or genetic algorithms, have also been proposed to refine the hyperparameter selection process, reducing computational costs while improving predictive performance (Bergstra & Bengio, 2012).

Of the black box models utilised in this work, the Support Vector Machine is the most sensitive to tuning. Using a Gaussian kernel, we tune the hyperparameter related to the width of the kernel, γ. This hyperparameter also controls the linearity of the model, with smaller values making the model more linear. This meant our selection of γ had to be non-linear, as there is no point in trying to interpret a linear black box model; at that point it would be wiser to use a glass box logistic regression model. This hyperparameter was tuned along with the Cost hyperparameter, C. The cost hyperparameter can cause the model to overfit, as having a large value can cause the SVM soft margin classifier to become hard margin, in which the boundaries become stricter. The cross-validation that was implemented for the hyperparameter tuning throughout this thesis should reduce the chances of overfitting for both hyperparameters.

2.2.3 Interpretability and practical considerations

Despite their strong theoretical foundation and high performance, SVMs face challenges related to interpretability and scalability. Unlike decision trees or logistic regression, SVMs do not provide explicit feature importance measures, making them less transparent. Additionally, the computational cost of training SVMs increases with the dataset size, particularly when employing non-linear kernels. To address these issues, techniques such as feature selection, dimensionality reduction (e.g., PCA), and approximations like the Nyström method have been proposed to improve efficiency (Williams & Seeger, 2000).

In summary, SVMs remain a powerful tool for classification, offering robust generalisation capabilities through margin maximisation and kernel-based transformations. The choice of kernel function plays a critical role in determining model performance, with the RBF kernel often emerging as the preferred option for complex datasets. However, careful hyperparameter tuning is essential to balance accuracy and generalisability. While SVMs are highly effective in many scenarios, their interpretability and computational efficiency must be considered when applying them to large-scale or real-time applications.

2.3 Random Forest

Random Forest (RF) is a powerful ensemble learning method introduced by Breiman (2001) to mitigate the overfitting problem inherent in traditional decision trees. It operates by constructing an ensemble of decision trees, each trained on a different subset of the data, with the aim of improving generalisation and reducing variance. The final prediction is obtained through majority voting in classification tasks or averaging in regression tasks. Class probabilities are computed as the average class frequencies across all trees.

2.3.1 Construction of a Random Forest

A Random Forest consists of multiple decision trees, where each tree is built using a bootstrap sample of the training data. Additionally, at each node of a tree, only a random subset of the features is considered for splitting, which helps to reduce correlation among individual trees and enhance the model's robustness.

For a given dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i represents the feature vector and y_i represents the corresponding label, the Random Forest algorithm can be described as follows:

1. **Bootstrap Sampling**: Generate B bootstrap samples D_b by randomly sampling instances with replacement from D.
2. **Tree Construction**: For each bootstrap sample, construct an unpruned decision tree:
    - At each node, randomly select m features from the total p available features, where typically $m = \sqrt{p}$ for classification tasks.
    - Determine the best split among the selected features using an impurity measure such as Gini impurity or entropy
    - Grow the tree until the stopping criterion is met (e.g., maximum depth or minimum number of samples per leaf).
3. **Aggregation**: For classification, the final prediction is obtained through majority voting:

$$\hat{y} = arg\max_{k} \sum_{b=1}^{B} 1(T_b(x) = k) \qquad \textbf{Equation 5}$$

where $T_b(x)$ is the prediction from the b-th tree, and k represents the class labels.

2.3.2 Hyperparameter Tuning

Two crucial hyperparameters in Random Forest models are the number of trees and the maximum depth of each tree. Increasing the number of trees typically improves performance by reducing variance. However, beyond a certain point, adding more trees results in diminishing returns while increasing computational cost. The maximum depth of each tree controls the complexity of the model. Shallow trees may underfit the data, failing to capture important patterns, whereas excessively deep trees may overfit by capturing noise rather than signal. The optimal depth is problem-dependent and should be tuned based on validation performance.

2.3.3 Interpretability and advantages of Random Forest

The algorithm provides insights into feature significance by analysing the decrease in impurity (e.g., Gini importance) when a feature is used for splitting. Gini importance measures how frequently a feature is used to split the data and how much it improves the classification performance at each split. This provides a global ranking of the features in the dataset. However, it can be biased to features with high-cardinality and can be unreliable for correlated features.

Random Forest is extremely robust to overfitting. By aggregating multiple trees, Random Forest mitigates the tendency of individual trees to overfit. The random feature selection strategy adopted by the methodology allows Random Forests to perform well even with high-dimensional data.

In summary, Random Forest is a versatile and powerful model that achieves strong predictive performance while maintaining interpretability through feature importance analysis. By carefully tuning hyperparameters such as the number of trees and maximum depth, the model can be optimised for a given dataset.

2.4 Gradient Boosting Machine

Gradient Boosting Machines (GBMs) were introduced by Friedman (2001) for function estimation. He approached the problem from the perspective of numerical optimisation in the function space rather than the parameter space. The work details how additive models can be trained iteratively by optimising a differentiable loss function, rather than just exponential loss (as in AdaBoost), using gradient descent techniques, generalising Boosting (Schapire, 1990; Freund & Schapire, 1997).

2.4.1 Construction of a Gradient Boosting Machine

The GBM algorithm is defined as follows:

1. Initialise the model with a constant function that minimises the loss:

$$F_0(x) = arg\min_c \sum_{i=1}^{N} L(y_i, c) \qquad \textbf{Equation 6}$$

where $L(y_i, c)$ represents the loss function (e.g. squared error, log loss), in which y_i are the target values and c is the constant that minimises the loss function over the entire dataset.

2. For each iteration m:
    a. Compute the negative gradient of the loss function, also known as the pseudo-residuals:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F_{m-1}} \qquad \textbf{Equation 7}$$

    b. Fit a weak learner (e.g. a decision tree) to the pseudo-residuals r_{im}, producing $h_m(x)$. The weak learner is typically shallow to avoid overfitting.
    c. Compute the optimal step size γ_m by solving:

$$\gamma_m = arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \qquad \textbf{Equation 8}$$

    d. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \qquad \textbf{Equation 9}$$

3. Repeat until convergence or stopping criteria are met.

The GBM outputs a log-odds score which is then passed through a logistic sigmoid function to get binary class probability.

2.4.2 Regularisation Techniques

To further mitigate the risk of overfitting in gradient boosting models, several regularisation strategies are commonly employed. These techniques serve to constrain model complexity and enhance generalisability, particularly when working with noisy or limited datasets.

One such method is shrinkage, often referred to as the learning rate. Shrinkage operates by scaling the contribution of each individual tree in the ensemble by a factor v, typically within the range $[0.01, 0.1]$.

By dampening the impact of each update, shrinkage helps to prevent the model from fitting too aggressively to anomalies in the training data, thereby reducing overfitting and improving predictive performance on unseen observations.

Another widely used approach involves imposing structural constraints on the trees themselves. This may include limiting the maximum depth of each tree or restricting the number of terminal (leaf) nodes. Similar to techniques applied in Random Forests, these constraints serve to simplify the decision rules and prevent overly complex models that memorise training examples rather than learning general patterns. Shallow trees, in particular, encourage the model to learn broader, more robust relationships across the feature space.

A third strategy is subsampling, wherein each tree is trained on a randomly selected subset of the available training data (Friedman, 2002). This method introduces a degree of randomness that helps to decorrelate the trees and reduce variance in the ensemble's predictions. By training on partially overlapping subsets of data, the model is less likely to overfit to particular patterns that may not generalise.

Collectively, these regularisation techniques play a pivotal role in enhancing the robustness of gradient boosting models, ensuring that they maintain high predictive accuracy without sacrificing interpretability or generalisation.

2.4.3 Interpretability

The interpretability of GBMs arises from their composition: an ensemble of decision trees trained in a sequential manner, with each tree aiming to correct the residual errors of its predecessors (Friedman, 2001).

While individual decision trees are inherently interpretable due to their hierarchical structure and clear decision paths (Breiman et al., 1984), the aggregation of numerous trees into an ensemble complicates direct interpretation. Nevertheless, the tree-based foundation of GBMs enables several post hoc techniques that provide meaningful insight into model behaviour. These include feature importance metrics, partial dependence plots (Friedman, 2001), and more sophisticated methods such as SHAP, which can attribute predictive power back to individual input features (Lundberg & Lee, 2017).

One of the most widely used interpretability tools for GBMs is feature importance scoring. These scores quantify the relative influence of each input variable on the model's predictions, aggregated across all trees in the ensemble (Chen & Guestrin, 2016). The gain metric measures the total improvement in the model's objective function brought about by splits on a particular feature. It reflects the feature's contribution to reducing prediction error. The frequency, or split count, refers to how often a feature is used to split nodes across all trees. A higher frequency suggests greater reliance on that feature, though it may not account for the quality of splits. The cover metric assesses the number of samples affected by splits involving the feature, offering insight into the breadth of its influence.

These importance metrics provide a global view of how the model uses different features, and are often visualised using bar plots to support interpretability in applied contexts (Ke et al., 2017; Prokhorenkova et al., 2018).

Despite their relative interpretability compared to deep learning models, GBMs have several limitations that should be acknowledged. First, the additive nature of boosting and the presence of interaction effects between features across multiple trees make it difficult to isolate the precise role

of individual variables, particularly in high-dimensional or non-linear settings. Second, when GBMs overfit to training data, they may assign undue importance to irrelevant or noisy features. This can mislead interpretations based on feature importance scores unless appropriate regularisation and validation are applied (Hastie et al., 2009). Third, while global feature importance scores offer a broad understanding of the model, they may fail to capture the nuances of individual predictions. For applications requiring transparency at the level of specific decisions, such as in clinical or legal settings, global metrics alone are often insufficient (Doshi-Velez & Kim, 2017). Finally, techniques like partial dependence plots assume independence between features, which may not hold in real-world datasets, potentially leading to misinterpretations of model behaviour (Molnar, 2022).

In light of these considerations, it is advisable to complement traditional importance metrics with model-agnostic or local interpretability methods, such as SHAP or Individual Conditional Expectation (ICE) plots, to ensure a more robust and nuanced understanding of GBM decision-making processes (Goldstein et al., 2015; Lundberg & Lee, 2017).

2.5 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) is a foundational model in the field of deep learning, forming the basic architecture for many modern neural networks. It is a class of feedforward artificial neural network (ANN) that consists of multiple layers of nodes, with each layer fully connected to the next. The core components of an MLP include input layers, hidden layers, output layers, activation functions, and a training algorithm, most famously backpropagation.

2.5.1 Construction of a network

An MLP consists of: An input layer that accepts inputted features $x \in \mathbb{R}^n$; one or more hidden layers where each neuron applies a linear transformation followed by a non-linear transformation; and an output layer that produces the final output, which in a classification task such as ours would be class probabilities.

Each layer performs:

$$a^{(l)} = \phi(W^{(l)}a^{(l-1)} + b^{(l)}) \qquad \textbf{Equation 10}$$

Where $a^{(l)}$ is the activation of layer l, $W^{(l)}$ and $b^{(l)}$ are the associated weights and biases associated with layer l, and ϕ is the activation function. Popular activation functions include the Rectified Linear Unit (ReLU), the hyperbolic tangent function (tanh) and the sigmoid function (Rumelhart et al., 1986).

The algorithm is trained using backpropagation by computing gradients of the loss function with respect to each weight. This involves: a forward pass in which outputs are produced as normal; loss computation using a differentiable loss function $L(y, \hat{y})$; a backward pass in which we update the weights by computing the gradient of the loss:

$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)}(a^{(l-1)})^T \qquad \textbf{Equation 11}$$

Where $\delta^{(l)}$ is the error signal at layer l (Rumelhart et al., 1986). Finally, we update the weights using gradient descent or an optimiser such as Adam. Class probabilities are obtained by passing the final layer's output through a sigmoid activation function, for binary classification.

2.5.2 Hyperparameter Tuning

There are many key hyperparameters that are tuneable when it comes to MLP models, ranging from numerical hyperparameters such as the number of layers or the learning rate, to categorical hyperparameters such as the activation function or the optimiser. We focus on tuning some of the numerical hyperparameters that affect the depth of the network and width of each layer, as well as the learning rate. The learning rate controls the step size when weights are updated by multiplying it by a value, typically in the range]0,1[. L^2 regularisation is applied to penalise the magnitude of the weights, which is controlled by the hyperparameter α. A high value of α can risk underfitting in the model, whereas a low value can risk overfitting. Increasing the number of layers or units per layer means the model can extract more information, and deeper networks are better suited for complex tasks. However, having too many layers or units can lead to overfitting and longer training times.

2.5.3 Interpretability

Interpreting MLP models remains a substantial challenge in the field of machine learning due to the distributed nature of their internal representations. Each hidden unit within an MLP typically captures only a partial aspect of the input signal, meaning that the model's final prediction results from a highly non-linear combination of numerous partial transformations. This distributed processing makes it difficult to attribute clear semantic meaning to the contribution of any single unit or feature. Moreover, the use of non-linear activation functions, such as ReLU or tanh, compounds this difficulty by

introducing complex, non-transparent interactions between inputs and outputs across multiple hidden layers. As a result, simple or intuitive relationships between input variables and predicted outcomes are often obscured, limiting the direct interpretability of MLPs.

Despite these challenges, several approaches have been developed to extract interpretable insights from MLPs. One class of techniques includes involves saliency maps, which use gradients to highlight the regions of the input that most influence the model's prediction (Simonyan et al., 2014). This method is particularly common in image-related tasks. For a more general, global perspective on model interpretability, feature importance techniques such as SHAP (Lundberg & Lee, 2017) have gained prominence, while methods like LIME (Ribeiro et al., 2016) offer insights into individual predictions at a local level. These model-agnostic methods estimate the influence of individual input features by leveraging game-theoretic principles to distribute credit among features or by approximating the local decision boundary of the model. Additionally, gradient-based methods, such as computing the derivative of the output with respect to input features, can offer insights into the sensitivity of the model's predictions to small changes in input values (Baehrens et al., 2010).

Overall, while MLPs are inherently more opaque than simpler models, the growing ecosystem of interpretability techniques allows users to derive meaningful explanations from their predictions, thus improving model transparency and enhancing trust in high-stakes applications.

2.6 Other relevant definitions

2.6.1 AUC

Area Under the Receiver Operating Characteristic (ROC) curve, also known as AUC, is a widely used metric for evaluating the performance of binary classifiers. First investigated by Bradley (1997) as a single scalar value summarizing the performance of classifiers across all possible threshold values, it was demonstrated to be particularly useful is situations in which class distributions are imbalanced. The study also compared AUC with other evaluation metrics and concluded that AUC provides a more comprehensive measure of a classifier's performance in various scenarios. The confidence intervals are calculated using the DeLong method, a non-parametric approach that is widely used in medical statistics and ROC analysis (DeLong et al., 1988).

2.6.2 K-fold Cross-validation

K-fold cross-validation is a widely used model evaluation technique that provides a robust estimate of a model's generalisation performance by systematically partitioning the available data into training and validation sets. In this method, the dataset is randomly divided into k equally (or nearly equally) sized folds. The model is trained k times, each time using $k-1$ folds for training and the remaining fold for validation. The performance metric is then averaged across all k trials to obtain a more stable and less biased estimate of model accuracy (Kohavi, 1995). This approach helps to mitigate issues associated with overfitting to a single train-test split and is especially beneficial when working with limited data.

K-fold cross-validation is commonly used in model selection, hyperparameter tuning, and comparative evaluation of learning algorithms (Hastie et al., 2009). Variants such as stratified k-fold cross-validation offer refinements for specific data scenarios, and should be implemented when care must be taken to maintain class balances.

2.6.3 Hosmer-Lemeshow Test

The Hosmer-Lemeshow test is a widely used statistical procedure for evaluating the goodness of fit of logistic regression models (Hosmer & Lemeshow, 1980; Hosmer et al., 2013). It provides a means of assessing how well the model's predicted probabilities correspond to the observed outcomes in the

data. Specifically, the test divides the dataset into a series of ordered subgroups based on the predicted probabilities of the outcome. Within each subgroup, the observed number of events (i.e., cases where the outcome of interest occurs) is compared with the expected number of events predicted by the model.

A chi-squared test statistic is then computed to quantify the discrepancies between observed and expected event rates across all subgroups. A high p-value suggests that there is no significant difference between the observed and expected frequencies, indicating that the model fits the data well. Conversely, a low p-value would suggest that the model does not adequately capture the structure of the data and that its predictive accuracy may be lacking. The Hosmer-Lemeshow test thus serves as a valuable diagnostic tool, particularly in applied research contexts where model calibration is crucial for reliable decision-making.

3 The Partial Responses

3.1 Methodology

3.1.1 Partial Responses

Based on the work done by Lisboa, Ortega-Martorell, & Olier (2020), we have extended the partial response methodology, including Lasso regularisation, to other non-linear machine learning methods (SVM, RF and GBM) to show that the methodology is model agnostic for tabular data. A detailed numerical example of the following methodology is provided in Appendix A.

The implementation of the partial response methodology is to decompose the logit probability of the positive class membership into its component functions of one or two variables, also known as an ANOVA decomposition. This is done by iteratively anchoring all but one or two variables at their median value, taking the logit, and subtracting any additional effects present, such as the effect of all variables set to their median and in the case of bivariate components, their corresponding univariate components. These components are additive contributions to the original class membership probability. The ANOVA decomposition is defined by:

$$\begin{aligned} logit\big(P(C|x)\big) &\equiv \log\left(\frac{P(C|x)}{1-P(C|x)}\right) \\ &= \varphi_0 + \sum_i \varphi_i(x_i) \\ &+ \sum_{i\neq j} \varphi_{ij}(x_i, x_j) + \cdots + \sum_{i_1\neq\cdots\neq i_P} \varphi_{\,i_1\ldots\,i_P}(x_{i_1}, \ldots, x_{i_P}) \end{aligned} \qquad \textbf{Equation 12}$$

Where the general form of the terms is a recursive function of subsets of the covariates $\{x_{i_1}, \ldots, x_{i_P}\}$ up to the dimensionality of the data, P. We define the partial responses using a given measure $\mu(x)$ as:

$$\varphi_0 = \int_{[x]^P} logit\big(P(C|x)\big)\, d\mu(x) \qquad \textbf{Equation 13}$$

$$\varphi_S(x_s) = \int_{[x]^{P-|S|}} logit\big(P(C|x)\big)\, d\mu(x_{-S}) - \sum_{T\subset S} \varphi_T(x_T) \qquad \textbf{Equation 14}$$

where $S \in R^S$ represents a subset of variables with dimensionality $|S| \leq d$. The terms x_s and x_{-s} denote, respectively, the subspace spanned by $S: |S| = n$ and its complement $-S: |-S| = d - n$. In order to obtain interpretable functions $\varphi_S(x_s)$ we will restrict the cardinality $|S|$ of the set of variables of interest for each component function to be $n = 1, 2$.

There are two natural choices of measure. The first one I implemented is known as the Dirac measure (Zhang et al., 2010), which forms the anchored ANOVA decomposition:

$$d\mu(x) = \delta(x - x_c)dx \qquad \textbf{Equation 15}$$

for an arbitrary point x_c that is called an anchor point. This measure implies that the integrals above amount to evaluating the function at the anchor point for all inputs in the argument of the integral, leaving the remaining variables free. Our choice of anchor point is the median of the data, but it can

be any value of central tendency in theory. The data is first median centred and scaled so that the median of each covariate is 0.

The second choice of measure is known as the Lebesgue measure. While the Dirac measure in **Equation 15** forms the anchored ANOVA decomposition, the Lebesgue measure can be seen as an unanchored ANOVA decomposition.

$$d\mu(x) = \rho(x)dx \qquad \textbf{Equation 16}$$

where $\rho(x)$ is the density function of the variables in the argument of the integral. This measure calculates the weighted mean of the integrand.

In the case of the Lebesgue measure the integrals in **equations (13)-(14)** are calculated empirically using the training data, with sample size N observations

$$\hat{F}_S(x_s) = \frac{1}{N}\sum_{k=1}^{N} logit\left(P\left(C\middle|x_S, x_{-S}^k\right)\right) \qquad \textbf{Equation 17}$$

where the variables with dimensions x_s take any desired values but those in the complement set with dimension x_{-S}^k are fixed at their actual values in the training set $k = 1..N$ (Friedman, 2001). This corresponds to shifting all onto the coordinate(s) x_s so that in **Equation 17** every data point for a variable has the same value of this input dimension while retaining the original values for all other variables.

The orthogonalised partial responses $\varphi_S(x_s)$ follow by using **Equation 14**.

$$\hat{\varphi}_0 = \frac{1}{N}\sum_{k=1}^{N} logit\left(P\left(C\middle|x^k\right)\right) \qquad \textbf{Equation 18}$$

$$\hat{\varphi}_i(x_i) = \hat{F}_i(x_i) - \hat{\varphi}_0 \qquad \textbf{Equation 19}$$

$$\hat{\varphi}_{ij}(x_i, x_j) = \hat{F}_{ij}(x_i, x_j) - \hat{\varphi}_i(x_i) - \hat{\varphi}_j(x_j) - \hat{\varphi}_0 \qquad \textbf{Equation 20}$$

The Lebesgue measure is more rigorous than the Dirac measure, as it calculates partial responses over all combinations of values within the dataset. The Dirac measure, however, takes a 1-dimensional view of the data. This will be discussed further in Chapter 6, when the Lebesgue measure is formally utilised, and Chapter 8, in which we summarise the benefits and drawbacks of each measure.

The partial responses for each data observation become the input to a logistic regression Lasso, in which model coefficients are aggressively pruned via L1-regularisation for feature selection. We use cross-validation to tune the hyperparameter λ using the AUC metric. We can then choose our preferred model that is within 1 standard error of the best performing model, in order to comprise a minimal set. The aim of this method is to extract the predictive power of a black box model into a sparse white-box model. We can also plot the contributions of the univariate and bivariate components for both global and local interpretability purposes.

Given an observation described by an input vector, the model prediction is calculated as follows:

1. Take the value of each input variable and find its contribution to the $logit(P(C|x))$.
2. Include the contributions from all univariate terms and also bivariate terms.
3. Add these contributions and also the β_0 from the Lasso. This addition forms the risk index, which is the full $logit(P(C|x))$.
4. Feed the logit into a sigmoid function. The result is the predicted posterior probability of class membership, in this case, the probability of mortality, $P(C|x)$.

3.1.2 Summary of method

Algorithm Partial Response Models $pr(BB, D)$

Input: Set D of training examples; Predictions $P(C|x)$ from a pre-trained black box model BB.

1. ANOVA decomposition: Apply the equations mentioned above for the $logit\big(P(C|x)\big)$. The Dirac measure leads to an anchored decomposition referenced to the choice of anchor point, whereas the Lebesgue measure has no anchor point and therefore the resulting partial responses are unanchored.

2. Model selection with the Lasso: Input the set of univariate and bivariate partial responses $\varphi_i(x_i)$ and $\varphi_{ij}\big(x_i, x_j\big)$ calculated over the training data set D as new inputs to a logistic regression Lasso, with the target variable remaining the same class memberships from the original data. L1-regularisation is utilised by the Lasso in order to perform variable selection of the partial responses. The Lasso will also output a linear coefficient for each partial response, $\beta_i \; and \; \beta_{ij}$ as well as an intercept β_0, generally resulting in good calibration.

Output $\boldsymbol{prBB(BB, D)}$: This is the output of the Lasso in step 2 which has the form of a GAM shown in the equation above truncated to the selected subset of functions $\varphi_i(x_i)$ and $\varphi_{ij}\big(x_i, x_j\big)$:

$$logit\big(prBB(C|x)\big) \approx \varphi(0) + \sum_i \beta_i \varphi_i(x_i) + \sum_{i \neq j} \beta_{ij} \varphi_{ij}\big(x_i, x_j\big) \qquad \textbf{Equation 21}$$

Each partial response comprises a non-linear function of its arguments. Consequently, the model prediction equals the sum of all partial responses plus the intercept, weighted by the linear coefficients from step 2, followed by application of the sigmoid function, which inverts the $logit\big(P(C|x)\big)$. Algorithmically, the methodology can be expressed as:

ALGORITHM: PARTIAL RESPONSE MODELS $pr(BB, D)$

Input: *Set D of training examples size (k, i); Predictions $P(C|x)$ from a pre-trained black box model BB*
Output: *Interpretable Logistic Regression surrogate $pr(BB, D)$*
Initialisation: *Standardise all features x_i in D to have median 0*
Dirac Measure
    Null Term: $D = 0$
        Get $P(C|x^k)$ *from BB*
        $\varphi_0 = logit\left(P(C|x^k)\right)$
    Univariate Terms: For *each feature x_i in D*
        $x_i = x_i, x_{-i} = 0$
        Get $P(C|x^k)$ *from BB*
        Compute $logit\left(P(C|x^k)\right)$
        $\varphi_i(x_i) = logit\left(P(C|x^k)\right) - \varphi_0$
    End For
    Bivariate Terms: For *each pair of features (x_i, x_j) in D*
        $x_i = x_i, x_j = x_j, x_{-i,-j} = 0$
        Get $P(C|x^k)$ *from BB*
        Compute $logit\left(P(C|x^k)\right)$
        $\varphi_{ij}(x_i, x_j) = logit\left(P(C|x^k)\right) - \varphi_i(x_i) - \varphi_j(x_j) - \varphi_0$
    End For

Lebesgue Measure
    Null Term: $D = D$
        Get $P(C|x^k)$ *from BB*
        $\varphi_0 = \frac{1}{N}\sum_{k=1}^{N} logit\left(P(C|x^k)\right)$
    Univariate Terms: For *each observation k in x_i*
        $x_i = x_i^k, x_{-i} = x_{-i}$
        Get $P(C|x^k)$ *from BB*
        Compute $logit\left(P(C|x^k)\right)$
        $\varphi_i(x_i) = \frac{1}{N}\sum_{k=1}^{N} logit\left(P(C|x^k)\right) - \varphi_0$
    End For
    Bivariate Terms: For *each pair of observations (k, l) in (x_i, x_j)*
        $x_i = x_i^k, x_j = x_j^l, x_{-i,-j} = x_{-i,-j}$
        Get $P(C|x^k)$ *from BB*
        Compute $logit\left(P(C|x^k)\right)$
        $\varphi_{ij}(x_i, x_j) = \frac{1}{N}\sum_{k=1,l=1}^{N} logit\left(P(C|x^k)\right) - \varphi_i(x_i) - \varphi_j(x_j) - \varphi_0$
    End For
Logistic Regression Lasso
    Input the new partial response dataset into a Logistic Regression Lasso model
    Using cross-validation, tune the hyperparameter λ using the AUC metric
    We can select a preferred model within 1 standard error of the best, to comprise a minimal set

A complete numerical example of both measures is provided in Appendix A.

3.2 Optimising the black box

To achieve optimal performance from our partial response models, it is essential to first optimise the underlying black box models through rigorous hyperparameter tuning. This process ensures that the black box models, which serve as the foundation for generating partial responses, are performing at their highest potential before interpretability is applied.

The hyperparameter tuning is conducted via an exhaustive grid search procedure, wherein a predefined range of values for two key hyperparameters is systematically evaluated for each black box model. This approach enables a comprehensive exploration of the hyperparameter space, increasing the likelihood of identifying the combination that yields the best predictive performance.

In order to enhance the robustness and generalisability of the selected hyperparameters, we employ stratified k-fold cross-validation. This method divides the dataset into k equally sized folds while preserving the original class distribution within each fold, thereby mitigating potential biases introduced by imbalanced datasets. By iteratively training and validating across all folds, we obtain a reliable estimate of model performance and ensure that the chosen hyperparameters are not overfitted to a specific subset of the data.

This systematic approach to model optimisation provides a strong foundation for the subsequent generation and interpretation of partial response functions.

3.3 Optimising the Lasso

Optimising the Logistic Regression Lasso can help to reduce the number of covariates in the model, which fits with our desiderata of parsimony. To do this, cross-validation is utilised to tune the hyperparameter λ, which governs coefficient shrinkage in the model. As the value of λ increases, the penalty applied to the shrinkage increases, meaning coefficients tend towards zero and some are even removed from the model altogether, resulting in a sparser model.

The value of λ is selected based on the best performing AUC metric. We also investigate the model that is 1 standard error away from the best performing model, in the direction of a higher λ and therefore a sparser model. We do this as is maintains a significant performance while aiming to further shrink the coefficients and achieve a sparser model.

4 Extension of partial responses to SVM

4.1 Introduction

To check the validity of the partial responses we obtain, we start with two-variable artificial datasets. This is for two reasons: This work only goes as far as bivariate interaction terms, to reduce complexity and the fact that higher order interactions don't add anything significant (Van Belle et al., 2016); The logit probability for two variables is an identity. We then extend to three variables to test the methodology, and finally introduce different levels of randomness to each variable. Datasets with more than two variables will include interactions higher than bivariate in the logit probability. To account for this, we calculate a "catch-all" residual term which contains all higher order interactions.

4.2 Artificial Datasets

4.2.1 Data Description

The two circles artificial dataset consists of 1000 observations and 3 variables, two of which are relevant to the outcome variable. The third variable is random noise that is irrelevant to the classes. A visualisation of the two relevant variables is shown in **Figure 2 (a)**, in which the two classes are separated as two concentric circles.

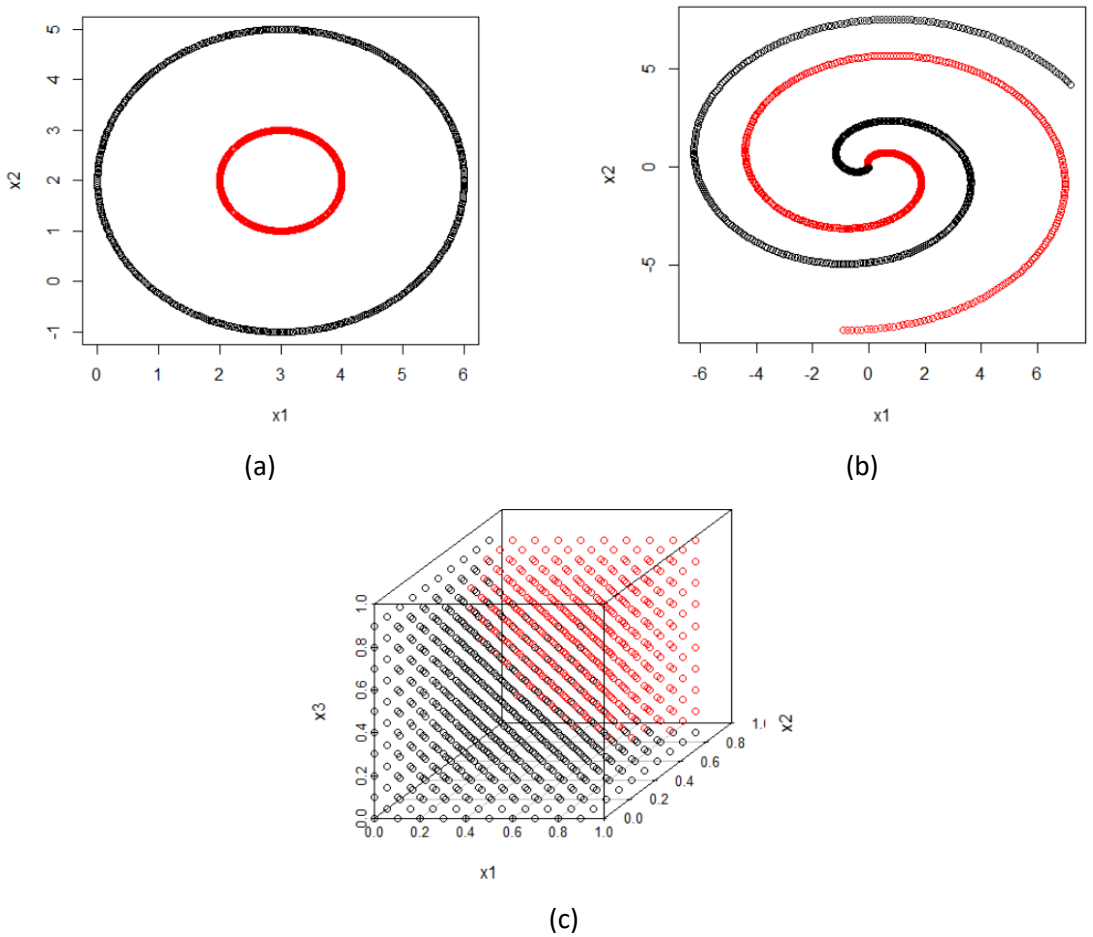


Figure 2. Visualisation of the relevant variables in (a) the two circles artificial dataset, (b) the Swiss roll artificial dataset, (c) the three-way interaction artificial dataset

The Swiss roll artificial dataset consists of 1000 observations and 3 variables, two of which are relevant to the outcome variable. The third variable is random noise that is irrelevant to the classes. A

visualisation of the two relevant variables is shown in **Figure 2 (b)**, in which the two classes are separated as two spirals.

The three-way interaction artificial dataset consists of 1000 observations and 3 variables, all of which are relevant to the outcome variable, with prevalence of 42%. A visualisation of the dataset and class boundary is shown in **Figure 2 (c)**, in which the two classes are separated by the interaction between the three variables.

The fuzzy three-way interaction artificial dataset is similar to the three-way interaction artificial dataset described previously, with a small amount of randomness added from a normal distribution multiplied by a constant. We tested a variety of different constant values, namely 0.1, 0.2 and 0.4, which are visualised in **Figure 3**.

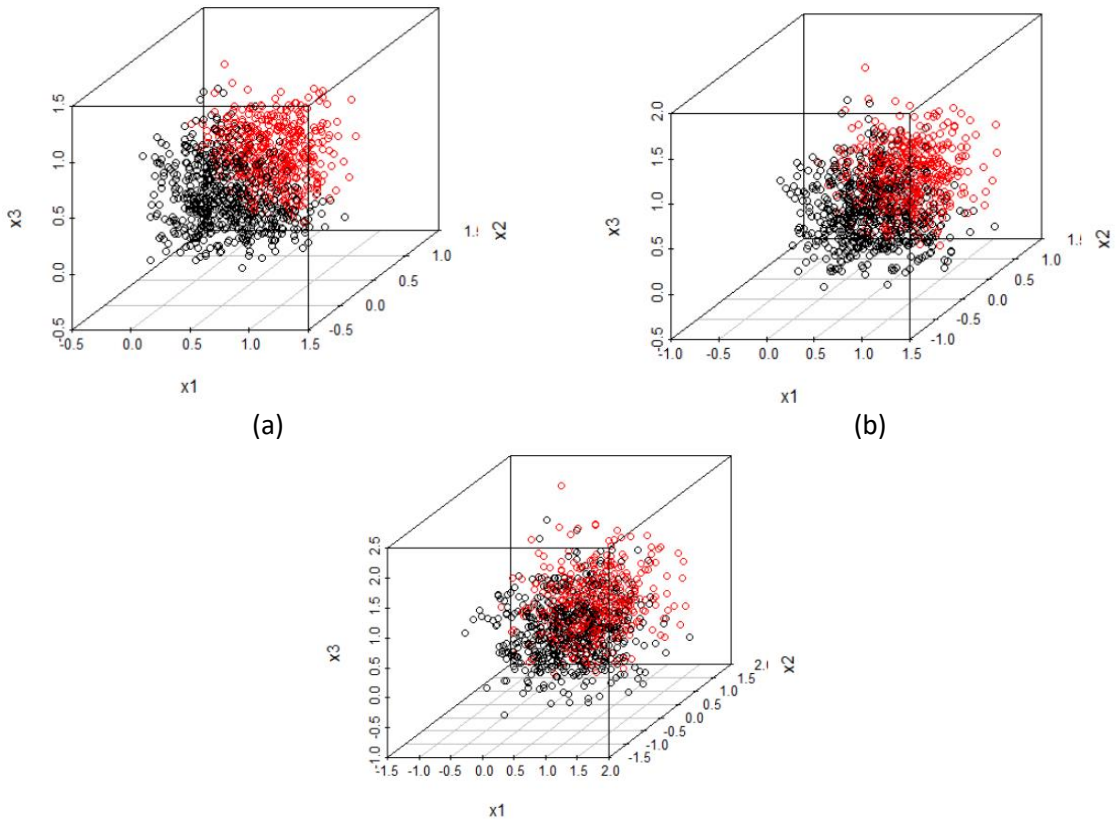


Figure 3. Visualisation of the fuzzy three-way interaction artificial datasets with different amounts of randomness added from a normal distribution, multiplied by (a) 0.1, (b) 0.2, (c) 0.4

4.2.2 Classification Performance

In **Table 1**, we compare the classification performance of our partial response model prSVM against its black box counterpart. For the Three-way Interaction dataset, we test with and without the residual term to check for relevance. We also compare the number of variables in the SVM to the number of partial responses in our prSVM models, after LASSO regularisation.

Our partial response models perform as well as the black box models, while being fully interpretable. We also see that our models contain, in most cases, equal or fewer components compared to the black box, adhering to our desiderata around parsimony.

Table 1. Performance comparison of the partial response SVM (prSVM) against the black box SVM for the two circles, Swiss roll and three-way interaction artificial datasets

Dataset	Model	Number of variables/ Partial Responses	Test AUC (CI)
Two Circles	SVM	3	1 (1, 1)
	prSVM	2	1 (1, 1)
Swiss Roll	SVM	3	0.9988 (0.9973, 1)
	prSVM	3	1 (1, 1)
Three-way Interaction	SVM	3	0.9998 (0.9993, 1)
	prSVM	7 – Residual inc.	0.9997 (0.9993, 1)
	prSVM	6	0.9978 (0.9955, 1)

Table 2. Performance comparison of the prSVM against the black box SVM for the fuzzy three-way interaction artificial datasets

Dataset	Model	Number of variables/ Partial Responses	Test AUC (CI)
Three-way Interaction (0.1)	SVM	3	0.9407 (0.9172, 0.9643)
	prSVM	5 – Residual inc.	0.9488 (0.9274, 0.9702)
	prSVM	5	0.9475 (0.9259, 0.9691)
Three-way Interaction (0.2)	SVM	3	0.8675 (0.8281, 0.9070)
	prSVM*	3	0.8749 (0.8368, 0.9130)
Three-way Interaction (0.4)	SVM	3	0.7517 (0.6968, 0.8067)
	prSVM*	3	0.7513 (0.6965, 0.8061)

* best model with or without residual included in the process.

4.2.3 Visualisation and Interpretability

Figure 4 shows heatmaps of the mean AUC values of each corresponding pair of hyperparameters, for the training split of each dataset. These helped to inform our choice of hyperparameters for our black box models. Our hyperparameter choices were based on which set had the highest mean AUC whilst also maintaining some non-linearity. If our hyperparameters were indicative of a linear model, then there would be no need for a black box model, and thus no need to interpret one.

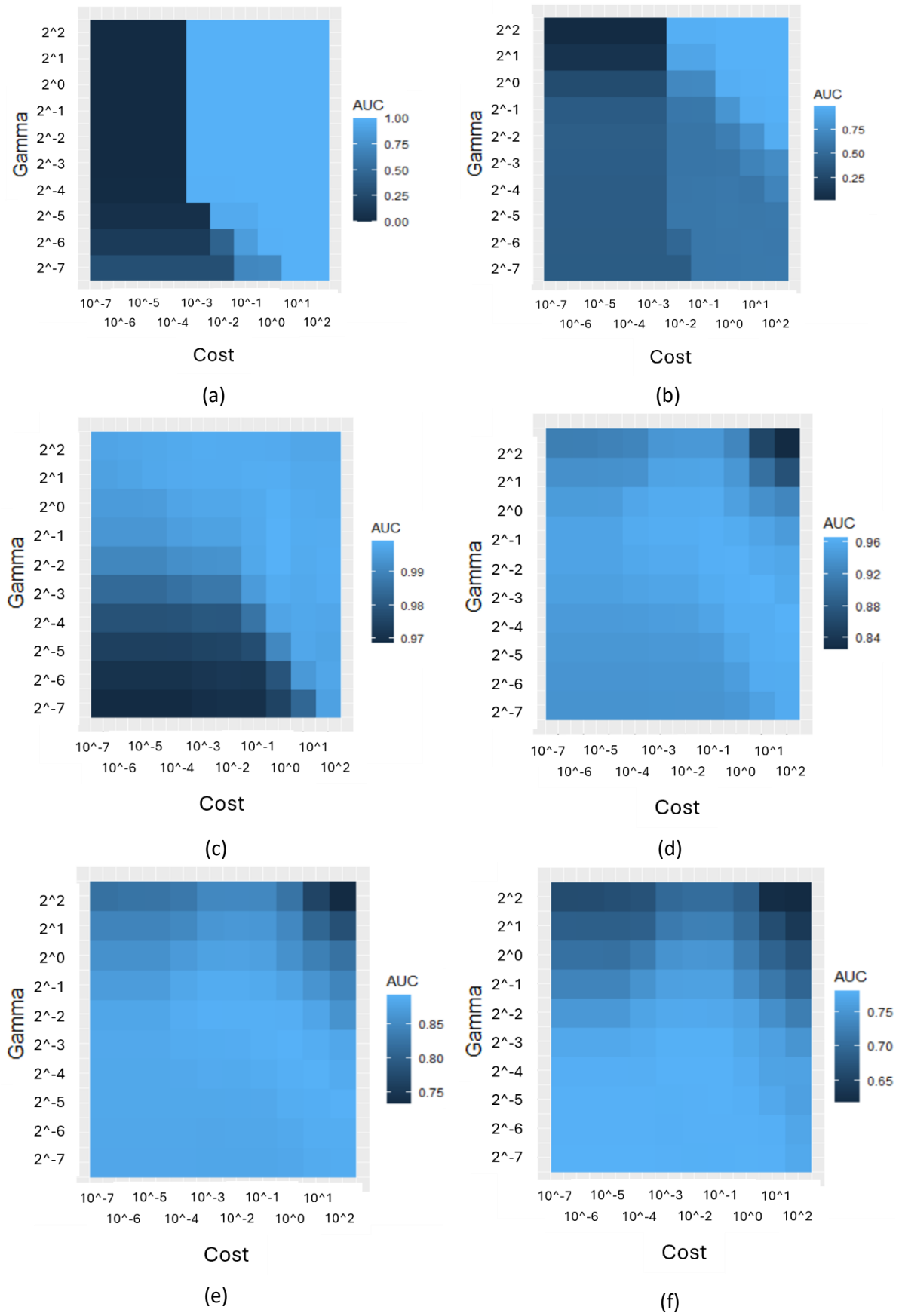
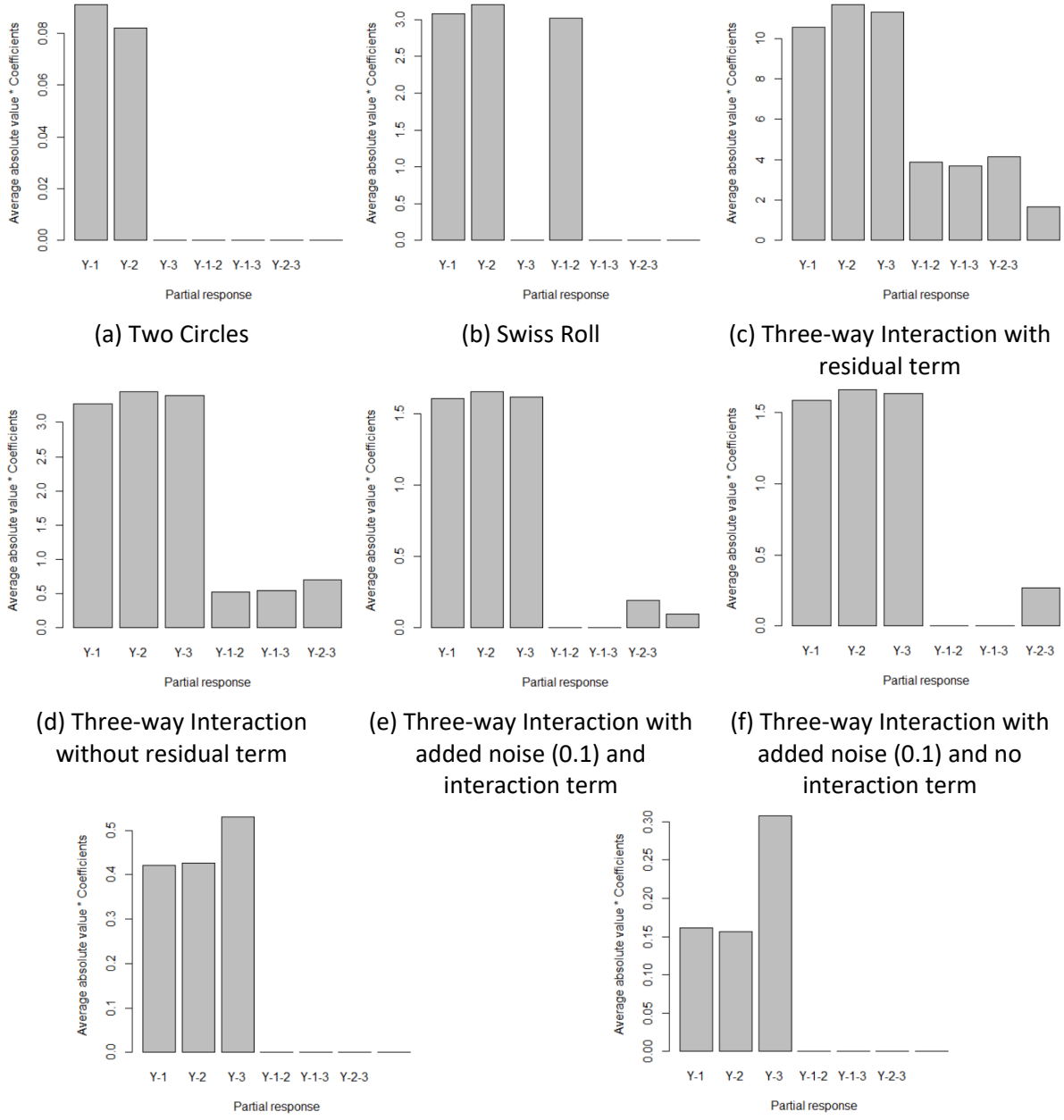


Figure 4. Hyperparameter tuning heatmaps for the SVM hyperparameters C and γ, for the following artificial datasets: (a) two circles, (b) swiss roll, (c) three-way interaction, (d) fuzzy three-way interaction (0.1), (e) fuzzy three-way interaction (0.2), (f) fuzzy three-way interaction (0.4)

Next, we look at the variable importance for each partial response model within each dataset. We include the three-way interaction with and without the residual term to observe its effect on the partial responses. We calculate the variable importance as the average absolute value of the partial response multiplied by its Logistic Regression coefficient after LASSO regularisation.



(a) Two Circles (b) Swiss Roll (c) Three-way Interaction with residual term

(d) Three-way Interaction without residual term (e) Three-way Interaction with added noise (0.1) and interaction term (f) Three-way Interaction with added noise (0.1) and no interaction term

(g) Three-way Interaction with added noise (0.2) (h) Three-way Interaction with added noise (0.4)

Figure 5. Variable importance for the best performing prSVM models for each artificial dataset; (a) Two Circles, (b) Swiss Roll, (c) Three-way Interaction with residual term, (d) Three-way Interaction without residual term, (e) Three-way Interaction with added noise (0.1) and interaction term, (f) Three-way Interaction with added noise (0.1) and no interaction term, (g) Three-way Interaction with added noise (0.2) and (h) Three-way Interaction with added noise (0.4)

Plots (a) and (b) show that the Lasso regularisation has selected the only partial responses that were relevant to the output class when the data was generated. In (c) and (d) we compare the effect of the residual term, noting how the weight placed on the partial responses is reduced when the residual

term is removed. Comparing (e) and (f), we see that the residual term has no major effect on the partial responses, and that the inclusion of random noise has seen the removal of some bivariate interaction terms. Increasing the random noise further, as seen in (g) and (h), and the interaction terms disappear completely.

Finally, we check the calibration curves for each model. The curves are a plot of predicted probability versus actual probability, so we can check how well the model is predicting across the whole dataset. Better calibration is visible as a perfect diagonal line in the form $y = x$, shown by the light grey "Ideal" line in the plots of **Figure 6** below. Also plotted along the x-axis is a histogram of the model probabilities to show the spread.

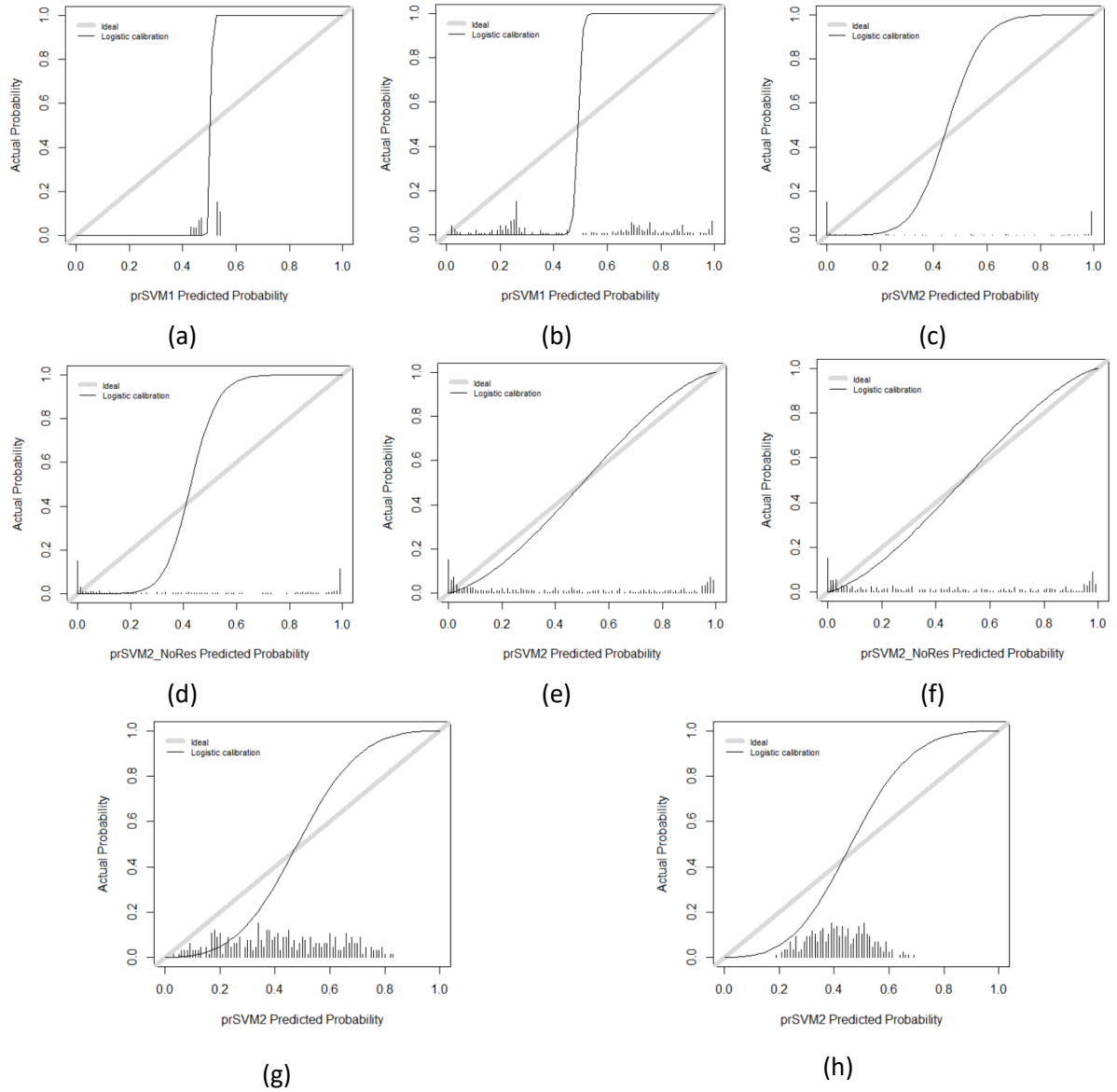


Figure 6. Calibration curves of predicted probability versus actual probability for the best performing prSVM models for each artificial dataset; (a) Two Circles, (b) Swiss Roll, (c) Three-way Interaction with residual term, (d) Three-way Interaction without residual term, (e) Three-way Interaction with added noise (0.1) and interaction term, (f) Three-way Interaction with added noise (0.1) and no interaction term, (g) Three-way Interaction with added noise (0.2) and (h) Three-way Interaction with added noise (0.4)

The results obtained for the Two Circles, Swiss Roll, and Three-way Interaction artificial datasets demonstrate near-perfect classification performance, with AUC values equal to or extremely close to 1. Such results indicate that the models are capable of distinguishing between classes with near-complete accuracy. As a consequence, the corresponding calibration curves offer limited additional insight, given that perfect or near-perfect classification implies optimal probability estimates across all thresholds. Nevertheless, for the sake of completeness and methodological transparency, the calibration curves for these datasets are presented in **Figure 6** plots (a)-(d).

Plots (e) and (f) show almost perfect calibration, showing the models can handle some amount of added noise. And again, the residual term seems to have no effect on the calibration. Plot (g) shows that this particular model underpredicts for lower actual probabilities, and overpredicts for higher actual probabilities. The histogram adds further weight to this, as we can see that there are very few predictions above 0.8. Finally, plot (h) shows similarly to (g); underpredicting for lower actual probabilities and overpredicting for higher actual probabilities. The histogram shows that the model is not predicting any low or high probabilities at all, being spread mainly between 0.2 and 0.6. The calibration change for the Three-way Interaction dataset as the amount of noise increases is interesting. Both ends of "no noise" and "most noise" are poorly calibrated, whereas a little bit of noise is capable of being handled by the SVM model.

4.3 Real world datasets

4.3.1 Data Description

The prSVM performance is compared with that of the original SVM with a Gaussian kernel and the SVM nomogram model (Van Belle et al., 2016) using the same two real-world datasets.

The Pima Diabetes dataset (Ripley, 2007; Smith et al., 1988) comprises measurements recorded from 768 women, who were at least 21 years old, of Pima Indian heritage, and tested for diabetes using World Health Organization criteria. One of the variables, "Blood Serum" Insulin, has significant amounts of missing data. This variable was removed along with all entries with missing values of "Plasma Glucose Concentration" in a tolerance test, "Diastolic Blood Pressure" (BP), "Triceps Skin Fold Thickness" (TSF) or "Body Mass Index" (BMI), resulting in a reduced data set with n = 532. In line with common practice, a 70% stratified subset was selected for training (n = 372), and the remaining 30% were used for testing (n = 160). The additional variables available are "Age", "Number of Pregnancies", and "Diabetes Pedigree Function" (DPF), a measure of family history of diabetes. A binary target variable indicated diabetes status, with a positive class prevalence of 33% indicating that a person has diabetes.

The Statlog German Credit Card dataset (Kelly et al., n.d.) (n = 1000) contains information on people taking credit by a bank. The data contains 24 variables, however we use the same 6 covariates as Van Belle et al. (2016) for comparability, specifically: the status of applicant's account in the bank, credit duration in months, credit purpose, credit amount, current employment duration and current residence duration. The binary outcome classes each person as either a good or bad credit risk according to their attributes. The data is stratified into a 70% training set (n = 700) and a 30% out of sample test set (n = 300) with a positive class prevalence of 30%, the positive class in this case indicating a person being a bad credit risk.

4.3.2 Classification Performance

The two models for real-world data were optimised by 4-fold cross validation on the training data. For both models the hyperparameter σ was tested in the range $[2^{-7}, 2^{2}]$ with the values 2^{-2} and 2^{-4} selected for the Pima and German Credit Card datasets respectively.

The relative performance compared to the original SVM and the values quoted by Van Belle et al. (2016) are listed in **Table 3**. Note the smaller number of variables selected, for a similar classification performance. This is important because smaller univariate and bivariate effects are more difficult to infer accurately and can be unstable.

The implementation of SVM in R by Karatzoglou et al. (2006) involves a cost parameter that penalises misclassifications. The effect on calibration of both the hyperparameters was considered and the calibration for the Pima dataset is shown in **Figure 7**. This is consistent with the hypothesis that modelling the SVM with component functions renders the prSVM a more accurate probabilistic model than resorting to the Platt approximation. The form of the partial responses provides valuable insights about the validity of the model predictions, as it can be verified by expert end-users.

Table 3. Results comparison between the original SVM, the prSVM and the SVM Approximation by Van Belle et al. (2016). The number of components is the number of covariates for the SVM and the number of partial responses for the rest.

Dataset	Model	AUC [CI]	Number of components	Hosmer-Lemeshow statistic (p-value)
PIMA Diabetes	SVM	0.801 [0.730,0.873]	7	26.5 (0.000867)
	prSVM	0.806 [0.737,0.876]	7	15.7 (0.0465)
	SVM Approx.	0.780	28	
German Credit Card	SVM	0.757 [0.696,0.818]	6	21.0 (0.00719)
	prSVM	0.754 [0.696,0.813]	18	11.2 (0.190)
	SVM Approx.	0.760	21	

4.3.3 Visualisation and interpretability

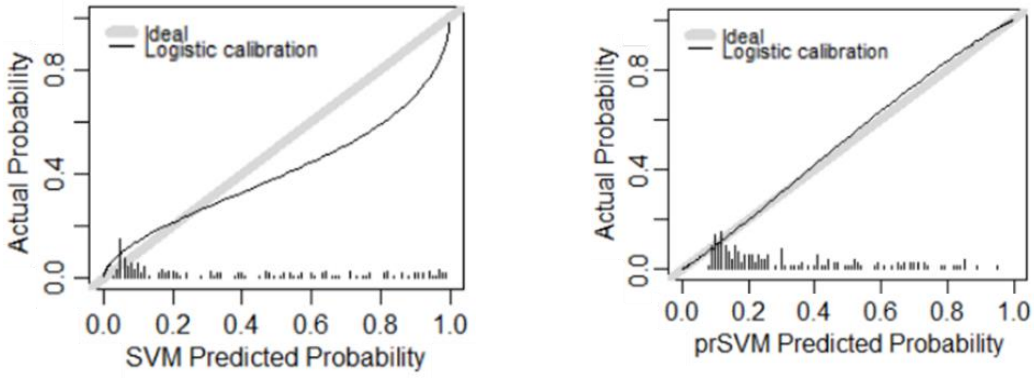


Figure 7. Calibration curves for the Pima diabetes data set, with hyperparameters $\gamma = 2^{-2}$ and $Cost = 10^{-2}$, showing an improvement for the prSVM compared with the original SVM with a Gaussian kernel.

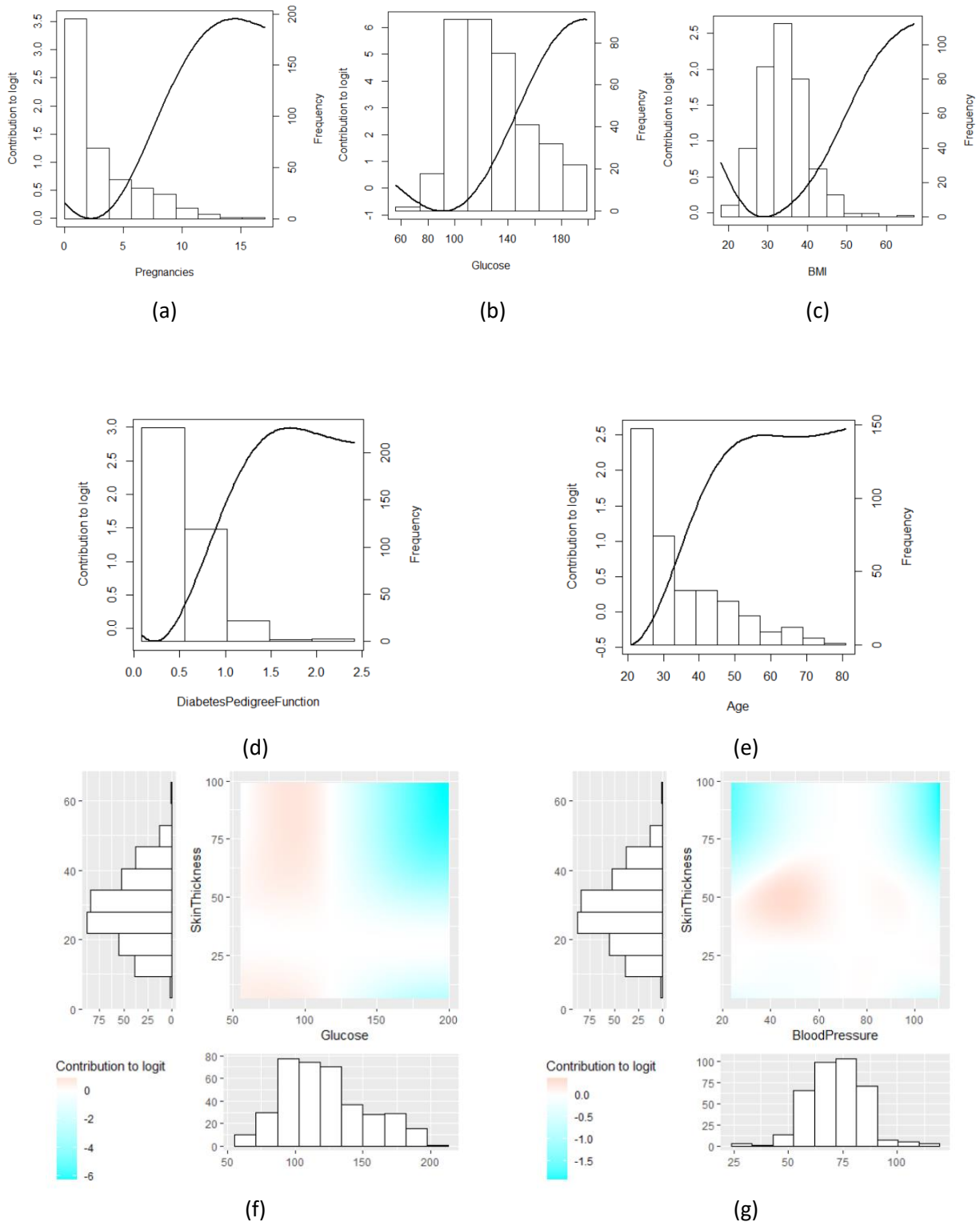


Figure 8. Partial responses in the nomogram for the Pima diabetes data set. The partial responses relevant to the prSVM for the Pima dataset, as selected by the Lasso regularisation, are plotted. For the univariate terms, the x-axis shows the range of each variable and its contribution to the logit probability is denoted on the y-axis. For the bivariate terms, the contribution to the logit is a colour mapping. It can also be plotted as a 3-dimensional surface with the z-axis being the contribution, as shown later in section 5.5.2.

4.3.4 Discussion

An examination of the results presented in **Table 3** demonstrates that our partial response Support Vector Machine (prSVM) models successfully retain the predictive accuracy of the original black box SVM classifiers, whilst offering full interpretability. Notably, for the Pima Indians Diabetes dataset, the prSVM model comprises the same number of components as the original SVM, namely a reduced subset of univariate and bivariate terms, highlighting that not all input features are necessary for accurate classification. This selective inclusion underscores the utility of our method in identifying and retaining only the most relevant variables, thereby enhancing model simplicity without sacrificing performance.

In contrast, the prSVM model trained on the German Credit Card dataset includes a greater number of components compared to its black box counterpart. However, this number remains lower than that of the SVM approximation method described by Van Belle et al. (2016)Van Belle et al. (2016), which exhaustively includes all univariate and bivariate terms. The relatively compact structure of our model not only enhances interpretability but also improves computational efficiency. Furthermore, the Hosmer-Lemeshow goodness-of-fit test yields a p-value that indicates a satisfactory fit to the data at the conventional 5% level of significance. In contrast, for the Pima prSVM model, the test result suggests that the model only fits the data adequately at the more stringent 1% level. This implies that the Pima model provides a comparatively poorer fit to its dataset than the prSVM model trained on the German Credit Card dataset.

Turning to the calibration plots illustrated in **Figure 7**, we observe that the original SVM exhibits a pattern of under-prediction in the lower probability range and significant over-prediction in the higher probability range. Conversely, the prSVM demonstrates notably improved calibration characteristics, with only a minor under-prediction observed in the upper probability range. This suggests that the prSVM not only preserves the predictive capabilities of the SVM but also offers enhanced reliability in terms of probability estimates, an important consideration in high-stakes decision-making environments.

In reviewing the univariate partial response plots in **Figures 8 (a)-(e)**, a consistent pattern emerges across most variables: as the value of a given predictor increases, its contribution to the likelihood of a positive classification for diabetes also increases. This trend is particularly notable in variables such as Body Mass Index (BMI) and Glucose. For these variables, we also observe a modest rise in positive class contribution at lower values. This behaviour may reflect instances where the variable values fall below the lower threshold of what is considered physiologically normal, suggesting a non-linear association between the predictor and the model's output. Alternatively, these early increases might be attributable to artifacts inherent in the underlying black box model.

Specifically, since the model in use employs a Support Vector Machine with a Radial Basis Function kernel, it is plausible that the observed patterns are influenced by the circular nature of the decision boundaries that the RBF kernel induces. The curvature observed in the partial response functions may thus reflect the geometry of the SVM decision space rather than a strictly interpretable causal or correlative relationship. This possibility highlights the need for caution when interpreting partial response functions, especially in regions with sparse data.

To enhance interpretability and provide users with additional context, we overlay histograms of the variable distributions beneath the partial response plots. These histograms serve as a visual cue for the density of observations across the range of each predictor. In regions where the frequency is high,

the corresponding partial responses can be interpreted with greater confidence due to the abundance of supporting data. Conversely, in areas with low observation density, the partial responses may be less reliable, and their interpretive value should be treated with appropriate scepticism. This integrative approach aids in grounding the interpretability of partial responses within the empirical characteristics of the dataset.

Numerous studies analysing the Pima dataset have identified glucose as the dominant predictor, with higher values strongly associated with increased diabetes prevalence (Smith et al., 1988; Zou et al., 2018). The steep, monotonic rise observed in the glucose partial response is therefore clinically expected and supports the validity of the extracted partial responses. Analyses of the Pima dataset have repeatedly shown higher mean BMI values among diabetic cases compared to non-diabetic controls (Smith et al., 1988). The partial responses for Pregnancies and Age also align with known clinical associations. Increasing age is a well-established risk factor for type 2 diabetes, reflecting cumulative metabolic stress and declining insulin sensitivity (ADA, 2014). Similarly, the number of pregnancies is a relevant predictor in the Pima population, as repeated pregnancies may be associated with gestational diabetes and long-term metabolic changes, a relationship previously reported in analyses of this dataset (Smith et al., 1988). The Diabetes Pedigree Function captures hereditary risk and family history, and its increasing partial response contribution at higher values is consistent with genetic susceptibility to diabetes observed in the Pima Indian population (Knowler et al., 1981).

Looking at the bivariate partial response plots in **Figures 8 (f)-(g)** reveals that the strongest contributions to the predicted probability of a positive classification predominantly occur in regions where both interacting variables exhibit extreme values. These regions correspond to areas of low data density. This pattern suggests that the model assigns higher predictive influence to combinations of variables that are less frequently observed, which may indicate a tendency to overemphasise outlying or less typical cases within the dataset.

In contrast, the contributions within regions of higher data density, where the values of both variables are closer to the central tendency, tend to be comparatively weaker. This observation could imply that the model treats more common or average observations with a more conservative weighting, potentially reflecting a bias towards robustness in regions with substantial empirical support.

Notably, an exception to this general trend is observed in **Figure 8 (g)**, where a slight positive contribution to the prediction is evident even around the central values of Skin Thickness. This suggests a nuanced relationship in this particular interaction, where average values of Skin Thickness, when considered in conjunction with its paired variable of Blood Pressure, may still meaningfully enhance the probability of a positive classification.

Such patterns underscore the importance of contextualising bivariate partial responses not only in terms of the magnitude of their contributions but also with respect to the distribution of the data. This enables a more informed interpretation of the interactions and supports more reliable conclusions about the behaviour of the underlying model.

4.4 Conclusion

We demonstrate that it is indeed feasible to provide meaningful explanations for the predictions generated by a black box Support Vector Machine model, both in synthetic and real-world datasets. The resulting interpretable models, derived through the application of partial responses, successfully preserve the predictive performance of the original SVM classifiers. Importantly, they achieve this while offering full transparency and interpretability.

To further enhance parsimony, we employ Logistic Regression with Lasso regularisation, which systematically eliminates redundant or non-informative partial responses. This results in a sparse, minimal model that retains only the most salient univariate and bivariate effects, thereby aligning with one of our primary desiderata, model simplicity without compromising accuracy.

The partial response visualisations offer an intuitive and accessible means of interpreting the influence of individual variables. These plots clearly illustrate how each predictor contributes to the model's output across the entire range of its values, thereby enabling a comprehensive understanding of variable behaviour. Such visual transparency is particularly valuable in applied contexts, where stakeholders may require interpretable explanations for predictive outcomes in order to support informed decision-making.

5 Model Agnostic Partial Response Models

5.1 Introduction

The partial response methodology is not inherently restricted to Support Vector Machine classifiers as the underlying black box model. Rather, it can be regarded as a flexible and generalisable framework that is applicable to a broad range of predictive models. In this chapter, we substantiate this claim by extending the methodology to encompass other widely-used black box models, specifically Random Forests, Gradient Boosting Machines, and Multi-Layer Perceptron Neural Networks.

Through these extensions, we demonstrate that the partial response framework retains its interpretive strengths and predictive performance across diverse modelling paradigms. This empirical evidence confirms the model-agnostic nature of the methodology, one of its key advantages. By enabling consistent interpretability regardless of the underlying algorithm, the partial response approach offers a unified and transparent means of demystifying complex predictive models, thereby broadening its applicability in both research and applied machine learning contexts.

5.2 Extension to other non-linear models

The extension of the methodology to other black box machine learning models is fairly straightforward, only needing to account for how each black box's function handles the outputted predictions from the models. Additionally, if the original black box model is an MLP, it is possible to construct a GANN/SENN to replicate the output of the logistic Lasso by replication of the weights from the MLP multiplied by the coefficients of the Lasso. The derivation of the Partial Response Network (PRN) proceeds as follows:

1. Train an MLP for binary classification;
2. Obtained the univariate and bivariate partial responses in **Equations (12)-(15)**.
3. Apply the Lasso to the partial responses;
4. Construct a second MLP as a linear combination of the partial responses to replicate the functionality of the Lasso. Each partial response, whether univariate or bivariate, is represented by a modular structure comprising the same number of hidden nodes as the original MLP. The modules are assembled into a single multi-layer structure represented as a GANN, shown in **Figure 9**.
5. Re-train the resulting multi-layer network by gradient descent. This results in the PRN.
6. Orthogonal Partial Responses can be obtained from the PRN and fed into the Lasso, leading to the PRN-Lasso.

The mapping of the partial responses onto the GANN requires matching the weights and bias terms as follows:

- Univariate partial responses:

$$v_j \rightarrow \beta_i * v_j \qquad \textbf{Equation 22}$$

$$v_0 \rightarrow \beta_i * \Big(v_0 - logit\big(P(C|0)\big)\Big) \qquad \textbf{Equation 23}$$

- Bivariate partial responses comprise two univariate and a bivariate block:

Univariate block weights:

$$v_j \rightarrow (\beta_k - \beta_{kl}) * v_j \qquad \textbf{Equation 24}$$

$$v_0 \rightarrow (\beta_k - \beta_{kl}) * \left(v_0 - logit\left(P(C|0)\right)\right)$$ **Equation 25**

Bivariate block weights:

$$v_j \rightarrow \beta_{kl} * v_j$$ **Equation 26**

$$v_0 \rightarrow \beta_{kl} * \left(v_0 - logit\left(P(C|0)\right)\right)$$ **Equation 27**

The input weights are the same as for the original, pre-trained MLP; the output weights correspond to the labels in fig 1, and the terms β_k, β_{kl} are the Lasso parameters for each partial response.

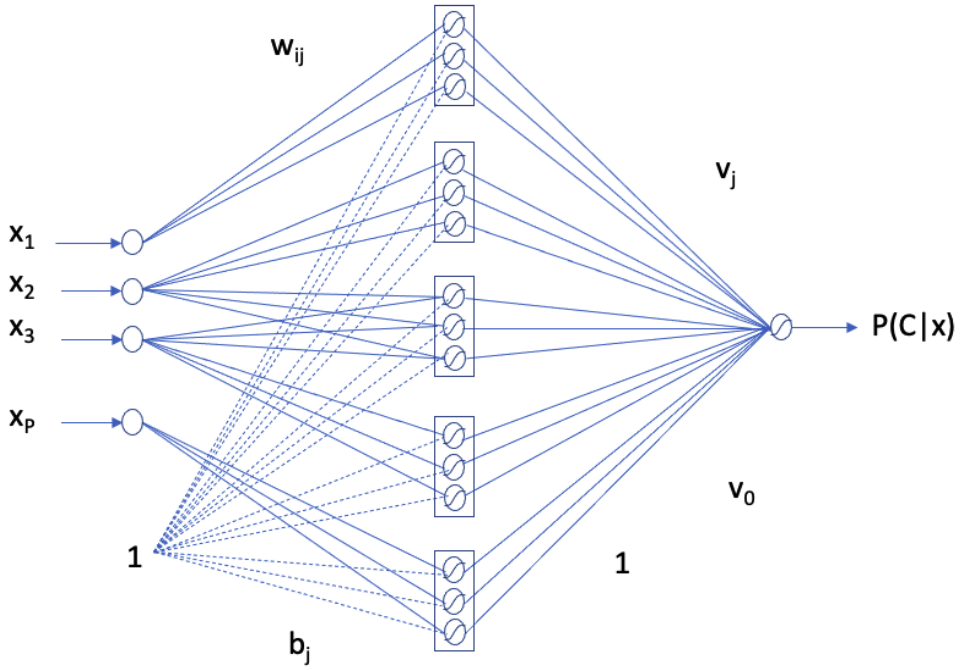


Figure 9. Structure of a Generalised Additive Neural Network (GANN), also known as a Self-Explanatory Neural Network (SENN). Each univariate effect, which we call a partial response, is modelled by a path with a separate block of hidden units. Bivariate terms involve three blocks of hidden units, one for each input and one receiving both inputs. The responses are added to make the input to the output node, i.e. the $logit(P(C|x))$.

The GANN architecture implements the calculation of the partial responses by embedding each partial response as an independent neural subnetwork whose output corresponds to a single ANOVA component.

The input-to-hidden weights of each subnetwork are copied from the original pre-trained MLP. This ensures that each block computes the same non-linear transformation of its input(s) as the black-box model. For a univariate partial response, only the corresponding input feature is active, while all other inputs are fixed at their anchor value through the bias terms, so the block output represents $\phi_i(x_i)$. For bivariate partial responses, three subnetworks are used: two univariate blocks and one joint block receiving both inputs, which together implement the inclusion–exclusion structure of the ANOVA decomposition.

The hidden-to-output weights of each block are then scaled by the Lasso coefficients associated with that partial response. As a result, the network output node computes a weighted sum of partial responses plus an intercept, which corresponds exactly to the additive decomposition of the logit. The

sigmoid activation applied at the output recovers the class probability. Thus, the GANN architecture realises the partial response calculations directly within the network structure while retaining interpretability through modular, additive components.

The additional variable reduction performed by the Lasso in the PRN-Lasso is not strictly necessary for interpretability, but it is useful for parsimony, stability, and generalisation. The PRN architecture already enforces an additive structure over univariate and bivariate partial responses, which substantially improves interpretability compared to the original black-box MLP. However, the number of candidate partial responses grows rapidly, particularly when bivariate terms are included. Many of these components may contribute little additional predictive power or may capture redundant information due to correlation between features. Applying Lasso after constructing the PRN serves to aggressively prune weak or redundant partial responses, yielding a sparser model that is easier to interpret and less prone to overfitting. From a practical perspective, the PRN-Lasso improves model robustness and calibration by reducing variance and multicollinearity among partial responses, while retaining the dominant effects learned by the black box.

When the PRN is retrained, its architecture is strongly constrained. Input-to-hidden weights are initialised from the original MLP and the network is trained to replicate an additive structure over a reduced set of partial responses. This retraining is therefore not equivalent to fitting a new unconstrained black box, but rather to fine-tuning a structured surrogate model.

Nevertheless, overfitting may occur if all stages are tuned on the same data without appropriate safeguards. In practice, this risk is mitigated by using validation splits or cross-validation for hyperparameter selection, limiting the order of interactions considered, and evaluating the final PRN-LASSO on held-out data. Under these conditions, the multiple fits act as regularising projections of the original model, rather than compounding sources of overfitting.

5.3 Data Description

The MIMIC-III clinical database (Johnson et al., 2016) is a large, publicly available database of critically ill patients who stayed in the intensive care units of the Beth Israel Deaconess Medical Centre between 2001 and 2012. The database is comprehensive and includes vital signs measurements, patient demographics, medications, procedure codes, diagnostic codes, laboratory measurements, imaging reports, hospital length of stay, and survival data, among others (Johnson et al., 2016). The variables for our study have been chosen based on a previous publication by Harutyunyan et al. (2019).

Outliers were removed during data cleaning, e.g. heart rate measurements below 0. We used information from the first 48 hours of the patients' admission to the ICU to model in-hospital mortality, also including the hour preceding the ICU admission for any prior information recorded in the ambulance.

In the cases where variables were time series, e.g. heart rate, we first calculated hourly means and then extracted the overall mean and standard deviation of each of these variables.

The Glasgow Coma Scale (GCS) scores, which relate to the level of consciousness of patients with acute brain injuries, were recorded following a standard clinical protocol (Teasdale & Jennett, 1974). GCS scores are treated as continuous since they are ordered from deep coma, at low scores, to fully conscious for high scores. Lower scores are known to be strongly associated with increased mortality risk.

Missing values are common in routinely collected clinical data. In this dataset, they were imputed with the same methods as Harutyunyan et al. (2019) namely using mean values. Patient records where the level of missingness exceeded 30% were discarded.

The overall mortality rate over the complete dataset is 11.3%. This is used in this study to illustrate how the proposed methods are robust against class imbalance, which is a common feature in clinical datasets. Moreover, our study measures calibration since this is a critical feature for the interpretation of posterior probabilities for patient stratification by risk. We have not extended the study into actual stratification, but we calculate the underpinning calibrated risk scores and correlate them with the additive response components for each statistically significant variable and pairwise interaction. To our knowledge, this level of analysis of this dataset is not available in the published literature, and it is also seldom published for clinical datasets generally, even though it is an essential component of performance validation for any probabilistic binary classifier for decision support in a high-stakes application.

The final dataset contains 7,532 observations (ICU patient admissions), 14 predictor variables and one binary response (1 = death before discharge, 0 = alive at discharge).

The study design involves splitting the data into three elements: a training dataset (n=4,519) and a validation dataset (n=1,506) which, together, form the model derivation database. This is used for model estimation and optimisation. However, the performance estimates may be optimistic on the validation dataset. Therefore, there is a third dataset, the test dataset (n=1,507). Cross-validation was not used as this is a large dataset, and training a model multiple times on large amounts of data requires a high computational cost. A hold-out set provides a good estimate and is sufficient enough to be representative of the entire population.

For each algorithm, a single model selected to be optimal as described in the next section was taken forward and applied to the out of sample dataset. This provides an unbiased estimate of generalisation performance. This aspect of our study is central to determining how well black boxes perform compared with the baseline models, Logistic Regression, SAM and EBM, and also with the interpretable models derived from pre-trained models.

The data are standardised by an affine transformation that consists of shifting the median to zero and scaling to unit variance.

5.4 Results

5.4.1 Classification Performance

This section benchmarks the classification performance of the PR models against two interpretable models, EBM (Lou et al., 2012) and SAM)(Ravikumar et al., 2009), as well as three state-of-the-art machine learning algorithms, GBM (Friedman, 2001), SVM (Vapnik, 1998) and RF (Breiman, 2001).

For each partial response model, we include two variants labelled 1 & 2 according to the selection of Lasso parameters: 1) best AUC on the validation set and 2) best AUC minus 1 standard error, with the aim of achieving a sparser model while maintaining significant performance.

Our results are shown in **Tables 4 & 5**, with only mean values of each covariate, and using both the mean and standard deviation. The 95% confidence intervals of the AUC are shown in brackets.

The variables selected by the best-performing interpretable models and the benchmark models are listed in **Tables 6 & 7**.

Table 4. Classification performance for MIMIC-III data with inputs as means only. C: Number of components

Model	C	Training AUC	Validation AUC	Test AUC
Interpretable models				
LR	9	0.774 (0.753, 0.796)	0.790 (0.752, 0.827)	0.785 (0.752, 0.818)
SAM	9	0.735 (0.711, 0.758)	0.742 (0.704, 0.780)	0.739 (0.702, 0.775)
EBM	19	0.828 (0.804, 0.850)	0.805 (0.764, 0.847)	0.790 (0.751, 0.829)
Black box models				
SVM	9	0.729 (0.705, 0.752)	0.726 (0.683, 0.768)	0.713 (0.674, 0.752)
RF	9	0.945 (0.935, 0.955)	0.806 (0.771, 0.841)	0.782 (0.747, 0.816)
GBM	9	0.813 (0.793, 0.833)	0.802 (0.767, 0.838)	0.787 (0.753, 0.820)
MLP	9	0.809 (0.786, 0.833)	0.790 (0.747, 0.833)	0.802 (0.763, 0.840)
Partial response models				
prSVM1	34	0.771 (0.747, 0.794)	0.778 (0.737, 0.818)	0.763 (0.727, 0.800)
prSVM2	19	0.755 (0.731, 0.779)	0.769 (0.730, 0.808)	0.755 (0.717, 0.792)
prRF1	43	0.923 (0.913, 0.934)	0.774 (0.735, 0.814)	0.778 (0.743, 0.813)
prRF2	36	0.905 (0.893, 0.917)	0.769 (0.728, 0.809)	0.775 (0.739, 0.811)
prGBM1	10	0.809 (0.789, 0.829)	0.804 (0.769, 0.838)	0.785 (0.751, 0.818)
prGBM2	6	0.786 (0.765, 0.807)	0.795 (0.761, 0.830)	0.768 (0.733, 0.803)
PRN	11	0.795 (0.771, 0.819)	0.791 (0.748, 0.834)	0.805 (0.768, 0.844)
PRN-Lasso	11	0.795 (0.771, 0.819)	0.789 (0.746, 0.832)	0.807 (0.768, 0.845)

Table 5. Classification performance for MIMIC-III data with means and standard deviations. C: Number of components

Model	C	Training AUC	Validation AUC	Test AUC
Interpretable models				
LR	14	0.790 (0.768, 0.812)	0.801 (0.765,0.837)	0.797 (0.763, 0.831)
SAM	14	0.749 (0.726, 0.773)	0.753 (0.716, 0.791)	0.744 (0.706, 0.782)
EBM	24	0.858 (0.837, 0.879)	0.812 (0.771, 0.853)	0.793 (0.754, 0.833)
Black box models				
SVM	14	0.989 (0.982, 0.995)	0.767 (0.724, 0.810)	0.732 (0.691, 0.772)
RF	14	0.960 (0.952, 0.968)	0.814 (0.779, 0.849)	0.797 (0.762, 0.832)
GBM	14	0.827 (0.807, 0.846)	0.805 (0.770, 0.841)	0.791 (0.756, 0.825)
MLP	14	0.828 (0.805, 0.850)	0.810 (0.769, 0.852)	0.815 (0.777, 0.853)
Partial response models				
prSVM1	54	0.830 (0.811, 0.850)	0.797 (0.759, 0.834)	0.794 (0.760, 0.828)
prSVM2	31	0.806 (0.785, 0.827)	0.786 (0.747, 0.825)	0.782 (0.746, 0.818)

prRF1	21	0.855 (0.839, 0.871)	0.770 (0.732, 0.808)	0.770 (0.733, 0.806)
prRF2	16	0.841 (0.824, 0.858)	0.761 (0.723, 0.799)	0.767 (0.731, 0.804)
prGBM1	15	0.817 (0.797, 0.837)	0.811 (0.777, 0.845)	0.783 (0.748, 0.818)
prGBM2	7	0.787 (0.766, 0.809)	0.802 (0.768, 0.836)	0.771 (0.734, 0.807)
PRN	12	0.810 (0.786, 0.833)	0.799 (0.756, 0.841)	0.807 (0.769, 0.845)
PRN-Lasso	12	0.811 (0.787, 0.834)	0.797 (0.755, 0.840)	0.812 (0.774, 0.850)

Table 6. Component functions selected by the sparse models and Partial Response models for MIMIC-III data with inputs as means only.

Model	**SAM**	**EBM**	**prGBM1**	**PRN-Lasso**
#Components	9	19	10	11
Univariate components				
Diastolic BP mean	✓	✓	✓	✓
Systolic BP mean	✓	✓	✓	
GCS Total mean	✓	✓	✓	✓
Glucose mean	✓	✓	✓	
Heart Rate mean	✓	✓	✓	✓
O2 Saturation mean	✓	✓	✓	✓
Respiratory Rate mean	✓	✓	✓	✓
Temperature mean	✓	✓	✓	✓
Weight	✓	✓	✓	✓
Two-way interactions				
Systolic BP mean X GCS Total mean		✓		✓
GCS Total mean X Heart Rate mean		✓		✓
GCS Total mean X Respiratory Rate mean		✓		✓
GCS Total mean X Temperature mean		✓		✓
O2 Saturation mean X Weight			✓	
Diastolic BP mean X GCS Total mean		✓		
GCS Total mean X Glucose mean		✓		
GCS Total mean X O2 Saturation mean		✓		
GCS Total mean X Weight		✓		
Systolic BP mean X Heart Rate mean		✓		
Heart Rate mean X Temperature mean		✓		

Table 7. Component functions selected by the sparse models and Partial Response models for MIMIC-III data with means and standard deviations.

Model	SAM	EBM	prGBM1	PRN-Lasso
#Components	14	24	15	12
Univariate components				
Diastolic BP mean	✓	✓	✓	
Diastolic BP st dev	✓	✓	✓	
Systolic BP mean	✓	✓	✓	✓
Systolic BP st dev	✓	✓		
GCS Total mean	✓	✓	✓	✓
GCS Total st dev	✓	✓	✓	
Glucose mean	✓	✓	✓	
Glucose st dev	✓	✓	✓	
Heart Rate mean	✓	✓	✓	✓
O2 Saturation mean	✓	✓		✓
O2 Saturation st dev	✓	✓	✓	✓
Respiratory Rate mean	✓	✓	✓	✓
Temperature mean	✓	✓	✓	✓
Weight	✓	✓	✓	✓
Two-way interactions				
Systolic BP mean X GCS Total mean		✓		✓
Systolic BP st dev X GCS Total mean		✓		✓
GCS Total mean X GCS Total st dev		✓		✓
GCS Total mean X Temperature mean		✓		✓
Systolic BP mean X Heart Rate mean			✓	
Systolic BP mean X O2 Saturation st dev			✓	
GCS Total mean X Weight			✓	
GCS Total mean X Glucose st dev		✓		
Diastolic BP st dev X GCS Total mean		✓		
GCS Total mean X Heart Rate mean		✓		
GCS Total mean X O2 Saturation st dev		✓		
Diastolic BP mean X GCS Total mean		✓		
GCS Total mean X Respiratory Rate mean		✓		

5.4.2 Visualisation and interpretability

Understanding the influence of individual physiological variables on mortality risk is essential for building interpretable and clinically meaningful predictive models. In this section, we examine the effect of key clinical variables in the PRN-Lasso model trained on the MIMIC-III dataset. By analysing univariate partial responses, we gain insights into how different features contribute to the model's predictions while holding other variables at their mean values.

One of the most critical factors in mortality prediction is the Glasgow Coma Scale (GCS) score, a widely used measure of consciousness in critical care. As illustrated in **Figure 10**, the model exhibits a

monotonic decrease in mortality probability with increasing GCS scores. The $logit(P(C|x))$ scale shows how this variable influences the logistic regression score index $\beta \cdot x$, providing a direct interpretation of its effect. The dashed line represents the initial response after the first iteration of the MLP, while the solid line shows the refined response following the GANN/SENN approach, which enhances interpretability and captures non-linear refinements. The smooth monotonic structure observed in both the initial MLP response and the refined PRN curve is clinically plausible, as neurological impairment tends to exert a graded and cumulative effect on mortality risk rather than abrupt threshold effects.

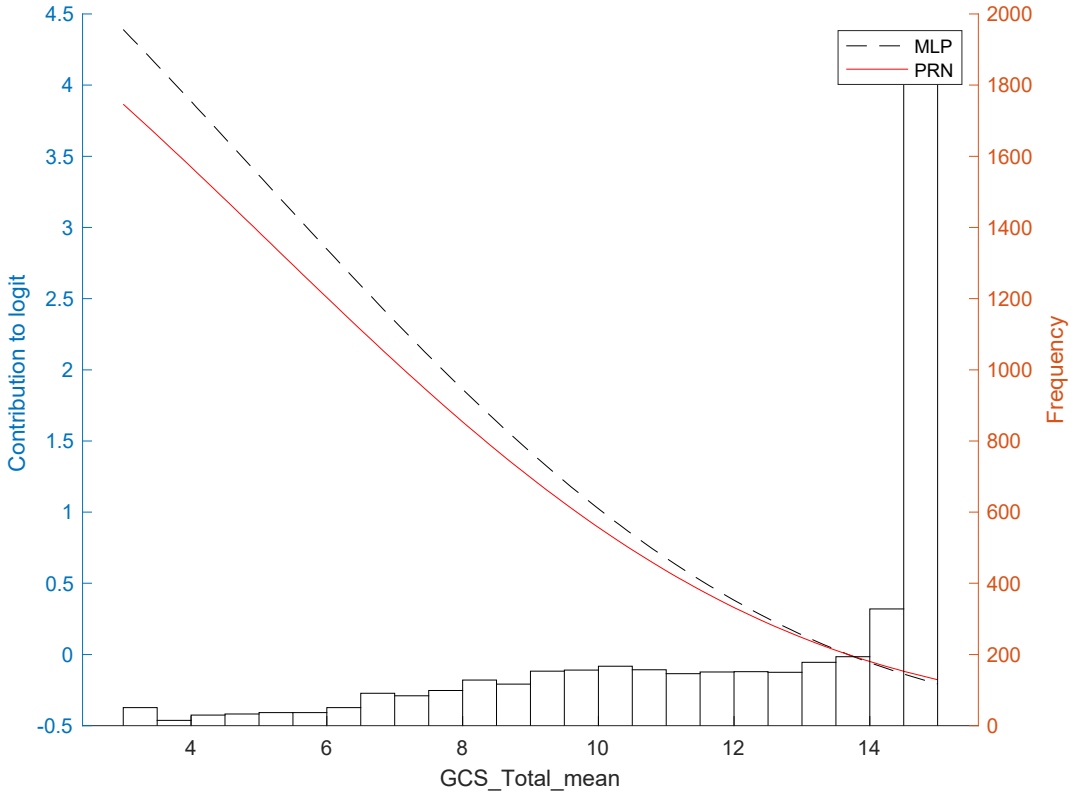


Figure 10. **Partial Response of the total GCS score in the PRN-Lasso model.** Example univariate partial responses from the PRN-Lasso model on the MIMIC III data using means only. The GCS score shows a monotonic decrease in mortality, as expected. The left hand side scale shows the contribution of this variable to the $logit(P(C|x))$, which corresponds directly to the score index $\beta.x$ in logistic regression. The dashed line is the initial partial response after the first iteration of the MLP and the solid line is the result after the second iteration using the GANN/SENN.

Beyond GCS, other physiological indicators also exhibit strong associations with mortality risk. **Figure 11** presents the effect of Respiratory Rate (RR), where the relationship is distinctly non-linear. Mortality risk increases away from the mean respiratory rate, but the effect is more pronounced at higher RR values, highlighting the asymmetry of risk factors in critically ill patients. Such non-linear dependencies, which standard logistic regression models struggle to capture, are effectively learned by the PRN-Lasso approach.

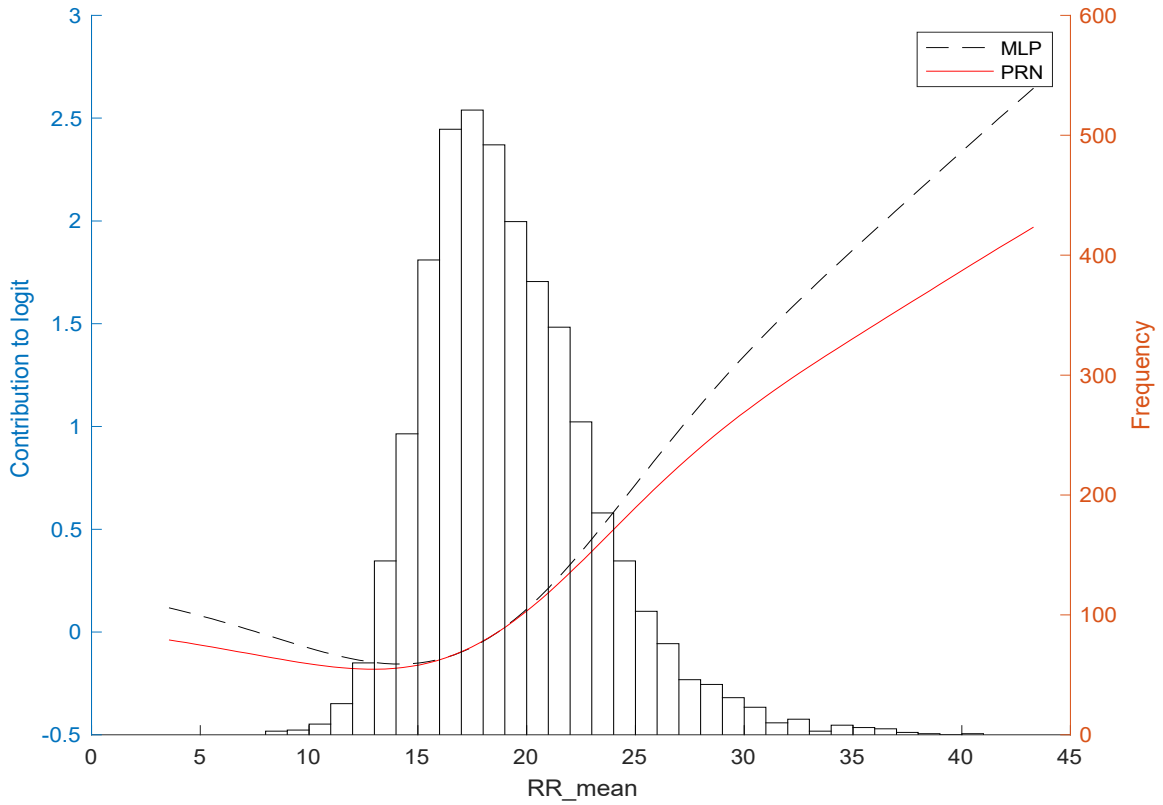


Figure 11. Partial Response of the Respiratory Rate in the PRN-Lasso model. Another important effect identified in the PRN-Lasso model is the Respiratory Rate (RR). This figure illustrates the non-linear nature of the partial responses. Mortality probability increases away from the mean respiratory rate, but the effect is more pronounced for higher RR values, highlighting the model's ability to capture non-linear trends in patient risk.

Similarly, **Figure 12** demonstrates the impact of core temperature on mortality. The model identifies a significant increase in mortality for lower temperatures. Notably, the partial response curve is approximately linear within the main body of the temperature distribution, suggesting why logistic regression performs well overall on this dataset. However, deviations at the extremes highlight the benefits of using a more flexible modelling approach like PRN-Lasso, which can capture subtle variations beyond simple linear trends.

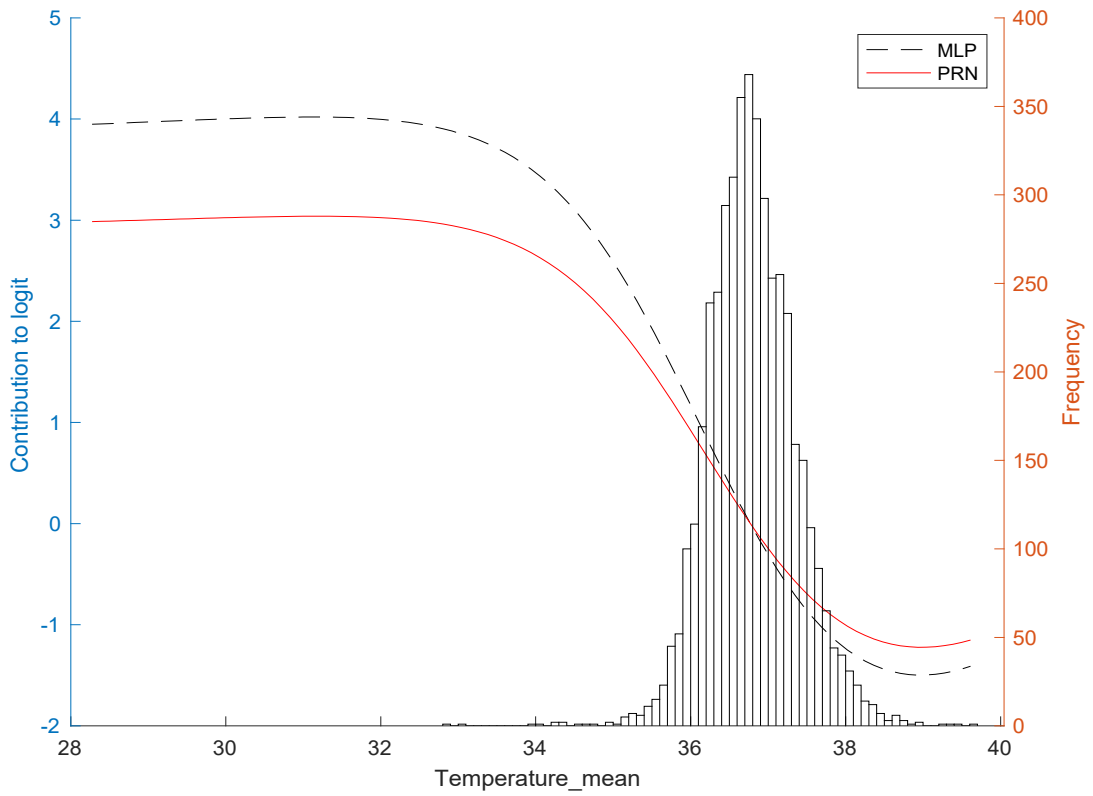
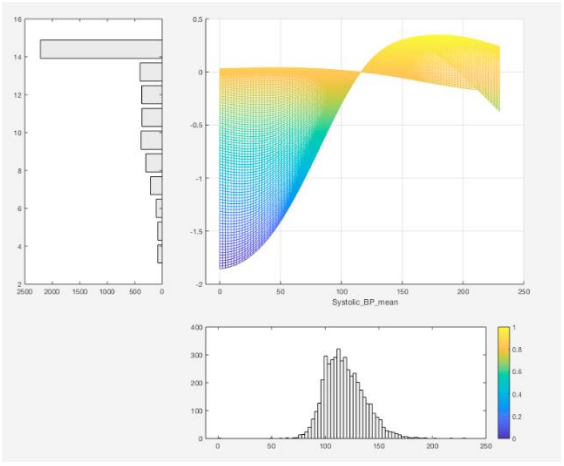


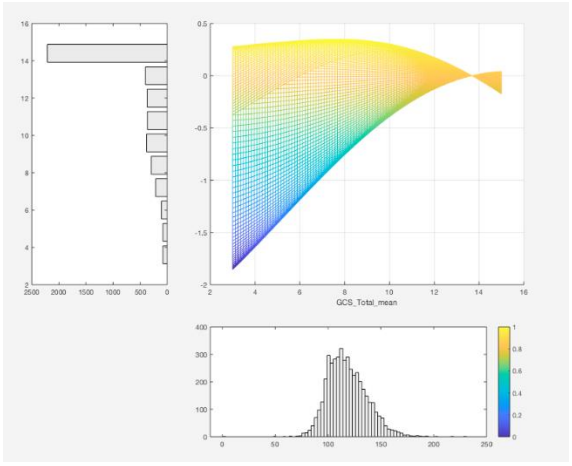
Figure 12. Partial Response of the core temperature in the PRN-Lasso model. Mean core temperature also has a statistically significant effect on mortality, as quantified in the PRN-Lasso model. Mortality risk increases for lower temperatures, with the response curve remaining approximately linear within the central temperature range. This explains why logistic regression performs well overall on this dataset while demonstrating the added value of more flexible models in capturing deviations at the extremes.

While univariate partial responses provide insight into individual feature effects, patient outcomes often depend on interactions between multiple physiological variables. **Figure 13** presents the bivariate partial response of the GCS score and Systolic Blood Pressure (SBP), two critical indicators in critical care. The figure includes two orthogonal views (**Figure 13 (a) & (b)**) along the main axes, demonstrating that the bivariate partial response vanishes along each axis, meaning that neither variable independently dominates mortality risk. Instead, their combined effect is essential in predicting patient outcomes.

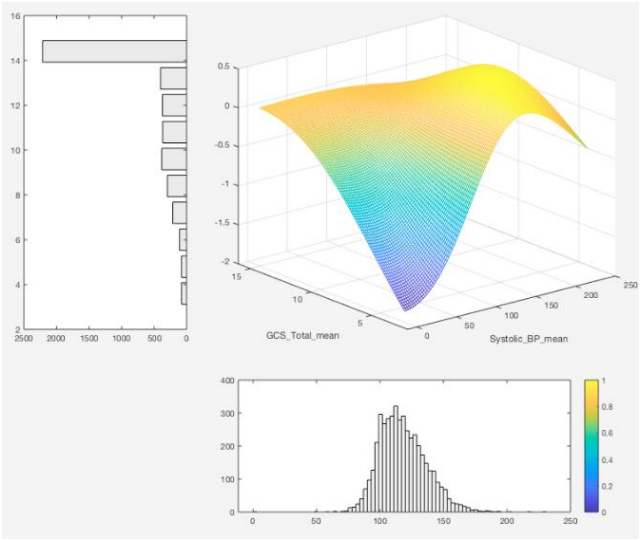
A crucial insight from the 3D representation (**Figure 13 (c)**) is the need for posterior probability calibration when both GCS score and SBP are low. This region of the response surface suggests a systematic underestimation of mortality risk, indicating that additional adjustments are required for optimal model performance. As with previous figures, histograms of the original variables are included to contextualise the data distribution, illustrating that these interactions are particularly relevant in clinically significant ranges.



(a)



(b)



(c)

Figure 13. Two-way interaction between the GCS score and Systolic Blood Pressure from the PRN-Lasso model. This graphic shows: (a) & (b) views along the main axis to show that the bivariate partial response vanishes along each axis; Note that the axes in the modelled data correspond to the values of the median in the original data, prior to standardisation by median centring and scaling to unit variance. (c) a 3D view. This graphic shows that a correction is required to ensure good calibration of the posterior probability for cases where the GCS score and Systolic BP are both low. In common with the other figures of the partial responses, the graphs show histograms of the original variables.

Beyond learning meaningful feature relationships, a well-calibrated predictive model should output probabilities that accurately reflect observed outcomes. **Figure 14** assesses the calibration of the PRN-Lasso model, demonstrating that its predictions are highly well-calibrated given the prevalence of mortality in the dataset—11.1% in the training data, 10.6% in validation, and 12.7% out-of-sample.

The histogram of output predictions is notably skewed toward lower values, reflecting the dataset's class imbalance. Importantly, the circles in the plot represent the proportion of mortality within each prediction bin, and they closely follow the ideal calibration line, confirming that the model provides reliable probability estimates across most prediction ranges. This high-quality calibration is essential for clinical deployment, as it ensures that probability outputs can be meaningfully interpreted for decision support.

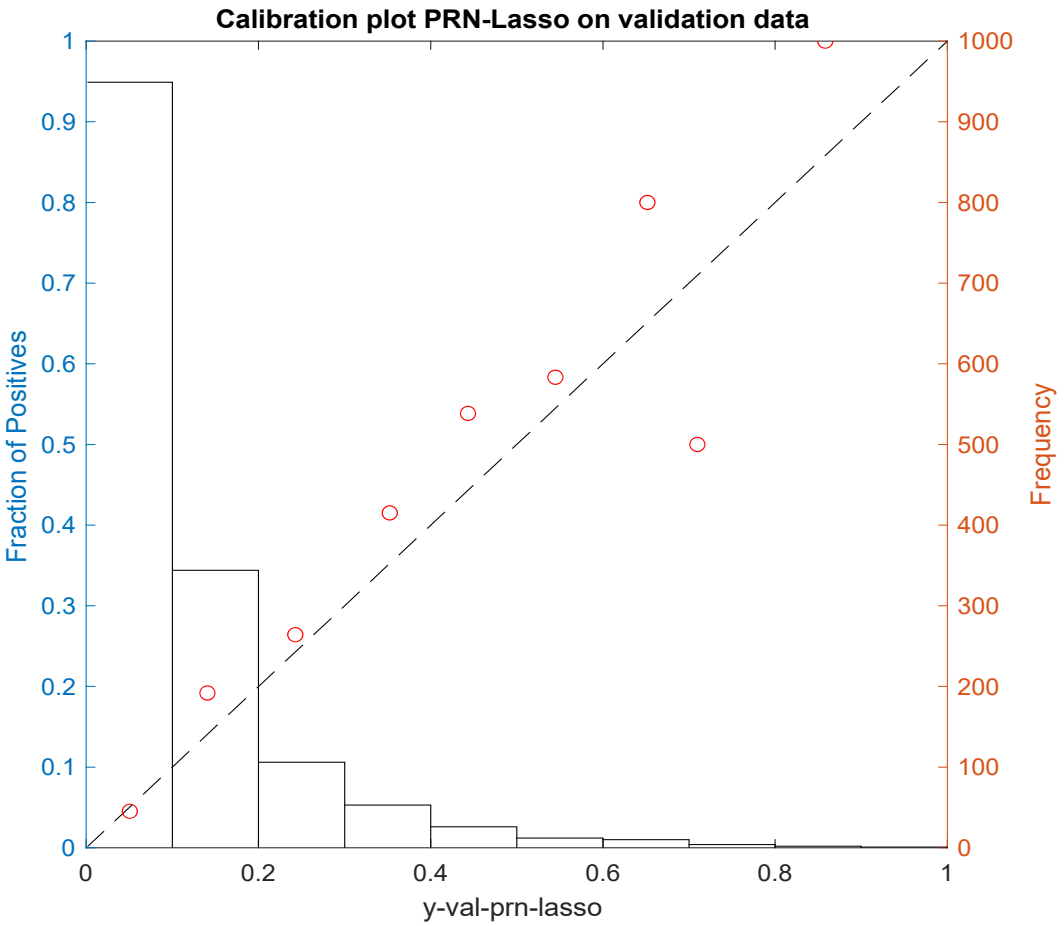


Figure 14. Calibration of the PRN-Lasso model. This figure evaluates the calibration quality of the PRN-Lasso model. The histogram of output predictions is heavily skewed toward lower values, reflecting the mortality prevalence in the dataset (11.1% in training, 10.6% in validation, 12.7% out-of-sample). The circles represent the proportion of observed mortality in each prediction bin and closely align with the ideal calibration line, confirming that the model produces well-calibrated probability estimates across most prediction intervals.

To evaluate the robustness and consistency of different predictive modelling approaches, we compare the univariate partial response of the GCS score across multiple models. Given its clinical significance as a measure of consciousness and neurological function, GCS serves as an essential variable for assessing model behaviour.

Figure 15 presents the univariate response of GCS score in four distinct models: prGBM, prSVM, EBM, and SAM. As observed previously in **Figure 10**, all models exhibit a monotonic decrease in mortality probability as the GCS score increases, consistent with clinical expectations. However, the specific shape of the response curve varies between models, reflecting differences in their underlying assumptions and learning mechanisms.

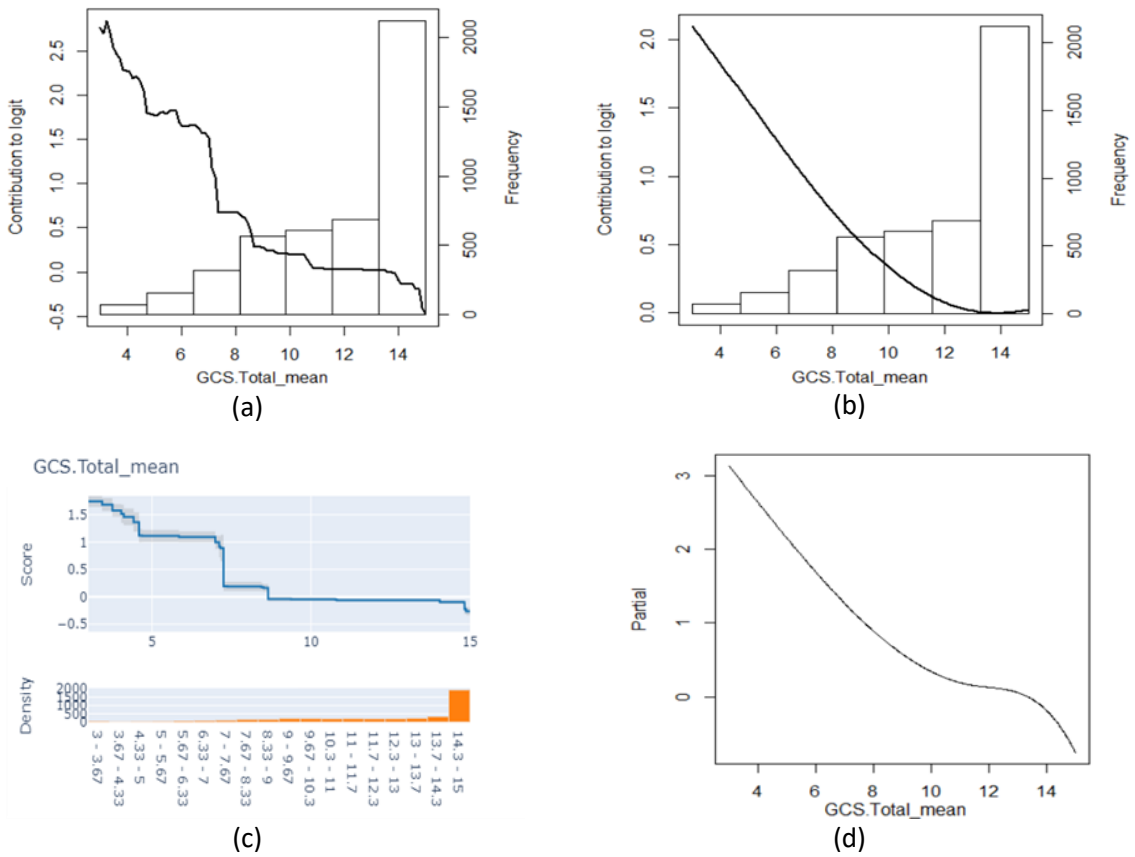


Figure 15. Example univariate responses for GCS score from the (a) prGBM, (b) prSVM, (c) EBM and (d) SAM models. Similarly to **Figure 10**, these plots show a decrease in mortality as the GCS score increases.

The prGBM model captures a smooth, non-linear decline in mortality risk, leveraging its boosted tree structure to represent flexible feature interactions. The prSVM model follows a similar trend, though the nature of its decision boundaries may introduce more rigid segmentations depending on the choice of kernel. The EBM model, designed for transparency and interpretability, maintains a largely linear decrease with slight deviations, illustrating its reliance on additive effects. The SAM model exhibits behaviour similar to prGBM, demonstrating both flexibility and smoothness while maintaining a degree of interpretability.

These findings highlight that while different machine learning models can effectively capture the expected relationship between GCS score and mortality risk, their representations of this effect differ. Some models prioritise flexibility, allowing for more nuanced responses, while others emphasise transparency and stability. Understanding these differences is crucial when selecting models for clinical applications, where both predictive performance and explainability are essential considerations.

5.5 Discussion

Most models have comparable performance, but they differ significantly in their interpretability. In particular, while LR is restricted to linear dependence on the covariates, it performs well for this dataset. This is explained by the partial responses, which show that the dependence on the individual variables is close to linear in the main body of the histogram for that variable. However, the responses

saturate either side of it and so flatten out, hence the marginally better performance of some of the partial response models.

The results show that SVM is not ideally suited when non-linearities are weak, with overfitting that results in the prSVM sometimes outperforming the original back box on the out of sample data. In contrast, the GBM generates partial responses that generalise well. The RF model has the highest performance degradation when the ANOVA decomposition is truncated. This is likely because the partial responses for the RF are staggered and not smooth due to the internal structure of the black box model.

The PRN and PRN-Lasso perform very well for this medical dataset. **Figures 10-12** show three of the component responses that together add to make the $logit(P(C|x))$ for the PRN-Lasso model. The findings in **Figure 10** are strongly supported by clinical literature, where GCS is a cornerstone prognostic indicator in intensive care, trauma, and neurological assessment. Lower GCS scores are consistently associated with increased mortality risk across diverse ICU populations, reflecting diminished neurological function and severity of illness (Knaus et al., 1985; Teasdale & Jennett, 1974). The asymmetric response shown in **Figure 11** is well documented in critical care literature. Tachypnoea is a sensitive marker of physiological deterioration, reflecting hypoxia, metabolic acidosis, sepsis, or respiratory failure, and has been repeatedly shown to be a stronger predictor of adverse outcomes than bradypnoea in ICU settings (Badawi et al., 2018; Cretikos et al., 2008). Similarly, the impact of core temperature on mortality shown in **Figure 12** is consistent with clinical knowledge about hypothermia as a risk factor (Peres Bota et al., 2004; Young et al., 2015).

The predicted probabilities can be compared with the observed occurrence of mortality to produce the calibration curve. This is shown in **Figure 14** for the out of sample dataset. Note as well the very good match between the predicted probability of mortality, in the x-axis, and the fraction of predicted cases in the same interval of predicted mortality, shown by the circles.

Calibration is vital for clinical applications where the quantitative inference of the output must be numerically accurate. It is very much possible to have poor calibration with excellent classification performance measured by the AUC. This occurs when the predictions are in the right ranking order but their numerical values may be very skewed, for instance, due to class imbalance. Not all classifiers are well-calibrated, but the MLP is, even for extreme imbalances of the order of 1/100, as is the case for instance when predicting event rates for short time intervals in survival modelling.

The AUC performance of the proposed approaches is in line with those reported by Harutyunyan et al. (2019). Nevertheless, the data structures used and precise experiments are not the same, and we used very simple compression of the time series. The paper makes the comment that “even a model with 0.91 AUC-ROC can make trivial mistakes and there is a lot of room for improvement”. We agree and suggest that the interpretability element is helpful to identify the precise weight that each input variable, or pair of input variables, contributes to the prediction, hence finding out what, if anything, misled the model.

The ability to diagnose the model, that is to say, to find out exactly why it was right or not, is important in order to improve it in a controlled manner or, even, to find issues in data collection e.g. artifacts or variables missing from the protocol, or unintended biases in the sampling process. Moreover, this also allows the clinician to integrate the machine learning model, including pre-trained black boxes, into the clinical reasoning process.

Compared with the state of the art, our models perform better than SAM and similarly to EBM. Both of these approaches in principle support univariate and bivariate effects, but in SAM the component additive elements are restricted to splines, which can be a limiting factor on performance.

In the case of the EBM, its partial response for the GCS is shown in **Figure 15**, alongside the corresponding functions derived from the SVM and GBM algorithms. These plots follow the same trend as **Figure 10**, which is the expected decrease in mortality for higher GCS scores due to how it is calculated. Interestingly, while the plot for the prSVM is smooth, it shows a curvature that may be an artifact resulting from the width of the original radial basis functions. A similar effect is present in all of the component functions from this model.

The component function for the GBM is noticeably noisy, and for the EBM it is staggered. This may lead to a loss in classification performance compared to a better estimate of the partial response. The example in **Figure 10** is consistent with the smooth interpolation of the curves in **Figure 15 (a) & (c)**.

The variables selected by the best performing interpretable models and the benchmark models are listed in **Tables 6 & 7**. While we have already noted that SAM supports bivariate effects in principle, we were unable to find any reference to these in the software used. The EBM and prGBM1 models, like the SAM, select all univariate components as well as several bivariate components. Glucose does not appear in any univariate or bivariate term of the PRN-Lasso.

The partial response models are sparser than the EBM, containing a similar number of terms to the SAM, while also including bivariate terms. The bivariate components selected by the PRN-Lasso suggest that they are corrections to the calibration of the GCS Total Mean. In contrast, the EBM utilises more than double the number of bivariate terms.

The additional step for the MLP of mapping the Lasso model onto a SENN and continuing training to result in the PRN model, led to only a small improvement in performance. This is apparent also from the small changes observed in the shape of the partial responses. This indicates that for real-world data sets such as MIMIC the noise present in the data limits performance to the extent that the significant predictive factors are well represented by just the univariate and bivariate terms in the ANOVA decomposition.

5.6 Conclusion

We show that it is possible to open any black box model, including pre-trained models, in cases where significant noise is present, without losing much predictive power, if any, but making the model transparent to the non-linearities in the data.

This involves the application of the ANOVA decomposition, anchored on the median of the data followed by the Lasso as a computationally efficient method to derive the structure of a GAM using the component functions derived from the ANOVA. The application of the LASSO to the univariate and bivariate terms in the ANOVA decomposition carries out the functions of model selection and re-calibration, resulting in a globally interpretable model with comparable performance to the original black box model. In this way, we buck the accuracy/interpretability trade-off for tabular data.

Furthermore, the performances of the resulting GAMs compare favourably with state-of-the-art sparse models from the statistical literature, SAM, and from machine learning, the EBM. The derived Partial Responses are consistent across the range of models and have plausible clinical interpretations.

The interpretability of the model by end-users is at the level of nomograms (Van Belle et al., 2016). Nomograms are familiar to clinicians as graphical implementations of logistic regression. GAMs are interpretable in the same way, except that the score for each variable is read from what we call the partial response plot, where the height of the plot directly measures the contribution to the logit, which is the nomogram score for that variable.

6 PRiSM

6.1 Introduction

As discussed in Section 3.1.1, there exist two natural choices of measure for computing partial responses. Thus far, the Dirac measure has served as the foundation for our methodology, offering a discrete and computationally efficient means of approximating the influence of individual variables across their respective domains. However, an alternative approach merits investigation: the Lebesgue measure (Zhang et al., 2010). This continuous measure provides a more theoretically grounded evaluation of variable contributions, capturing the true, unanchored behaviour of features over their full range.

In this section, we undertake a comparative analysis of the Dirac and Lebesgue measures, focusing on two key dimensions: classification performance and qualitative explanatory power. Specifically, we assess whether the more computationally intensive Lebesgue measure yields improvements in both predictive accuracy and visual interpretability over the Dirac-based approximation. In doing so, we aim to evaluate the trade-offs between theoretical precision and computational efficiency, and to determine whether the adoption of the Lebesgue measure offers tangible advantages for practical applications of the partial response methodology.

6.2 Data Description

6.2.1 Synthetic Data

The 2-D Circle artificial dataset contains 10,000 observations with unbalanced classes, which is consistent through all future artificial datasets. There are two covariates with randomly generated values, using $x_i = 0.5 * (u_i + w)$ where u_i and w are uniform distributions in the range [0,1]. This will demonstrate the prediction accuracy when the variables are correlated. The logit has two separate univariate components:

$$logit(P(C|(x_1, x_2)) = 10 * \left[\left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2 - 0.08\right] \quad \textbf{Equation 28}$$

We use noisy data by generating binary classes with a Bernoulli distribution, which is done for all future artificial datasets reported:

$$Y \sim Bin\left(n, P(C|(x_1, x_2))\right). \quad \textbf{Equation 29}$$

The factor of 10 in **Equation 28** is to reduce the amount of noise and ensure a reasonable AUC. There are two univariate main effects and no interaction terms.

The XOR function artificial dataset is a pure bivariate interaction, represented in the multilinear form appropriate for continuous Boolean algebra (Tsukimoto, 2000)(Tsukimoto, 2000):

$$P(C|(x_3, x_4)) = x_3 + x_4 - 2x_3x_4, x_i \in]0,1[\quad \textbf{Equation 30}$$

Each covariate is uniformly distributed in [0,1]. The density function has the property that

$$logit(P(C|(x_3, x_4)) = log\left(\frac{x_3 + x_4 - 2x_3x_4}{1 - x_3 - x_4 + 2x_3x_4}\right) \quad \textbf{Equation 31}$$

Therefore, $logit(P\left(C\middle|\left(x_3,\frac{1}{2}\right)\right)=0$ making it a pure interaction when the ANOVA decomposition with the Dirac measure is anchored at (1/2,1/2). For the Lebesgue measure:

$$logit(P\big(C\big|(x_3,x_4)\big)=-logit(P\big(C\big|(1-x_3,x_4)\big) \quad \textbf{Equation 32}$$

and similarly for the other dimension, therefore the integrals over [0,1] corresponding to the univariate terms vanish, once again leaving the pure interaction term.

The Logical AND function artificial dataset is a combination of univariate and bivariate terms in which data is generated based on the logical AND function, represented in continuous Boolean algebra by the following atomic term:

$$P\big(C\big|(x_5,x_6)\big)=x_5x_6, x_i\in]0,1[\quad \textbf{Equation 33}$$

The ANOVA expansion anchored at (1/2,1/2) is:

$$\begin{aligned} logit\left(P\big(C\big|(x_5,x_6)\big)\right) \\ &= -\log(3)+\sum_i\left[\log\left(\frac{x_i}{2-x_i}\right)+\log(3)\right] \\ &+\left[\log\left(\frac{(2-x_5)(2-x_6)}{1-x_5x_6)}-\log(3)\right)\right] \end{aligned} \quad \textbf{Equation 34}$$

The Lebesgue measure yields univariate terms given by

$$\varphi_i^{Lebesgue}=log\left({}^{x_i}/_{(1-x_i)}\right)+\frac{log(1-x_i)}{x_i}+Li_2(1) \quad \textbf{Equation 35}$$

where $Li_2(1)$ is the polylogarithm function of second order evaluated at 1. The bivariate term is given by the explicit ANOVA decomposition in eq. (5) and does not reduce to a simpler algebraic form.

The Three-way interaction artificial dataset comprises a dataset than cannot be modelled by only univariate and bivariate terms. The purpose is to see how well PRiSM models work to model high-order effects:

$$P\big(C|(x_7,x_8,x_9)\big)=x_7x_8x_9, x_i\in]0,1[\quad \textbf{Equation 36}$$

We combine these artificial datasets into a 9-dimensional input set for all classifier targets. In each case, only two or three variables are relevant to the classification and the others comprise noise. A minimum requirement of all classifiers is to identify the relevant input dimensions and discard the rest. In addition, since we have the data generators, we can calculate the optimal classification performance corresponding to allocating every data point to the correct class, irrespective of the stochastic label generated by the Bernoulli distribution.

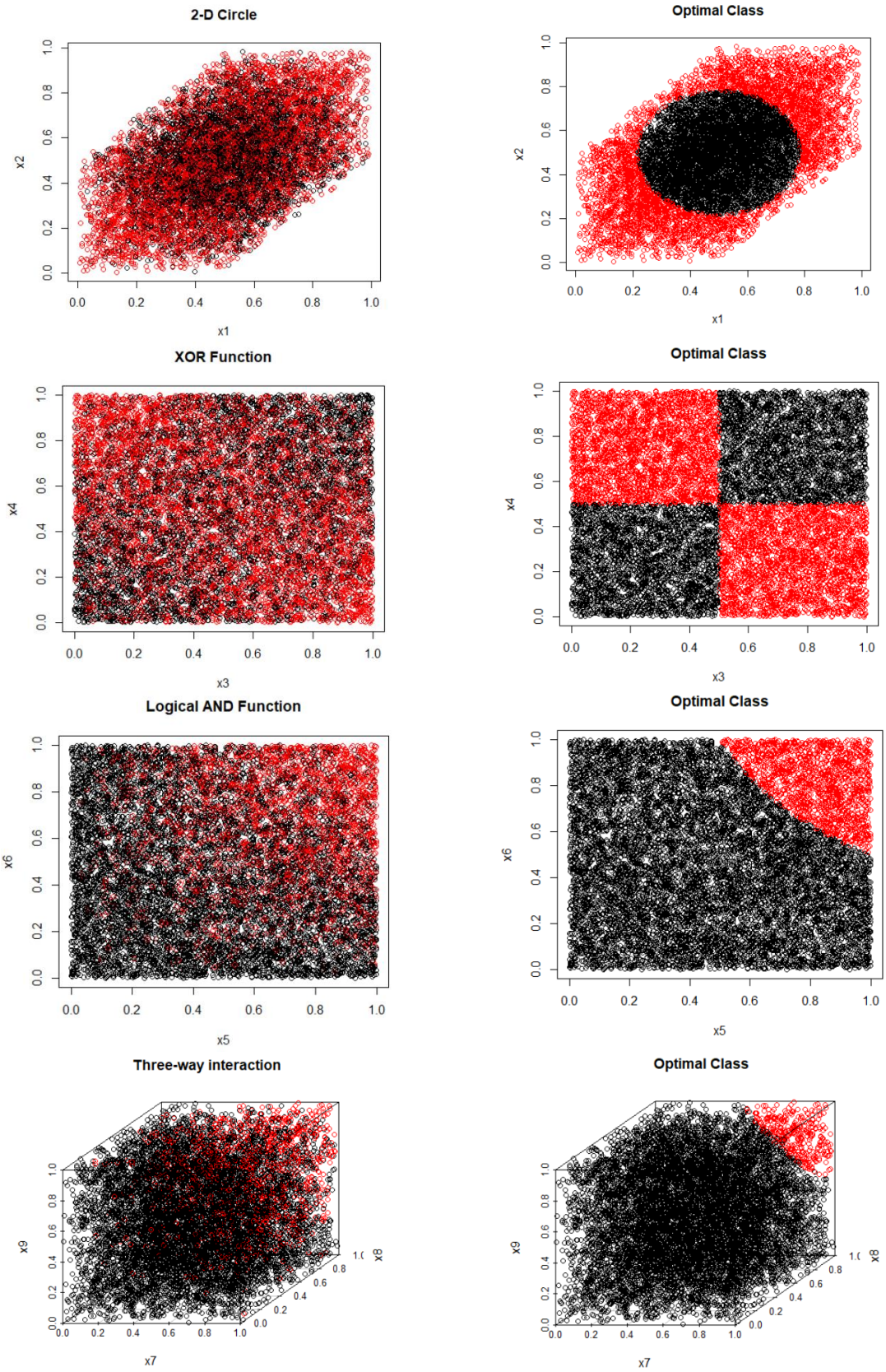


Figure 16. Two-dimensional plots of the relevant variables from the 9 input dimensions are plotted, showing the actual training data with Bernoulli noise and the ideal class allocations used to find the best achievable AUC.

6.2.2 Real World Data

The Statlog Shuttle dataset (Abe et al., 2006; Kelly et al., n.d.) from NASA comprises 9 numerical attributes and an outcome label. It is split into 43,500 cases for training and 14,500 for testing. There are 7 outcomes, of which 21% are in the category "Rad Flow". The binary classification task is to separate this category, Class 0, from the others, assigned to Class 1. Given the strong imbalance between classes, the default accuracy for a null model, i.e., predicting the predominant class for all rows, is 79%. The target accuracy is 99–99.9%. The Pima Diabetes and Statlog German Credit Card datasets are also utilised in this analysis, although we use the full version of the Statlog German Credit Card dataset as described below.

The numerical version of the **Statlog German Credit Card dataset** (Kelly et al., n.d.) contains n=1000 instances and 24 attributes. The first 700 observations were used for training, with a prevalence of bad credit risks being 29.6%. The remaining 300 observations for testing, with a prevalence of bad risks of 31%. The data set was used in the form created for the benchmarking study Statlog, where three categorical variables ("Other Debtors", "Housing" and "Employment") were coded in binary form with multiple columns.

6.3 Results

The following sections compare the performance and characteristics of different PRiSM models obtained by opening a range of frequently used black box algorithms, including the PrMLP which is the model created prior to the derivation of the PRN.

6.3.1 Classification Performance

The purpose of the benchmarking on the synthetic data is to ascertain how close each machine learning classifier and the corresponding interpretable PRiSM models get to the optimal classification accuracy, which is obtained using the known class membership probabilities given by the generating formulae for class membership, notwithstanding the presence of noise in the targets.

The classification performance of frequently used machine learning models and their interpretable versions applied to the four synthetic sets are listed in **Tables 8-11**. The generated data set was split into three parts: training, validation, and independent (out-of-sample) test set. It is interesting to see how much the optimal AUC varies between three slices from the same noisy data. This illustrates the importance of calculating confidence intervals. The model with marginally the best point estimate of the AUC for the optimisation data may not have the highest AUC estimated on the independent sample.

The benchmarking results for the interpretable models against the original black box classifiers for the real-world datasets are summarised in **Table 12**. All methods use the same data sets and the AUCs are quoted for test data only.

Table 8. Classification performance for the 2-D circle measured by the AUC [CI]. The input variables x1 and x2 are ideally selected solely for their univariate responses.

AUC [CI]	No. input variables	Training (n=6,000)	Validation (n=2,000)	Test set (n=2,000)
Optimal classifier	2	0.676 [0.662,0.689]	0.657 [0.634,0.681]	0.666 [0.643,0.690]
MLP	9	0.676 [0.663,0.690]	0.659 [0.635,0.682]	0.660 [0.636,0.684]
SVM	9	0.695 [0.682,0.708]	0.646 [0.622,0.670]	0.648 [0.624,0.672]
GBM	9	0.697 [0.684,0.710]	0.649 [0.625,0.673]	0.641 [0.617,0.665]
PRiSM models	***Components***		***Dirac measure***	
prMLP	2	0.675 [0.661,0.688]	0.658 [0.634,0.682]	0.661 [0.637,0.685]
PRN	2	0.676 [0.662,0.689]	0.659 [0.636,0.683]	0.664 [0.640,0.687]
PRN-Lasso	2	0.676 [0.662,0.689]	0.659 [0.636,0.683]	0.664 [0.640,0.688]
prSVM	2	0.676 [0.662,0.689]	0.658 [0.634,0.681]	0.664 [0.640,0.688]
prGBM	5	0.681 [0.667,0.694]	0.655 [0.631,0.679]	0.655 [0.632,0.679]
PRiSM models	***Components***		***Lebesgue measure***	
prMLP	2	0.675 [0.662,0.689]	0.659 [0.636,0.683]	0.661 [0.637,0.685]
PRN	2	0.676 [0.662,0.689]	0.659 [0.636,0.683]	0.664 [0.640,0.687]
PRN-Lasso	2	0.676 [0.662,0.689]	0.660 [0.636,0.683]	0.664 [0.640,0.687]
prSVM	3	0.675 [0.662,0.689]	0.657 [0.634,0.681]	0.665 [0.641,0.689]
prGBM	2	0.673 [0.659,0.686]	0.656 [0.632,0.679]	0.654 [0.630,0.678]

Table 9. Classification performance for the XOR function measured by the AUC [CI]. The input variables x3 and x4 are ideally selected solely for their bivariate response.

AUC [CI]	No. input variables	Training (n=6,000)	Validation (n=2,000)	Test set (n=2,000)
Optimal classifier	1	0.689 [0.675,0.702]	0.663 [0.639,0.687]	0.671 [0.648,0.695]
MLP	9	0.692 [0.678,0.705]	0.665 [0.641,0.688]	0.669 [0.646,0.693]
SVM	9	0.708 [0.695,0.721]	0.652 [0.628,0.676]	0.660 [0.637,0.684]
GBM	9	0.713 [0.700,0.726]	0.586 [0.561,0.610]	0.609 [0.584,0.633]
PRiSM models	***Components***		***Dirac measure***	
prMLP	1	0.688 [0.675,0.701]	0.663 [0.639,0.686]	0.672 [0.648,0.695]
PRN	1	0.690 [0.677,0.703]	0.664 [0.640,0.687]	0.670 [0.646,0.694]
PRN-Lasso	1	0.688 [0.675,0.702]	0.663 [0.639,0.686]	0.672 [0.648,0.695]
prSVM	14	0.691 [0.678,0.705]	0.663 [0.640,0.687]	0.671 [0.648,0.695]
prGBM	1	0.687 [0.674,0.700]	0.656 [0.633,0.680]	0.661 [0.638,0.685]
PRiSM models	***Components***		***Lebesgue measure***	
prMLP	1	0.689 [0.676,0.702]	0.664 [0.640,0.688]	0.670 [0.647,0.694]
PRN	1	0.690 [0.677,0.703]	0.664 [0.640,0.687]	0.670 [0.646,0.693]
PRN-Lasso	1	0.690 [0.676,0.703]	0.664 [0.641,0.688]	0.670 [0.647,0.694]
prSVM	7	0.690 [0.677,0.703]	0.633 [0.640,0.687]	0.672 [0.648,0.695]
prGBM	1	0.688 [0.675,0.702]	0.656 [0.632,0.680]	0.659 [0.635,0.682]

Table 10. Classification performance for the logical AND function measured by the AUC [CI]. The input variables x5 and x6 are ideally selected with two univariate responses and a bivariate response.

AUC [CI]	No. input variables	Training (n=6,000)	Validation (n=2,000)	Test set (n=2,000)
Optimal classifier	3	0.816 [0.802,0.830]	0.836 [0.813,0.860]	0.817 [0.793,0.841]
MLP	9	0.816 [0.803,0.830]	0.833 [0.809,0.857]	0.815 [0.791,0.839]
SVM	9	0.803 [0.790,0.817]	0.797 [0.772,0.821]	0.786 [0.762, 0.809]
GBM	9	0.822 [0.810,0.834]	0.826 [0.805,0.847]	0.808 [0.787,0.830]
PRiSM models	***Components***		***Dirac measure***	
prMLP	3	0.815 [0.801,0.828]	0.833 [0.809,0.857]	0.813 [0.789,0.837]
PRN	3	0.816 [0.802,0.829]	0.835 [0.811,0.858]	0.814 [0.790,0.838]
PRN-Lasso	3	0.816 [0.802,0.830]	0.835 [0.811,0.859]	0.814 [0.791,0.838]

prSVM	6	0.800 [0.787,0.813]	0.813 [0.790,0.835]	0.797 [0.774, 0.820]
prGBM	6	0.820 [0.807,0.832]	0.828 [0.807,0.848]	0.807 [0.786,0.829]
PRiSM models	***Components***		***Lebesgue measure***	
prMLP	3	0.815 [0.801,0.828]	0.832 [0.808,0.856]	0.813 [0.789,0.837]
PRN	3	0.816 [0.802,0.829]	0.835 [0.811,0.858]	0.814 [0.790,0.838]
PRN-Lasso	3	0.816 [0.802,0.830]	0.835 [0.811,0.858]	0.815 [0.791,0.839]
prSVM	4	0.799 [0.786,0.812]	0.812 [0.790,0.834]	0.796 [0.773,0.819]
prGBM	8	0.817 [0.805,0.829]	0.828 [0.808,0.849]	0.810 [0.789,0.831]

Table 11. Classification performance for the three-way interaction measured by the AUC [CI]. Three input variables are involved, x7, x8 and x9.

AUC [CI]	No. input variables	Training (n=6,000)	Validation (n=2,000)	Test set (n=2,000)
Optimal classifier	3	0.840 [0.822,0.859]	0.817 [0.783,0.851]	0.836 [0.805,0.868]
MLP	9	0.840 [0.822,0.859]	0.809 [0.775,0.843]	0.832 [0.801,0.864]
SVM	9	0.797 [0.779,0.815]	0.764 [0.729,0.798]	0.786 [0.755,0.817]
GBM	9	0.831 [0.816,0.847]	0.796 [0.767,0.826]	0.813 [0.786,0.840]
PRiSM models	***Components***		***Dirac measure***	
prMLP	3	0.837 [0.818,0.855]	0.811 [0.777,0.845]	0.821 [0.797,0.861]
PRN	3	0.837 [0.819,0.856]	0.812 [0.778,0.846]	0.830 [0.799,0.862]
PRN-Lasso	3	0.837 [0.819,0.856]	0.812 [0.778,0.846]	0.830 [0.799,0.862]
prSVM	6	0.813 [0.796,0.829]	0.777 [0.744,0.810]	0.807 [0.778,0.836]
prGBM	3	0.832 [0.817,0.847]	0.797 [0.768,0.826]	0.813 [0.786,0.841]
PRiSM models	***Components***		***Lebesgue measure***	
prMLP	3	0.834 [0.816,0.853]	0.808 [0.774,0.842]	0.828 [0.796,0.860]
PRN	3	0.837 [0.819,0.856]	0.812 [0.778,0.846]	0.831 [0.799,0.862]
PRN-Lasso	3	0.837 [0.819,0.856]	0.812 [0.778,0.846]	0.831 [0.799,0.862]
prSVM	6	0.808 [0.792,0.824]	0.776 [0.745,0.808]	0.805 [0.777,0.833]
prGBM	4	0.825 [0.809,0.841]	0.798 [0.768,0.828]	0.810 [0.781,0.839]

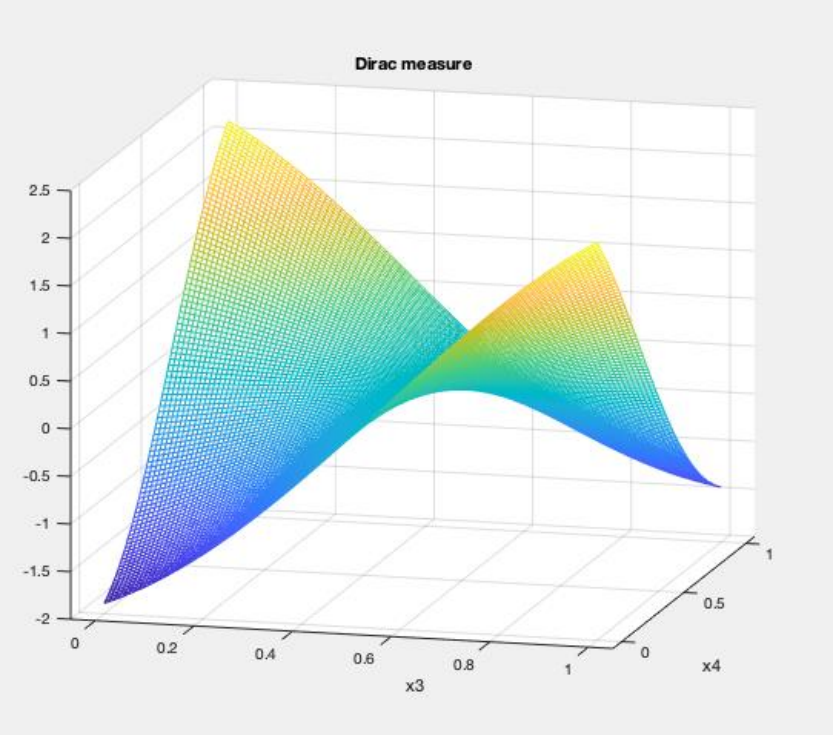
Table 12. Classification performance for the real world data sets. The label 'D' indicates the number of input variables for the black boxes and component functions for the PRiSM models.

AUC [CI]	D	Diabetes	D	Credit Card	D	Shuttle
MLP	7	0.902 [0.850,0.954]	24	0.811 [0.754,0.869]	6	0.999 [0.998,1.000]
SVM	7	0.817 [0.749,0.884]	24	0.793 [0.733,0.852]	6	0.999 [0.999,1.000]
GBM	7	0.816 [0.748,0.884]	24	0.784 [0.724,0.845]	6	1.000 [0.999,1.000]
PRiSM models			***Dirac measure***			
prMLP	5	0.902 [0.851,0.954]	14	0.808 [0.750,0.866]	3	0.999 [0.999,1.000]*
PRN	5	0.903 [0.851,0.954]	14	0.809 [0.752,0.867]	3	0.999 [0.998,1.000]*
PRN-Lasso	5	0.903 [0.851,0.955]	13	0.811 [0.754,0.868]	2	0.998 [0.997,0.999]*
prSVM	5	0.884 [0.829,0.940]	13	0.798 [0.739,0.857]	3	0.998 [0.997,0.999]
prGBM	8	0.847 [0.784,0.910]	10	0.763 [0.700,0.825]	2	0.998 [0.997,0.999]
PRiSM models			***Lebesgue measure***			
prMLP	4	0.889 [0.835,0.944]	14	0.807 [0.749,0.865]	3	0.999 [0.998,1.000]*
PRN	4	0.903 [0.852,0.955]	14	0.813 [0.756,0.870]	3	0.999 [0.998,1.000]*
PRN-Lasso	4	0.905 [0.853,0.956]	13	0.815 [0.758,0.872]	2	0.999 [0.998,1.000]*
prSVM	6	0.896 [0.842,0.949]	12	0.803 [0.745,0.861]	3	0.998 [0.997,0.999]
prGBM	7	0.881 [0.824,0.937]	9	0.791 [0.732,0.851]	2	0.997 [0.995,0.998]

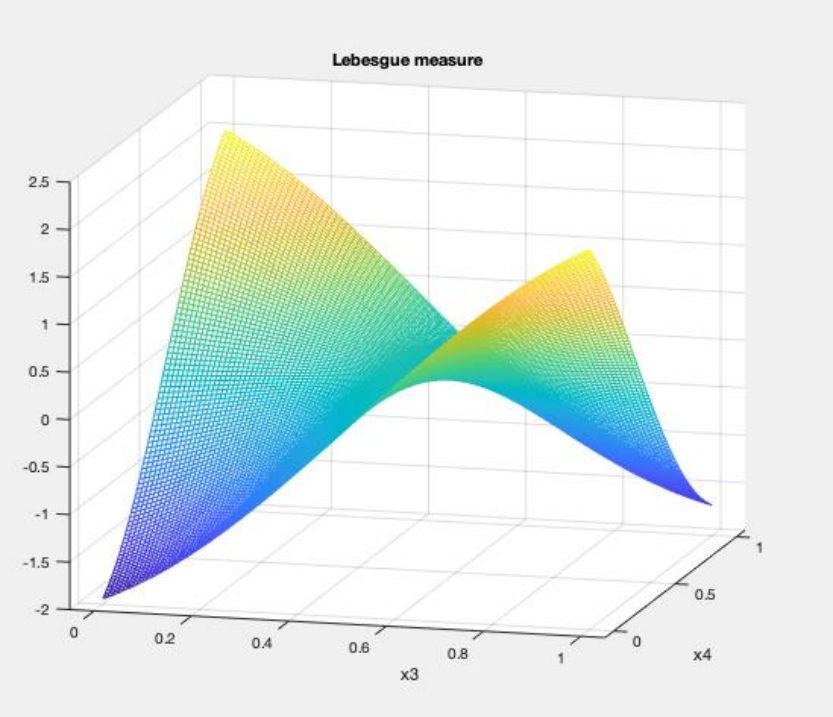
* Indicates a two-stage model selection process, explained in the text.

6.3.2 Visualisation and interpretability

The following figures compare the Dirac and Lebesgue measures for the same partial response model, the PRN.



(a)



(b)

Figure 17. (a) the two-way interaction term identified by the Dirac measure and (b) the interaction estimated with the Lebesgue measure, which is almost identical to the curve in (a). Both surfaces are the only terms in the GAM and closely correspond to the logit of the ideal XOR prediction surface. The main difference to theory is that the values at the four corners which saturate at finite values, whereas in theory they extend to infinity in both vertical directions. This, however, has little impact on the crucial region for classification which is the class boundary.

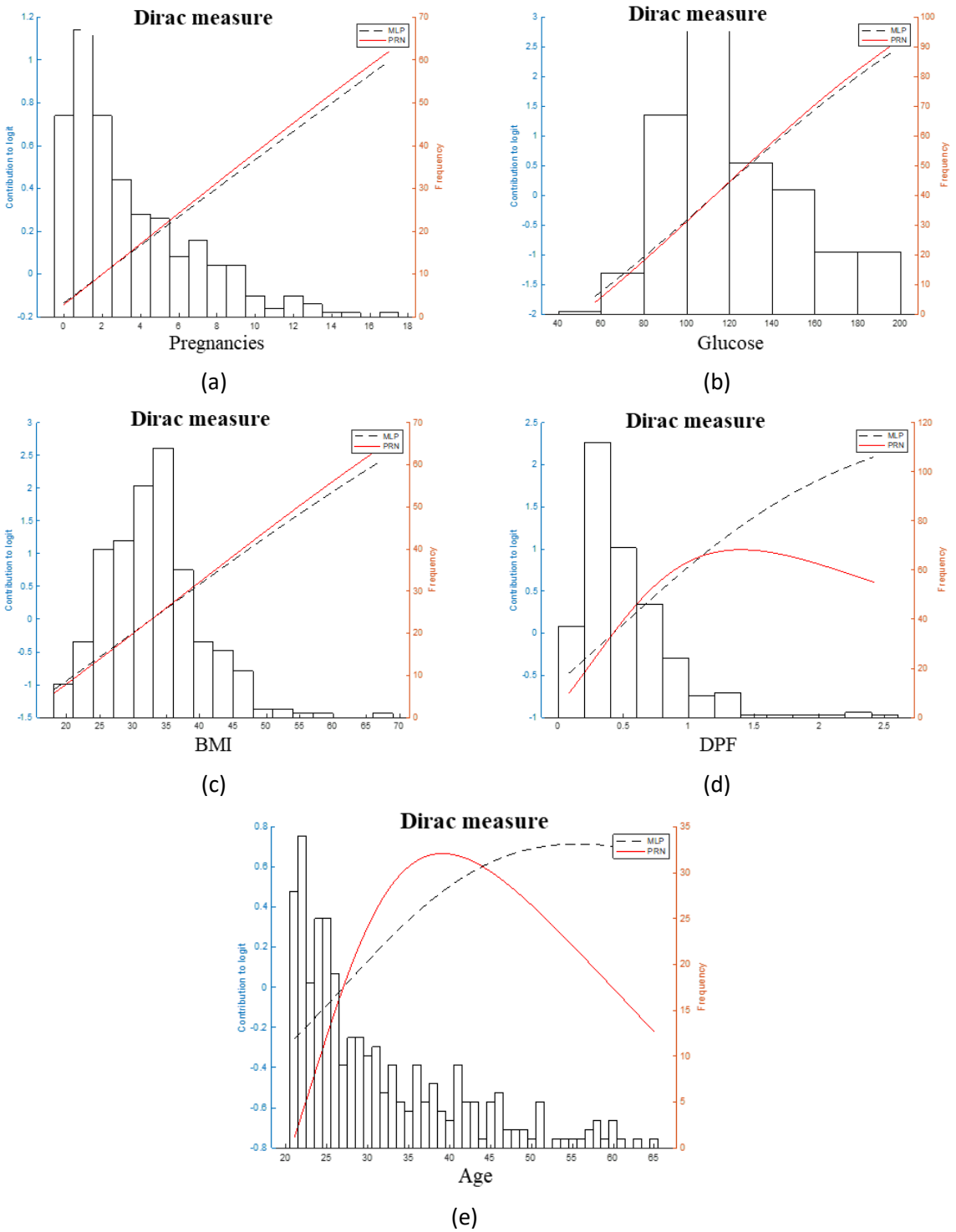


Figure 18. Contributions to the logit from partial responses to the logit (left axis) for the Diabetes data set obtained with the Dirac measure, overlapped with the histogram of the training data (right axis). The final partial responses derived at the second application gradient descent (solid lines) are shown alongside the partial responses from the original prMLP (dashed lines).

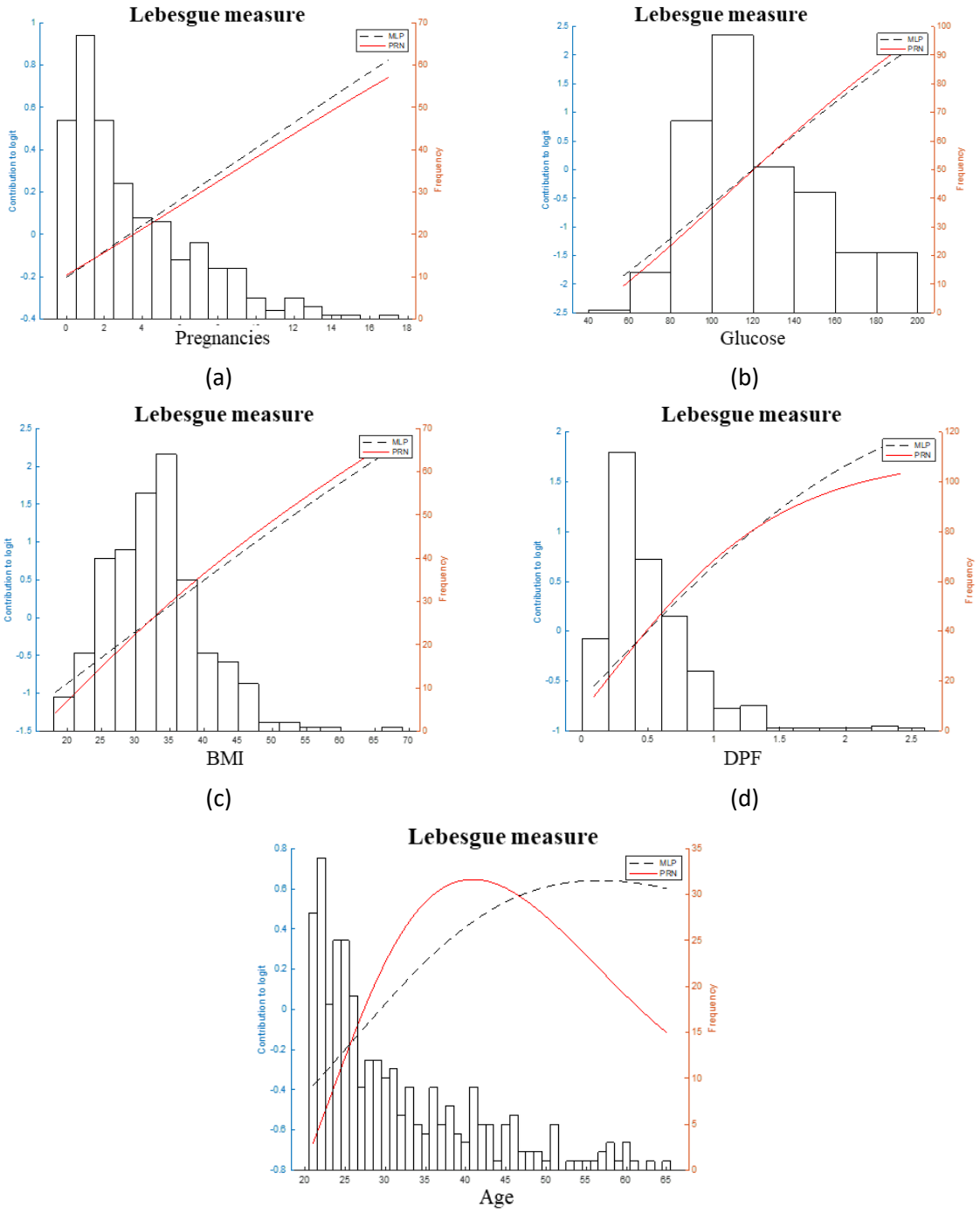


Figure 19. As for **Figure 18** with the Lebesgue measure. The component functions of the GAM are very similar for both measures. They have a similar structure and range of contributions to the logit. Despite being fitted with a generic non-linear model, the prMLP, several of the partial responses are linear. Variable "DPF" shows a saturation effect, as might be expected, while the log odds of "Age" as an independent effect peak around age 40. Note that data sparseness for higher values will result in greater uncertainty in the estimation of the partial response.

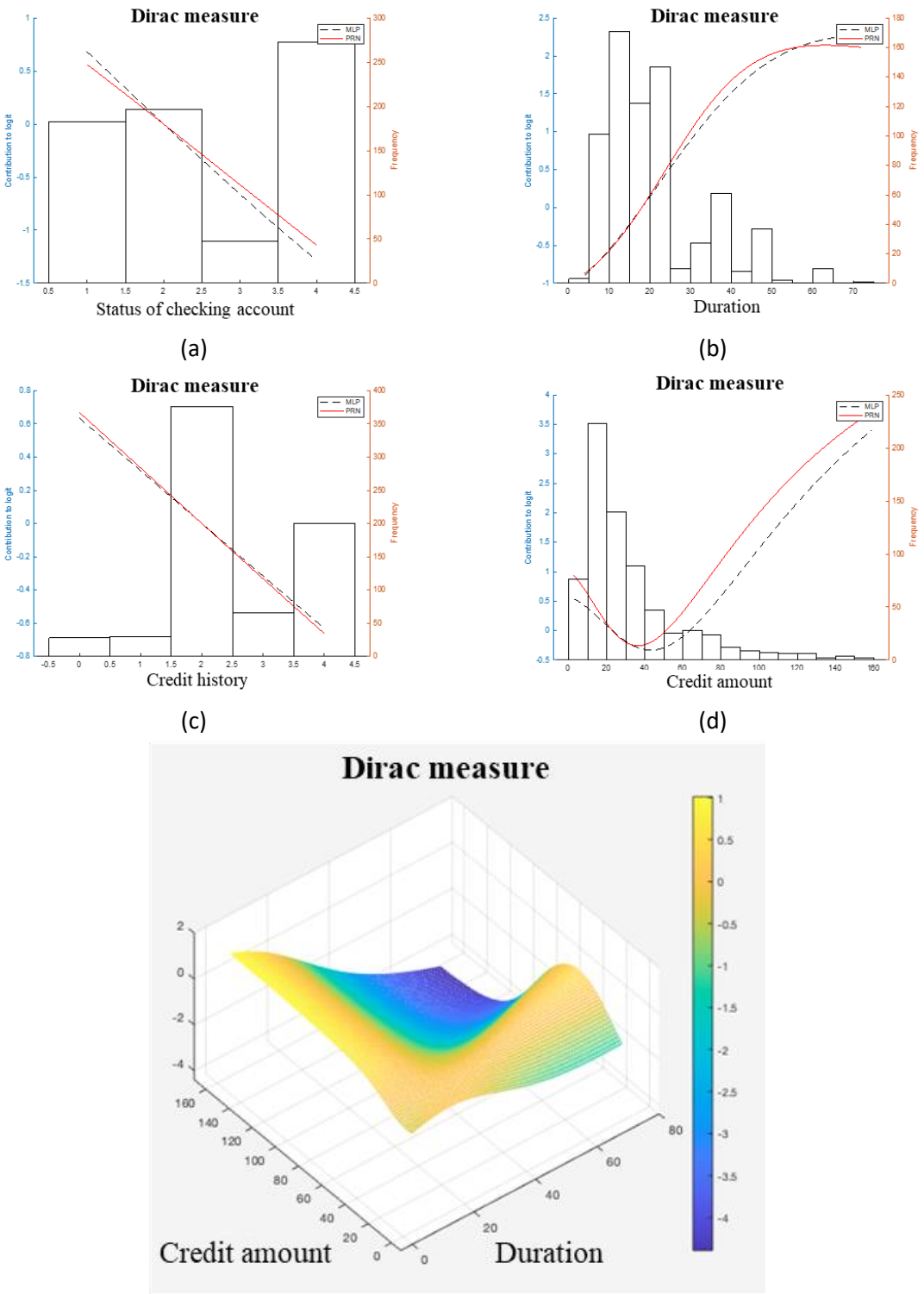
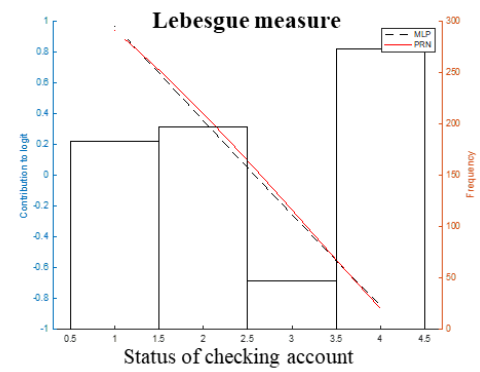
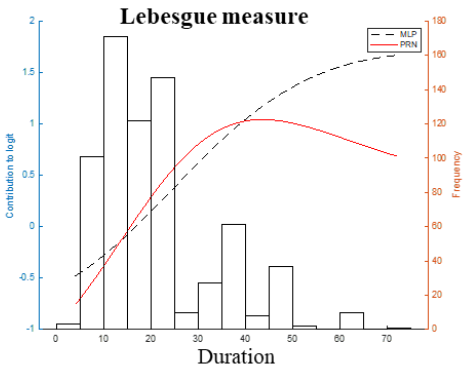


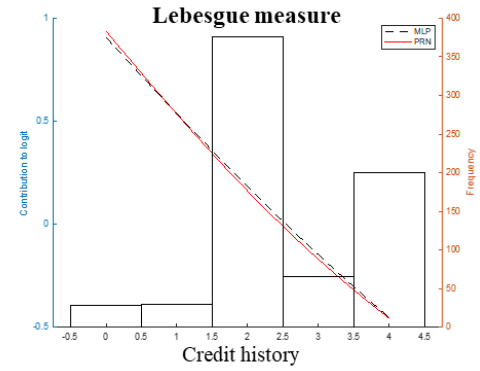
Figure 20. Partial responses for the German Credit Card data set, using the same notation as the previous figures.



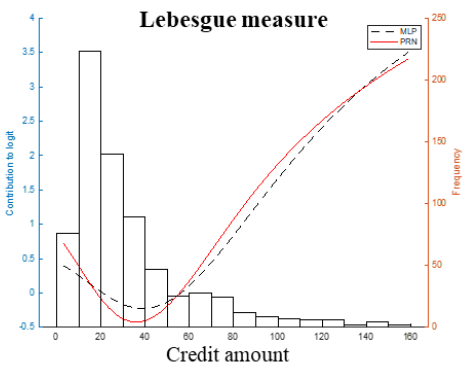
(a)



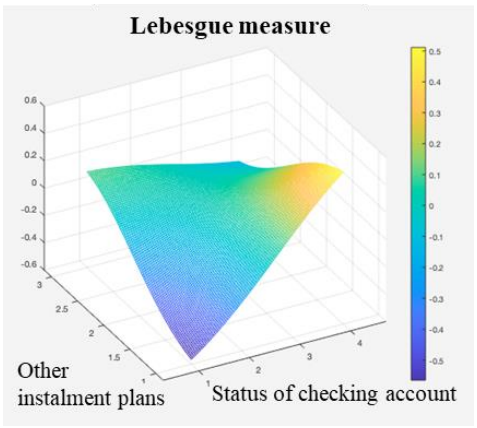
(b)



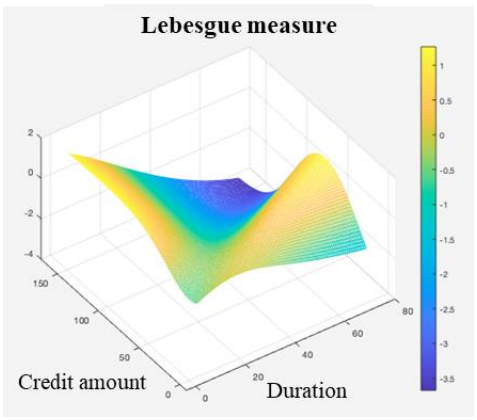
(c)



(d)



(e)



(f)

Figure 21. As for **Figure 20** with the Lebesgue measure. Despite the different nature of the two measures, they offer entirely consistent interpretations, with the only difference being the selection by the Lasso model of a second bivariate interaction term, albeit with a range in contribution to the logit that is five times smaller than for the interaction term involving “Credit amount” and “Duration”.

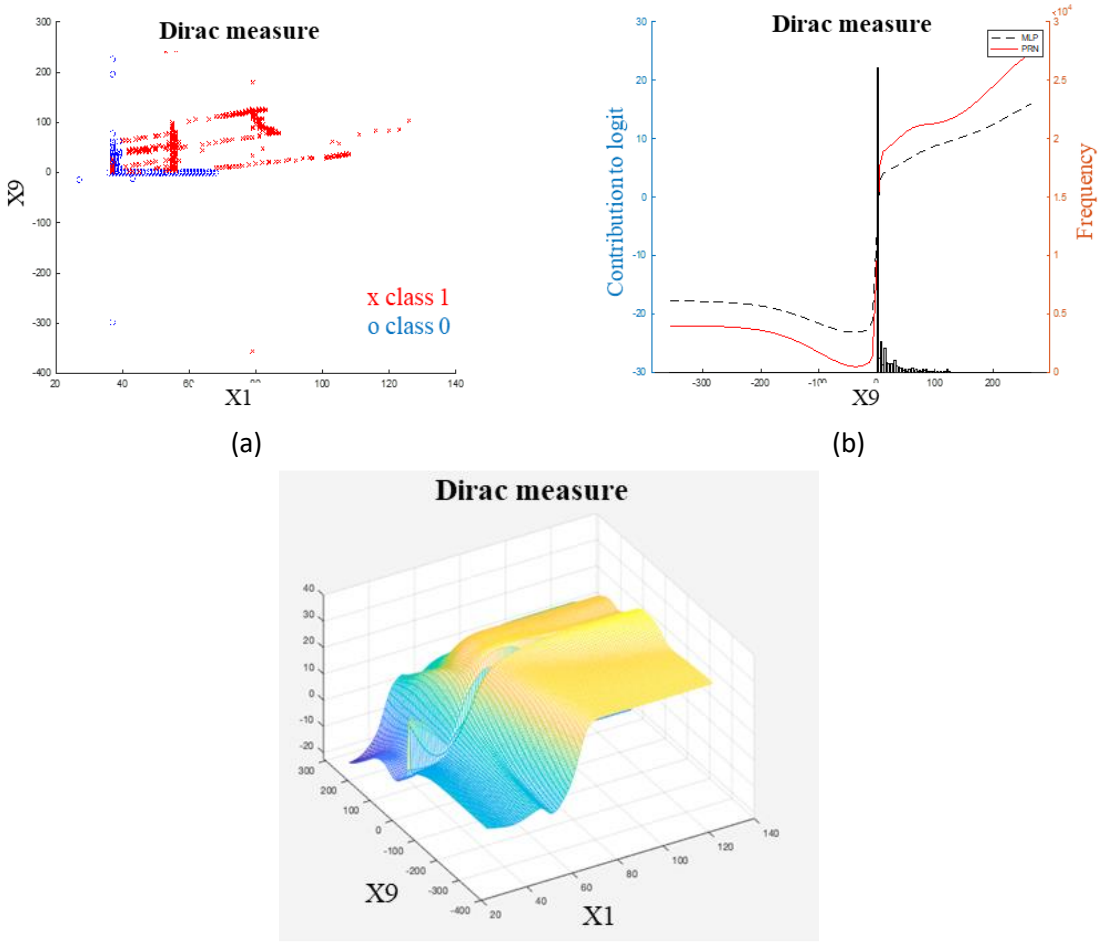


Figure 22. Nomogram of the PRN-Lasso model obtained for the Statlog Shuttle dataset using the Dirac measure with a training/test split of n=43,500 and 14,500 respectively; (a) shows the raw data for the two variables selected, which corresponds well with two partial responses in the final model, namely: (b) the main effect involving x9 and (c) two-way interaction x_1 against x_9.

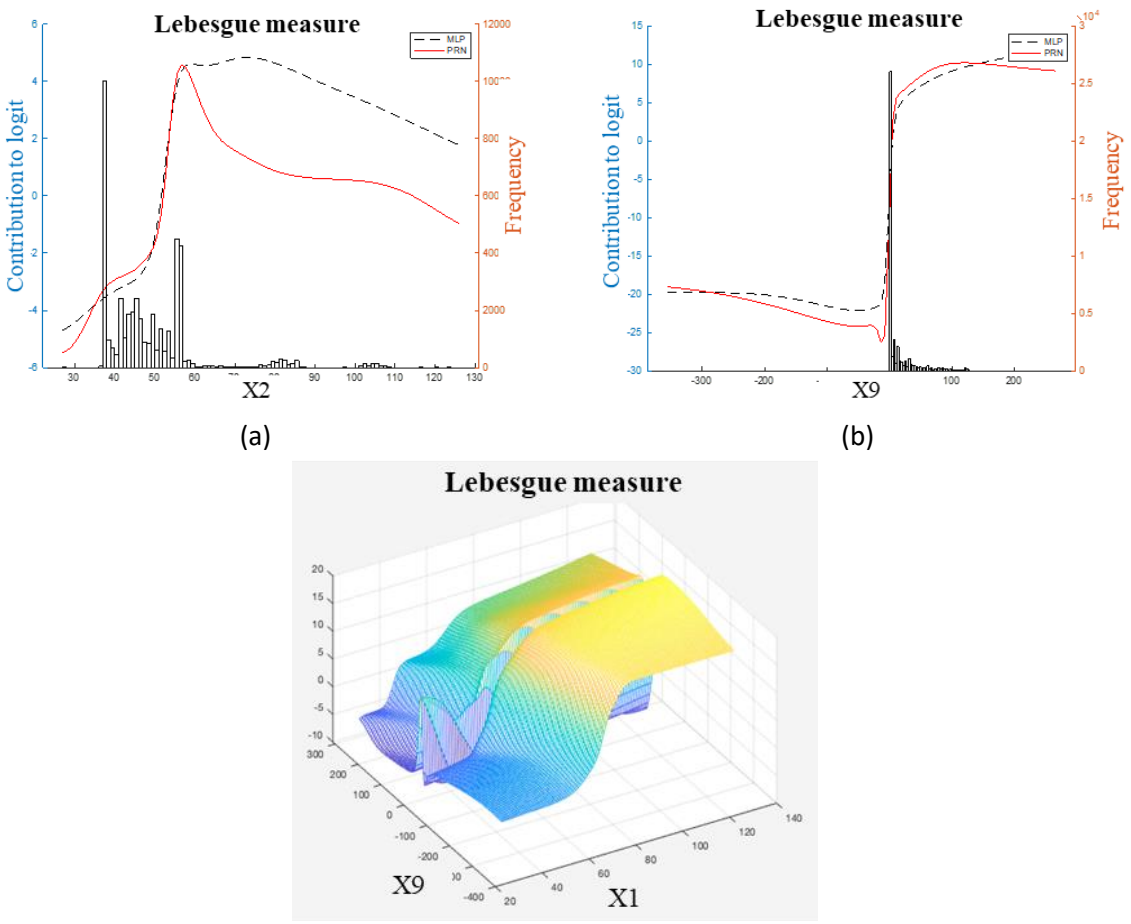


Figure 23. As for **Figure 22** with the Lebesgue measure. The same two variables were used as with the Dirac measure and similar AUC performance was achieved albeit involving an additional univariate term.

6.4 Discussion

The MLP-derived PRiSM models identified the correct ANOVA components for all data sets, with both the Dirac and Lebesgue measures. The three-way interaction term is a product of three inputs, not a pure interaction term. Therefore, its decision boundary can be approximated even in the absence of a third-order term, with three univariate partial responses sufficient to get close to the optimal prediction accuracy. Note that the predicted response in **Figure 17** for the XOR task is very close to the bilinear surface corresponding to the theoretically correct response given by multilinear algebra (Tsukimoto, 2000).

Some machine learning models can be prolific in model selection with the ANOVA decomposition, followed by the Lasso, for either measure. The models always include the relevant variables but may also suffer from overfitting. However, it is remarkable how the interpretable models frequently achieve AUC values within 1% of the optimal value.

The models that filtered out the correct number of components to model each data set all have consistent interpretations. The partial responses correspond to the data generators in the vicinity of the class boundary although the responses tend to level off away from the boundary, where the precise value of the logit is less important since the class membership probabilities are close to zero or one.

While the accuracy of all models is comparable, the PRiSM models use fewer variables and are intuitive to interpret. It is also apparent that the two different measures lead to very similar classification performances. The coefficients of the Lasso used for re-calibration are close to unity for all models.

The number of component functions in **Table 12** shows the effect of variable selection by the Lasso. The diabetes data set generates only univariate responses. However, the credit card and shuttle data sets require two-way interactions as well as univariate effects. Note that the credit card data set generates 300 partial responses to choose from. We also note when comparing the results in **Table 12** to those in **Table 3**, the difference in performance can be attributed to a difference in data splits as we have train, test and validation sets.

The GAMs seeded by the SVM and GBM are calibrated by the LASSO resulting in the prSVM and prGBM. The univariate and bivariate structure of these models can be used to define a PRN model which is a SENN with MLP components, initialised either with random weights or with univariate and bivariate modules trained to replicate each of the selected partial responses. This will replicate the PRN and, following orthogonalization, the PRN-Lasso.

The sparsity of the models and their potential for interpretation are illustrated by the partial responses of two models, the MLP-Lasso and the PRN, shown in **Figures 18-23**. These functions are derived from the training data and are always used for prediction on out-of-sample data. The corresponding component functions for the other PRiSM models have similar values, although, if derived from Random Forests, e.g. in the case of the prGBM, they are stepwise constant rather than smooth. This is shown in Walters et al. (2022) for a different data set.

Among the seven covariates in the diabetes data set, five occurred together as univariate responses in all of the random initialisations for both the prMLP and PRN-Lasso and the two measures. They are “Pregnancies”, “Glucose”, “BMI”, “DPF” and “Age”. An interaction term involving “Glucose” and “DPF” was present in three random initialisations. The set of models obtained is therefore remarkably stable. The partial responses for the recurrent univariate effects are shown in **Figures 18-19**.

The German Credit Card data set is more challenging. Out of 24 variables, six were present in all initialisations for both measures: “Duration”, “Credit history”, “Savings accounts”, “Period of employment” and the two variables labelled x_{16} and x_{17}. In the case of the Lebesgue measure, three more variables recurred in all 10 initialisations, namely “Status of checking account”, “Other instalment plans” and “Worker status”. In addition, the variable “Credit amount” featured as a univariate or a bivariate term in 8 initialisations. These ten variables were selected to obtain the models for which a selection of component functions is shown in **Figures 20-21**.

Four univariate functions for multi-valued input variables are consistently monotonically decreasing and very close to linear, suggesting that these indicators are well calibrated as independent effects on credit risk, quantifying reductions in risk with rising in-put values. “Duration” shows saturation in its contribution to risk for large values and “Credit amount” has a non-linear response with a minimum value. The bivariate responses suggest that, to optimise the overall calibration of the model, adjustments are required in addition to the main effects. This includes a risk reduction when the “Credit amount” and “Duration” are both high and a slight enhancement when either is small compared with the median value.

After mapping the structure derived with the Dirac measure from the prSVM onto the PRN-Lasso, good discrimination was achieved with an AUC of 0.812 [0.755,0.869] with just ten univariate effects. They comprised the six variables identified also by the PRN, together with the variables “Personal status”, “Property” and “Other instalment plans”.

For both the Diabetes and Credit Card data the other benchmarked algorithms SVM and GBM generally select more components than the MLP and have worse generalisation performance, as evident from **Table 12**. If the partial response models derived from each machine learning algorithm are mapped onto a SENN and further trained, then their performance becomes similar for all of the models and they select consistent input variables, although some models will include additional ones.

The scalability and power of the method can be illustrated using the Statlog Shuttle data set. The data set is challenging because all of the variables have non-normal distributions, often with highly peaked histograms. Such peaked distributions indicate that most observations fall within a narrow range of values resulting in low entropy, since the variables have low uncertainty. Also, two of the variables, x5 and x9 have a Pearson correlation of -0.875, indicating a high degree of linear dependence between them. This correlation and the low entropy are both characteristics that reflect limited variability in the data, which increases the modelling difficulty.

When applying MLP the weight decay parameters estimated for variables x2, x4 and x6 are noticeably larger than the others, indicating that these variables are less informative. They were therefore removed from the data. In the case of the Dirac measure, univariate component functions for x1 and x9 were selected by the PRN-Lasso with an AUC of 0.996 [0.994,0.998]. Selecting just these two variables as the inputs resulted in the performance listed in **Table 12**, involving a univariate effect for x9 together with the interaction between x1 and x9. The Lebesgue measure behaved similarly but for the same Lasso selection procedure, and included also a univariate effect for x1 albeit without an appreciable performance improvement.

The prSVM model selection process also followed a two-stage process, ending with the same two variables x1 and x9 for both measures, each time involved in two univariate effects and a bivariate term. Interestingly, the prGBM model converged straight away on the two-component solution involving a univariate effect for x9 and an interaction between x1 and x9 with the Dirac measure; with the Lebesgue measure it converged on two univariate effects.

6.5 Conclusion

We present a novel variation on our original partial response methodology by replacing the Dirac measure with the Lebesgue measure in the computation of partial responses. While the Dirac measure serves as an effective approximation of a variable's contribution across its entire domain, the Lebesgue measure provides a more rigorous and mathematically grounded assessment of the "true" contribution values.

By incorporating the Lebesgue measure, our revised partial response models continue to uphold the predictive accuracy of their respective black box counterparts. Furthermore, these models remain parsimonious, retaining the desirable characteristic of comprising a minimal set of explanatory components, consistent with the structure observed in the Dirac-based models.

A comparative analysis between the Dirac and Lebesgue formulations reveals minimal differences in both predictive performance and interpretability. The Dirac-based partial responses, although approximate in nature, closely mirror the outputs derived from the more analytically precise Lebesgue-based method. This proximity suggests that the Dirac measure offers a highly practical and computationally efficient alternative, particularly in scenarios where computational resources are constrained or rapid interpretability is of paramount importance.

In summary, the adoption of the Lebesgue measure refines the theoretical foundations of our partial response methodology, while the Dirac measure remains a viable and resource-efficient approximation with negligible trade-offs in model fidelity or interpretive clarity.

7 Bootstrapping

7.1 Introduction

A key challenge in the development of reliable data-driven systems is increasing user trust in partial responses, particularly in contexts where full information may be unavailable or computationally infeasible. To address this, it is essential to ensure that partial responses exhibit robustness to variations in the underlying data. Moreover, users must be presented with clear evidence regarding the stability of these responses across the relevant domain of the variable in question. These two criteria, consistency and stability, constitute central desiderata as outlined in Section 1.

To fulfil these requirements, we augment our methodological framework by integrating a bootstrapping approach. Bootstrapping enables us to empirically estimate the variability and reliability of partial responses by resampling from the observed data distribution. This statistical technique allows us to quantify the degree of uncertainty inherent in the partial responses and to communicate this uncertainty effectively to the user, bolstering confidence in the outputs provided.

7.2 Advantages and Limitations

The principal advantage of employing bootstrapping lies in the increased robustness and reliability of the resulting estimates. By generating a large number of resampled datasets, bootstrapping enables the empirical approximation of the distribution of partial responses, thereby allowing for more confident inferences about their stability. In particular, when a sufficiently large number of bootstrap iterations is employed, the outcomes tend to exhibit a high degree of consistency. This rigidity enhances our ability to characterise the 'true' partial responses with a greater level of statistical assurance.

However, the most significant limitation associated with bootstrapping is its computational intensity. Generating and processing a large number of resampled datasets can be time-consuming, particularly in high-dimensional settings or when working with complex models. As such, it is necessary to carefully evaluate the trade-off between the computational cost and the benefits conferred by the increased precision and stability of the estimates. In practical applications, this cost–benefit analysis will inform the decision as to whether the application of bootstrapping is justified in light of available resources and time constraints.

7.3 Results

Utilising the PIMA Indians dataset (Ripley, 2007; Smith et al., 1988), the data is split 70/30 into training and test sets; 100 bootstrapped samples are taken, with replacement, from the training set to the same number of observations as said training set. 100 hyperparameter sets are tested on the training set for the hyperparameters C $[10^{-2}, 10^{7}]$ and γ $[2^{-2}, 2^{7}]$. These values were chosen in order to avoid choosing a hyperparameter set that is too linear, as a non-linear model may not be required if the hyperparameters are linear. 10-fold cross-validation is performed for each hyperparameter set, in which 90% of the training set is used to train an SVM model with a hyperparameter set and 10% used to test, and the mean AUC of all 10 folds is taken for each set. The set with the highest mean AUC is chosen as the hyperparameter set for that bootstrapped sample. It should be acknowledged that computing hyperparameter tuning for each bootstrapped sample comes at a large computational cost, but inevitably leads to better model configuration and robustness.

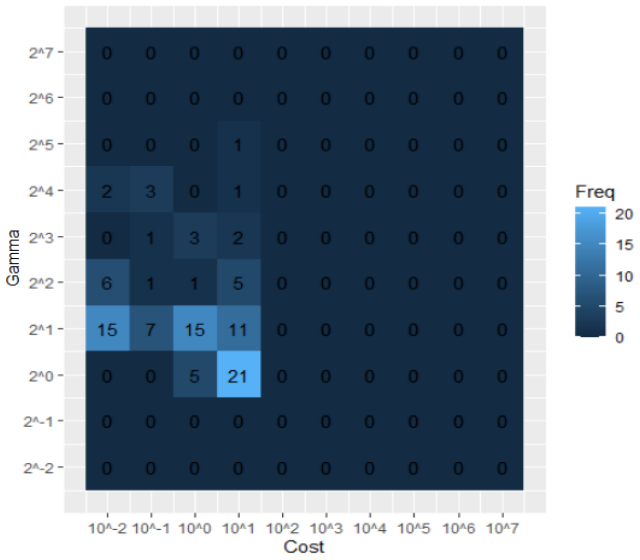


Figure 24. The count of chosen hyperparameter sets for each bootstrap.

Figure 24 presents the frequency distribution of selected hyperparameter configurations across all bootstrap iterations. The results reveal a clear concentration of selected values within a relatively constrained region of the hyperparameter space. Specifically, approximately 75% of the chosen hyperparameter sets featured a γ value no greater than 2^1 and a C value not exceeding 10^1. This clustering suggests a degree of stability in the hyperparameter selection process.

The most frequently selected hyperparameter set ($\gamma = 2^0, C = 10^1$) was subsequently adopted as the fixed configuration for subsequent analysis. Its consistent selection across bootstrap iterations provides empirical support for its robustness and suitability within the modelling framework.

An additional 100 bootstrapped samples were generated, with each sample used to train a black box SVM model alongside a corresponding partial response model. For every bootstrap iteration, both the partial response values and the AUC performance metrics of the resulting models were recorded.

To estimate the overall model performance, the mean AUC across the 100 bootstrap replicates was computed. Similarly, the average partial response values for each variable were aggregated across bootstraps and visualised with 95% confidence interval error bars. These intervals provide an indication of the variability and reliability of the estimated partial effects, thereby supporting a more nuanced interpretation of the model's behaviour across the input space. The 95% confidence interval for each point in each variable's range is calculated using the formula:

$$95\%\ CI = \bar{x} \pm 1.96\frac{s}{\sqrt{n}} \qquad \textbf{Equation 37}$$

Where $\bar{x}$ is the mean AUC of the bootstraps, s is the sample standard deviation and n is the number of bootstraps.

7.3.1 Classification Performance

Table 13. The average training and test AUC for the 100 bootstraps with the SVM and prSVM, as well as a 95% confidence interval.

Model	Training AUC (CI)	Test AUC (CI)
SVM	1 (1, 1)	0.7466 (0.7421, 0.7511)
prSVM	0.9340 (0.9313, 0.9367)	0.7707 (0.7670, 0.7743)

7.3.2 Visualisation and interpretability

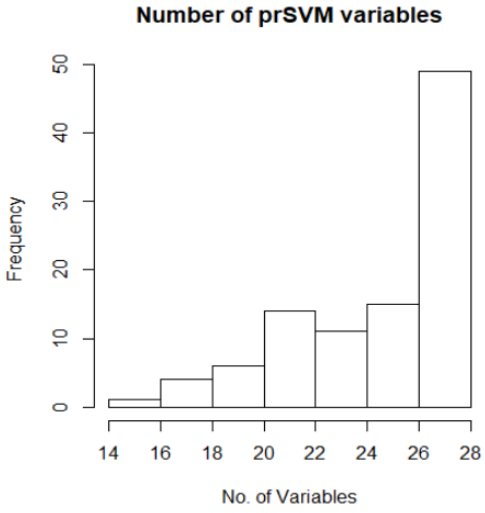


Figure 25. A histogram of the number of variables selected by the Logistic Regression Lasso in the 100 prSVM models.

Figure 25 displays a histogram illustrating the distribution of the number of variables selected by the Lasso-regularised logistic regression models applied to the 100 prSVM bootstrap instances. Each model incorporated univariate and bivariate partial response features, totalling 28 candidate variables.

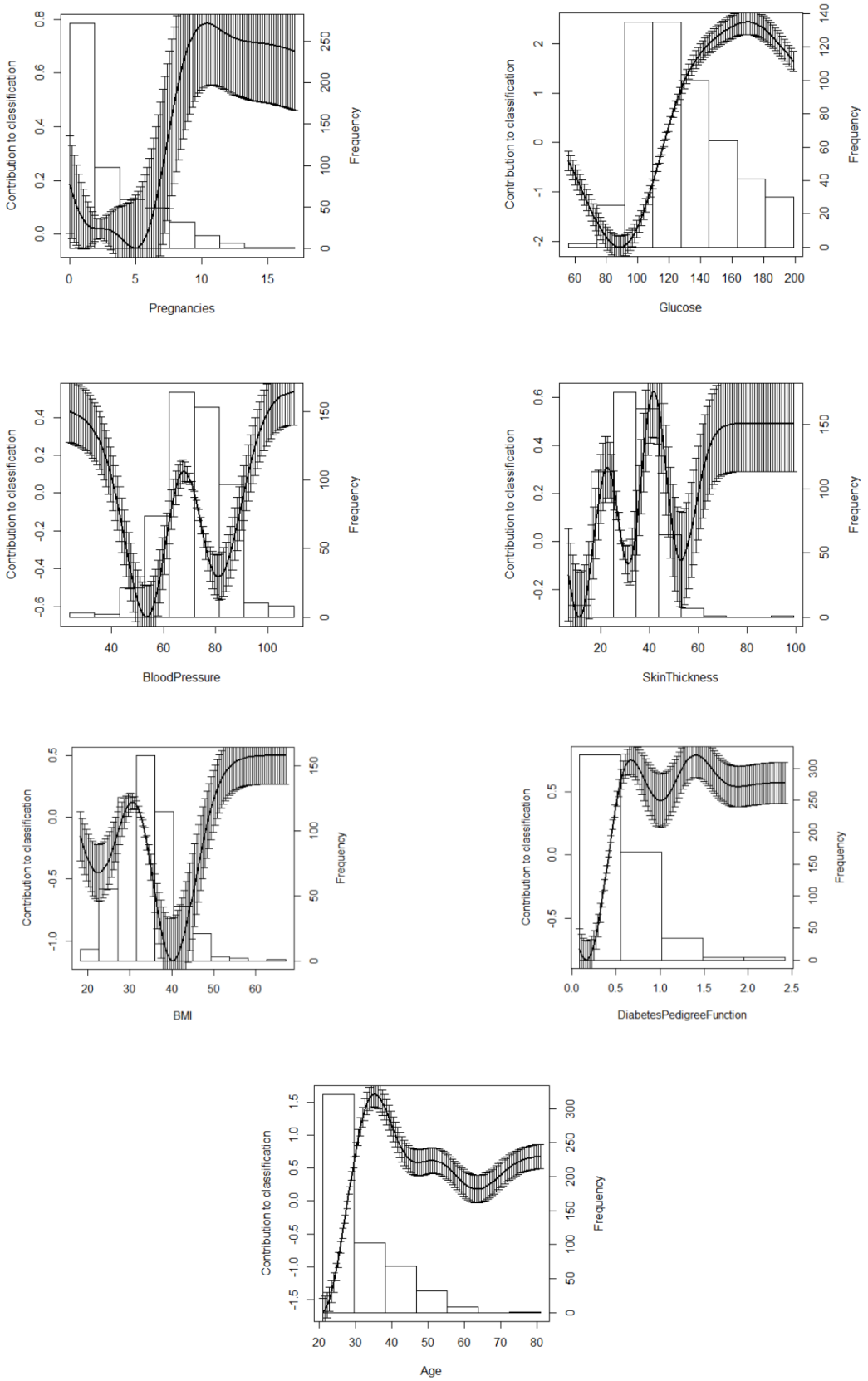


Figure 26. Average univariate partial responses for each variable, with error bars derived from the 100 bootstraps.

7.4 Discussion

The disparity observed between training and test performance metrics in **Table 13** provides clear evidence of substantial overfitting within the SVM model. This overfitting is subsequently inherited by the prSVM, as the latter is derived directly from the former. The inflated performance on the training set, coupled with a notable drop on the test set, underscores the SVM model's limited generalisability.

Interestingly, the prSVM model demonstrates superior performance on the out-of-sample test set compared to the original SVM. This finding is further reinforced by the non-overlapping confidence intervals associated with their respective test AUCs, suggesting that the improvement is statistically significant. The prSVM's enhanced generalisation may be attributed to the regularising effect of the partial response modelling process, which potentially mitigates some of the overfitting present in the base SVM.

The histogram in **Figure 25** reveals that the majority of models selected a large proportion of partial response features, indicating a consistent reliance on a broad set of partial responses across bootstrap iterations. Notably, none of the models selected fewer than 14 variables, and most utilised nearly all available features. This suggests that the partial responses capture diverse and complementary information, contributing meaningfully to the predictive performance of the prSVM framework. The consistency in variable selection also points to a degree of stability in the structure of the underlying relationships being modelled.

An analysis of **Figure 26** reveals that the error bars serve as a valuable indicator of the reliability of the partial response estimates across the range of each variable. Specifically, the error bars are narrower in regions where the dataset exhibits high observational density, suggesting greater confidence in the estimated contributions of those variables to the model's predictions. Notably, these high-density regions often coincide with the anchor points of each variable, which are set at their respective median values. This alignment reinforces the interpretability of the model in the central range of the data distribution.

Conversely, in areas characterised by lower data frequency, typically found at the upper and lower extremes of each variable, the error bars widen considerably. This increased uncertainty reflects the model's limited exposure to examples in these regions during training, thereby reducing the reliability of the corresponding partial response estimates. For end-users, this visual representation of uncertainty serves as a cautionary guide, highlighting the need for prudent interpretation of model behaviour in data-sparse regions. By explicitly incorporating and visualising these confidence intervals, the partial response framework enhances transparency and fosters a more nuanced understanding of model predictions.

7.5 Conclusion

Bootstrapping serves as a powerful statistical technique that enhances the robustness and credibility of our partial response explanations. By repeatedly resampling the training data with replacement, bootstrapping enables the estimation of variability and uncertainty associated with each partial response, thereby producing more reliable and generalisable insights into a variable's influence across its entire range.

This approach contributes to improved classification performance by stabilising model estimates, ultimately resulting in narrower confidence intervals and reduced variance. Consequently, users are provided with more trustworthy and consistent interpretive outputs. The enhanced partial response plots, augmented with bootstrapped error bars, offer an intuitive and user-friendly visualisation. These

error bars act as a critical interpretive guide, signalling regions of the input space where data is sparse and thus where the reliability of the explanations may be diminished.

Overall, the integration of bootstrapping not only strengthens the empirical foundation of the partial response methodology but also supports transparency and interpretability, key desiderata in the application of machine learning to high-stakes domains.

8 Summary and Conclusion

8.1 Summary

This study extended the partial response methodology to a range of non-linear black box models. In chapter 4, we demonstrated that it is indeed feasible to provide meaningful explanations for the predictions generated by a black box Support Vector Machine model, both in synthetic and real-world datasets. The resulting interpretable models, derived through the application of partial responses, successfully preserved the predictive performance of the original SVM classifiers. Importantly, they achieved this while offering full transparency and interpretability. To further enhance parsimony, we employed Logistic Regression with Lasso regularisation, which systematically eliminates redundant or non-informative partial responses. This resulted in a sparse, minimal model that retained only the most salient univariate and bivariate effects, thereby aligning with one of our primary desiderata, model simplicity without compromising accuracy. The partial response visualisations offered an intuitive and accessible means of interpreting the influence of individual variables. These plots clearly illustrated how each predictor contributes to the model's output across the entire range of its values, thereby enabling a comprehensive understanding of variable behaviour. Such visual transparency is particularly valuable in applied contexts, where stakeholders may require interpretable explanations for predictive outcomes in order to support informed decision-making.

In chapter 5, we showed that it is possible to open any black box model, including pre-trained models, in cases where significant noise is present, without losing much predictive power, if any, but making the model transparent to the non-linearities in the data. The performances of the resulting PR models compared favourably with state-of-the-art sparse models from the statistical literature, SAM, and from machine learning, the EBM. The derived Partial Responses are consistent across the range of models and have plausible clinical interpretations. The interpretability of the model by end-users is at the level of nomograms (Van Belle et al., 2016)(Van Belle et al., 2016), which are familiar to clinicians as graphical implementations of logistic regression. Partial Response models are interpretable in the same way, except that the score for each variable is read from what we call the partial response plot, where the height of the plot directly measures the contribution to the logit, which is the nomogram score for that variable.

In chapter 6, we presented a novel variation on our original partial response methodology by replacing the Dirac measure with the Lebesgue measure in the computation of partial responses. While the Dirac measure serves as an effective approximation of a variable's contribution across its entire domain, the Lebesgue measure provides a more rigorous and mathematically grounded assessment of the "true" contribution values. By incorporating the Lebesgue measure, our revised partial response models continued to uphold the predictive accuracy of their respective black box counterparts. Furthermore, these models remain parsimonious, retaining the desirable characteristic of comprising a minimal set of explanatory components, consistent with the structure observed in the Dirac-based models. A comparative analysis between the Dirac and Lebesgue formulations revealed minimal differences in both predictive performance and interpretability. The Dirac-based partial responses, although approximate in nature, closely mirror the outputs derived from the more analytically precise Lebesgue-based method. This proximity suggested that the Dirac measure offers a highly practical and computationally efficient alternative, particularly in scenarios where computational resources are constrained or rapid interpretability is of paramount importance. In summary, the adoption of the Lebesgue measure refined the theoretical foundations of our partial response methodology, while the Dirac measure remained a viable and resource-efficient approximation with negligible trade-offs in model fidelity or interpretive clarity.

In chapter 7, bootstrapping was applied as a powerful statistical technique to enhance the robustness and credibility of our partial response explanations. By repeatedly resampling the training data with replacement, bootstrapping enabled the estimation of variability and uncertainty associated with each partial response, thereby producing more reliable and generalisable insights into a variable's influence across its entire range. This approach contributed to improved classification performance by stabilising model estimates, ultimately resulting in narrower confidence intervals and reduced variance. Consequently, users were provided with more trustworthy and consistent interpretive outputs. The enhanced partial response plots, augmented with bootstrapped error bars, offered an intuitive and user-friendly visualisation. These error bars acted as a critical interpretive guide, signalling regions of the input space where data is sparse and thus where the reliability of the explanations may be diminished. Overall, the integration of bootstrapping not only strengthens the empirical foundation of the partial response methodology but also supports transparency and interpretability, key desiderata in the application of machine learning to high-stakes domains.

The framework enables the construction of intuitive univariate and bivariate visualisations derived from the partial response functions. These visual tools effectively communicate how individual variables, or pairs of variables, influence predictions across their entire respective domains. By providing a detailed, range-wide view of the variables, these plots support more comprehensive insights into model behaviour and facilitate informed decision-making.

This research makes several significant contributions to the field of interpretable machine learning. Firstly, it introduces a generalisable, model-agnostic framework that bridges the gap between predictive performance and interpretability, two objectives often seen in tension. Unlike many interpretability techniques which are either specific to certain model architectures or limited in scope, the proposed partial response methodology applies broadly across different black box models without sacrificing accuracy. Secondly, by enabling clear visualisation of univariate and bivariate effects, the framework supports both diagnostic understanding and practical decision-making. The incorporation of bootstrapped uncertainty estimates further enhances the credibility and trustworthiness of the explanations—an essential quality for deployment in high-stakes domains such as healthcare or finance. Collectively, these contributions advance the goal of transparent and accountable AI, paving the way for interpretable solutions in increasingly complex machine learning systems.

8.2 Future Directions

There are numerous pathways through which the partial response methodology can be further adapted and extended. One such direction involves its application to alternative predictive modelling tasks, such as survival analysis or regression problems, where explaining variable contributions remains equally critical. A particularly promising continuation within the domain of classification is to explore its integration with deep learning models applied to image and time series data. These data types are characterised by high dimensionality and complex temporal or spatial structures, which pose additional challenges to interpretability.

Such extensions are especially pertinent in healthcare and medical domains, where the ability to provide transparent and trustworthy explanations for model predictions is of paramount importance. As deep learning continues to gain traction in clinical decision support systems, there is a growing demand for interpretability methods that can bridge the gap between predictive accuracy and human understanding.

9 References

Abe, N., Zadrozny, B., & Langford, J. (2006). Outlier detection by active learning. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *2006*, 504–509. https://doi.org/10.1145/1150402.1150459

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., & Hinton, G. (2020). *Neural Additive Models: Interpretable Machine Learning with Neural Nets*. http://arxiv.org/abs/2004.13912

Alvarez-Melis, D., & Jaakkola, T. S. (2018). Towards robust interpretability with self-explaining neural networks. *ArXiv*, (NeurIPS).

Association, A. D. (2014). Diagnosis and classification of diabetes mellitus. *Diabetes Care*, *37*(SUPPL.1), 81–90. https://doi.org/10.2337/dc14-S081

Badawi, O., Liu, X., Hassan, E., Amelung, P. J., & Swami, S. (2018). Evaluation of ICU Risk Models Adapted for Use as Continuous Markers of Severity of Illness Throughout the ICU Stay*. *Critical Care Medicine*, *46*(3). https://journals.lww.com/ccmjournal/fulltext/2018/03000/evaluation_of_icu_risk_models_adapted_for_use_as.3.aspx

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K. R. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research*, *11*, 1803–1831.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*, 281–305.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 144–152. https://doi.org/10.1145/130385.130401

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Cretikos, M. A., Bellomo, R., Hillman, K., Chen, J., Finfer, S., & Flabouris, A. (2008). Respiratory rate: the neglected vital sign. *Medical Journal of Australia*, *188*(11), 657–659. https://doi.org/10.5694/j.1326-5377.2008.tb01825.x

Cristianini, N., & Schölkopf, B. (2002). Support vector machines and kernel methods: The new generation of learning machines. *AI Magazine*, *23*(3), 31–41.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves : A Nonparametric Approach. *Biometrics*, *44*(3), 837–845.

Doshi-Velez, F., & Kim, B. (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. (Ml), 1–13. http://arxiv.org/abs/1702.08608

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). Visualizing higher-layer features of a deep network. *Bernoulli*, (1341), 1–13. http://igva2012.wikispaces.asu.edu/file/view/Erhan+2009+Visualizing+higher+layer+features+of+a+deep+network.pdf

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Https://Doi.Org/10.1214/Aos/1013203451*, *29*(5), 1189–1232. https://doi.org/10.1214/AOS/1013203451

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, *24*(1), 44–65. https://doi.org/10.1080/10618600.2014.907095

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets Ian. *ArXiv*.

Guenther, N., & Schonlau, M. (2016). Support vector machines. *Stata Journal*, *16*(4), 917–937. https://doi.org/10.1177/1536867x1601600407

Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., & Galstyan, A. (2019). Multitask learning and benchmarking with clinical time series data. *Scientific Data*, *6*(1), 1–18. https://doi.org/10.1038/s41597-019-0103-9

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.

Hosmer, D. W., & Lemeshow, S. (1980). Goodness of Fit Tests for the Multiple Logistic Regression Model. *Communications in Statistics - Theory and Methods*, *9*(10), 1043–1069. https://doi.org/https://doi.org/10.1080/03610928008827941

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). John Wiley & Sons, Inc.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A Practical Guide to Support Vector Classification. *BJU International*.

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). Data Descriptor: MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3:160035*, 1–9.

Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Support Vector Machines in R. *Journal of Statistical Software*, *15*(9), 1–28.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*.

Kelly, M., Longjohn, R., & Nottingham, K. (n.d.). *The UCI Machine Learning Repository*. Retrieved 1 January 2022, from https://archive.ics.uci.edu

Knaus, W. A., Draper, E. A., Wagner, D. P., & Zimmerman, J. E. (1985). APACHE II: A severity of disease classification system. *Critical Care Medicine*, *13*(10). https://journals.lww.com/ccmjournal/fulltext/1985/10000/apache_ii__a_severity_of_disease_classification.9.aspx

Knowler, W. C., Pettitt, D. J., Savage, P. J., & Bennett, P. H. (1981). Diabetes incidence in Pima indians: contributions of obesity and parental diabetes. *American Journal of Epidemiology*, *113*(2), 144–156. https://doi.org/10.1093/oxfordjournals.aje.a113079

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *IJCAI International Joint Conference on Artificial Intelligence*, *2*(June), 1137–1143.

Kontijevskis, A., Wikberg, J. E. S., & Komorowski, J. (2007). Computational Proteomics Analysis of HIV-1 Protease Interactome. *Proteins: Structure, Function, and Bioinformatics*, *68*(1), 305–312.

Kovalev, M. S., Utkin, L. V., & Kasimov, E. M. (2020). SurvLIME: A method for explaining machine learning survival models. *Knowledge-Based Systems*, *203*, 106164. https://doi.org/10.1016/j.knosys.2020.106164

Liang, Y., Li, S., Yan, C., Li, M., & Jiang, C. (2021). Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing*, *419*, 168–182. https://doi.org/10.1016/j.neucom.2020.08.011

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, *23*(1), 1–45. https://doi.org/10.3390/e23010018

Lisboa, P. J.G., Ortega-Martorell, S., Jayabalan, M., & Olier, I. (2020). Efficient Estimation of General Additive Neural Networks: A Case Study for CTG Data. *Communications in Computer and Information Science*, *1323*, 432–446. https://doi.org/10.1007/978-3-030-65965-3_29

Lisboa, Paulo J.G., Ortega-Martorell, S., & Olier, I. (2020). Explaining the Neural Network: A Case Study to Model the Incidence of Cervical Cancer. *Communications in Computer and Information Science*, *1237 CCIS*, 585–598. https://doi.org/10.1007/978-3-030-50146-4_43

Lou, Y., Caruana, R., & Gehrke, J. (2012). Intelligible models for classification and regression. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 150–158. https://doi.org/10.1145/2339530.2339556

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *2017-Decem*, 4766–4775. https://github.com/slundberg/shap

Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., & Carvalho, A. C. P. L. F. (2015). To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. *Proceedings of the International Joint Conference on Neural Networks*, *2015-Septe*. https://doi.org/10.1109/IJCNN.2015.7280644

Marcinkevičs, R., & Vogt, J. E. (2020). *Interpretability and Explainability: A Machine Learning Zoo Mini-tour*. http://arxiv.org/abs/2012.01805

Molnar, C. (2022). *Interpretable Machine Learning* (2nd ed.). https://christophm.github.io/interpretable-ml-book/

Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). *InterpretML: A Unified Framework for Machine Learning Interpretability*. 1–8. http://arxiv.org/abs/1909.09223

Peres Bota, D., Lopes Ferreira, F., Mélot, C., & Vincent, J. L. (2004). Body temperature alterations in the critically ill. *Intensive Care Medicine*, *30*(5), 811–816. https://doi.org/10.1007/s00134-004-2166-z

Platt, J. C. (1999). *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, *2018-Decem*(Section 4), 6638–6648.

Ravikumar, P., Lafferty, J., Liu, H., & Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *71*(5), 1009–1030. https://doi.org/10.1111/j.1467-9868.2009.00718.x

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). 'Why should i trust you?' Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Richard, A., Mayag, B., Talbot, F., Tsoukias, A., & Meinard, Y. (2020). Transparency of Classification Systems for Clinical Decision Support. *Communications in Computer and Information Science*, *1239 CCIS*, 99–113. https://doi.org/10.1007/978-3-030-50153-2_8

Ripley, B. D. (2007). Pattern Recognition via Neural Networks. *Cambridge University Press*, (1987).

Rögnvaldsson, T., Etchells, T. A., You, L., Garwicz, D., Jarman, I., & Lisboa, P. J. G. (2009). How to find simple and accurate rules for viral protease cleavage specificities. *BMC Bioinformatics*, *10*, 1–17. https://doi.org/10.1186/1471-2105-10-149

Rögnvaldsson, T., You, L., & Garwicz, D. (2007). Bioinformatic approaches for modeling the substrate specificity of HIV-1 protease: an overview. *Expert Review of Molecular Diagnostics*, *7*(4), 435–451.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Ruiz, H., Etchells, T. A., Jarman, I. H., Martín, J. D., & Lisboa, P. J. G. (2013). A principled approach to network-based classification and data representation. *Neurocomputing*, *112*, 79–91. https://doi.org/10.1016/j.neucom.2012.12.050

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature*, *323*(9), 533–536.

Sangroya, A., Rastogi, M., Anantaram, C., & Vig, L. (2020). Guided-LIME: Structured sampling based hybrid approach towards explaining blackbox machine learning models. *CEUR Workshop Proceedings*, *2699*.

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, *5*(2), 197–227. https://doi.org/10.1023/A:1022648800760

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, 1–8.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. *Proceedings - Annual Symposium on Computer Applications in Medical Care*, 261–265.

Teasdale, G., & Jennett, B. (1974). ASSESSMENT OF COMA AND IMPAIRED CONSCIOUSNESS. A Practical Scale. *Lancet*, *304*(7872), 81–84.

Tsukimoto, H. (2000). Extracting rules from trained neural networks. *IEEE Transactions on Neural Networks*, *11*(2), 377–389.

Van Belle, V., Van Calster, B., Van Huffel, S., Suykens, J. A. K., & Lisboa, P. (2016). Explaining support vector machines: A color based nomogram. *PLoS ONE*, *11*(10), 1–33. https://doi.org/10.1371/journal.pone.0164568

Vapnik, V. (1998). *Statistical learning theory*.

Visani, G., Bagli, E., & Chesani, F. (2020). OptiLIME: Optimized lime explanations for diagnostic computer algorithms. *CEUR Workshop Proceedings*, *2699*.

Walters, B., Ortega-Martorell, S., Olier, I., & Lisboa, P. (2021). *The partial response SVM*. (October), 575–580. https://doi.org/10.14428/esann/2021.es2021-36

Walters, B., Ortega-Martorell, S., Olier, I., & Lisboa, P. J. G. (2022). Towards interpretable machine learning for clinical decision support. *Proceedings of the International Joint Conference on Neural Networks*, *2022-July*. https://doi.org/10.1109/IJCNN55064.2022.9892114

Walters, B., Ortega-Martorell, S., Olier, I., & Lisboa, P. J. G. (2023). How to Open a Black Box Classifier for Tabular Data. *Algorithms*, *16*(4). https://doi.org/10.3390/a16040181

Williams, C. K. I., & Seeger, M. (2000). Using the Nyström Method to Speed Up Kernel Machines. *Advances in Neural Information Processing Systems*, *13*.

Young, P., Saxena, M., Bellomo, R., Freebairn, R., Hammond, N., van Haren, F., Holliday, M., Henderson, S., Mackle, D., McArthur, C., McGuinness, S., Myburgh, J., Weatherall, M., Webb, S., & Beasley, R. (2015). Acetaminophen for Fever in Critically Ill Patients with Suspected Infection.

New England Journal of Medicine, *373*(23), 2215–2224. https://doi.org/10.1056/nejmoa1508375

Zhang, Z., Choi, M., & Karniadakis, G. E. (2010). *Anchor Points Matter in ANOVA Decomposition*. https://doi.org/10.1007/978-3-642-15337-2_32

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2921–2929. https://doi.org/10.1109/CVPR.2016.319

Zhou, Z., Cai, H., Rong, S., Song, Y., Ren, K., Wang, J., Zhang, W., & Yong, Y. (2017). Activation maximization generative adversarial nets. *ArXiv*, 1–24.

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting Diabetes Mellitus With Machine Learning Techniques. *Frontiers in Genetics*, *9*(November), 1–10. https://doi.org/10.3389/fgene.2018.00515

10 Appendix A: Worked Example of the Dirac and Lebesgue Partial Response Measures

Looking at a few example observations from the "Three-way Interaction" artificial dataset D, which has been stratified into a train/test split of 70/30. From the training set:

Table A.2. Example observations from the Three-way Interaction artificial dataset

x_1	x_2	x_3	Outcome
0.2	0.1	0.9	0
0.9	0.4	0.2	1
0.4	0.6	0.1	0

Initialisation: Standardise all features to have median 0

Table A.14. Standardised values for the example observations

x_1	x_2	x_3	Outcome
-1.0591428	-1.0515425	1.7639473	0
1.4121904	0.0000000	-0.7055789	1
-0.3530476	1.0515425	-1.0583684	0

For a Support Vector Machine model using the Radial Basis Function kernel, with hyperparameters tuned:

Dirac Measure

Utilising the Dirac measure. For the null term, $D = 0$:

Table A.15. Setting all values to their feature median, to be inputted into the model

x_1	x_2	x_3	Outcome
0	0	0	0
0	0	0	1
0	0	0	0

Gives a null term $\varphi_0 = logit\left(P\left(C\middle|x^k\right)\right) = 4.385349$.

For the univariate terms, $\varphi_i(x_i) = logit\left(P\left(C\middle|x^k\right)\right) - \varphi_0$. When computing for x_1:

Table A.16. Target feature remains the same, while all others are set to their median

x_1	x_2	x_3	Outcome
-1.0591428	0	0	0
1.4121904	0	0	1
-0.3530476	0	0	0

Computing all the univariate terms gives partial responses:

Table A.17. Univariate partial responses for the example observations

$\varphi_1(x_1)$	$\varphi_2(x_2)$	$\varphi_3(x_3)$	$\varphi_{12}(x_1, x_2)$	$\varphi_{13}(x_1, x_3)$	$\varphi_{23}(x_2, x_3)$	Outcome
-13.906120	-15.683308	9.760864				0
6.411547	0.000000	-10.577995				1
-4.445737	8.051492	-15.388180				0

For the bivariate terms, $\varphi_{ij}(x_i, x_j) = logit\left(P(C|x^k)\right) - \varphi_i(x_i) - \varphi_j(x_j) - \varphi_0$. When computing for the bivariate terms between x_1 and x_2:

Table A.18. Target feature pair remains the same, while all others are set to their median

x_1	x_2	x_3	Outcome
-1.0591428	-1.0515425	0	0
1.4121904	0.0000000	0	1
-0.3530476	1.0515425	0	0

Computing all the bivariate terms gives partial responses:

Table A.19. Final univariate and bivariate partial responses for the example observations

$\varphi_1(x_1)$	$\varphi_2(x_2)$	$\varphi_3(x_3)$	$\varphi_{12}(x_1, x_2)$	$\varphi_{13}(x_1, x_3)$	$\varphi_{23}(x_2, x_3)$	Outcome
-13.906120	-15.683308	9.760864	8.37312745	1.83212465	0.109211842	0
6.411547	0.000000	-10.577995	0.00000000	0.98014234	0.000000000	1
-4.445737	8.051492	-15.388180	-0.19617049	2.35780017	-3.692965341	0

Lebesgue Measure

Utilising the Lebesgue measure. For the null term, $D = D$:

Table A.20. For the null term with the Lebesgue measure, all values stay the same to be inputted into the model

x_1	x_2	x_3	Outcome
-1.0591428	-1.0515425	1.7639473	0
1.4121904	0.0000000	-0.7055789	1
-0.3530476	1.0515425	-1.0583684	0

Gives a null term $\varphi_0 = \frac{1}{N}\sum_{k=1}^{N} logit\left(P(C|x^k)\right) = -0.9765357$.

For the univariate terms, when computing for x_1 we must compute multiple times, once for each value of x_1 as follows:

Tables A.21-11. Each individual value for a feature becomes all values for that feature, as is shown here for x_1

x_1	x_2	x_3	Outcome
-1.0591428	-1.0515425	1.7639473	0
-1.0591428	0.0000000	-0.7055789	1
-1.0591428	1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
1.4121904	-1.0515425	1.7639473	0
1.4121904	0.0000000	-0.7055789	1
1.4121904	1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
-0.3530476	-1.0515425	1.7639473	0
-0.3530476	0.0000000	-0.7055789	1
-0.3530476	1.0515425	-1.0583684	0

Computing all univariate terms, $\varphi_i(x_i) = \frac{1}{N}\sum_{k=1}^{N} logit\left(P\left(C\middle|x^k\right)\right) - \varphi_0$, gives partial responses:

Table A.12. Univariate partial responses for the example observations

$\varphi_1(x_1)$	$\varphi_2(x_2)$	$\varphi_3(x_3)$	$\varphi_{12}(x_1,x_2)$	$\varphi_{13}(x_1,x_3)$	$\varphi_{23}(x_2,x_3)$	Outcome
6.4109767	3.5391162	-10.9162100				0
5.9241409	0.8455565	-5.8387308				1
0.3515465	5.4602873	-8.7208140				0

For the bivariate terms, when computing for the bivariate terms between x_1 and x_2 we must compute multiple times, once for each pair of values of x_1 and x_2 as follows:

Tables A.13-21. Each pair of values for the target features becomes all values for their corresponding features. For x1 and x2 with our example observations, there are 9 pairs of values.

x_1	x_2	x_3	Outcome
-1.0591428	-1.0515425	1.7639473	0
-1.0591428	-1.0515425	-0.7055789	1
-1.0591428	-1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
-1.0591428	0.0000000	1.7639473	0
-1.0591428	0.0000000	-0.7055789	1
-1.0591428	0.0000000	-1.0583684	0

x_1	x_2	x_3	Outcome
-1.0591428	1.0515425	1.7639473	0
-1.0591428	1.0515425	-0.7055789	1
-1.0591428	1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
1.4121904	-1.0515425	1.7639473	0
1.4121904	-1.0515425	-0.7055789	1
1.4121904	-1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
1.4121904	0.0000000	1.7639473	0
1.4121904	0.0000000	-0.7055789	1
1.4121904	0.0000000	-1.0583684	0

x_1	x_2	x_3	Outcome
1.4121904	1.0515425	1.7639473	0
1.4121904	1.0515425	-0.7055789	1
1.4121904	1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
-0.3530476	-1.0515425	1.7639473	0
-0.3530476	-1.0515425	-0.7055789	1
-0.3530476	-1.0515425	-1.0583684	0

x_1	x_2	x_3	Outcome
-0.3530476	0.0000000	1.7639473	0
-0.3530476	0.0000000	-0.7055789	1
-0.3530476	0.0000000	-1.0583684	0

x_1	x_2	x_3	Outcome
-0.3530476	1.0515425	1.7639473	0
-0.3530476	1.0515425	-0.7055789	1
-0.3530476	1.0515425	-1.0583684	0

Computing bivariate terms, $\varphi_{ij}(x_i, x_j) = \frac{1}{N}\sum_{k=1,l=1}^{N} logit\left(P\left(C\middle|x^k\right)\right) - \varphi_i(x_i) - \varphi_j(x_j) - \varphi_0$, gives partial responses:

Table A.22. Final univariate and bivariate partial responses for the example observations

$\varphi_1(x_1)$	$\varphi_2(x_2)$	$\varphi_3(x_3)$	$\varphi_{12}(x_1, x_2)$	$\varphi_{13}(x_1, x_3)$	$\varphi_{23}(x_2, x_3)$	Outcome
6.4109767	3.5391162	-10.9162100	1.90025029	-3.27609408	-3.974452522	0
5.9241409	0.8455565	-5.8387308	1.83216213	0.74589583	-1.410241510	1
0.3515465	5.4602873	-8.7208140	1.06213457	-1.6120217	-3.276256722	0

These partial responses are then input into a Logistic Regression model with Lasso regularisation. The model parameter λ is tuned using cross-validation, and the model is selected based on the AUC metric. We can select a preferred model within 1 standard error of the best AUC performing model, in order to comprise a minimal set.

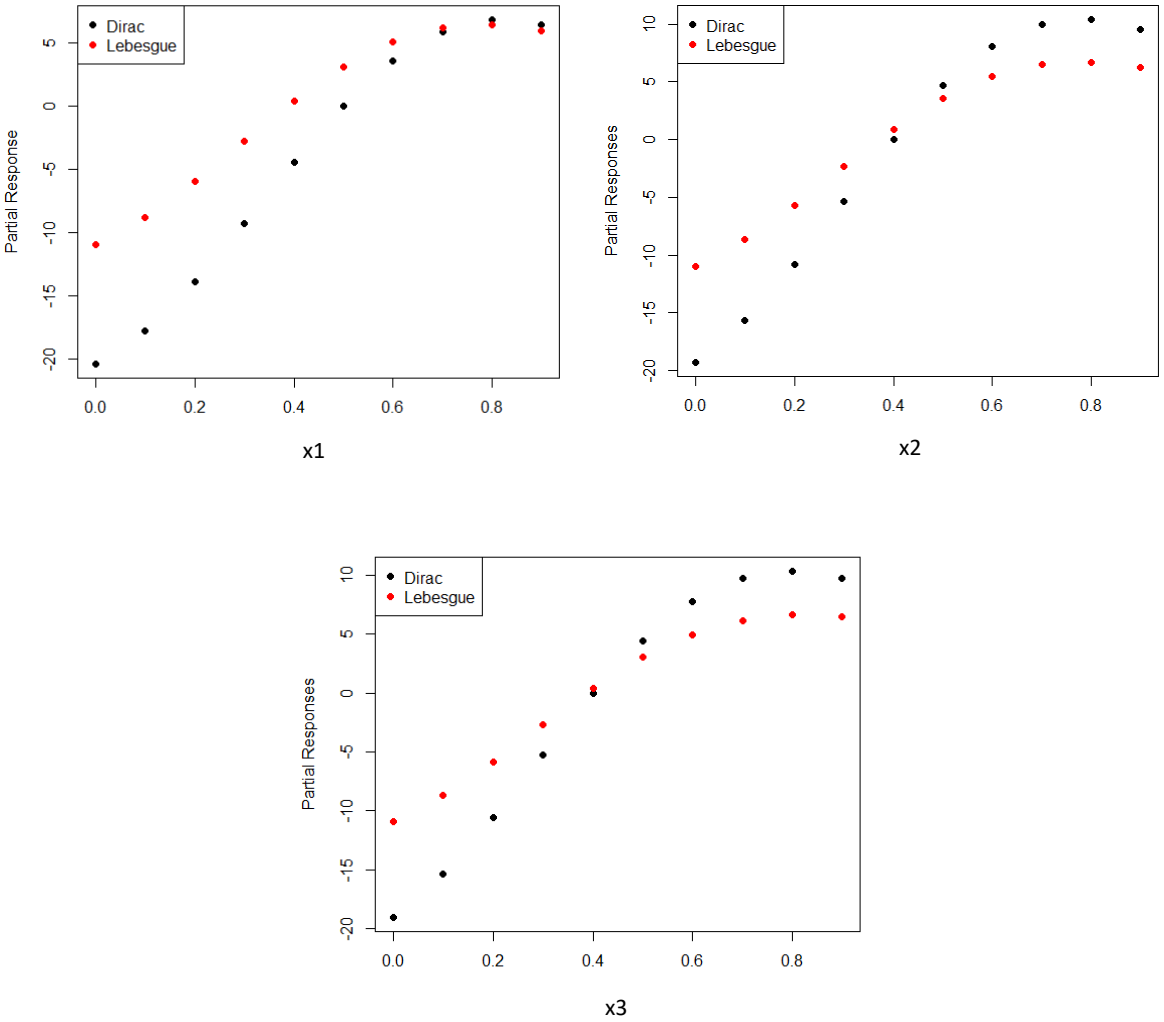


Figure A.2. Plots comparing partial responses for each variable, between Dirac and Lebesgue measures.