10-16-2023 11:30 AM

Local Model Agnostic XAI Methodologies Applied to Breast Cancer Malignancy Predictions

Heather Hartley, *Western University*

Supervisor: Michael Bauer, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

This thesis examines current state-of-the-art Explainable Artificial Intelligence (XAI) methodologies applicable to breast cancer diagnostics, as well as local model-agnostic XAI methodologies more broadly. It is well known that AI is underutilized in healthcare due to the fact that black box AI methods are largely uninterpretable. The potential for AI to positively affect health care outcomes is massive, and AI adoption by medical practitioners and the community at large will translate to more desirable patient outcomes. The development of XAI is crucial to furthering the integration of AI within healthcare, as it will allow medical practitioners and regulatory bodies to become more comfortable and trusting with respect to AI. The scope of this thesis is to examine XAI as it applies to breast cancer diagnostics specifically. However, as we have chosen to discuss local model-agnostic XAI methodologies, the techniques outlined in this thesis will be applicable to all medical domains.

Keywords: Explainable Artificial Intelligence (XAI), local model-agnostic explanations, breast cancer diagnostics

Summary for Lay Audience

The main achievements of this thesis are as follows;

(1) Provide an in-depth technical overview of the theory behind state of the art local XAI methodologies

(2) Extensively apply local XAI methodologies to unveil the inner workings of a XGBoost black box model used to diagnose breast cancer with 96% accuracy, using the Breast Cancer Wisconsin Diagnostic data set (BCW-D). This thesis is the most exhaustive analysis of local XAI methodologies applied to breast cancer diagnostics to date.

(3) Present a novel modification of the Biased Kernel SHAP algorithm called Fixed Biased Kernel SHAP, used to efficiently and accurately approximate true Kernel SHAP values, and evaluate the performance of this algorithm as compared to the original Biased Kernel SHAP algorithm.

Acknowledgements

To my supervisor Mike, I would like to thank you for all of your invaluable expertise and guidance throughout this process. To my loved ones, thank you for your continued encouragement and unwavering support.

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

1.1 Thesis overview and Contribution

In 2020, approximately 27,400 women were diagnosed with breast cancer in Canada [20]. Every 1 out of 8 women will receive a positive breast cancer diagnosis at some point in their life [20]. For women, breast cancer is the leading cause of cancer worldwide. Early detection, accurate diagnostics, and precisely selecting the most effective treatments for breast cancer are the most important tasks researchers must address in order to further favourable outcomes for breast cancer patients. As the intersection of artificial intelligence and breast cancer research continues to grow, we expect to see improvements in current mortality statistics.

Advancements in artificial intelligence stand to benefit humanity at large with its potential applications for healthcare. Predicting and diagnosing disease, as well as advising state of the art treatments and predicting subsequent patient outcomes to these treatments are the two domains in which AI stands to make its biggest

impact.

The contributions of this thesis include; (1) Provides a thorough technical run-down of current state of the art local XAI methodologies (2) Provides the most in depth research to date on applied local XAI methodologies for breast cancer diagnostics from fine-needle aspirate data, using the BCW-D dataset [26]. (3) Thorough and novel illustrations that help visualize the inner workings of an XGBoost model that achieves 96% diagnostic accuracy. Methods used include ICE & C-ICE, LIME, exact (true) Kernel SHAP, Biased Kernel SHAP approximation, and a novel algorithm called Fixed Biased Kernel SHAP. (4) Provides a novel algorithm to efficiently approximate Kernel SHAP values, called Fixed Biased Kernel SHAP. (5) A thorough analytical comparison between the various XAI methodologies.

1.2 Thesis organization

In Chapter 2 we discuss XAI in the broad sense and how various XAI methodologies may be categorized, as well as the current literature surrounding the application of XAI to breast cancer analytics. In Chapter 3 we take a deep dive into the theory behind the current leading XAI methods including Individual Conditional Expectation (ICE), Centered Individual Conditional Expectation (C-ICE), LIME, and Kernel SHAP. In Chapter 4 we discuss methods for approximating Kernel SHAP, a notoriously computationally expensive local XAI method. We present a modified version of the Biased Kernel SHAP approximation method, called Fixed

Biased Kernel SHAP. Chapter 4 discusses these Kernel SHAP approximations from a theoretical view point. Chapter 5 applies the approximation methods to a patient from the BCW-D test set with a malignant diagnosis that was diagnosed using an underlying XGBoost black box model, including our novel Fixed Biased Kernel SHAP algorithm. Additionally in Chapter 5, we apply all of the local XAI methods discussed in Chapter 3 to the same malignant case, providing an in depth analysis of the results and comparison between these methods.

Chapter 5 is where we evaluate and apply our novel Fixed Biased Kernel SHAP algorithm to explain the XGBoost algorithm used to predict breast cancer for the Breast Cancer Wisconsin Diagnostic dataset (BCW-D) dataset with 96% accuracy. We compare the performance of this algorithm versus a typical Biased Kernel SHAP approximation algorithm. We provide conclusory explanation tables at the end of each section in Chapter 5 to summarize the explanations provided by the local XAI methods on the XGBoost model for a malignant case. Finally, we compare the explanations and computational efficiency for all of the local XAI methods presented in this paper. In Chapter 6 we conclude our thesis and provide avenues for further research.

Chapter 2

Related Work

2.1 XAI Overview

Explainable AI (XAI) are methods that help users understand how black box artificially intelligent models arrive at their conclusions. It exists as a realm of artificial intelligence research in and of itself, as practitioners across the board try to unveil how these incredibly complex methods function internally. While inherently interpretable models such as linear regression, logistic regression, GLM, and simple decision trees have their own merits, they typically do not compete with more complicated black box models such as random forests, XGBoost, and neural nets of varying kinds. These black box models often offer superior accuracy at the cost of training speed, and model complexity. XAI seeks to bridge the gap between these convoluted black box models and human interpretability.

XAI methods can be broken down into two major categories; local and global

explanations. Local explanations are XAI methods that explain how a black box model arrived at its output for a single, localized instance. For example, a local XAI method may aim to explain exactly why an XGBoost model has predicted that a patient has a malignant breast tumor. Global XAI methods on the other hand, seek to explain how AI models make predictions on a comprehensive basis over an entire set of predictions. The focus of this thesis will be local XAI methodologies, for more information on global XAI methodologies please consult the research of Christopher Molnar [15].

Both global and local XAI methods may be either model-agnostic or model specific. Model specific XAI methods are those that may only be applied to certain underlying black box models, such as Tree SHAP which uses a Shapley Value approach to demystifying deep tree-based machine learning models [13]. Model agnostic XAI methods are those which may be applied to any underlying black box model, and ensemble machine learning models which may use a series of models. This thesis focuses specifically on local model agnostic XAI methods as it applies to breast cancer diagnosis.

The three major local XAI methodologies used at present are local Interpretable model-agnostic explanations (LIME), Individual Conditional Expectation plots (ICE) and Centered Individual Conditional Expectation plots (C-ICE), and variations of model agnostic Shapley value methods. LIME will be discussed in further detail both theoretically in Section 3.2 and as applied to breast cancer diaganostics in Subsection 5.2.2. ICE and C-ICE plots will be discussed in further detail both theoretically in Section 3.1 and applied to breast cancer diagnostics in

Subsection 5.2.1. Kernel SHAP, a model-agnostic XAI application of Shapley values, will be discussed both theoretically in Section 3.3 and applied to breast cancer diagnostics in Subsection 5.2.3.

In a time where Artificial Intelligence (AI) is exploding in popularity, there is an ever increasing need for methods that uncover the inner workings of artificially intelligent black box models. Some believe that the accuracy of AI models should be enough to warrant the adoption of these technologies without understanding of their inner workings, that we should simply trust AI to give us all the answers. However, we believe that there is a growing need for methods that attempt to dissect these algorithms to better understand their rationale. In the realm of medicine and health care, both the legality and ethics of using AI to diagnose and treat patients is of utmost consideration by healthcare and AI practitioners alike. Additionally, methods that seek to expose the contributing factors to a particular diagnosis may be of great importance when attempting to determine the best course of treatment for a particular patient. As it relates to breast cancer, treatment will vary greatly based on tumor subtypes and tumor presentation [25]. Furthermore, the goal of this thesis is to provide the most robust analysis and application of local model-agnostic XAI methods as it applies to breast cancer diagnostics to date.

2.2 XAI Applied to Breast Cancer

There are a multitude of data types and various kinds of datasets as it relates to breast cancer. Although healthcare data is scarce relative to other sectors due

to privacy and data collection concerns, there are various types of breast cancer datasets such as those consisting of mammography images, fine needle aspirate (FNA) images, FNA tabular data derived from FNA images, clinical information from patients, gene expression data of patients, and RNA sequencing data (this list is not exhaustive).

The current literature on XAI as it is applied to breast cancer is centered around a few different purposes; recurrence prediction, diagnostic prediction, treatment outcome prediction, and survival prediction.

Underlying models used in breast cancer related to XAI research include XGBoost, Random Survival Forests, Survival Support Vector Machines, Decision Tree Induction Algorithms, Classification and Regression Trees (CART) , Multilayer Perceptron, Deep multi-layer perceptron, and Radial Basis Function Networks [7] [16] [16] [8] [9] [4].

Depending on the underlying classification method, the XAI methodology applied to the problem may vary. As mentioned previously, XAI methodologies may be applied post-hoc or a practitioner may just opt to use an intrinsically interpretable model, meaning the underlying classification method is not a black box. Most of the literature on XAI and breast cancer related issues make use of post-hoc XAI methodology layered on top of black box underlying classification methods. This is due to the fact that black box methodologies typically provide far superior predictive accuracy compared to intrinsically interpretable models, and predictive accuracy is of utmost importance for most medical use cases. The black box nature of AI algorithms is a core reason for slow uptake of in-practice AI within

healthcare, and furthermore, discovering appropriate XAI methodologies to use in conjunction with these black box techniques is of utmost importance.

The literature presents a multitude of XAI methodologies used with breast cancer datasets; SHAP, LIME, feature importance, partial dependence plots, case-based reasoning, rainbow boxes, polar multi-dimensional scaling scatter plots, linear projections, radviz, and heatmap visualizations [7] [8] [9] [11] [4] [3]. This thesis takes its analysis and application of local XAI methods to the problem of breast cancer diagnostics one step further by discussing and applying all of the existing major methods, comparing the results between these methods, and providing a novel algorithm for Kernel SHAP approximation called Fixed Biased Kernel SHAP.

When considering mammographic images, convolutional neural networks (CNNs) are a frequented choice by practitioners. Heatmap visualizations are typically used in conjunction with these CNN models used to detect breast cancer in images, as is presented in the research by Binder et al. [3] and Montebello et al [10]. Heatmap visualization is a particularly useful tool when examining mammographic images as it highlights the pixels that contribute most heavily to the classification of either malignant or benign.

It is important to note that mammographic images are not the only diagnostic tool available for breast cancer. Typically when a tumor may be detected in a mammographic image, a biopsy will be conducted to determine if the breast tissue is malignant or benign. Fine-needle aspiration (FNA) is a technique wherein a fine needle is used to extract suspicious breast tissue, and the tissue is then examined

under a microscope for biopsy. Suspicious breast tissue in this case may be detected when a physical examination is performed and a lump is found, or may be as a result of image detection. Fine-needle aspiration is considered a minimally invasive biopsy method compared to incisional or excisional biopsy [19].

The Breast Cancer Wisconsin datasets, both the original (BCW-O) and the diagnostic dataset (BCW-D), have been integral in the development of machine learning models for breast cancer research. Both datasets contain features that are computed from digitized FNA images. BCW-D is cited in the literature on XAI and breast cancer research through the works of Lamy et al. [11], Brito-Sarracino et al. [4], Hakkoum et al. [8]. The BCW-D dataset is used throughout this thesis.

Chapter 3

Local Model Agnostic XAI Methods, Technical Overview

3.1 ICE and C-ICE

Individual Conditional Expectation (ICE) plots are the local alternative to Partial Dependence Plots (PDP). Partial Dependence plots, while limited in their use due to the assumption that features are independent [15], provide a relatively straightforward way of describing the relationship between the target variable (output of the black box model) and a feature input when considering the entirety of the dataset. PDP plots are generated by marginalizing the complement set of features out, and perturbing the feature to be explained so that the average marginal effect of changing this feature on the black box output is calculated and plotted in a graph [12]. Put simply, PDPs help describe the average marginal effect on

the target variable with increases or decreases in value for a given feature. The focus of this thesis is on local explainable AI methodologies as it is applied to breast cancer diagnosis, and as such we will limit our explanation of PDPs to the aforementioned.

ICE plots are useful as they can uncover heterogenous effects of a feature, which PDPs cannot [12]. ICE plots are most often graphically represented by plotting a single line for each instance, and the line is formulated by holding the complement set of features constant, and using the black box model to generate new predictions while varying the value of the feature in question. Although ICE plots are of use in determining heterogeneous trends in the data, they can sometimes be difficult to read or interpret due to the fact that the starting point for the prediction for each line will differ [15], in other words the trajectory of various lines may be similar between individual lines but not necessarily easily to see.

To solve this visualization problem that standard ICE plots face, we may use a centered ICE plot as an alternative (C-ICE). A centered ICE plot will show the difference in the prediction of $f(x)$ from an anchor point, whereas an ICE plot will show the total change in the prediction of $f(x)$ from a starting point. The formula for plotting a C-ICE plot for a given feature is as follows [15];

$$f_{centered} = f(x_{feature}, x_{complement}) - f(x_{anchor}, x_{complement}) \tag{3.1}$$

An more in depth analysis of ICE and C-ICE for breast cancer diagnosis in a patient with a malignant diagnosis from the BCW-D dataset can be found in

Subsection 5.2.1. We provide a sample ICE & C-ICE graph below for reference.

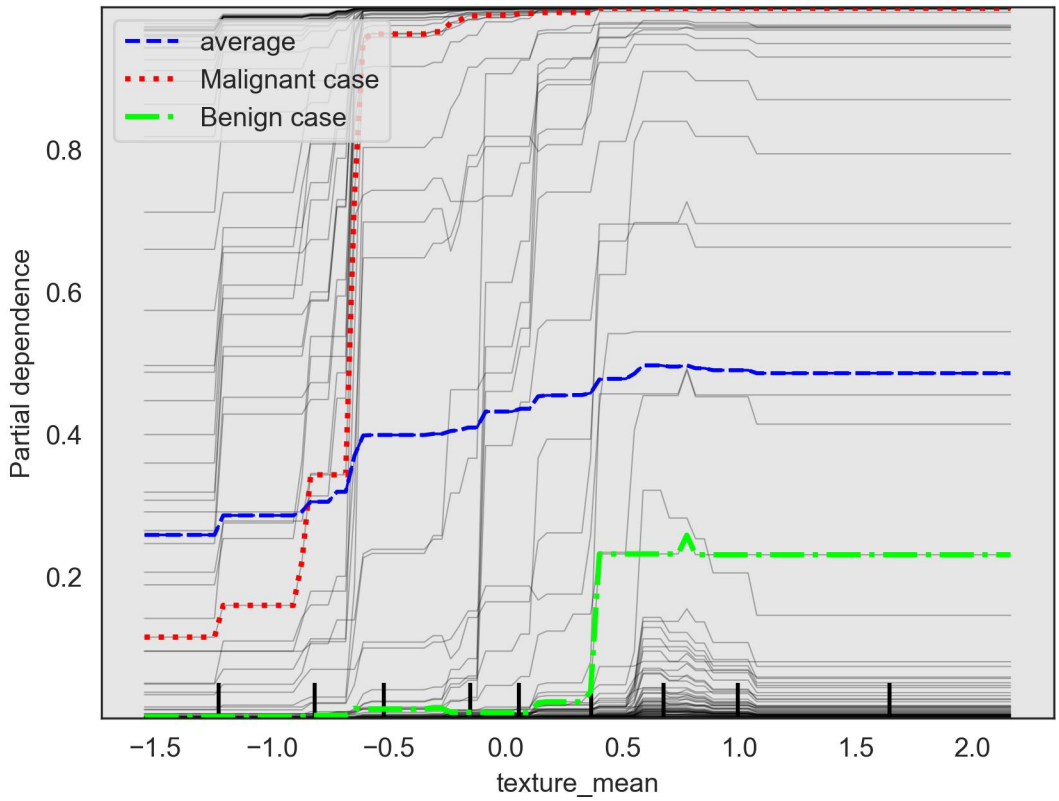

Figure 3.1: ICE for texture mean, XGBoost.

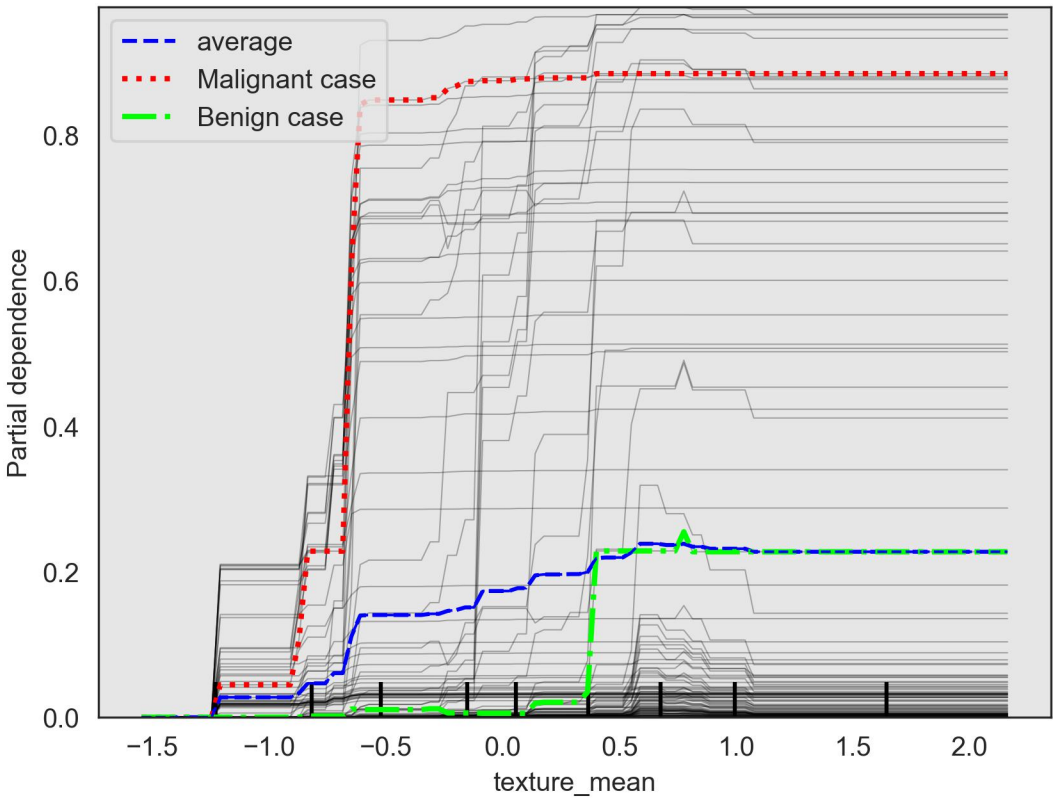

Figure 3.2: C-ICE for texture mean, XGBoost.

3.2 LIME

Local Interpretable Model-Agnostic Explanations (LIME), is an explainable AI method that was first introduced by Ribeiro et al. in 2016 [17]. LIME generates neighbours around an explanation instance, runs these neighbours through the underlying black box model to get an expected prediction for each, and then fits a local surrogate model on these generated neighbours. The local surrogate model chosen may be any model that is considered interpretable, not limited to but including; linear regression, logistic regression, decision trees, or generalized linear models. When fitting the local surrogate model, LIME uses an exponential kernel to assign higher weight to generated points that are closer to the explanation point [15]. The formula for the exponential LIME kernel is shown below;

$$\text{Weight} = \sqrt{\exp\left(-\frac{\text{distance}}{\text{kernel width}^2}\right)} \tag{3.2}$$

As we see from the above equation, we must choose a kernel width when calculating the weight for a generated neighbour. A larger kernel width equates to a wider radius from the explanation point being considered as part of the neighbourhood. This kernel width is often arbitrarily chosen, though more sophisticated methods for choosing kernel width have been proposed [24].

The goal of LIME is to minimize a loss function, that considers the explanation model g, the original model f, and the kernel defining the local neighbourhood around the explanation point $\pi_{x'}$. The below Equation 3.3 is the generic LIME formula. It states that the optimal locally interpretable model agnostic ex-

planation model represented by g, will minimize the difference between itself and the black box model f, for some given neighbourhood $\pi_{x'}$. Additionally, there is a penalty term added for surrogate models that are more complex represented by $\Omega(g)$.

$$\varepsilon = \arg\min_{g \in G} L(f, g, \pi_{x'}) + \Omega(g) \tag{3.3}$$

Criticisms of LIME are that it may violate local accuracy, an axiom of additive feature attribution methods, when the selected kernel weighting function and the loss function are chosen arbitrarily, thus producing a surrogate model that is not tangent to the black box model [14].

An applied example of LIME for breast cancer diagnosis can be found in Subsection 5.2.2.

3.3 Shapley Values, SHAP, and Kernel SHAP

SHAP (SHapley Additive exPlanation) values are an explainable AI framework first introduced by Lundberg et al. [14] in their paper entitled 'A Unified Approach to Interpreting Model Predictions'. SHAP values are related closely to Shapley values, a concept derived from cooperative game theory by Lloyd Shapley in 1953 in his paper entitled 'A Value for n-Person Games' [18].

A cooperative game requires players and a game, which in the context of machine learning are the features of the model and the model prediction respectively. For the purposes of this thesis we will not delve into the differences between

cooperative games and other types of games that exist in game theory such as non-cooperative games.

The payout from the game is the model prediction itself, and this payout must be distributed equitably amongst the players based on their respective contributions. Shapley values are a unique solution that tell us how we should divide this payout amongst the players and it is based on each feature's contribution to the game. For example, if we have a machine learning model $f(x)$, then the Shapley values will tell us how to divide the prediction of $f(x)$ less its base value into the contributions made by each of the features.

The formal definition for a Shapley value is shown below in Equation 3.4. θ_i represents the Shapley value for feature i. S is a subset of players in a cooperative game, and F is the set of all possible permutations of subsets. $|S|$ and $|F|$ are the number of players in the subset and total number of players respectively.V is the value function that takes in a subset of players in the game, and in the context of machine learning it is the black box model output on a subset of features. Ultimately, the equation details the marginal contribution of a player to the game.

$$\theta_i = \sum_{S \subseteq F - \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!}(V(S \cup i) - V(S)) \tag{3.4}$$

Shapley values differ from SHAP values in that the formal Shapley value first proposed by Lloyd Shapley [18] may use any value function that maps a coalition of players (ie. a subset of players in the powerset of all possible permutations of feature subsets) to the real number space $V(S)$ that satisfies the axioms of a

Shapley value; efficiency, symmetry, dummy, and additivity. $V(S)$ represents the value function for the coalition S. SHAP values are effectively the application of Shapley values to machine learning models, wherein the machine learning model is the value function $V(S)$.

When trying to evaluate the Shapley values of some cooperative game, we must choose a value function that satisfies the axioms of a Shapley value. One possible value function that returns Shapley values is shown below in Equation 3.5. The x_{S_1} through x_{S_M} represent the actual feature values for a particular coalition/subset, where M is the total number of features. $f_S(x_S)$ denotes the marginal function of f with respect to the subset S, meaning it takes in a subset of the total features in the grand coalition. In order to calculate $f_S(x_S)$ we either need to retrain f using only the features in the subset, or use the original model to calculate the marginal value of f with respect to the subset. $E[f(X)]$ is the average target value of the machine learning model training set.

$$V(S) = V(x_{S_1}, x_{S_2}, x_{S_3}, ..., x_{S_m}) = f_S(x_S) - E[f(X)] \tag{3.5}$$

We know that training machine learning models is more computationally expensive than producing predictions, so let's suppose that $f_S(x_S)$ is computed by using the original model f which is able to take in missing values for features. Most machine learning models are not equipped to handle missing features and this is a theoretical example to showcase the differences between Shapley values and SHAP values. There are various ways in which a practitioner may simulate a

feature being missing, however in this thesis we will use the background training data set to simulate missing features.

So we've established by the formal definition of a Shapley value that we may use any value function that satisfies the Shapley value axioms. And we also know that one possible value for the value function that satisfies these axioms is given in Equation (3.5) . Where SHAP values branch off from the general form of a Shapley value, is that they tell us how to calculate the marginal value of f with respect to the subset S. SHAP values make use of the fact that training new models is an expensive feat, and that it's faster to use the original model to calculate the marginal function of the subset. SHAP says that we may use the expected value of the function given the subset of features present in the subset.

$$f_S(x_S) \approx E[f(x)|x_S] \tag{3.6}$$

Now suppose we have some variable z' that is a binary coalition vector that represents the features present in the subset with 1 where the feature is present and 0 where the feature is absent. If we had an instance $x = [17, 4, 5, 9, 3]$ then x' would be the grand coalition vector $[1, 1, 1, 1, 1]$. z' is defined to be any coalition (ie. subset) of the grand coalition vector, for example $z' = [1, 0, 1, 0, 1]$. z is the value representation of the binary coalition vector, so if $z' = [1, 0, 1, 0, 1]$, then $z = [17, missing, 5, missing, 3]$. In this thesis we use the background data set feature values, ie. the training set used to train the black box model, in order to simulate missing features. This means that in order to calculate a coalitions value

we will replace the feature values in the background dataset with the feature values from the explanation point where the feature is present. The mapping function $h_x(z')$ takes in a binary coalition vector and returns the value coalition vector z (ie. $h_x(z') = z$). With this new binary coalition notation we can assert that $f_x(z') = f_S(x_S) = f(h_x(z')) = f(z)$ are all equivalent, but written in different notation. Therefore we can rewrite Equation (3.6) as follows;

$$f_x(z') = f(h_x(z')) \approx E[f(z)|z_s] \tag{3.7}$$

The formal definition of SHAP values can be written as follows if we're using coalition notation, but this may also be rewritten in subset notation using x_s instead of z_s:

$$\theta_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(|x'| - |z'| - 1)!}{|x'|!}(E[f(z)|z_s] - E[f(z)|z_{s-\{i\}}]) \tag{3.8}$$

Equation (3.4) initially looks different than Equation (3.8), but we know that $|S|$ is equal to $|z'|$ and $|F|$ is equal to $|x'|$. $|x'|$ is equal to the number of elements in the grand coalition (ie. the total number of players) and $|z'|$ is equal to the number of elements in the subset z'. Equation (3.8) is asserting that to calculate the contribution of feature i, we must average the contributions of feature i to all the coalitions/subsets of x'. Another difference to note between Equation (3.8) and (3.4) is that the Shapley value in Equation (3.8) is a function of f (our original machine learning model), and x (the point of interest to be explained). The Shapley value in Equation (3.8) is a function of f and x because we are using a

specified value function similar to (3.6) but where $f_S(x_S)$ is defined to be the conditional expectation of $f(z)$ given some subset z_s as shown in Equation (3.7) . In short, SHAP values are the machine learning application of Shapley values.

Now the computation of these conditional expectations of f based on some subset z_s is a computationally exhaustive task, particularly when features are not independent and the black box model is not linear. This is because as the number of features in a machine learning model increases, the number of possible feature permutations rises exponentially. SHAP may be approximated in different ways depending on whether or not we apply the optional assumption of feature independence, however this is beyond the scope of this thesis. If one assumes feature independence, one may approximate SHAP by employing the Shapley Sampling Values method described in the original SHAP paper[14]. However, only assuming feature independence and not model linearity requires us to use Equation (3.8) which is a permutation form of computing Shapley values, and thus it requires an exponentially large number of model evaluations, which is extremely expensive and impractical in a higher dimension feature space.

This thesis focuses on how we may approximate SHAP under the assumptions of both feature independence and model linearity by employing Kernel SHAP. Kernel SHAP does not need to evaluate $f(x)$, the original machine learning model as many times as the permutation version of Shapley values described in equations (3.4) and (3.8) [14]. Additionally, Kernel SHAP has been shown to be a faster approximation of SHAP values with similar accuracy as permutation based approximation methods [14].

Now let's define an explainer model called g that takes in a binary coalition vector as its argument;

$$g(z') = g(z'_1, z'_2, z'_3, ..., z'_M) \tag{3.9}$$

By the definition of what constitutes an additive feature attribution method, g is a linear function that takes in a binary coalition vector as follows:

$$g(z') = c_0 + \sum_{i=1}^{M} c_i z'_i \tag{3.10}$$

There are 3 desirable properties of feature attribution methods; local accuracy, missingness, and consistency. These properties are explained further in the original paper presenting SHAP [14]. As this thesis focuses on local explanation methods, we highlight that the local accuracy property refers to the fact that we want our explanation model to explain some model f at a particular point x; $g(x') = f(x)$ for some particular instance x.

$$f(x) = g(x') = \theta_0 + \sum_{i=1}^{M} \theta_i x'_i \tag{3.11}$$

Lundberg et al. highlight that Shapley values are the only solution that satisfy the 3 aforementioned desirable properties of additive feature attribution methods, and that any explanation methods that do not follow the Shapley value formula violate local accuracy and/or consistency. Furthermore, Lundberg et al. proposed SHAP as an adaptation of Shapley values with a defined value function (the value function noted in Equation (3.6)), that satisfy the 3 desirable properties of an

additive feature attribution method. As previously mentioned, SHAP values are very computationally expensive and Kernel SHAP provides a way to approximate SHAP values that is faster than permutation based Shapley value methods, while retaining similar accuracy.

Kernel SHAP assumes model linearity in addition to feature independence. It is shown in the SHAP paper that Kernel SHAP minimizes a loss function using certain parameters, and in doing so it recovers the SHAP values. The loss function considers as arguments the original function, the explainer model g, and some measure of distance between the original model and the explainer model. The kernel which minimizes the loss function L is called the Shapley kernel and is detailed below along with the implementation of the loss function which considers the squared distance between the explainer model output for some coalition z' and the original model output for that coalition. The Shapley kernel proof may be found in the original SHAP paper [14]. $\Omega(g) = 0$ simply states that the Shapley kernel does not use a regularization term on the explainer model. $\pi_x(z')$ represents the kernel weight applied to each subset z', taking into account the total number of features M, previous referred to as $|x'|$, and the number of subsets of size $|z'|$. $L(f, g, \pi_{x'})$ is the loss function that is to be minimized, and in layman's terms it is essentially stating that we want the smallest distance between the predictions of the black box model and the explainer model that aims to emulate the black box model.

$$\Omega(g) = 0 \tag{3.12}$$

$$\pi_{x'}(z') = \frac{M-1}{\binom{M}{|z'|}|z'|(M-|z'|)}$$

$$L(f, g, \pi_{x'}) = \sum_{z' \subseteq x'} \pi_{x'}(z')(g(z') - f_x(z'))^2$$

So we know that by minimizing the loss function detailed in Equation (3.12), we will recover the SHAP values under the assumption of model linearity and feature independence (Kernel SHAP). Additionally, as Lundberg et al. point out, using the Shapley Kernel definitions for the regularization term, the weighting term, and the loss function are the solution to LIME that achieve local accuracy, missingness, and consistency [14]. The algorithm for the computation of Kernel SHAP values is discussed by Covert et al. in their research regarding Unbiased versus Biased Kernel SHAP computations [5], and we provide the Biased Kernel SHAP algorithm for posterity in Section 4.3.

In order to solve for exact (true) Kernel SHAP values we must evaluate our machine learning model 2^M times, where M is the number of features in the model. For models containing a small number of features this is a feasible computation, however as the dimensionality of a machine learning model increases the number of model evaluations we must perform for true Kernel SHAP increases exponentially. Thus the task of approximating true Kernel SHAP values in an computationally efficient manner is of utmost importance.

This thesis focuses on examining and modifying the Biased Kernel SHAP

algorithm as presented by Covert et al. and discussed in Section 4.3, and also the Ensemble of Random SHAPs methods, first introduced by Utkin et al. and discussed in Section 4.2 [23] [5].

Chapter 4

Approximating Kernel SHAP

4.1 Existing Methods

As noted in the original SHAP paper [14], practitioners may approximate SHAP under the assumption of model independence, and not linearity, by the Shapley sampling values method. In this context, the Shapley sampling values method attempts to estimate the SHAP values using Equation (3.8), a permutation version of the classic Shapley values under the assumption of feature independence. The estimation of each feature's SHAP value is done by sampling the marginal contribution of a feature to all possible permutations of feature coalitions. This task is often done by Monte Carlo integration, as noted by Aas et al. [1], and Strumbelj et al. [21]. While the aforementioned sampling method does not assume model linearity, it is shown in the original SHAP paper that Kernel SHAP, which does assume model linearity, does not need to evaluate the original black box model as many times in order to approximate SHAP, and it does so with competing accuracy [14].

Aas et al. propose a sampling method regarding relaxing the assumption of feature independence in Kernel SHAP. They do so by estimating the conditional probability distribution of the complement coalition $x_{\bar{s}}$ given that x_s is equal to some coalition subset of features. This research provides quite interesting insight to the effect of assuming feature independence when trying to predict SHAP values, however the authors note that it is still very computationally expensive and does not provide a computational time benefit over Kernel SHAP [1]. Furthermore, the approximation methods studied in this thesis focus on approximating Kernel SHAP under the assumptions of feature independence and model linearity.

We believe that a focus on model agnostic Kernel SHAP approximations is imperative due to the fact that in practice, machine learning practitioners may opt for ensemble pipelines containing different types of machine learning models [6]. As such we will not provide further detail on model-specific SHAP approximations such as DASP, [2], Deep SHAP [14], or Tree SHAP [13].

Covert et al. discuss the differences between the Kernel SHAP algorithm which some practitioners have asserted is a biased algorithm [5]. However, they detail that a Biased Kernel SHAP algorithm sustains a trivial increase in bias for remarkably lower variance than an Unbiased Kernel SHAP algorithm, and thus converges to true Kernel SHAP values much faster than its unbiased counterpart. Producing true Kernel SHAP values for a particular explanation point requires producing an exponentially large number of model predictions, specifically $2^M \cdot$ number of training samples, where M is the total number of features. This is a challenging feat computationally if the number of features used as black box

inputs is relatively high. Thus, using a sample of coalitions rather than the entire power set is necessary to achieve computational efficiency. Knowing how many coalitions samples to use to reasonably approximate true Kernel SHAP values is also challenging. Covert et al. also address this challenge by suggesting the use of a convergence detection algorithm such as Welford's algorithm. More information on how to use Welford's aglorithm to detect Kernel SHAP convergence may be found in their paper [5]. In addition to the convergence detection solution to understanding when a sufficient number of coalition samples have been reached, the authors also offer a variance reduction technique that involves coalition complement sampling.

4.2 Ensemble of Random SHAPs

The Ensemble of Random SHAPs method was first introduced by Utkin et al. [23]. In their research they present several algorithms that take the following generalized approach; for a chosen level of t, construct coalitions of the original explanation point x of size t, wherein the features that are selected for each of the coalitions are chosen by some probability distribution and with replacement. For example, if we have an original explanation point $x = [1, 3, 7, 8, 9, 5, 0, 8, 11, 3]$, we could construct $t = 3$, $N = 4$ number of coalitions of the original vector as follows; $z'_1 = [1, 0, 0, 1, 1, 0, 0, 0, 0, 0]$, $z'_2 = [1, 1, 0, 1, 0, 0, 0, 0, 0, 0]$, $z'_3 = [0, 1, 0, 1, 1, 0, 0, 0, 0, 0]$, $z'_4 = [1, 1, 1, 0, 0, 0, 0, 0, 0, 0]$. The value representation of the coalition vectors (ie. the vectors corresponding to z' that have the values

from the original explanation point substituted where the indices of $z'_i = 1$) are then run through Kernel SHAP separately to produce 3 different sets of exact Kernel SHAP values. Using these algorithms, we are not returned Kernel SHAP approximations for the features not present in a chosen coalition, and for a chosen coalition we replace the data in the training set for each feature that is missing with its average feature value. In the case of standardized data we would replace the training data fed into the explainer model with 0 for features that are not present in the given coaltion. Over N number of iterations, we are left with a set of Kernel SHAP values from each z that correspond to the features present in each of these coalitions. We then combine these Kernel SHAP values by simple averaging to produce a final set of Kernel SHAP approximations.

ER-SHAP is the most rudementary algorithm presented in their paper [23], and it uses a uniform distribution when selecting t number of features from the explanation point to construct N coalition vectors, and the sets of Kernel SHAP approximations produced by each of the N iterations are then combined by a simple average. In the example cited earlier wherein we have the coalition vectors $z'_1 = [1, 0, 0, 1, 1, 0, 0, 0, 0, 0]$, $z'_2 = [1, 1, 0, 1, 0, 0, 0, 0, 0, 0]$, $z'_3 = [0, 1, 0, 1, 1, 0, 0, 0, 0, 0]$, and $z'_3 = [1, 1, 1, 0, 0, 0, 0, 0, 0, 0]$, we would produce 3 approximations for feature 1, 3 approximations for feature 2, 1 approximation for feature 3, 3 approximations for feature 4, 2 approximations for feature 5, and no approximations for features 6-10. We average the approximations based on how many times each feature is selected to be a part of a coalition vector, for example for feature 5 we would sum the 2 Kernel SHAP approximations produced by ER-SHAP and then we would

divide by 2. The time complexity of this particular example would be $4 \cdot 2^3$ since we performed 4 iterations of ER-SHAP and selected 3 features for each coalition, so we perform 2^3 model evaluations each time that we iterate. The total number of model evaluations for this example is 32.

4.3 Biased Kernel SHAP

This algorithm is the algorithm presented and analyzed in depth by Covert et al in their research paper entitled 'Improving Kernel SHAP: Practical Shapley Value Estimation via Linear Regression' [5]. We provide the algorithm for Biased Kernel SHAP as presented by the aforementioned authors for posterity and to compare with the Fixed Biased Kernel SHAP algorithm presented in this research paper.

To reiterate, a Biased Kernel SHAP algorithm sustains a trivial increase in bias for remarkably lower variance than an Unbiased Kernel SHAP algorithm, and thus converges to true Kernel SHAP values much faster than its unbiased counterpart. Producing true Kernel SHAP values for a particular explanation point requires producing an exponentially large number of model predictions, specifically $2^M \cdot$ number of training samples, where M is the total number of features. We do not detail the technical difference between Unbiased and Biased Kernel SHAP approximation algorithms, as it is out of scope for this thesis and covered extensively by Covert et al [5].

It is important to note that in a Biased or Unbiased Kernel SHAP algorithm, $p(Z)$ is the probability distribution for which we use to sample coalitions, and

it is determined by the Shapley kernel from Equation 3.12. Recall that $\pi_{x'}(z') = \frac{M-1}{\binom{M}{|z'|}|z'|(M-|z'|)}$. The probability distribution of the coalitions is thus defined as follows [5];

$$p(Z) = \begin{cases} N^{-1}\pi_{x'}(z') & \text{if } 0 < 1^T z' < M \\ 0 & \text{otherwise} \end{cases}$$

As in the above equation, N is the sum of the kernel weights for all coalition sizes. We must divide the coalitions kernel weights by this sum in order to get their respective probabilities. Furthermore, N is defined as follows;

$$N = \sum_{|z'|=1}^{M-1} \frac{M-1}{|z'|(M-|z'|)} \tag{4.1}$$

Algorithm 1 Biased Kernel SHAP

Data: Training Data Set
Input: Explanation Point, Trained Black Box Model
Result: Kernel SHAP approximations

//Initialize
$numFeatures$ = number of features in training set
$numSamples$ = number of coalitions to sample $2^{\text{numFeatures}}$
$n = 0$
$A = 0$
$b = 0$

while $n \leq numSamples$ **do**
  //Sample a coalition, compute matrix A, compute vector b
  Sample $z' \sim p(Z)$
  $Asample = z'z'^T$
  $bsample = z'(v(z) - v(0))$
  $n = n + 1$
  $A+ = (Asample - A)/n$
  $b+ = (bsample - b)/n$

end

// Get Shapley value estimates
$B = A^{-1}(b - 1 \cdot \frac{1^T A^{-1} b - v(1) + v(0)}{1^T A^{-1} 1})$
return B

4.4 Fixed Biased Kernel SHAP

Unfortunately, the Ensemble of Random SHAPs algorithms presented by Utkin et al. leave much to be desired. In particular, they give no explanation as to whether or not the calculation of Kernel SHAP approximations for each iteration shall be biased or unbiased, or whether the calculation of each set of Kernel SHAP approximations shall adhere to a t sized calculation or a full scale Kernel SHAP approximation using a weight matrix that has dimensionality equal to the number of features. Additionally, the authors provide no insight into how these algorithms perform against Biased Kernel SHAP. As an alternative, we present a novel algorithm called Fixed Biased Kernel SHAP that draws upon the advantages of both the Biased Kernel SHAP algorithm and the front-loading aspect of the Ensemble of Random SHAPs methods.

The algorithm is similar to Biased Kernel SHAP in that it uses the same calculation for matrix A and and vector b, but it is also similar to the Ensemble of Random SHAPs methods in that we select a t size coalition, and then we sample only coalitions of that size in addition to their enumerations. Similarly to Biased Kernel SHAP, Matrix A is the matrix representation of kernel weights, and b is the vector representation of the average of the coalitions values less the black box base value. The base value $v(0)$, is the average target value of the training samples fed into the black box model.

Fixed Bised Kernel SHAP differs from Biased Kernel SHAP in that the coalitions to be sampled are largely pre-determined by selecting a t coalition size to

exclusively sample and enumerate. In Fixed Biased Kernel SHAP we do not double count coalitions, for example if we have a 10 feature model and select $t = 2$, then the maximum number of coalitions that could be sampled would be 55 since $\binom{10}{2} = 45$ and $\binom{10}{1} = 10$. Note that in a 10 feature model, there are exactly 45 coalitions of size 2 and exactly 10 coalitions of size 1.

The algorithm for Fixed Biased Kernel SHAP can be found below, and the evaluation of the algorithm's performance versus Biased Kernel SHAP can be found in Subsection 5.3.2.

Algorithm 2 Fixed Biased Kernel SHAP

Data: Training Data Set
Input: Explanation Point, Trained Black Box Model
Result: Kernel SHAP approximations

//Initialize
$numFeatures$ = number of features in training set
t = selected coalition size, $2 \leq t \leq numFeatures$
$featuresList$ = [i for i in range(0, $numFeatures$)]
$grandCoalitionPowerSet$ = list(powerset($featuresList$))
$tCoalitions$ = [all t size coalitions from grandCoalitionPowerSet]
$numSamples$ = total number of coalitions to sample $\leq \sum_{i=1}^{t} \binom{\text{numFeatures}}{i}$
$A = 0$
$b = 0$
$n = 0$

$enumeratedCoalitions$ = set()
while $n \leq numSamples$ **do**
  // Sample a t size coalition
  $sampleCoalition$ = sample coalition from $tCoalitions \sim p(Uniform)$
  $coalitionSet$ = list(powerset($sampleCoalition$))
  $tCoalitions$.pop($sampleCoalition$)

  foreach *coalition in coalitionSet* **do**
    if *coalition not in enumeratedCoalitions* **then**
      // Compute matrix A, compute vector b
      z' = binary vector representation of coalition
      $Asample = z'z'^T$
      $bsample = z'(v(z) - v(0))$
      $n = n + 1$
      $A+ = (Asample - A)/n$
      $b+ = (bsample - b)/n$
      $enumeratedCoalitions$.add($coalition$)
    end
  end
end

// Get Shapley value estimates
$B = A^{-1}(b - 1 \cdot \frac{1^T A^{-1} b - v(1) + v(0)}{1^T A^{-1} 1})$
return B

Chapter 5

Applying Local XAI Methods to Breast Cancer Diagnostics

5.1 Description of Data Set and Preprocessing

The experiments in this Chapter all use the Breast Cancer Wisconsin Diagnostic (BCW-D) data set. This data set consists of features computed from fine needle aspirate (FNA) tumor collection, wherein the values of the features describe the nuclei characteristics found in the FNA image [26]. This data set has 569 records, and 10 features reported by their mean, standard-error, and worst values (average of the biggest 3 nuclei) [26]. The target variable for the BCW-D data set is a binary class indicating whether or not the patient has a malignant or benign tumor. The 10 features are; Radius, Texture, Perimeter, Area, Smoothness, Compactness, Concavity, Concave Points, Symmetry, and Fractal Dimension. The class distri-

bution is somewhat imbalanced as there are 357 benign and 212 malignant cases. Since this thesis computes exact Kernel SHAP results, we construct our models using only the mean values of the 10 aforementioned features and leave out the standard-error and worst feature values. Furthermore, we construct our models for the BCW-D dataset using 10 features.

For the purposes of comparing local model agnostic XAI methods as in section 5.2, we use an XGBoost model to produce predictions of tumor malignancy from the BCW-D dataset. Prior to evaluating our model we standardize all 10 features to a mean of 0 and standard deviation of 1 following a normal distribution. Since the focus of this thesis is on Local Model Agnostic XAI methods and approximating Kernel SHAP, we will not provide supplementary detail on the inner workings of XGBoost itself, since the XAI methods discussed in this paper may be applied to any underlying black box model. The accuracy of our underlying XGBoost model is 96%. The aforementioned black box model is trained using an 80/20 train-test split from the BCW-D dataset, which translates to 455 training samples and 114 test set samples.

This chapter applies local XAI methodologies to the same malignant case and analyzes and summarizes the differences between the various methods. We provide extensive and novel illustration to help visualize the explanation of the underlying XGBoost model. Additionally, we provide a novel modification of the Biased Kernel SHAP algorithm, called Fixed Biased Kernel SHAP, to efficiently approximate Kernel SHAP values while retaining satisfactory accuracy.

5.2 Method Application & Analysis

5.2.1 ICE and C-ICE

Figures 5.1, 5.2, 5.3, and 5.4, detail both the ICE & C-ICE plots for the top two features contributing towards malignancy in the patient studied. The top contributing feature according to ICE and C-ICE is texture, followed by smoothness. This is different than the average most impactful feature contributing towards malignancy, which is area. For a complete technical rundown on the inner workings of ICE & C-ICE please reference Section 3.1. The ICE & C-ICE plots for the remaining features may be found in Appendix A.

Each of the black lines constitute a single patient's ICE trajectory for the 114 patients in the BCW-D test set. The dashed blue line represents the average ICE plots for all 114 patients. Since we standardized our features prior to training our XGB model, the average feature value is zero across all features. For the texture feature in the malignant case studied, we see that at around -1 standardized deviations, the probability of malignancy starts to rise dramatically. To produce the ICE & C-ICE graphs below, we provide a novel modification of the sklearn.inspection.PartialDependenceDisplay graphs in order to highlight the local malignant case studied in this chapter, as well as the benign case referenced in Appendix A.

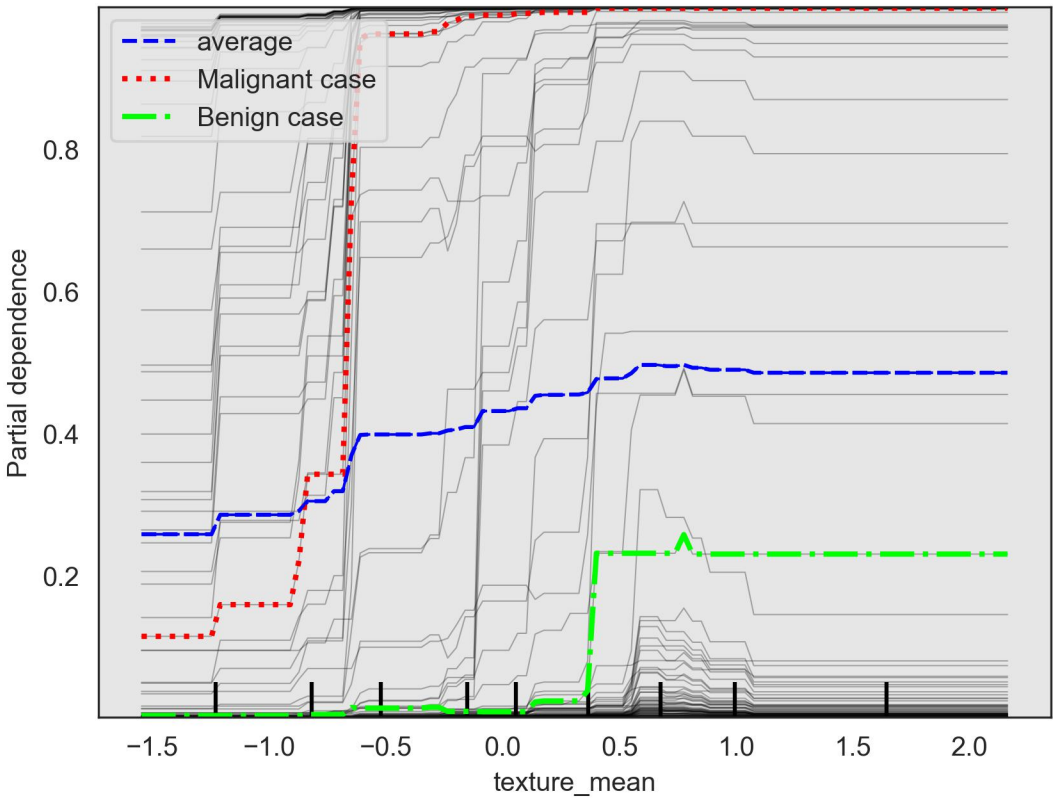

Figure 5.1: ICE for texture mean, XGBoost.

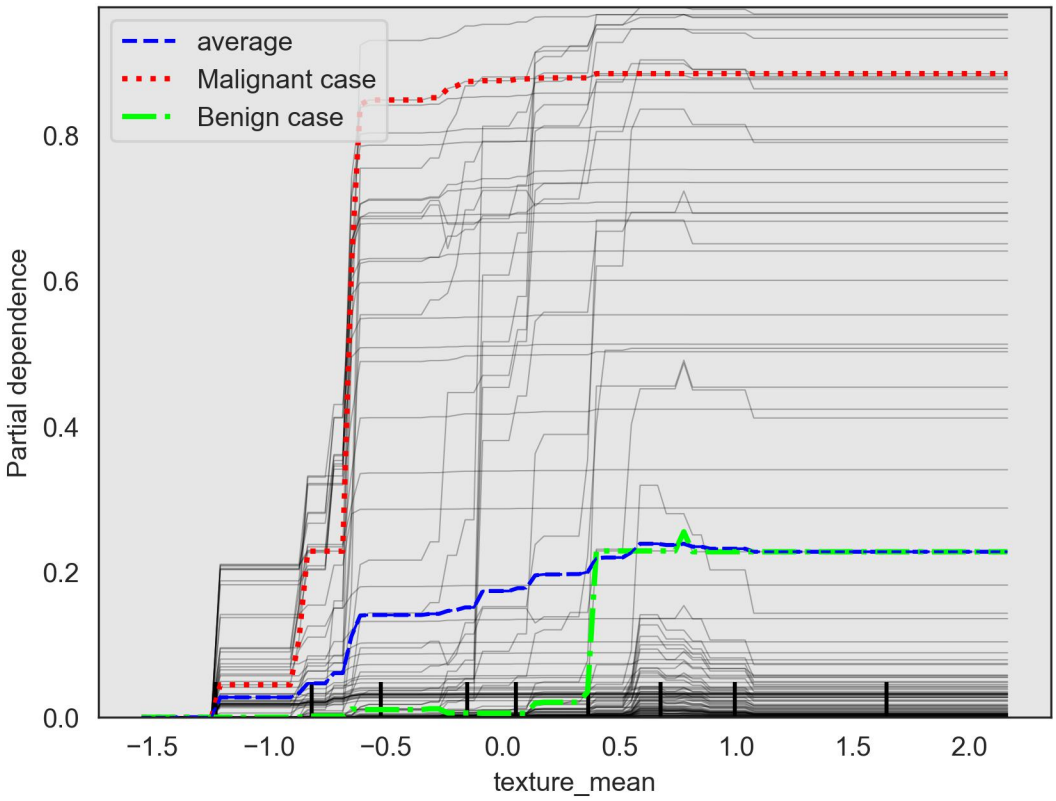

Figure 5.2: C-ICE for texture mean, XGBoost.

For the smoothness feature seen below on the malignant case studied, we see that at around -0.5 standard deviations, the probability of malignancy starts to increase.

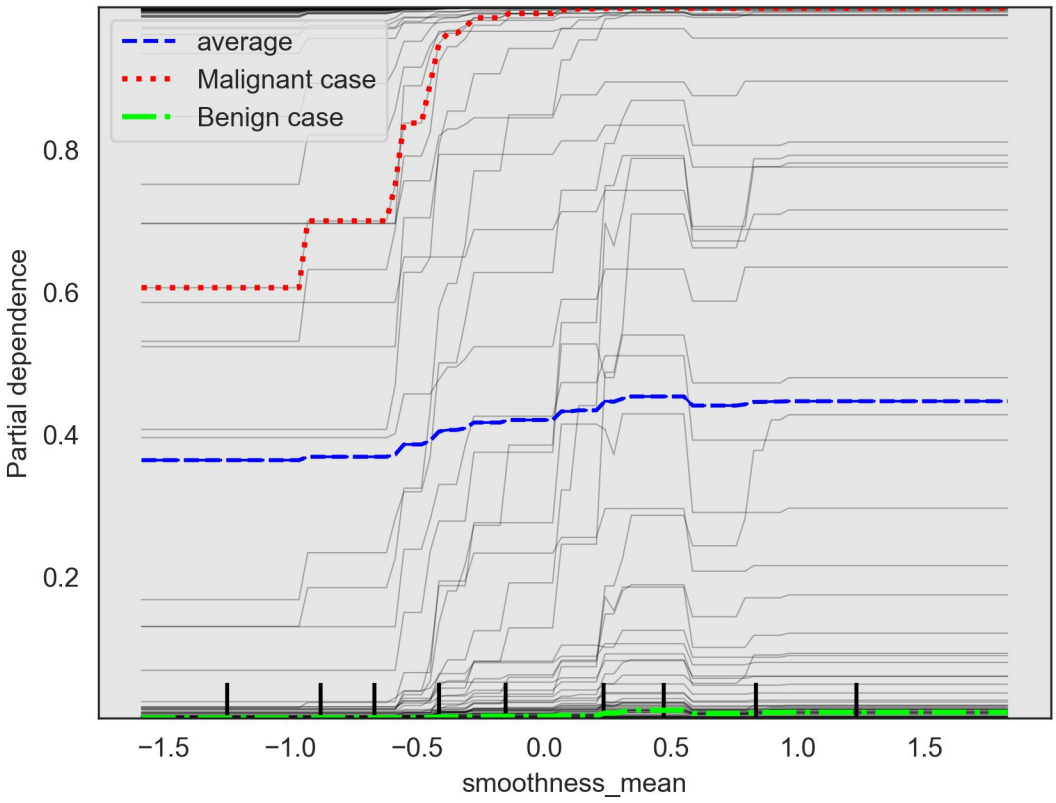

Figure 5.3: ICE for smoothness mean, XGBoost.

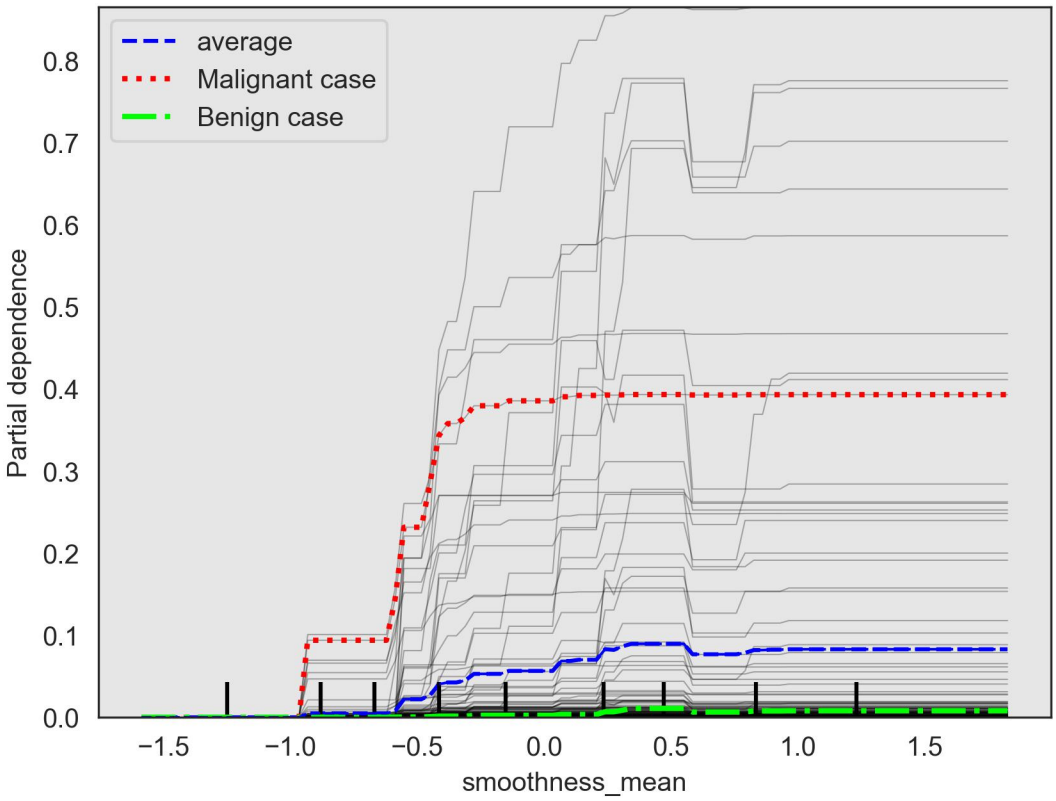

Figure 5.4: C-ICE for smoothness mean, XGBoost.

This thesis provides a conclusory explanation table for the patient we study with a positive malignancy diagnosis, for each of the local XAI methods presented. These tables aim to provide both clarity and ease of explanatory comparison between methods. The conclusory explanation table for ICE and C-ICE is shown in the table below.

Conclusory Explanation	**Feature Importance Ordering (Contributing To Malignancy)**	**Number of Model Predictions Used To Generate Explanation (Per Instance)[1]**	**Time To Completion In Seconds (Per Instance) [2]**
According to the ICE & C-ICE analysis on the XGBoost model, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is texture, followed by smoothness.	1. Texture 2. Smoothness 3. Concave points 4. Concavity 5. Area 6. Symmetry 7. Radius 8. Compactness 9. Fractal dimension 10. Perimeter	1000	2.934
[1] 100 permutations per feature, holding all other features constant. Multiplied by 10 features = 1000 predictions per instance.			
[2] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds			

Table 5.1: XGBoost Black Box - ICE & C-ICE Summary Table, Malignant Case.

5.2.2 LIME

This section details the outcome of applying LIME using a weighted logistic regression, with no penalty, to a malignant case from the BCW-D dataset. We chose not to construct our logistic regression using a penalty as it has been shown to be disadvantageous to LIME performance [24]. For both the malignant and the benign case, we compare and evaluate the outcome of using LIME using a kernel width of 2.3717 and a kernel width of 0.9. The kernel width of 2.3717 was selected as it is the default heuristic used in the LIME package created for python by Ribiero et al.[22]., and it is roughly 75% of the square root of the number of features (M=10). The kernel width of 0.9 was chosen randomly but with the intent of choosing a significantly more localized kernel than the default kernel width. The neighbourhood generated for the logistic regression used in LIME consisted of 4000 randomly generated data points following a normal distribution with mean 0 and standard deviation of 1. The weights used were generated using an exponential kernel as detailed in Equation 3.2. We provide extensive and novel illustration to help explain the inner workings of the XGBoost model on the malignant case studied.

As interpreting a logistic regression is not as straight forward as a simple linear regression, we will go into some detail regarding logistic regression interpretation. The formula for logistic regression is shown below;

$$y = \frac{1}{1 + e^{-z}} \tag{5.1}$$

In the above equation, z is the linear combination of coefficients produced by the logistic regression multiplied by the feature values;

$$z = z_0 + z_1 x_1 + z_2 x_2 + ... + z_m x_m \tag{5.2}$$

Furthermore, it is naive to take the value of the coefficients in z and use them as our feature importance interpretation, as the relationship between the coefficients and the output y is not linear [15].

Logistic regression gives us the probability that y = 1, in other words, the probability of some event y occuring. We can get a better understanding of how our coefficients influence the outcome of our model by looking at the odds of our model y.

The odds of some event y occurring is defined as follows;

$$Odds = \frac{P(y = 1)}{1 - P(y = 1)} = \frac{P(y = 1)}{P(y = 0)} \tag{5.3}$$

The equation for P(y=1) is given in 5.1. Furthermore, if we subtract P(y=1) from 1, we know that by the law of total probability that this gives us P(y=0);

$$P(y = 0) = 1 - \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{z}} \tag{5.4}$$

The odds of our model is then defined as follows;

$$Odds = \frac{1}{1 + e^{-z}} \div \frac{1}{1 + e^{z}} = e^{z} \tag{5.5}$$

If we want to understand how the odds change when a feature increases by a factor of 1 unit, we can look at the ratio of odds [15];

$$\frac{Odds_{x_{i+1}}}{Odds_{x_i}} = \frac{z_0 + z_1x_1 + z_ix_{i+1} + ... + z_mx_m}{z_0 + z_1x_1 + z_ix_i + ... + z_mx_m} \tag{5.6}$$

The above equation reduces to the following;

$$\frac{Odds_{x_{i+1}}}{Odds_{x_i}} = exp(z_i) \tag{5.7}$$

This means that for every unit increase for a given feature, the odds increase by a factor of $exp(z_i)$ [15].

However, we assert that looking at the absolute change in odds is also necessary to fully understand the explanation model, as a base odds that is relatively high or relatively low will skew the significance of the change. For example, if our base odds are equal to 50 (relatively high), and feature x_{i+1} increases the odds to 60, then our odds ratio will be 1.2. But let's say that feature x_{j+1} increases our odds to 100, then our odds ratio is 2. Feature x_i increased absolute odds by 10 but feature x_j increased odds by 50 which is 5 times as much as feature x_i. Furthermore, we also want to look at the absolute change in odds for each feature after making isolated unit increases to each feature. The absolute change in odds is defined as follows;

$$\Delta Absolute = Odds_{xi+1} - Odds_{xi} \tag{5.8}$$

In Figure 5.2 and in Figure 5.3 below, we summarize the results of our local

weighted logistic regression for the malignant case from BCW-D using a kernel width of 2.3717 (default width) and 0.9 respectively. The column 'Value' details the feature values for the malignant case, and the right-adjacent column details the logistic regression coefficients from our local LIME model. Additionally the table includes the black box prediction which is 1 (malignant), the local prediction which is also 1.

Black box accuracy details the accuracy between the predictions from our XGB model on the BCW-D test set and the true BCW-D target values. Local accuracy is the accuracy between the true BCW-D target values and the predictions for these target values generated by the local logistic LIME models. The LIME neighbourhood accuracy is the accuracy between the predictions from our XGB model and the logistic LIME model on the 4000 generated neighborhood points, taking the XGB predictions as our true values since the local interpretable LIME model seeks to explain the XGB model. The LIME R^2 score is the R^2 score of the predictions from our XGB model and the logistic LIME model on the 4000 generated neighborhood points.

Upon reading the below LIME summary tables for KW = 2.3717 and KW = 0.9, we take note that feature importance order based solely on the logistic regression coefficient values varies between the two differing kernel widths, though the top 5 contributing features are the same. The top 5 contributing features based solely on our Logistic LIME model coefficients are concave points, texture, area, concavity, and smoothness.

Malignant Case LIME Summary
Logistic Regression as Local Interpretable Model , Kernel Width = 2.3717 (Default)

Blackbox Prediction	1
Local Prediction	1
Blackbox Accuracy	96%
Local Accuracy	92%
LIME Neighbourhood Accuracy	84%
LIME R^2 Score	50%

Feature	Value	Logistic Regression Coefficient
Texture	0.76	2.16
Area	-0.18	2.00
Concave Points	0.43	1.72
Concavity	0.66	0.84
Smoothness	1.71	0.65
Perimeter	-0.05	0.49
Symmetry	0.09	0.15
Radius	-0.10	0.00
Fractal Dimension	0.90	-0.16
Compactness	0.46	-0.23

Table 5.2: XGBoost Black Box - Malignant Case LIME Summary, Kernel Width=2.3717

Malignant Case LIME Summary
Logistic Regression as Local Interpretable Model , Kernel Width = 0.9

Blackbox Prediction	1
Local Prediction	1
Blackbox Accuracy	96%
Local Accuracy	94%
LIME Neighbourhood Accuracy	84%
LIME R^2 Score	57%

Feature	Value	Logistic Regression Coefficient
Concave Points	0.43	3.19
Texture	0.76	3.04
Area	-0.18	2.83
Concavity	0.66	2.17
Smoothness	1.71	0.86
Perimeter	-0.05	0.60
Symmetry	0.09	0.60
Radius	-0.10	0.04
Compactness	0.46	0.03
Fractal Dimension	0.90	-0.62

Table 5.3: XGBoost Black Box - Malignant Case LIME Summary, Kernel Width=0.9

It is important to consider more than just the logistic regression coefficient itself when evaluating a LIME model that uses a logistic regression as its interpretable model. The below graphs detail the relative change in odds that each feature contributes when considering a logistic LIME model with kernel widths 2.3717 and 0.9 respectively. As a reminder, this calculation is simply e^{z_i}, in otherwords it is e to the exponent of the respective feature coefficient value as shown in Equation 5.7. The relative change in odds for all features while using a smaller kernel width of 0.9 are higher for all features than that of the default kernel width of 2.3717. As we use smaller kernel widths in LIME we tend to see higher adherence to the local ML model, which in our case is evidenced by an R^2 score of .57 for our 0.9 model and .5 for our default kernel width model.

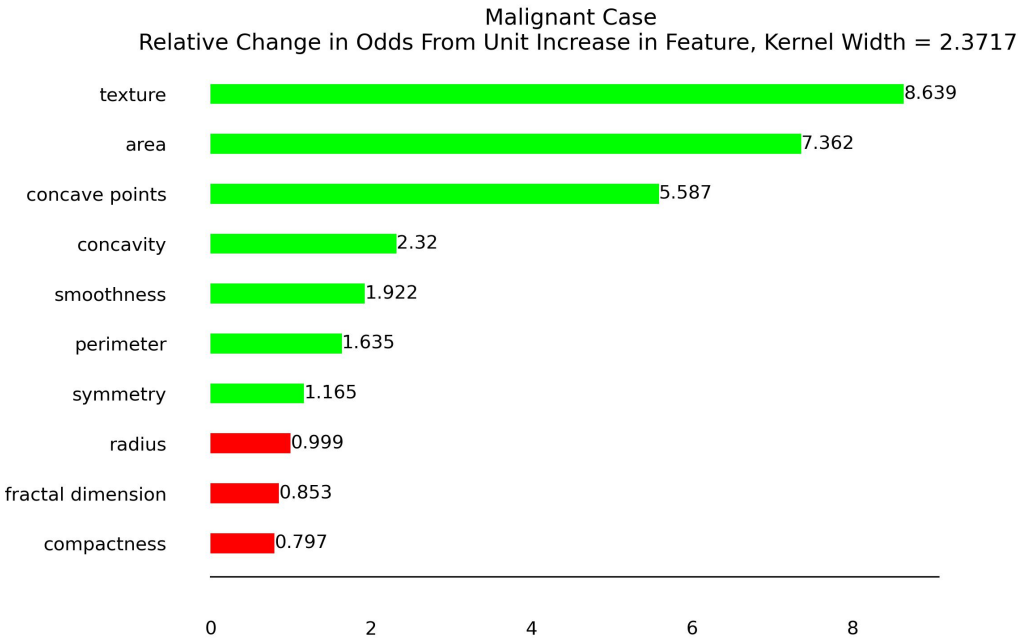

Figure 5.5: XGBoost Black Box - Malignant Case, Relative Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 2.3717

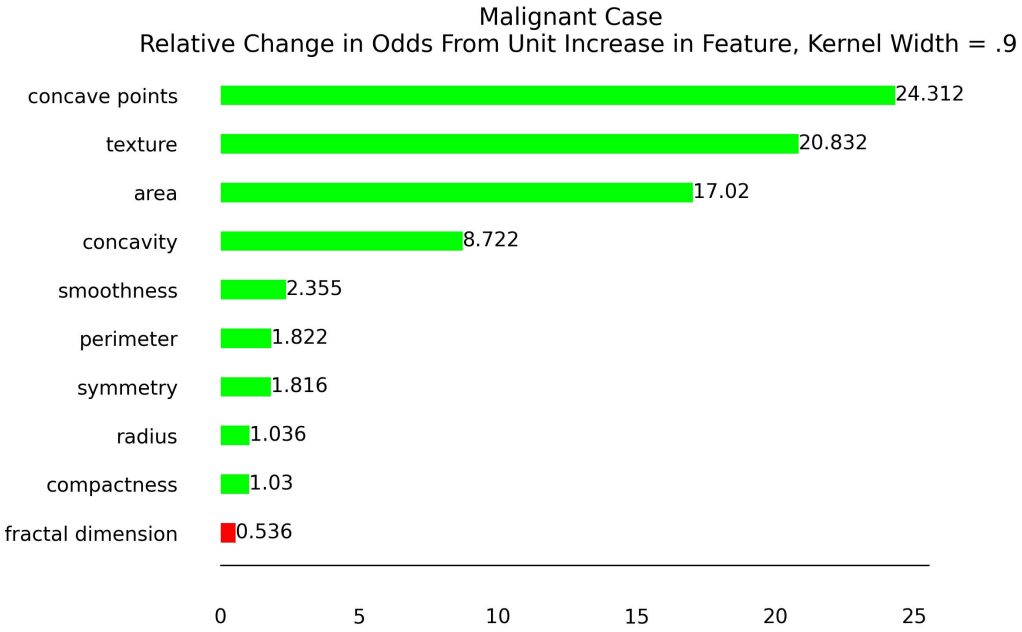

Figure 5.6: XGBoost Black Box - Malignant Case, Relative Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 0.9

Next we take a look at the equivalent graphs demonstrating the absolute change in odds from isolated unit increases in each of the feature values. Here we can understand the impact that each feature has on the change in odds, while eliminating the issue of base odds skewing relative importance. We notice that both the 0.9 and 2.3717 kernel width models have the same top 5 and bottom 5 contributing features. This relationship holds true for the relative change in odds as well. Furthermore, because the features contained in each of the aforementioned feature importance buckets are the same for both kernel width LIME models, we can predict that there is a fair chance that the feature importance ordering of true Kernel SHAP values may be somewhat similar. We will examine this patient's malignancy diagnosis from a Kernel SHAP perspective in Subsection 5.2.3.

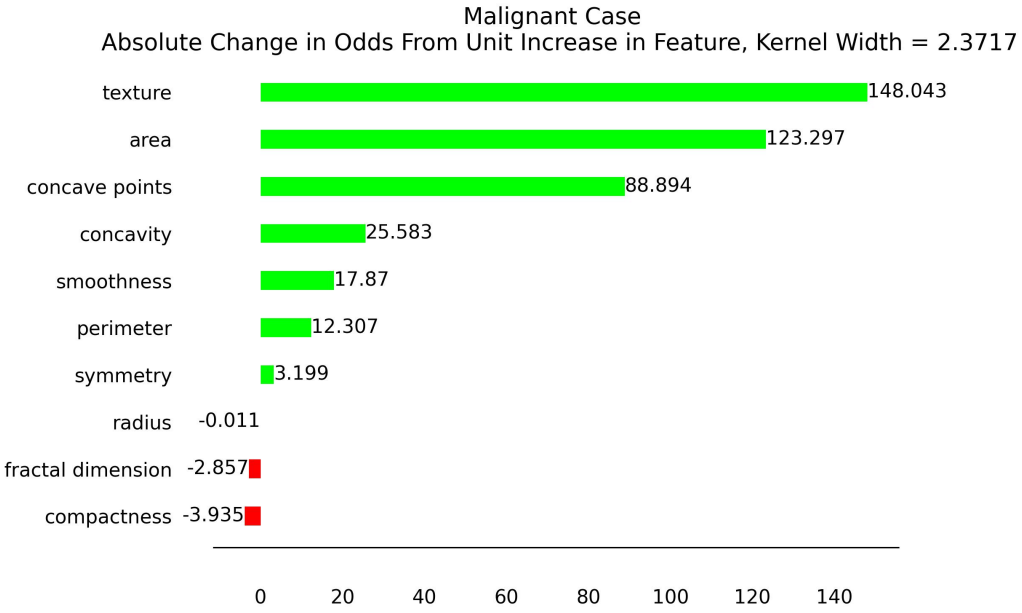

Figure 5.7: XGBoost Black Box - Malignant Case, Absolute Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 2.3717

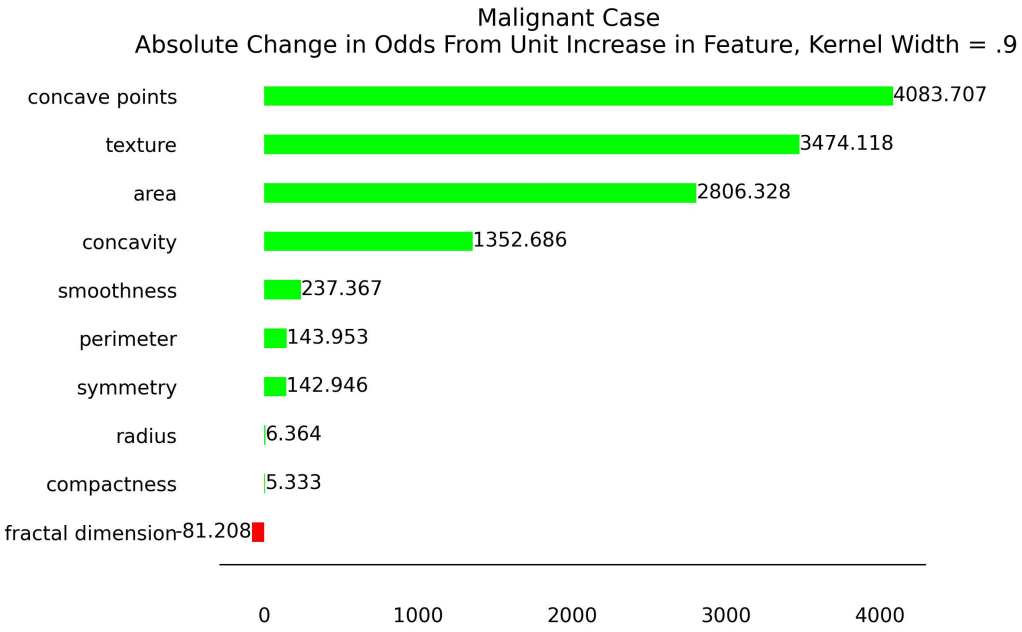

Figure 5.8: XGBoost Black Box - Malignant Case, Absolute Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 0.9

In Figures 5.9, and 5.10, we take the absolute value (modulus) of the absolute change in odds from each of the isolated unit increases and compare them to each other on a percentage basis of the sum of all the absolute values of absolute change in odds. We use the absolute value of the absolute change in odds, because producing a proportional visualization such as a pie chart requires this stipulation. We see that in the 2.3717 kernel width LIME model that texture is ranked as the highest as a percentage of the total sum of absolute value of the absolute change in odds, whereas in the 0.9 kernel width LIME model the top percentage contribution is concave points. The equivalent tables and figures for the benign case can be found in Appendix B.

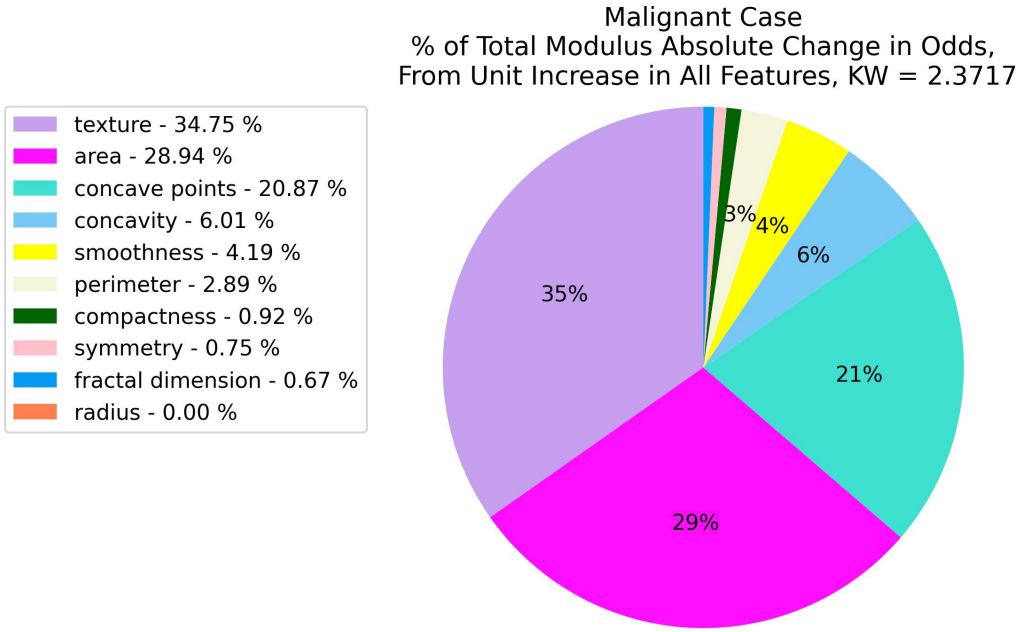

Figure 5.9: XGBoost Black Box - Malignant Case, % Percent of Total Modulus (Absolute Value) Absolute Change in Odds from Isolated Unit Increase For All Features, Kernel Width = 2.3717

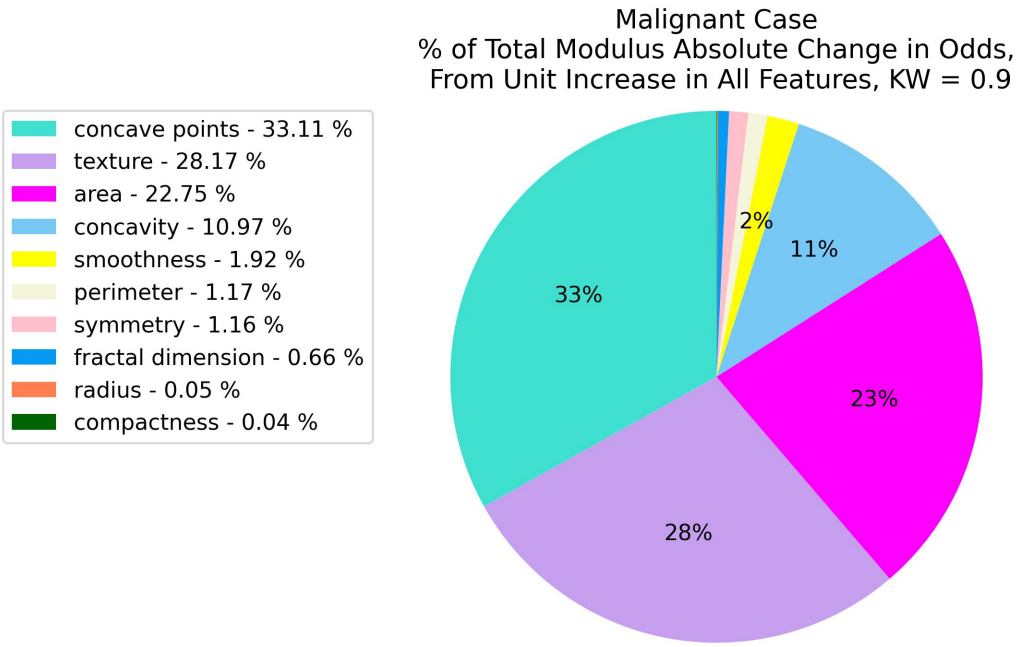

Figure 5.10: XGBoost Black Box - Maligant Case, % Percent of Total Modulus (Absolute Value) Absolute Change in Odds from Isolated Unit Increase For All Features, Kernel Width = 0.9

The conclusory explanation tables for the LIME method using a kernel width of 2.3717 and 0.9 are shown below for the malignant case studied in this thesis. We note that the top 5 features contributing to malignancy are the same for both LIME models, however the ordering is different and the top contributing feature is texture for the 2.3717 kernel width model and concave points for the 0.9 model.

Conclusory Explanation	Feature Importance Ordering (Contributing To Malignancy)	Number of Model Predictions Used To Generate Explanation (Per Instance)	Time To Completion In Seconds (Per Instance) [1]
According to LIME applied to the XGBoost model, using a kernel width of 2.3717, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is texture, followed by area.	1. Texture 2. Area 3. Concave points 4. Concavity 5. Smoothness 6. Perimeter 7. Symmetry 8. Radius 9. Fractal dimension 10. Compactness	4000	11.736
[1] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds			

Table 5.4: XGBoost Black Box - LIME Summary Table, Malignant Case, Kernel Width 2.3717

Conclusory Explanation	Feature Importance Ordering (Contributing To Malignancy)	Number of Model Predictions Used To Generate Explanation (Per Instance)	Time To Completion In Seconds (Per Instance) [1]
According to LIME applied to the XGBoost model, using a kernel width of 0.9, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is concave points, followed by texture.	1. Concave points 2. Texture 3. Area 4. Concavity 5. Smoothness 6. Perimeter 7. Symmetry 8. Radius 9. Compactness 10. Fractal dimension	4000	11.736
[1] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds			

Table 5.5: XGBoost Black Box - LIME Summary Table, Malignant Case, Kernel Width 0.9

5.2.3 Kernel SHAP

In this Section we detail the result of generating exact (true) Kernel SHAP values for our malignant case studied in this thesis. In this Section we provide novel illustrations to enhance explanatory usefulness of the underlying XGBoost model as it relates to true Kernel SHAP values. We will use these same illustrations in Section 5.3 wherein we attempt to approximate these true Kernel SHAP values. The equivalent figures for the benign case can be seen in Appendix C. In Figure 5.11 we see that the largest contributing feature to the positive malignant diagnosis is texture, as noted by the purple coloured bar with a Kernel SHAP value of 0.232. The second highest contributing feature to the diagnosis is concave points, as noted by the aqua bar with a Kernel SHAP value of 0.189. The third highest contributing feature toward the diagnosis is concavity, and it is followed then smoothness in fourth place. Figure 5.12 shows an alternative visualization of the contributing Kernel SHAP values for this patient. We note that the top 5 and bottom 5 contributing features for this patient's malignant diagnosis based on Kernel SHAP are the same as the top 5 contributing factors in both of the LIME models, however the exact ordering of the feature importance does vary.

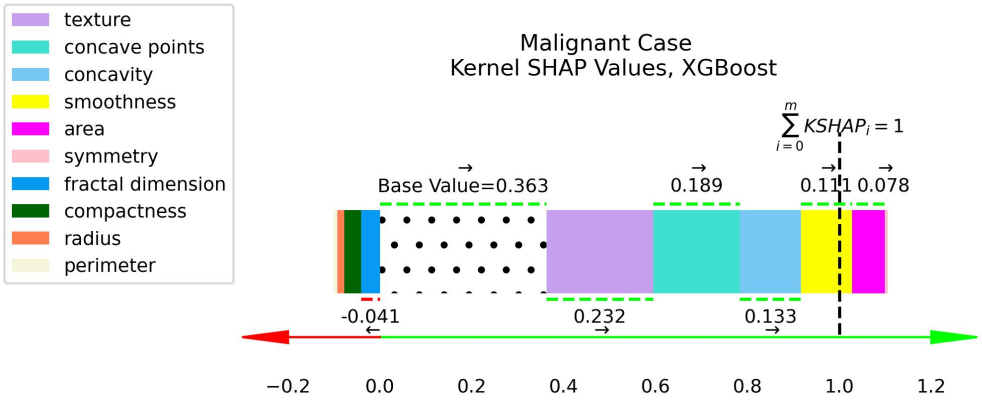

Figure 5.11: XGBoost Black Box - Exact Kernel SHAP values - Malignant Case, Kernel SHAP Additive Illustration

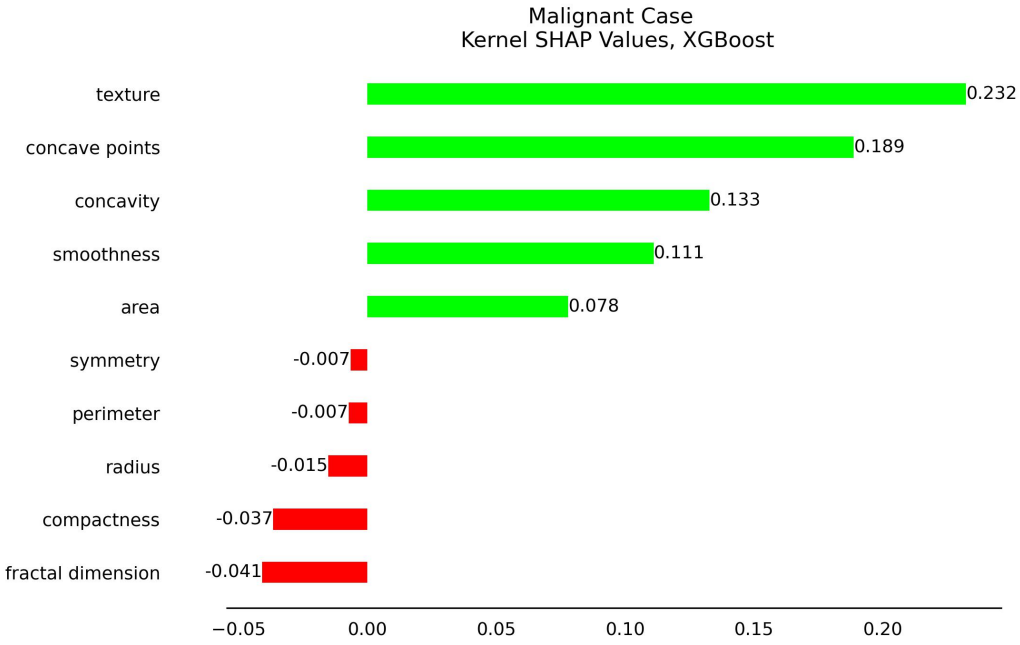

Figure 5.12: XGBoost Black Box - Exact Kernel SHAP values - Malignant Case, Kernel SHAP Bars Illustration

This next visualization is generated by summing the absolute value of the Kernel SHAP values for this patient, and then expressing the contribution of each individual Kernel SHAP value as a percentage of that sum. This is similar to the visualization that we produce in our LIME analysis of this patient in Figures 5.9, and 5.10. We see that texture and concave points dominate the malignant diagnosis for this patient with a 50% contribution to the absolute value sum. The top three contributing features, which includes concavity, contribute 65% of the sum of the absolute Kernel SHAP values. The top four features contribute 78% of this sum. Our Kernel SHAP analysis of this patient demonstrates that the contribution of each feature is non-uniform.

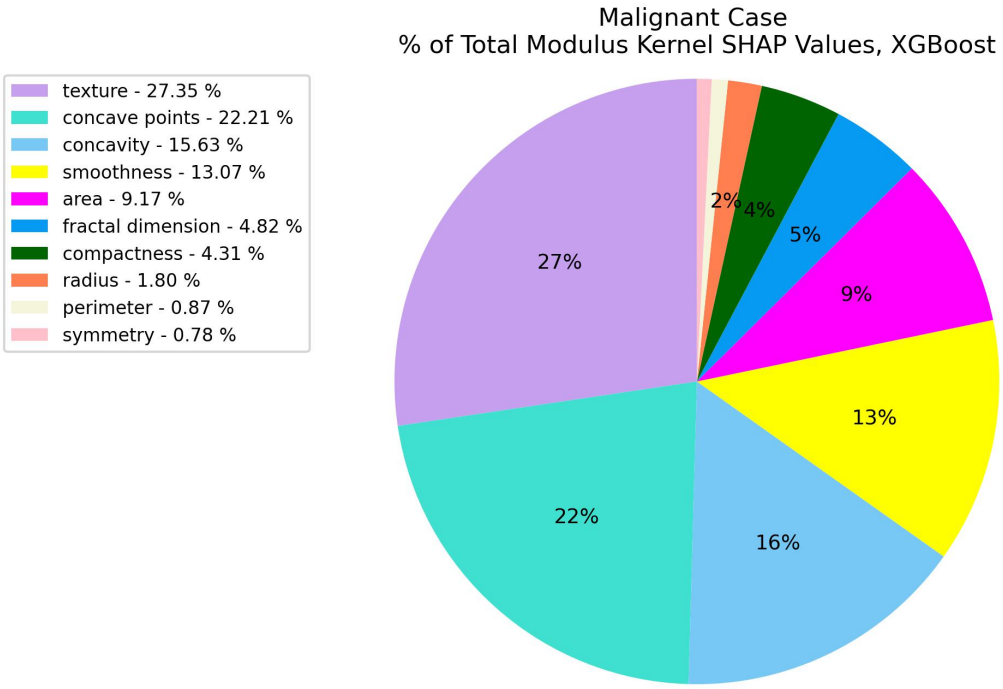

Figure 5.13: XGBoost Black Box - % of Total Modulus Kernel SHAP Values - Malignant Case

The conclusory explanation table for true Kernel SHAP values for the malignant case studied in this thesis is shown below. We note that the top 5 features contributing to malignancy are the same for both LIME models, however the ordering differs.

Conclusory Explanation	**Feature Importance Ordering (Contributing To Malignancy)**	**Number of Model Predictions Used To Generate Explanation (Per Instance)** [1]	**Time To Completion In Seconds (Per Instance)** [2]
According to Kernel SHAP applied to the XGBoost model, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is texture, followed by concave points.	1. Texture 2. Concave points 3. Concavity 4. Smoothness 5. Area 6. Symmetry 7. Perimeter 8. Radius 9. Compactness 10. Fractal dimension	465920	1367.00928
[1] The total number of coalitions is 2^{10} , muliplying this by 455 training samples gives us 465920 model predictions per instance			
[2] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds			

Table 5.6: XGBoost Black Box - True Kernel SHAP Summary Table, Malignant Case

5.3 Kernel SHAP Approximation Methods

This section details the results of applying three Kernel SHAP approximation techniques; (1) Using LIME to infer Kernel SHAP, (2) the Biased Kernel SHAP algorithm presented by Covert et al. [5] and (3) Our algorithm, Fixed Biased Kernel SHAP.

5.3.1 Using LIME to predict Kernel SHAP

There are two general metrics of comparison we will use to evaluate the differences between the explanations produced by our logistic LIME models using 2.3717 and 0.9 kernel widths versus true Kernel SHAP values for the BCW-D test set. These two metrics are 1) concordance index, and 2) Mean Squared Error.

The first metric we want to consider is concordance index, because it will give us insight into how the ordering of feature importance compares between the two methods. Concordance index is a measure of the number of concordant pairs between two vectors, divided by the total number of pairs. Applied in our context, concordance index for a given test set explanation is the number of concordant pairs between the Kernel SHAP values and the LIME model feature importance. If the true Kernel SHAP values for a given instance are such that $KSHAP_{xi} > KSHAP_{xj}$ for any two features of a given instance, and the LIME model feature importance for that instance indicate that $LIME_{xi} > LIME_{xj}$ then this constitutes a single concordant pair. To find the concordance index, we total the number of concordant pairs between Kernel SHAP and LIME and divide it by the total

possible pairs of features. This metric gives us insight into how the ordering of Kernel SHAP values compare to the order of LIME feature importance values. Naturally, a concordance index of 1 is the highest attainable value and a higher concordance index indicates better performance.

In order to compare our LIME generated feature importances to Kernel SHAP, we use the non-absolute value of the absolute change in odds from isolated unit increases in each of the features from our LIME models. Since we know that the sum of Kernel SHAP values for a given instance will equal the black box prediction less the base value from the training set, we can use this sum alongside our absolute change in odds to generate Kernel SHAP predictions from our LIME models. Because we are using a logistic model as our explainable model, we cannot simply take the coefficient values as our feature importance. As a reminder, the absolute change in odds for a isolated unit increase in a given feature is given by Equation 5.8. If we take the sum of these isolated increases and then divide each of the absolute change in odds for each of the features by this sum, then multiply these percentages by the sum of the Kernel SHAP values for a given instance (which we know by taking the predicted value less the base value) and then we have inferred the Kernel SHAP values using LIME for that instance. Naturally, if the sum of the absolute change in odds is relatively high or relatively low, this could inflate some of the inferred Kernel SHAP values. Furthermore, this Kernel SHAP approximation method prioritizes speed over accuracy. Of course we recognize this as an imperfect approximation, as the percentage contributions from LIME are generated taking the sum of isolated unit increases in each fea-

ture. However, using such an imperfect approximation of Kernel SHAP values by LIME inference is still useful as we demonstrate below. Using these inferred Kernel SHAP values for logistic LIME models that use a kernel width of 2.3717 and 0.9, we compare the inferred Kernel SHAP approximations with the actual Kernel SHAP values for each of the test set instances in BCW-D. We plot the results of the concordance index for each of the test set instances in the histograms below. We observe an average concordance index of 0.76 for inferred Kernel SHAP values using both the default kernel width and a kernel width of 0.9. However, there is less variability from the inferred Kernel SHAP predictions using the default kernel width versus a kernel width of 0.9, as evidenced by a lower total sum of squares of 1.27 versus 1.37.

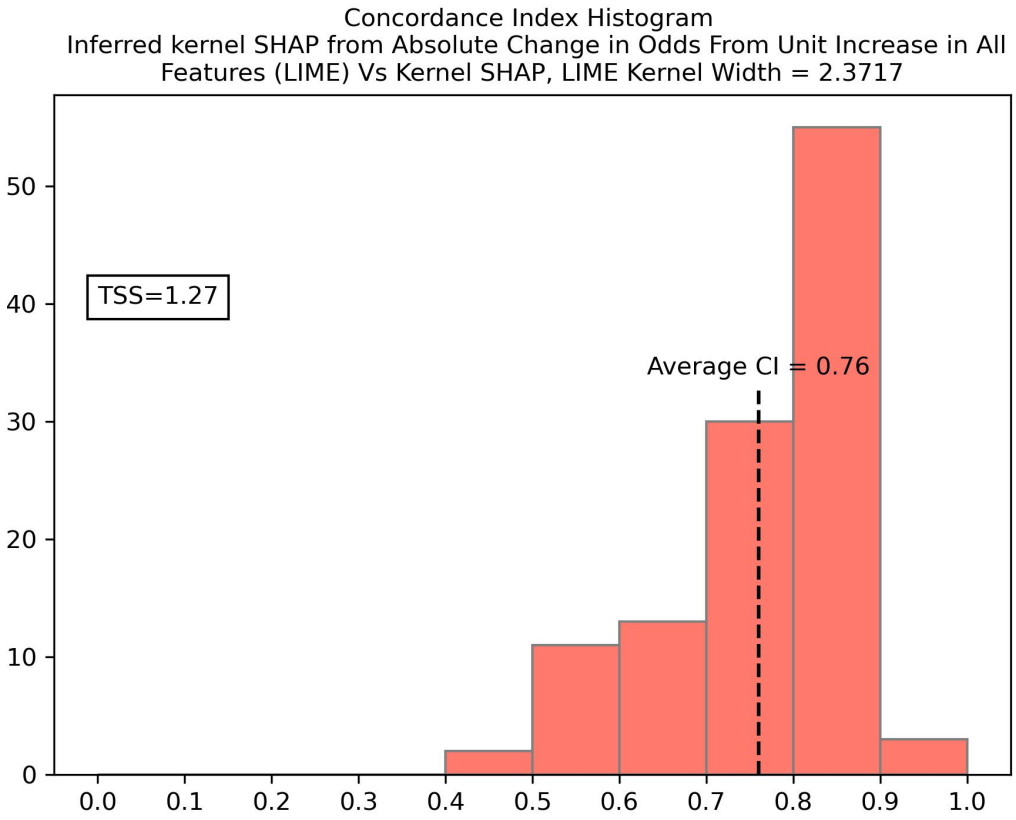

Figure 5.14: XGBoost Black Box - Comparing LIME vs Kernel SHAP - Concordance Index of Inferred Kernel SHAP from Absolute Change in Odds From Isolated Unit Increase in All Features (LIME) Vs Kernel SHAP, Kernel Width = 2.3717

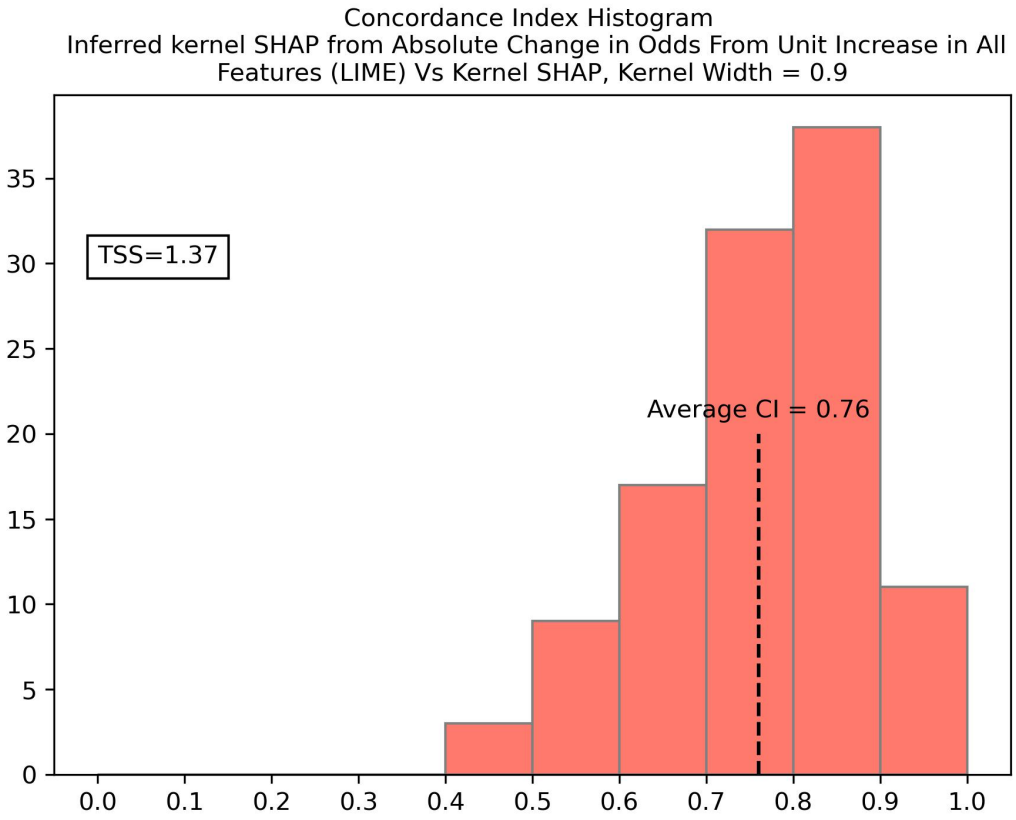

Figure 5.15: XGBoost Black Box - Comparing LIME vs Kernel SHAP - Concordance Index of Inferred Kernel SHAP from Absolute Change in Odds From Isolated Unit Increase in All Features (LIME) Vs Kernel SHAP, Kernel Width = 0.9

The second metric we want to evaluate is the mean squared error between our inferred Kernel SHAP values using our two logistic LIME models of varying kernel widths. The histogram of these results can be seen in the below figures. We observe an average MSE between inferred Kernel SHAP values and actual Kernel SHAP values of 0.04 while using a LIME kernel width of 2.3717, and 0.06 while using a LIME kernel width of 0.9. These results, compared with the concordance index histograms from above, suggest that using a wider LIME kernel width such as 2.3717 in conjunction with a logistic explainable model is more ideal than using smaller kernel widths if the goal is to predict Kernel SHAP values using LIME.

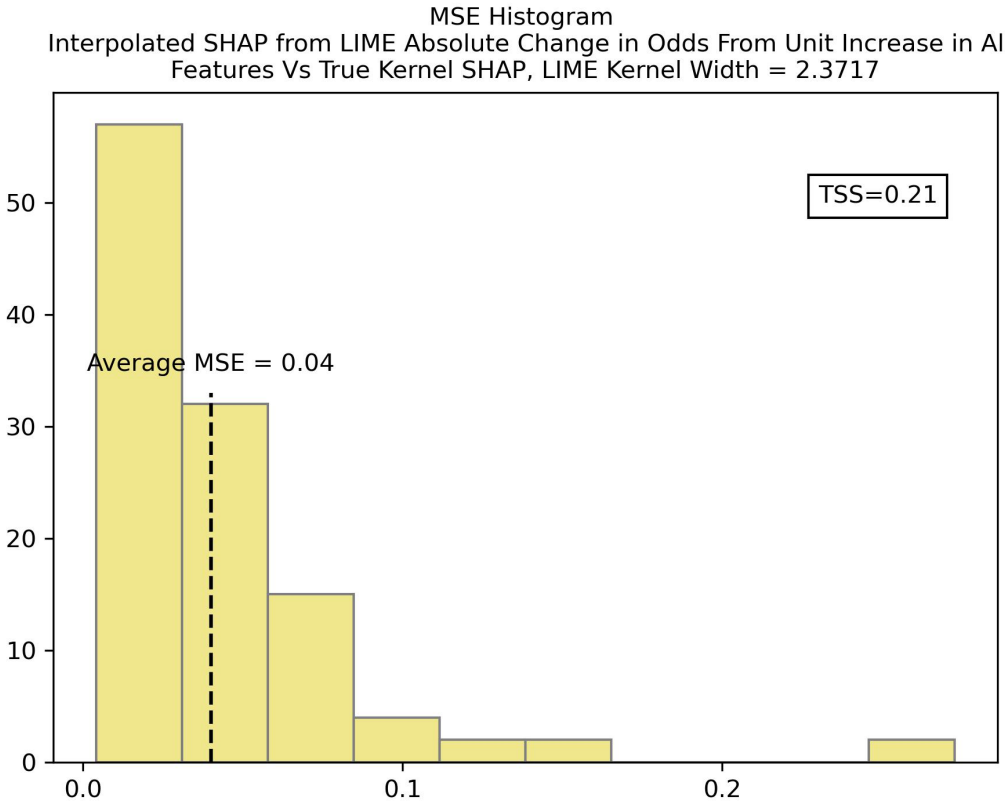

Figure 5.16: XGBoost Black Box - Comparing LIME vs Kernel SHAP - Mean Squared Error of Inferred Kernel SHAP from Absolute Change in Odds From Isolated Unit Increase in All Features (LIME) Vs Kernel SHAP, Kernel Width = 2.3717

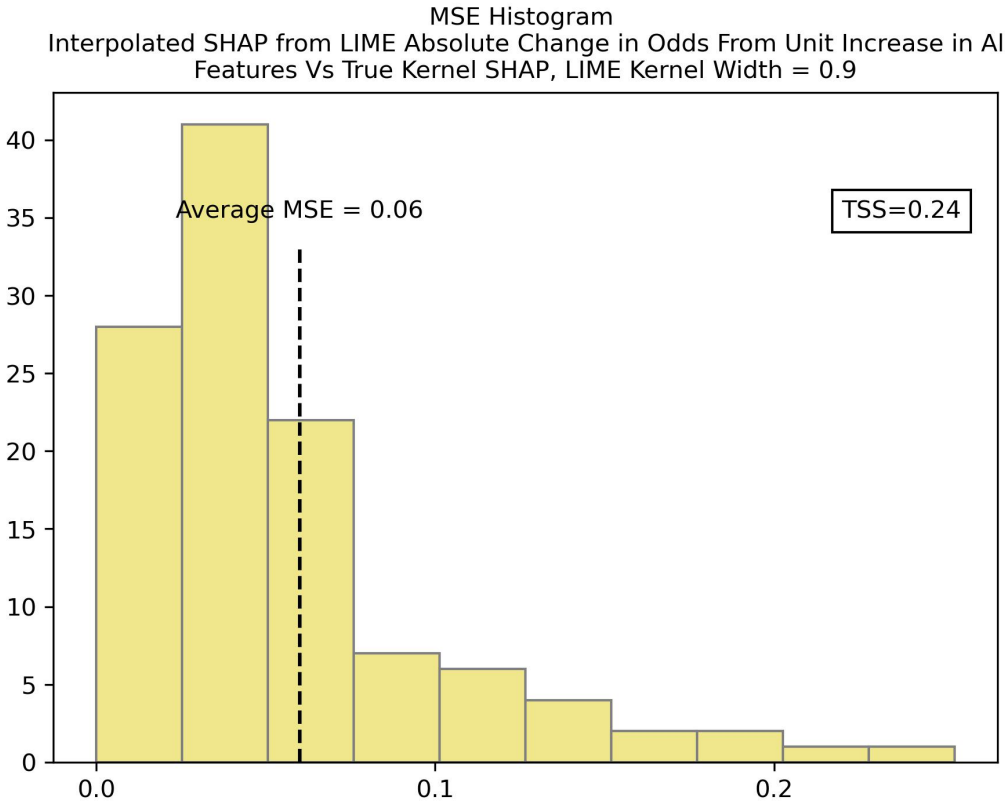

Figure 5.17: XGBoost Black Box - Comparing LIME vs Kernel SHAP - Mean Squared Error of Inferred Kernel SHAP from Absolute Change in Odds From Isolated Unit Increase in All Features (LIME) Vs Kernel SHAP, Kernel Width = 0.9

The following graphs depict the Kernel SHAP approximation using LIME to infer the Kernel SHAP values on the patient with a malignant tumor studied in this thesis, using a kernel width of 2.3717. The equivalent figures for the benign case can be seen in Appendix D. In figure 5.18 we see that the largest contributing feature to the positive malignant diagnosis is texture, as noted by the purple coloured bar with an approximated Kernel SHAP value of 0.228. The second highest contributing feature to the diagnosis is area, as noted by the magenta bar with an approximated Kernel SHAP value of 0.19. The third highest contributing feature toward the diagnosis is concave points, and it is followed then concavity in fourth place. Figure 5.19 shows an alternative visualization of the contributing Kernel SHAP value approximations for this patient.We note that the top 5 and bottom 5 contributing features for this patient's malignant diagnosis are the same as the top 5 contributing factors when compared to the true Kernel SHAP values and both of the LIME models, however the exact ordering of the feature importance does vary.

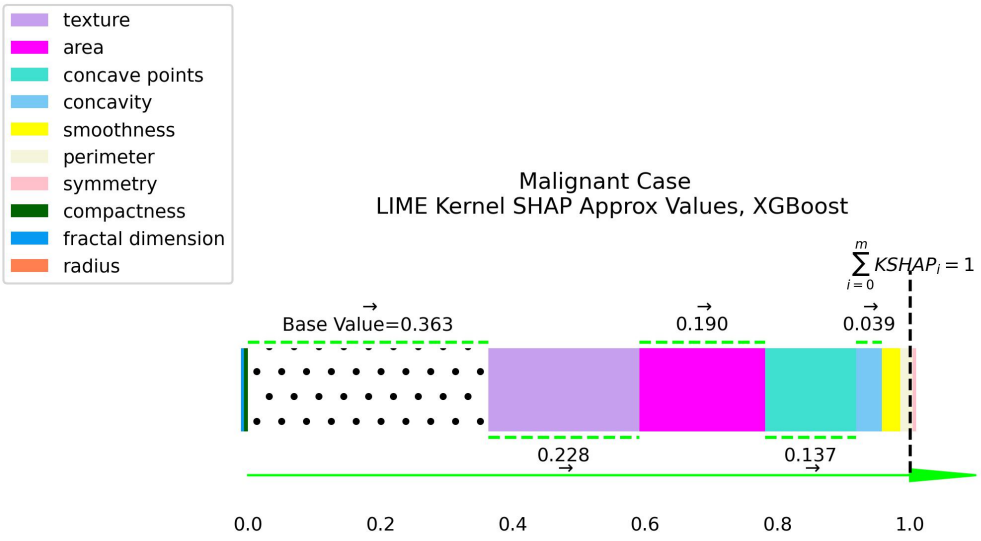

Figure 5.18: XGBoost Black Box - LIME Inferred Kernel SHAP Approximation - Malignant Case, Additive Illustration, Kernel Width = 2.3717

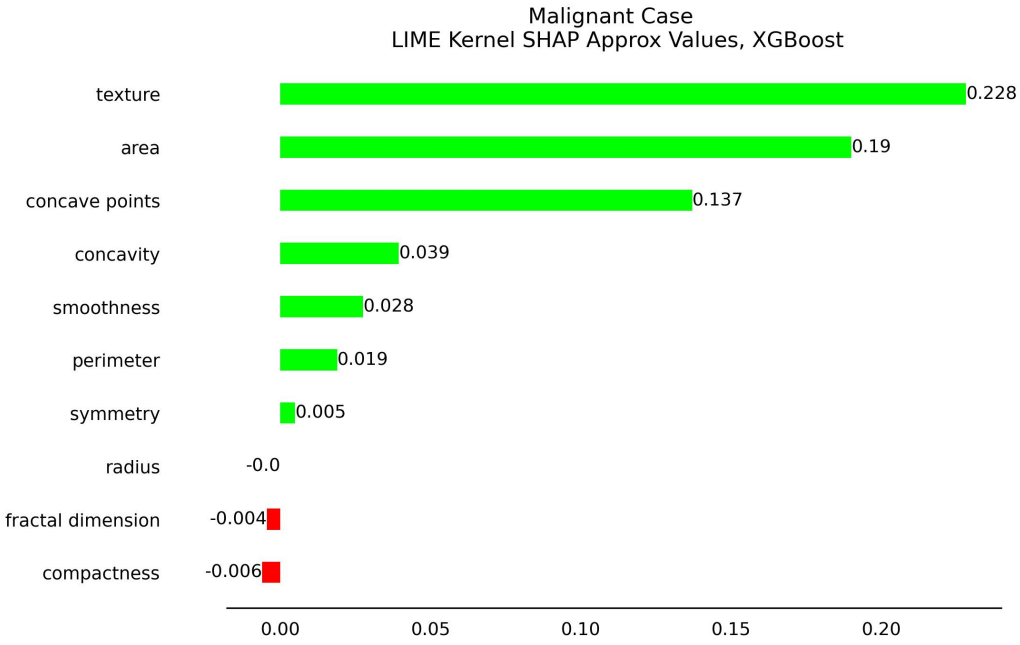

Figure 5.19: XGBoost Black Box - LIME Inferred Kernel SHAP Approximation - Malignant Case, Bars Illustration, Kernel Width = 2.3717

As mentioned previously, this next visualization is generated by summing the absolute value of the Kernel SHAP approximation values for this patient, and then expressing the contribution of each value as a percentage of that sum. We see that texture and area dominate the malignant diagnosis for this patient with a 64% contribution to the absolute value sum. The top three contributing features, which includes concave points, contribute 85% of the sum of the absolute Kernel SHAP values. The top four features, which includes concavity, contribute 91% of this sum. Ultimately, the results of the Kernel SHAP approximation using LIME inference inflates the contribution of the top contributing features when compared to the true Kernel SHAP values. For the malignant case, using LIME inference to approximate Kernel SHAP values produces a squared error of 0.034 for the Kernel SHAP value approximations for all 10 features.

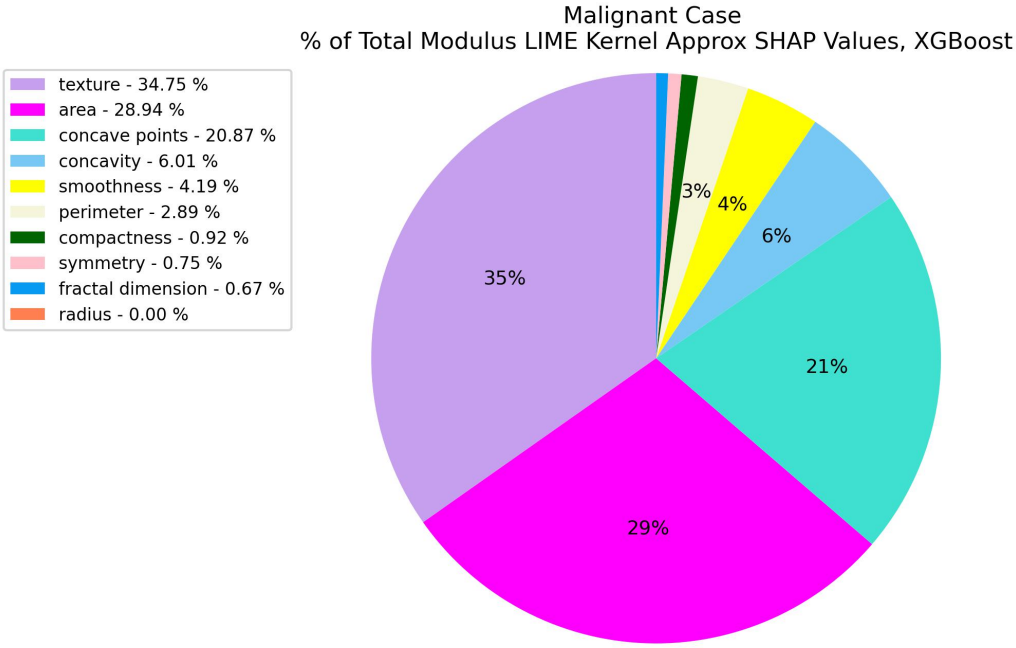

Figure 5.20: XGBoost Black Box - % of Total Modulus Approx Kernel SHAP Values using LIME Inferred Kernel SHAP Approximation - Malignant Case, Kernel Width =2.3717

The conclusory explanation table for the Lime Inferred Kernel SHAP approximation algorithm is presented below.

Conclusory Explanation	Feature Importance Ordering (Contributing To Malignancy)	Number of Model Predictions Used To Generate Explanation (Per Instance)	Time To Completion In Seconds (Per Instance) [1]	Squared Error To True Kernel SHAP Values (Per Instance) [2]
According to the LIME Inference Kernel SHAP approximation method applied to the XGBoost model, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is texture, followed by area.	1. Texture 2. Area 3. Concave points 4. Concavity 5. Smoothness 6. Perimeter 7. Symmetry 8. Radius 9. Fractal dimension 10. Compactness	4000	11.736	0.034
[1] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds				
[2] This is the squared error between the Kernel SHAP approximation values and the true Kernel SHAP values, across all 10 features				

Table 5.7: XGBoost Black Box - LIME Inferred Kernel SHAP Approximation Summary Table

5.3.2 Fixed Biased Kernel SHAP versus Biased Kernel SHAP

In this Subsection we look at how our Fixed Biased Kernel SHAP approximation method compares to Biased Kernel SHAP.

To do so we ran our approximation algorithm and Kernel SHAP on the BCW-D test set. We performed our Fixed Biased Kernel SHAP approximation method, and Biased Kernel SHAP, on each of the 114 samples in the test set from the BCW-D dataset, for an underlying XGB black box model. We evaluate both of these algorithms compared to exact (true) Kernel SHAP on the basis of mean squared error. We compared the results of these algorithms using a varying number of coalition samples to see how they performed respective to each other when using relatively low versus relatively high number of coalition samples.

To evaluate the effectiveness of both Fixed Biased Kernel SHAP and Kernel SHAP, we calculated the true (exact) Kernel SHAP values for the BCW-D test set values for the underlying black box model, XGBoost. Exact Kernel SHAP values are found by enumerating the entire powerset of coalitions. This is very computationally expensive as for a single instance that you wish to explain, it requires $2^M - 2$ model evaluations for every instance in the training data set, where M is the number of features in the dataset. We subtract 2 from 2^M because the model evaluation for the grand coalition is known (it is simply the output of the black box model with all features present), and the evaluation of the 0 or null coalition is simply the average of the model output over the training data. For simplicity, we will write 2^M going forward. So for a given instance in the test set, we completely enumerate its powerset, simulating 'missing' features by using

the training set used to train the black box model. Thus we must perform this enumeration of a single explanatory point for each of the training samples that the black box model was built upon. So in fact, to get exact Kernel SHAP values for a single instance, it is not simply 2^M number of model evaluations we must perform, but rather we must multiply 2^M by the number of training samples as each coalition will require us to produce model predictions in the amount of however large the training set is. To reduce the additional computational expense due to training data size, a machine learning practitioner may opt to apply clustering techniques to summarize the training data. Since the size of the training data set does not add exponentially higher computational expense but rather linearly higher computational expense, we do not focus on reducing the impact on run time from training data size in this thesis.

The results of our Fixed Biased Kernel SHAP approximation method at t=2 versus original Biased Kernel SHAP is seen below. The mean squared error is the average across the BCW-D test set of 114 patients, wherein the squared error between the Kernel SHAP approximation methods and true Kernel SHAP values is summed for all 10 features. Thus this is the average (mean) across the BCW-D test set, of the total squared error between the Kernel SHAP approximations and true Kernel SHAP.

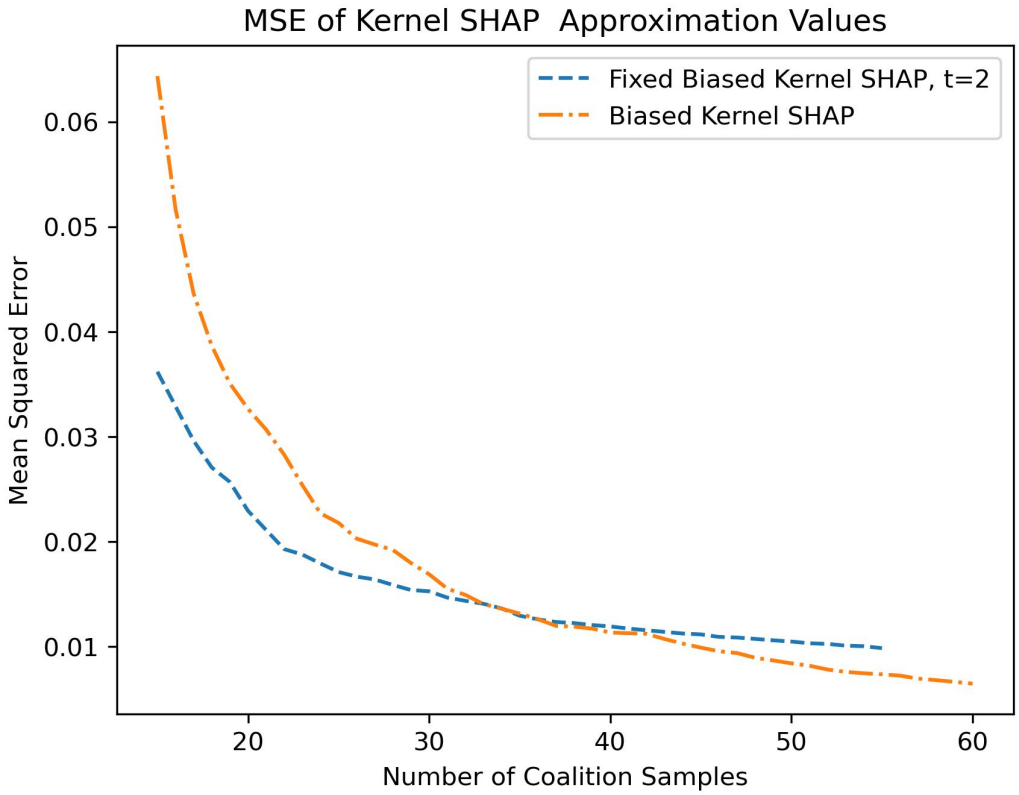

Figure 5.21: Mean Squared Error of Fixed Biased Kernel SHAP approximations and Biased Kernel SHAP for t=2, XGBoost

We observe that our algorithm, Fixed Biased Kernel SHAP, outperforms Biased Kernel SHAP for a low number of coalition samples at t=2. This is evidenced by observing the intersection of the blue line (our algorithm) with the orange line (Kernel SHAP) at approximately 36 sample coalitions. Furthermore we can conclude that our algorithm provides superior Kernel SHAP approximations when considering a relatively low number of samples and a t value of size 2.

The results of our Fixed Biased Kernel SHAP approximation method versus original Biased Kernel SHAP is seen below for t=3.

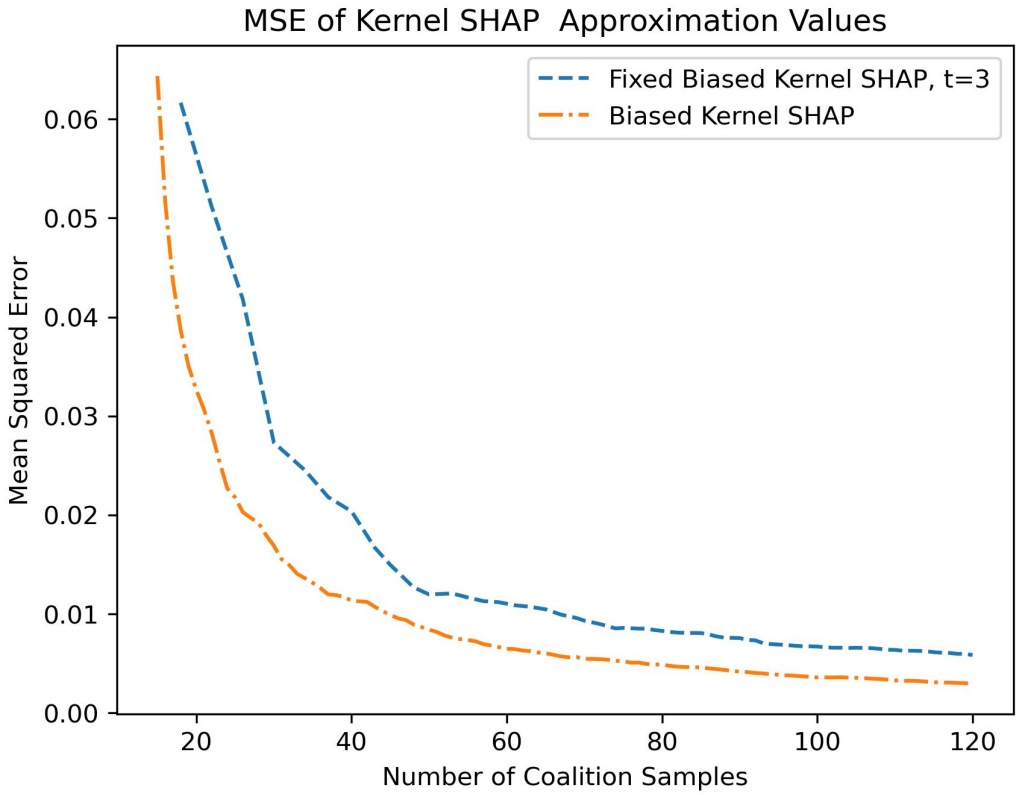

Figure 5.22: Mean Squared Error of Fixed Biased Kernel SHAP approximations and Biased Kernel SHAP for t=3, XGBoost

It is clear that the Biased Kernel SHAP algorithm is sensitive to the chosen level of t, as our algorithm does not provide superior Kernel SHAP approximations to Biased Kernel SHAP at t=3 based on mean squared error. If a practitioner has a large number of explanation points for which they wish to produce Kernel SHAP approximations for, and if run time per explanation is of utmost consideration, they may take a smaller subset of such explanations and fine tune the t parameter using Bayesian optimization or any number of other optimization techniques. Ultimately, it is a difficult task to approximate true Kernel SHAP values in a more efficient and accurate manner than the Biased Kernel SHAP algorithm. However, our algorithm provides an efficient way of producing reasonably accurate Kernel SHAP approximations for a small number of coalitions. For example, at t=2 and 25 coalitions, the MSE between Fixed Biased Kernel SHAP and true Kernel SHAP across the entire 114 test set instances in the BCW-D dataset is .017. At t=3 and 25 coalitions, the MSE between Fixed Biased Kernel SHAP and true Kernel SHAP across the entire 114 test set instances is approximately .042. This same figure at 25 coalitions for Biased Kernel SHAP is .022. Recall that 25 coalitions corresponds to 11,375 model predictions when we account for the 455 training samples used to train the XGBoost model. We computed an average model prediction speed of .002934 and multiplying this number by 11,375 gives us an approximate time to completion of 33.37 seconds per explanation point. Our algorithm outperforms Biased Kernel SHAP for t=2 up until approximately 36 sampled coalitions. Of course, if the highest accuracy is of utmost consideration then a higher number of coalition samples and using the original Biased

Kernel SHAP algorithm will lead to more accurate approximations to true Kernel SHAP values, but will come at a cost of lower computational efficiency.

The following graphs depict the Kernel SHAP approximation for the malignant case using the Fixed Biased Kernel SHAP approximation method presented in the algorithm table in Section 4.2, at a t=2 and sampling 25 coalitions. Recall that 25 coalitions is equivalent to 11,375 model predictions as each coalition has 455 training samples. The equivalent figures for the benign case can be seen in Appendix D. In Figure 5.23 we see that the largest contributing feature to the positive malignant diagnosis is texture, as noted by the purple coloured bar with an approximated Kernel SHAP value of 0.145. The second highest contributing feature to the diagnosis is concave points, as noted by the aqua bar with an approximated Kernel SHAP value of 0.142. The third highest contributing feature toward the diagnosis is concavity, and it is followed then smoothness in fourth place. In fact, the top 5 contributing features are ordered in the exact same manner as the true Kernel SHAP values for this patient, as noted in Figure 5.11 & 5.12. Figure 5.24 shows an alternative visualization of the contributing Kernel SHAP value approximations for this patient. We note that the top 5 and bottom 5 contributing features for this patient's malignant diagnosis are the same as the top 5 contributing factors when compared to the true Kernel SHAP values and both of the LIME models, however the exact ordering of the feature importance does vary.

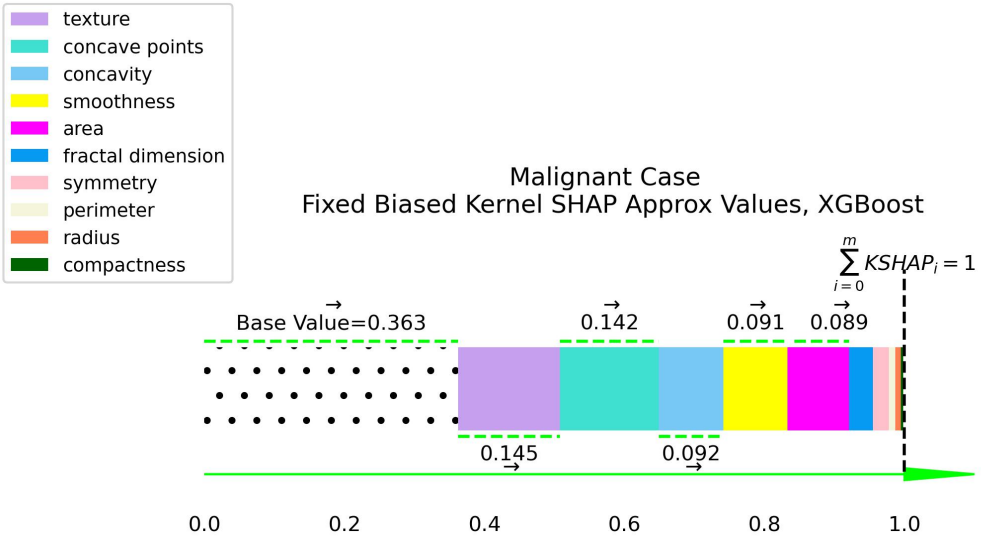

Figure 5.23: XGBoost Black Box - Fixed Biased Kernel SHAP Approximation - Malignant Case, Additive Illustration

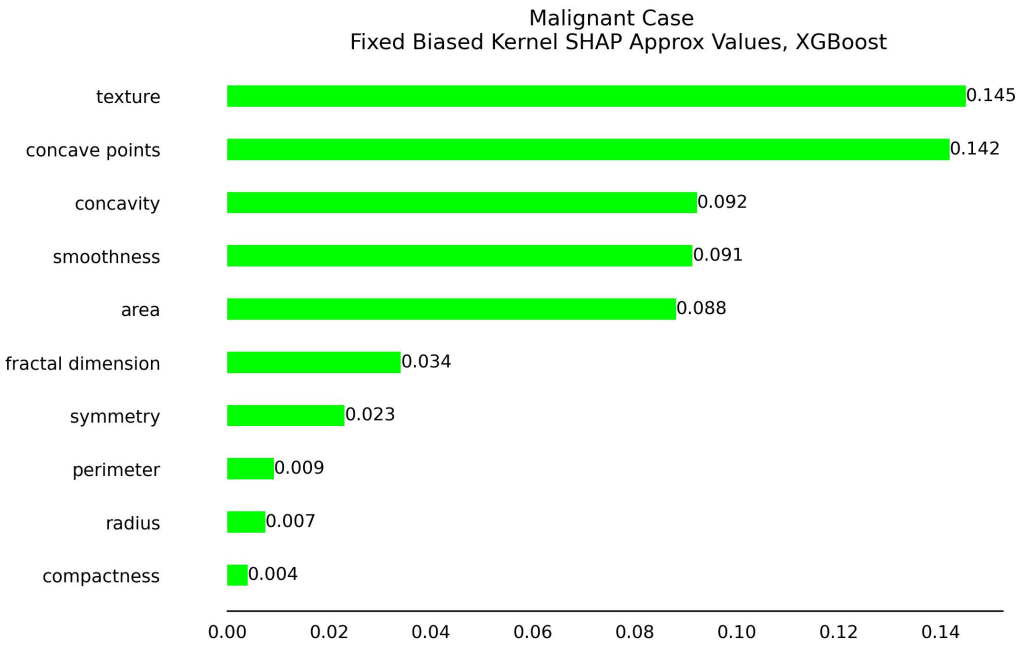

Figure 5.24: XGBoost Black Box -Fixed Biased Kernel SHAP Approximation - Malignant Case, Bars Illustration

As mentioned previously, this next visualization is generated by summing the absolute value of the Kernel SHAP approximation values for this patient, and then expressing the contribution of each value as a percentage of that sum. We see that texture and concave points dominate the malignant diagnosis for this patient with a 45% contribution to the absolute value sum, and this is similar to the true Kernel SHAP values for this patient wherein texture and concave points represent a 50% contribution. The top three contributing features, which includes concavity, contribute 60% of the sum of the absolute Kernel SHAP approximation values. The top four features contribute 74% of this sum. Ultimately, the results of the Kernel SHAP approximation using the fixed biased algorithm are quite similar to the true Kernel SHAP values for this patient presented in Section 5.2.3. Using the fixed biased algorithm produces a squared error of 0.021 for the Kernel SHAP value approximations for all 10 features.

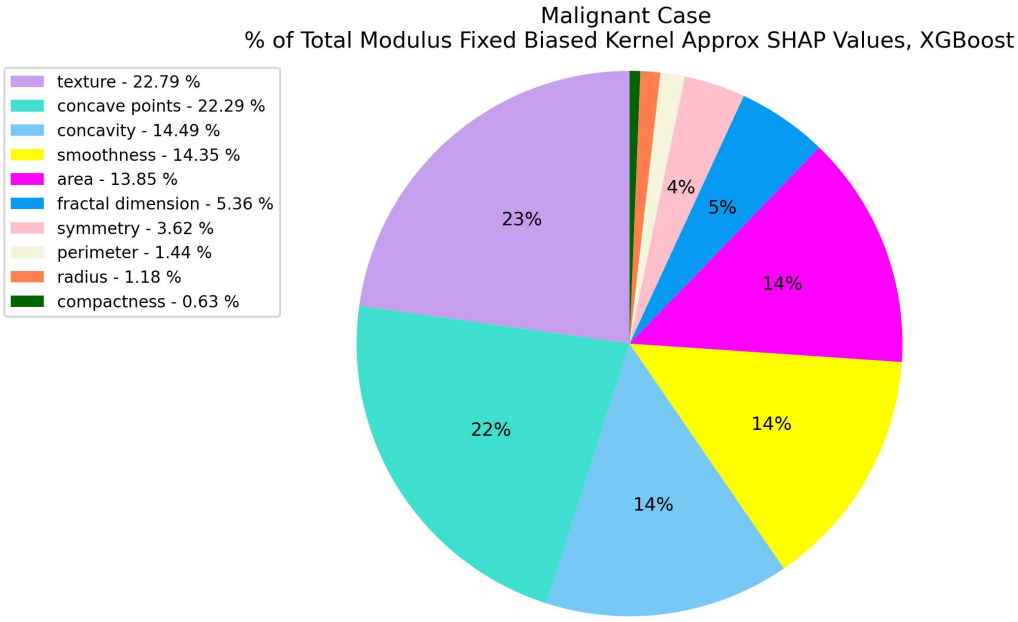

Figure 5.25: XGBoost Black Box - % of Total Modulus Approx Kernel SHAP Values using Fixed Biased Kernel SHAP - Malignant Case

The conclusory explanation table for the Fixed Biased Kernel SHAP approximation algorithm applied to our malignant case using t=2, and 25 sampled coalitions is presented below.

Conclusory Explanation	Feature Importance Ordering (Contributing To Malignancy)	Number of Model Predictions Used To Generate Explanation (Per Instance)[1]	Time To Completion In Seconds (Per Instance) [2]	Squared Error To True Kernel SHAP Values (Per Instance) [3]
According to the Fixed Biased Kernel SHAP approximation method applied to the XGBoost model, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is texture, followed by concave points.	1. Texture 2. Concave points 3. Concavity 4. Smoothness 5. Area 6. Fractal dimension 7. Symmetry 8. Perimeter 9. Radius 10. Compactness	11375	33.37425	0.021
[1] 25 coalitions from the powerset of 10 features, multiplied by 455 (number of training samples).				
[2] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds				
[3] This is the squared error between the Kernel SHAP approximation values and the true Kernel SHAP values, across all 10 features				

Table 5.8: XGBoost Black Box - Fixed Biased Kernel SHAP Approximation Summary Table, Malignant Case, t=2, 25 coalition samples

The following graphs depict the Kernel SHAP approximation using the biased method presented in the paper by Covert et al. [5]. We use a sample size of 25 coalitions to compare with our t=2 Fixed Biased Kernel SHAP approximation above. To reiterate, 25 coalitions is equivalent to 11,375 model predictions when we account for the 455 training samples. The equivalent figures for the benign case can be seen in Appendix D. In Figure 5.26 we see that the largest contributing feature to the positive malignant diagnosis is concave points, as noted by the aqua coloured bar with an approximated Kernel SHAP value of 0.168. The second highest contributing feature to the diagnosis is concavity, as noted by the blue bar with an approximated Kernel SHAP value of 0.167. The third highest contributing feature toward the diagnosis is smoothness, and it is followed then texture in fourth place. The top 5 contributing features are the same as the true Kernel SHAP values for this patient, however the ordering is not exactly alike. In the biased Kernel SHAP approximation, concave points and concavity are ranked as the top two contributing features. Figure 5.27 shows an alternative visualization of the contributing Kernel SHAP value approximations for this patient. We note that the top 5 and bottom 5 contributing features for this patient's malignant diagnosis are the same as the top 5 contributing factors when compared to the true Kernel SHAP values and both of the LIME models, however the exact ordering of the feature importance does vary.

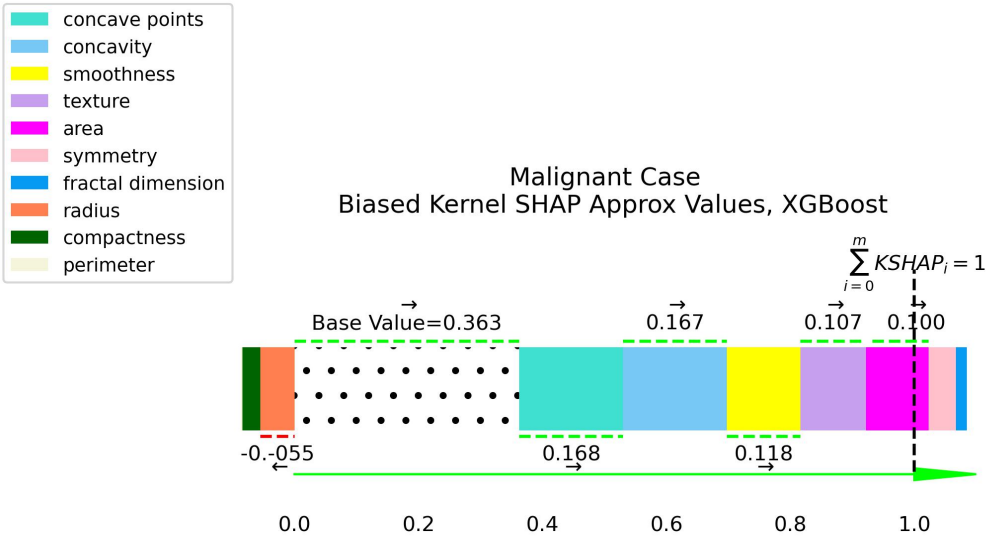

Figure 5.26: XGBoost Black Box - Biased Kernel SHAP Approximation - Malignant Case, Additive Illustration

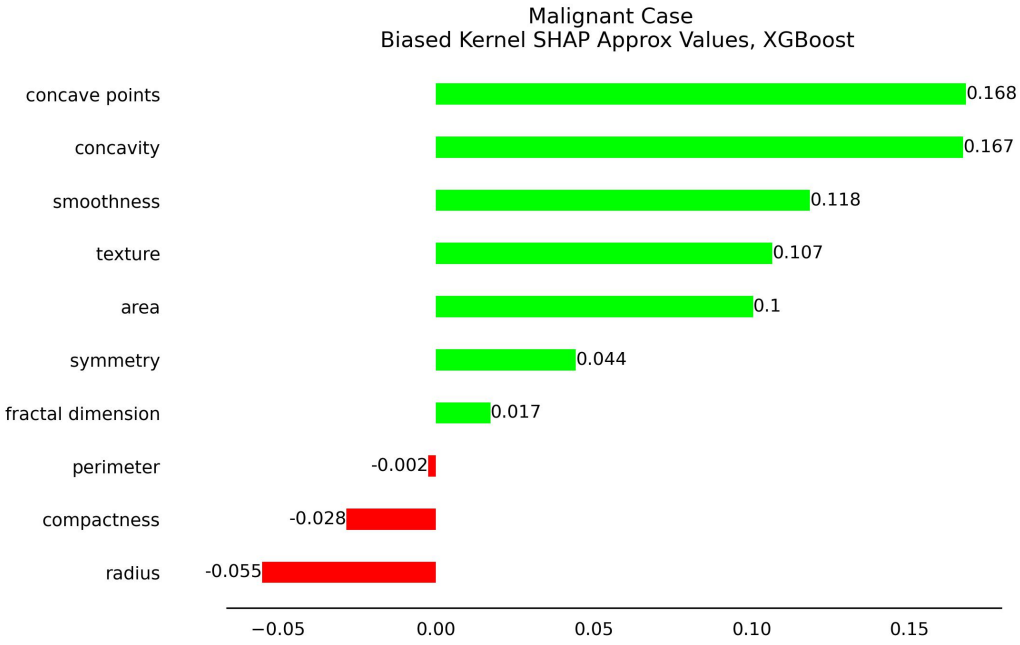

Figure 5.27: XGBoost Black Box - Biased Kernel SHAP Approximation - Malignant Case, Bars Illustration

As mentioned previously, this next visualization is generated by summing the absolute value of the Kernel SHAP approximation values for this patient, and then expressing the contribution of each value as a percentage of that sum. The top two contributions to this sum are concave points and concavity, and collectively they represent 41%.The top three contributing features, which includes smoothness, contribute 56% of the sum of the absolute Kernel SHAP approximation values. The top four features, which includes texture contribute 69% of this sum. Using the biased algorithm produces a squared error of 0.026 for the Kernel SHAP value approximations for all 10 features.

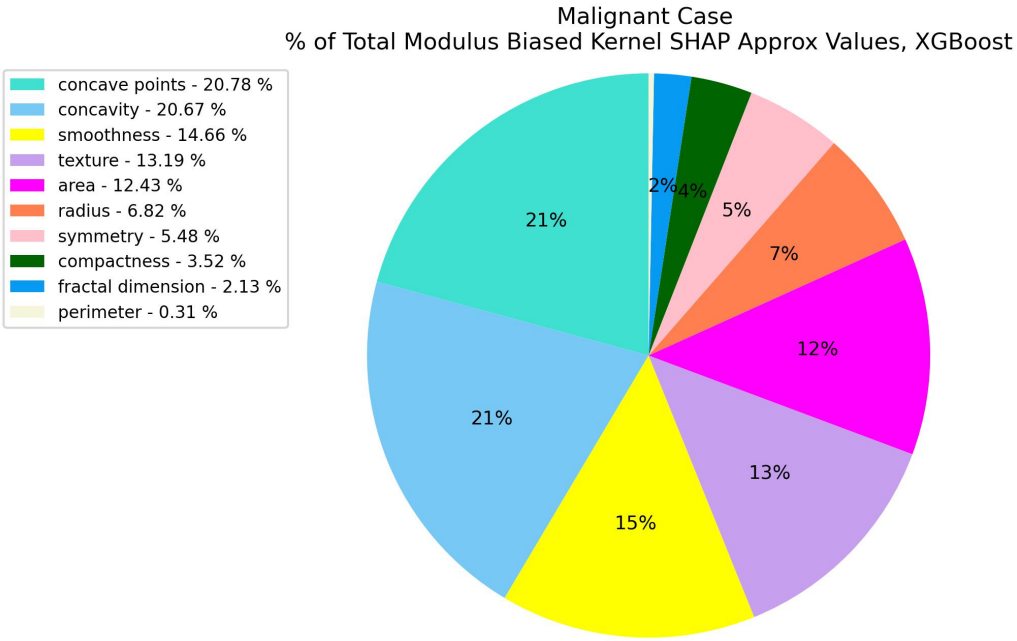

Figure 5.28: XGBoost Black Box - % of Total Modulus Approx Kernel SHAP Values using Biased Kernel SHAP - Malignant Case

The conclusory explanation table for the Biased Kernel SHAP approximation algorithm applied to our malignant case using 25 sampled coalitions is presented below.

Conclusory Explanation	Feature Importance Ordering (Contributing To Malignancy)	Number of Model Predictions Used To Generate Explanation (Per Instance)[1]	Time To Completion In Seconds (Per Instance) [2]	Squared Error To True Kernel SHAP Values (Per Instance) [3]
According to the Biased Kernel SHAP approximation method applied to the XGBoost model, the feature contributing most toward the positive diagnosis in the malignant patient studied in this paper is concave points, followed by concavity.	1. Concave points 2. Concavity 3. Smoothness 4. Texture 5. Area 6. Symmetry 7. Fractal dimension 8. Perimeter 9. Compactness 10. Radius	11375	33.37425	0.026
[1] 25 coalitions from the powerset of 10 features, multiplied by 455 (number of training samples).				
[2] Average Speed Per Model Prediction x Num Model Predictions. Average speed per instance model prediction is 0.002934 seconds				
[3] This is the squared error between the Kernel SHAP approximation values and the true Kernel SHAP values, across all 10 features				

Table 5.9: XGBoost Black Box - Biased Kernel SHAP Approximation Summary Table, Malignant Case, 25 coalition samples

5.4 Method Comparisons and Summary

This Section details a brief conclusion overview of the XAI Methods studied in this thesis, as well as their application to the patient studied with a malignant diagnosis from the BCW-D test set. Below we show a side-by-side comparison of the feature importance ordering generated by each of the XAI methodologies applied to the malignant case. As you can see, the XAI method selected to gain a better understanding of a local prediction will greatly impact the ordering of the feature importance. What should be understood by the practitioner is that there is a direct tradeoff between computational speed and accuracy of feature importance produced by local XAI methods, when using true Kernel SHAP values as the ground truth explanation. Of course, producing true Kernel SHAP values is the most accurate depiction of black box model reasoning, though it is an exponentially expensive task as the number of features of a model grows.

We note from the tables below that for the malignant case studied in this thesis, all of the seven XAI methods including the Kernel SHAP approximation methods concluded the same top 5 and bottom 5 contributing features towards the malignancy diagnosis, though the ordering within each of these buckets differs.

Feature Importance Ordering For Patient With Malignant Diagnosis Produced By Local XAI Methods			
ICE & C-CICE	**LIME, KW = 2.3717**	**LIME, KW = 0.9**	**True Kernel SHAP**
1. Texture	1. Texture	1. Concave points	1. Texture
2. Smoothness	2. Area	2. Texture	2. Concave points
3. Concave points	3. Concave points	3. Area	3. Concavity
4. Concavity	4. Concavity	4. Concavity	4. Smoothness
5. Area	5. Smoothness	5. Smoothness	5. Area
6. Symmetry	6. Perimeter	6. Perimeter	6. Symmetry
7. Radius	7. Symmetry	7. Symmetry	7. Perimeter
8. Compactness	8. Radius	8. Radius	8. Radius
9. Fractal dimension	9. Fractal dimension	9. Compactness	9. Compactness
10. Perimeter	10. Compactness	10. Fractal dimension	10. Fractal dimension

Table 5.10: XGBoost Black Box - Feature Importance Ordering for Malignant Case Using Local XAI Methods

Feature Importance Ordering For Patient With Malignant Diagnosis Produced By Kernel SHAP Approximations		
LIME Inferred Kernel SHAP Approximation	**Fixed Biased Kernel SHAP Approximation, t=2, N=25**	**Biased Kernel SHAP Approximation, N=25**
1. Texture	1. Texture	1. Concave points
2. Area	2. Concave points	2. Concavity
3. Concave points	3. Concavity	3. Smoothness
4. Concavity	4. Smoothness	4. Texture
5. Smoothness	5. Area	5. Area
6. Perimeter	6. Fractal dimension	6. Symmetry
7. Symmetry	7. Symmetry	7. Fractal dimension
8. Radius	8. Perimeter	8. Perimeter
9. Fractal dimension	9. Radius	9. Compactness
10. Compactness	10. Compactness	10. Radius

Table 5.11: XGBoost Black Box - Feature Importance Ordering for Malignant Case Using Kernel SHAP Approximation Methods

Tables 5.12 & 5.13 provide side-by-side comparison of the number of model predictions used for each of the local XAI methodologies applied to the malignant case studied in this paper. Producing true Kernel SHAP values is a highly computationally expensive task, as each instance will require $2^{NumFeatures}$ model predictions. Furthermore, Kernel SHAP approximation methods that balance accuracy and computational expense are crucial to the development of trustworthy XAI explanations.

As we can see in Table 5.13 below, using our Fixed Biased Kernel SHAP algorithm at $t = 2$ for a relatively small coalition sample size equal to 25 produces more accurate approximations to true Kernel SHAP values than the Biased Kernel SHAP algorithm for the malignant case studied. In fact, Fixed Biased Kernel SHAP provided consistently superior approximations to true Kernel SHAP at t=2 for our entire 114 test set samples in the BCW-D dataset, up to and including 36 coalitions, as evidenced in Figure 5.21. This is particularly useful if a practitioner would like to produce fast approximations while also improving accuracy to true Kernel SHAP.

	ICE & C-CICE	LIME, KW = 2.3717	LIME, KW = 0.9	True Kernel SHAP
Number of Model Predictions Used	1000	4000	4000	465920
Time to Completion Using Average Model Prediction Speed of .002934 seconds	2.934	11.736	11.736	1367.00928

Table 5.12: XGBoost Black Box - Number of Model Predictions Used and Time to Completion Using Local XAI Methods for Patient with Malignant Diagnosis

	LIME Inferred Kernel SHAP Approximation	Fixed Biased Kernel SHAP Approximation, t=2, N=25	Biased Kernel SHAP Approximation, N=25
Number of Model Predictions Used	4000	11375	11375
Time to Completion Using Average Model Prediction Speed of .002934 seconds	11.736	33.37425	33.37425
Squared Error to True Kernel SHAP	0.034	0.021	0.026

Table 5.13: XGBoost Black Box - Number of Model Predictions Used and Time to Completion Using Kernel SHAP Approximations for Patient with Malignant Diagnosis

Chapter 6

Conclusion And Future Work

6.1 Conclusion

The development of accurate and efficient Local Model Agnostic Explanations for black box Artificial Intelligence Models is of utmost importance. While there are many options available to Machine Learning practitioners, the leading XAI methodology for local explanations is based on Shapley Values. While there are model-specific techniques available to approximate SHAP values, they do not offer practitioners any solace to the conundrum of computationally expensive local model agnostic explanations. Currently, the best option available remains the Biased Kernel SHAP algorithm as noted by Covert et al [5].

In regards to breast cancer diagnosis using black box methods and using local XAI methods to explain these results, Fixed Biased Kernel SHAP provides a superior approximation true Kernel SHAP values for t=2 and a low number of coalition

samples. This may be useful to practitioners in situations where a large number of explanations must be generated and where limiting run time is of the highest consideration. However, we recognize that it is still very challenging to provide consistently superior model-agnostic approximations to true Kernel SHAP values than the approximations generated by the Biased Kernel SHAP algorithm.

In conclusion, this thesis makes several important contributions which include; (1) Provides a thorough technical run-down of current state of the art local XAI methodologies (2) Provides the most in depth research to date on applied local XAI methodologies for breast cancer diagnostics from fine-needle aspirate data, using the BCW-D dataset [26]. (3) Thorough and novel illustrations that help visualize the inner workings of an XGBoost model which achieves 96% diagnostic accuracy. Methods used include ICE & C-ICE, LIME, and Kernel SHAP, Biased Kernel SHAP approximation, and a novel algorithm called Fixed Biased Kernel SHAP. (4) Provides a novel algorithm to efficiently approximate Kernel SHAP values, called Fixed Biased Kernel SHAP (5) A thorough analytical comparison between the various leading XAI methodologies.

6.2 Future Work

There are a variety of techniques a practitioner may use to improve Kernel SHAP approximations, as evidenced by Covert et al. [5]. One such technique is a variance reduction technique that involves sampling the complement coalition in addition to the coalition sampled at each iteration in the Biased Kernel SHAP al-

gorithm. They detail in their work how although this technique increases the number of model evaluations required at each game evaluation, it still provides superior convergence to true Kernel SHAP values when accounting for the number of coalitions sampled. In addition to this variance reduction technique, a practitioner may opt to use a convergence detection algorithm such as Welford's to understand when to optimally stop coalition sampling and produce final Kernel SHAP approximations [5]. Future research may include an analysis of how both complement sampling and convergence detection impact the results of our Fixed Biased Kernel SHAP algorithm, and how these techniques perform when used to generate explanations for black box models used for breast cancer detection.

Bibliography

[1] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

[2] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

[3] Alexander Binder, Michael Bockmayr, Miriam Hägele, Stephan Wienert, Daniel Heim, Katharina Hellweg, Masaru Ishii, Albrecht Stenzinger, Andreas Hocke, Carsten Denkert, et al. Morphological and molecular breast cancer profiling through explainable machine learning. *Nature Machine Intelligence*, 3(4):355–366, 2021.

[4] Tamires Brito-Sarracino, Moisés Rocha dos Santos, Eric Freire Antunes, Iury Batista de Andrade Santos, Jonas Coelho Kasmanas, André Carlos Ponce de Leon Ferreira, et al. Explainable machine learning for breast

cancer diagnosis. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 681–686. IEEE, 2019.

[5] Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR, 2021.

[6] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, 2020.

[7] Dongxiao Gu, Kaixiang Su, and Huimin Zhao. A case-based ensemble learning system for explainable breast cancer recurrence prediction. *Artificial Intelligence in Medicine*, 107:101858, 2020.

[8] Hajar Hakkoum, Ali Idri, and Ibtissam Abnane. Artificial neural networks interpretation using lime for breast cancer diagnosis. In *World Conference on Information Systems and Technologies*, pages 15–24. Springer, 2020.

[9] Hajar Hakkoum, Ali Idri, and Ibtissam Abnane. Assessing and comparing interpretability techniques for artificial neural networks breast cancer classification. *Computer methods in biomechanics and biomedical engineering: imaging & visualization*, 9(6):587–599, 2021.

[10] Michele La Ferla, Matthew Montebello, and Dylan Seychell. An xai approach to deep learning models in the detection of ductal carcinoma in situ. *arXiv preprint arXiv:2106.14186*, 2021.

[11] Jean-Baptiste Lamy, Boomadevi Sekar, Gilles Guezennec, Jacques Bouaud, and Brigitte Séroussi. Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. *Artificial intelligence in medicine*, 94:42–53, 2019.

[12] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[13] Scott M Lundberg and Su-In Lee. Consistent feature attribution for tree ensembles. *arXiv preprint arXiv:1706.06060*, 2017.

[14] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[15] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[16] Arturo Moncada-Torres, Marissa C van Maaren, Mathijs P Hendriks, Sabine Siesling, and Gijs Geleijnse. Explainable machine learning can outperform cox regression predictions and provide insights in breast cancer survival. *Scientific reports*, 11(1):6968, 2021.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[18] Lloyd S Shapley. A value for n-person games. *Classics in game theory*, 69, 1997.

[19] David F Sigmon and Saira Fatima. Fine needle aspiration. 2020.

[20] L Smith, S Bryan, P De, et al. Canadian cancer statistics advisory committee. canadian cancer statistics 2018. 2018.

[21] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[22] Marco Tulio Correia Ribeiro. LIME Library, 2018.

[23] Lev V Utkin and Andrei V Konstantinov. Ensembles of random shaps. *arXiv preprint arXiv:2103.03302*, 2021.

[24] Giorgio Visani, Enrico Bagli, and Federico Chesani. Optilime: Optimized lime explanations for diagnostic computer algorithms. *arXiv preprint arXiv:2006.05714*, 2020.

[25] Adrienne G Waks and Eric P Winer. Breast cancer treatment: a review. *Jama*, 321(3):288–300, 2019.

[26] William Wolberg, Nick Street, and Olvi Mangasarian. Breast cancer wisconsin (diagnostic).

Appendix A

ICE and C-ICE Plots

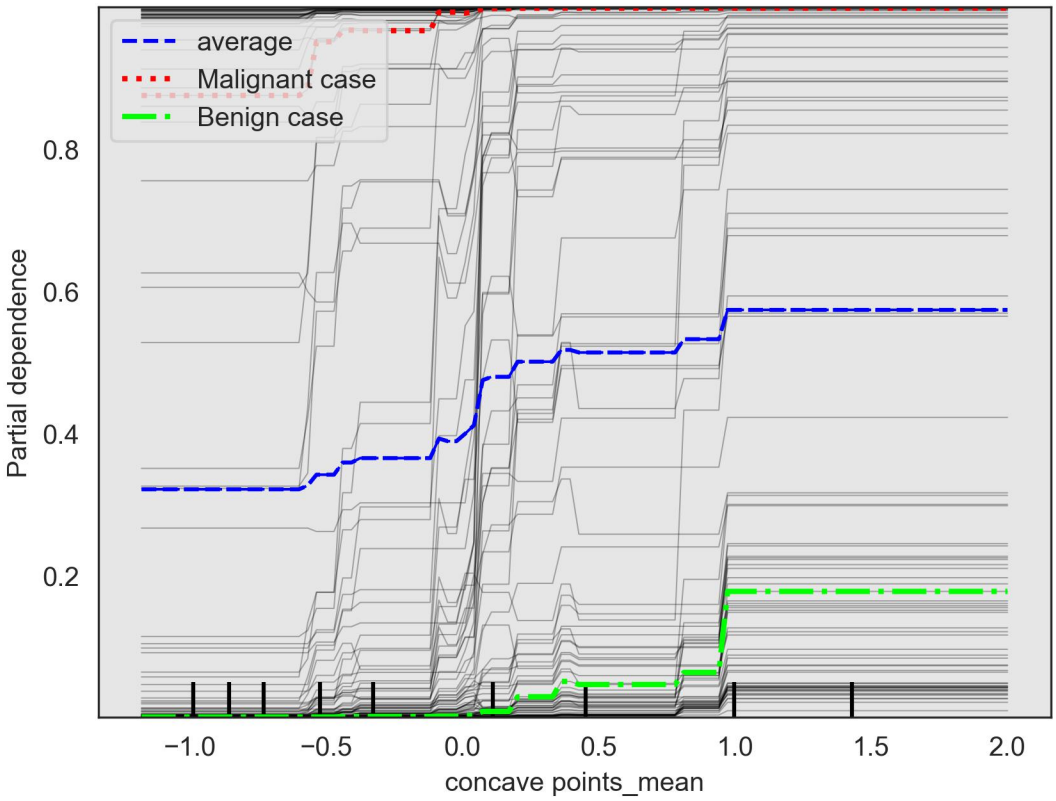

Figure A.1: ICE for concave points mean, XGBoost.

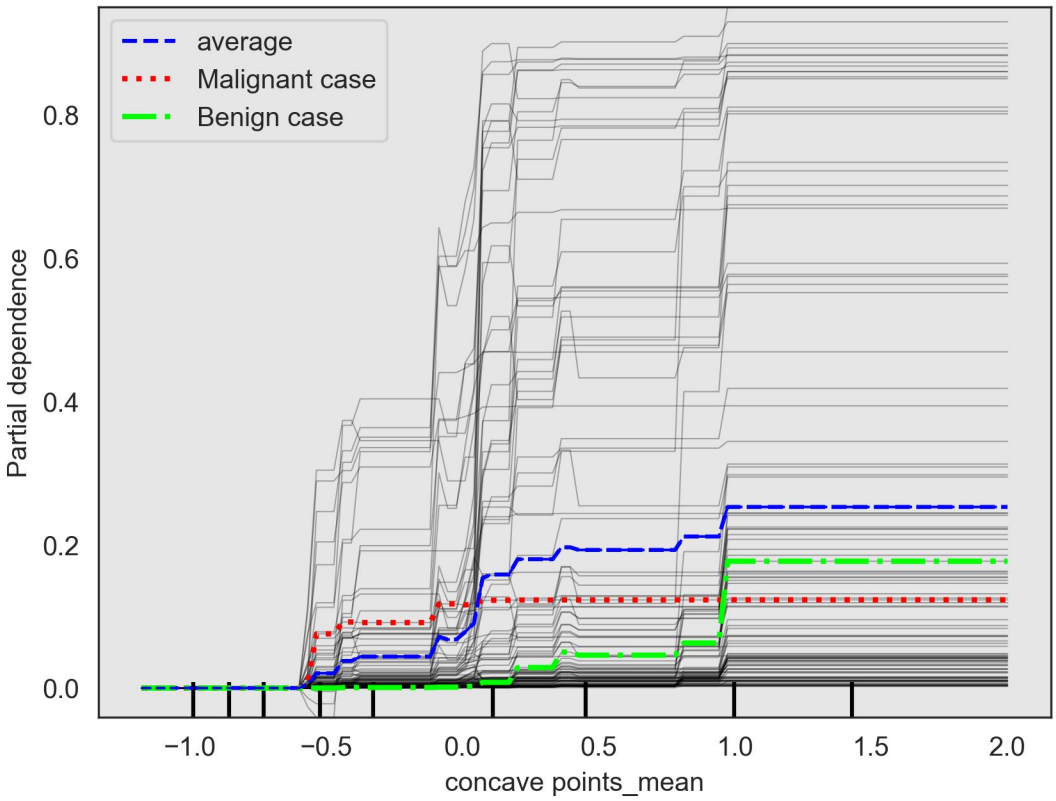

Figure A.2: C-ICE for concave points mean, XGBoost.

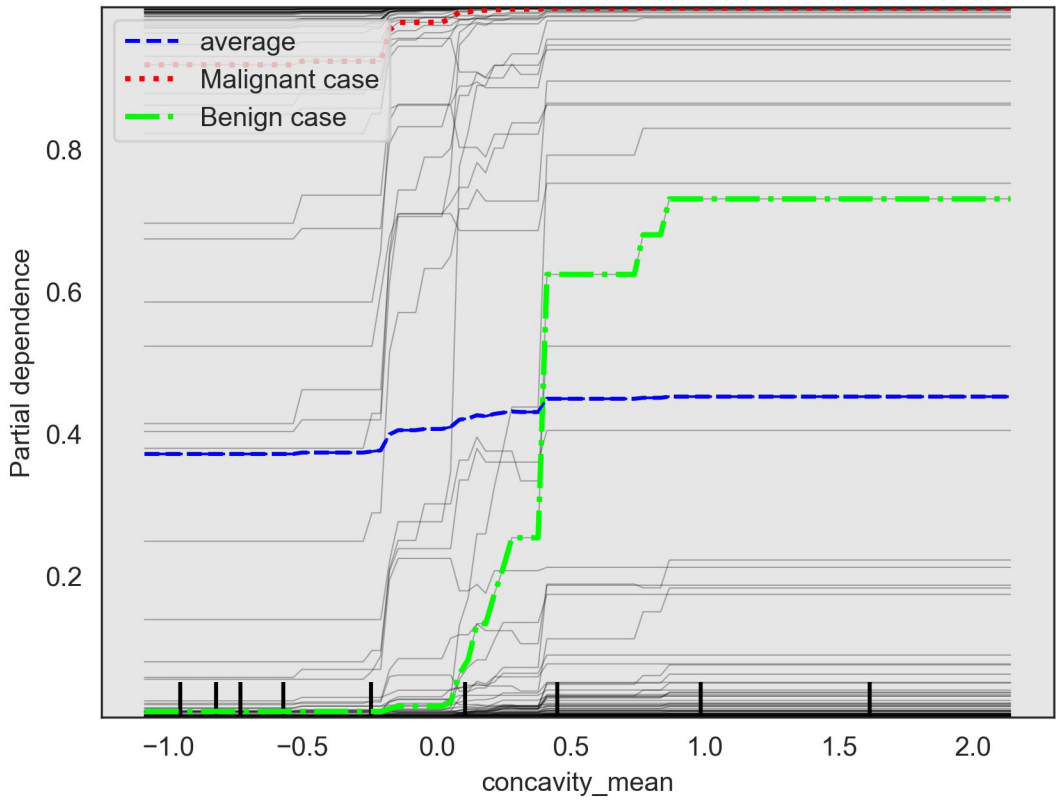

Figure A.3: ICE for concavity mean, XGBoost.

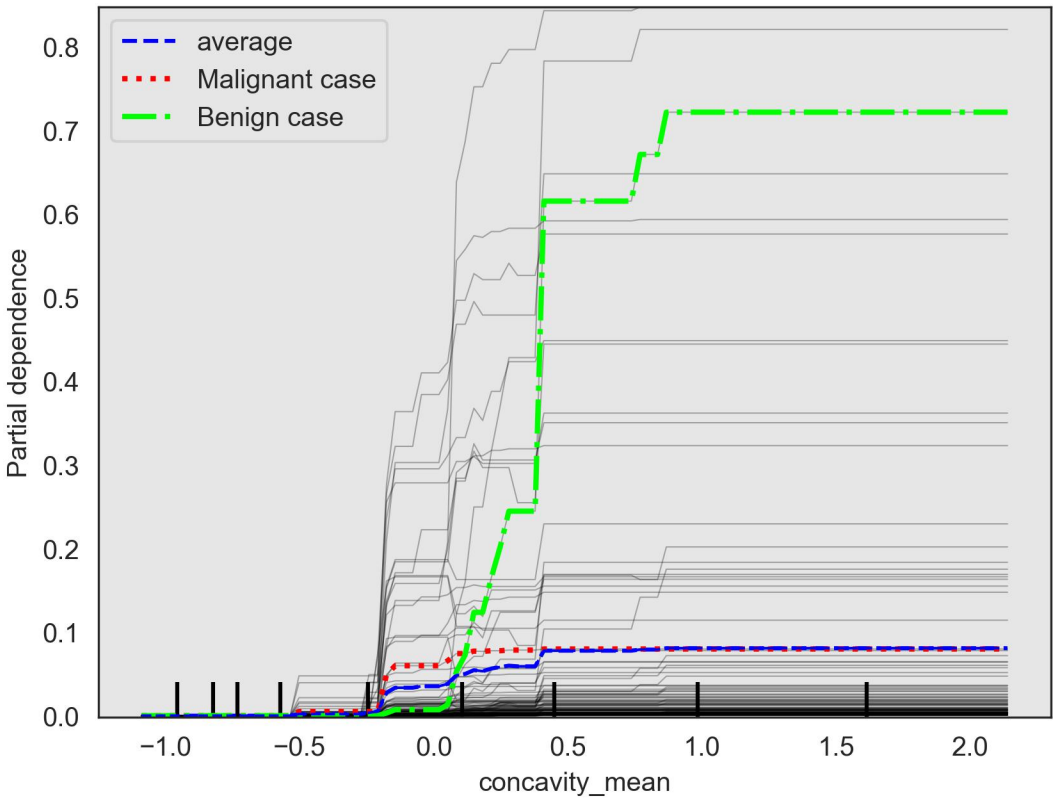

Figure A.4: C-ICE for concavity mean, XGBoost.

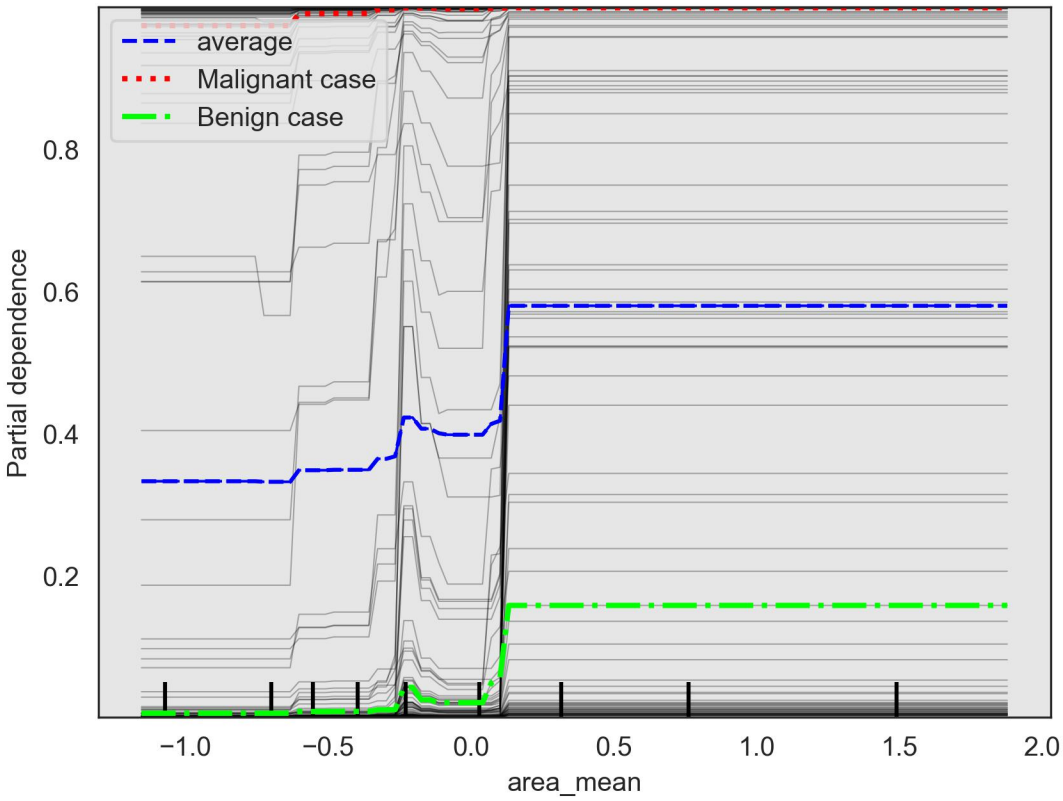

Figure A.5: ICE for area mean, XGBoost.

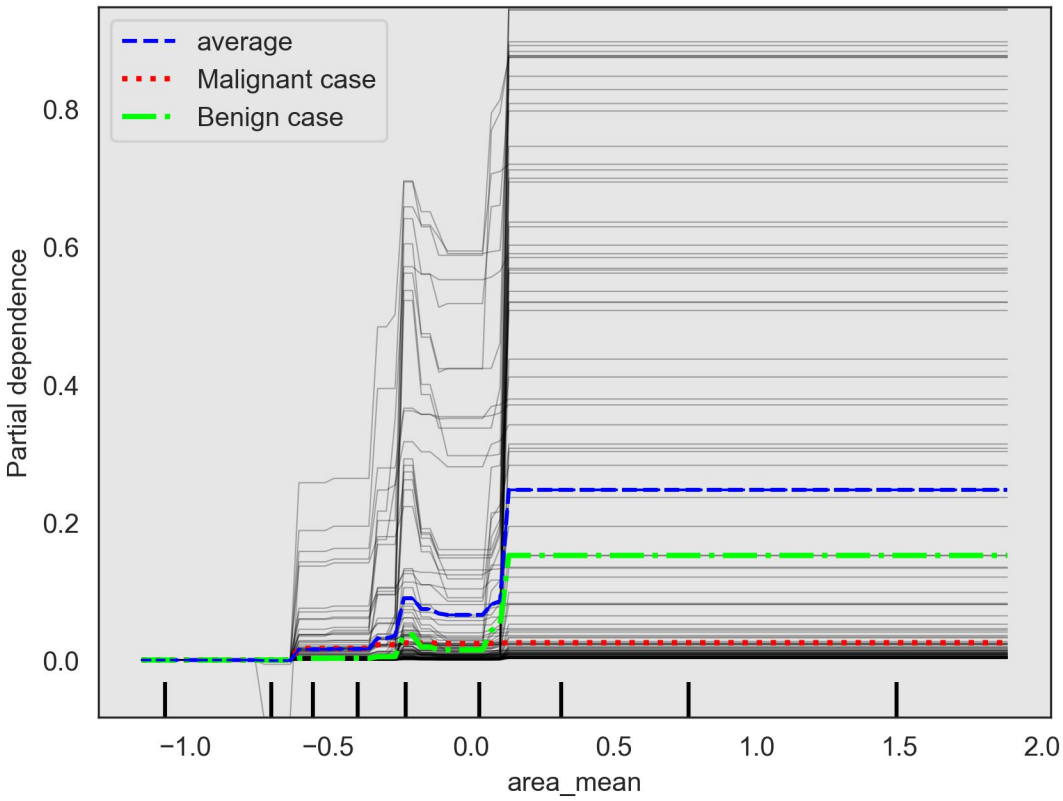

Figure A.6: C-ICE for area mean, XGBoost.

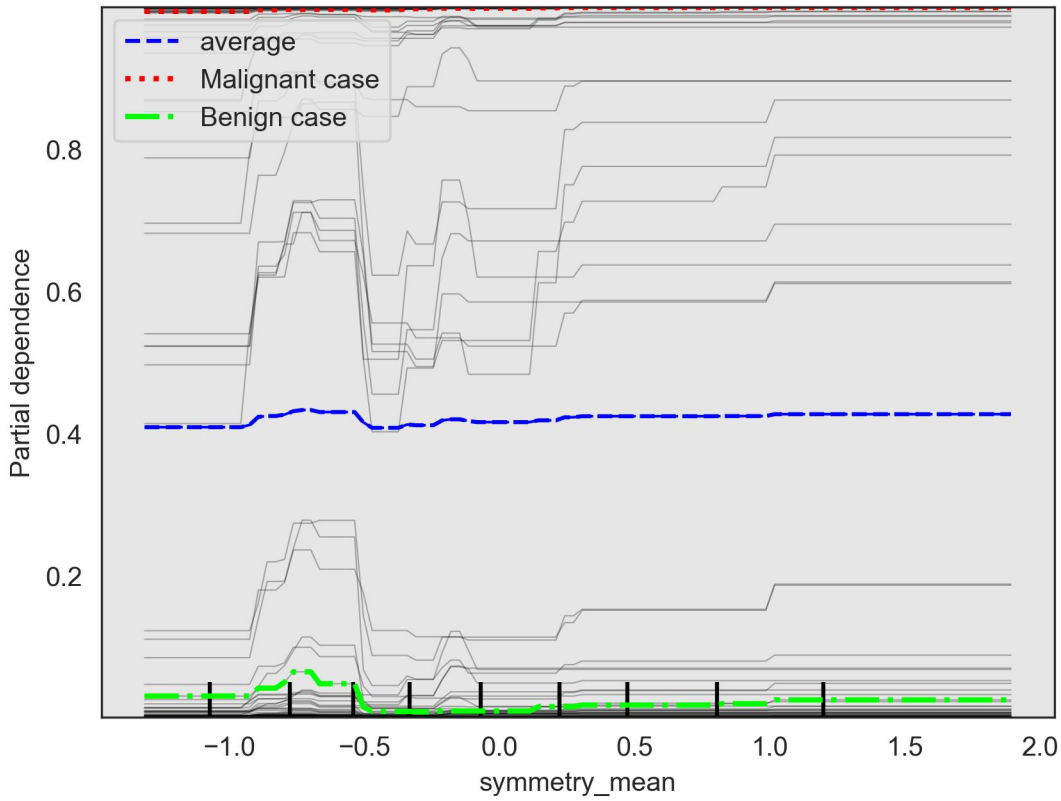

Figure A.7: ICE for symmetry mean, XGBoost.

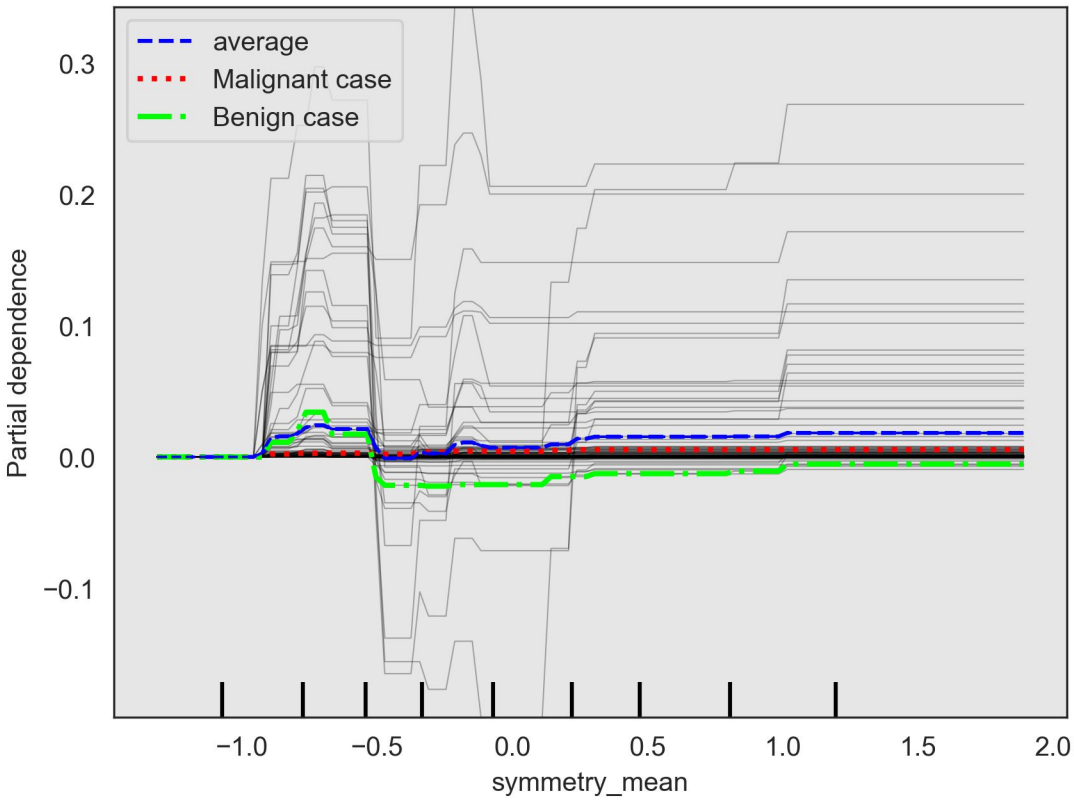

Figure A.8: C-ICE for symmetry mean, XGBoost.

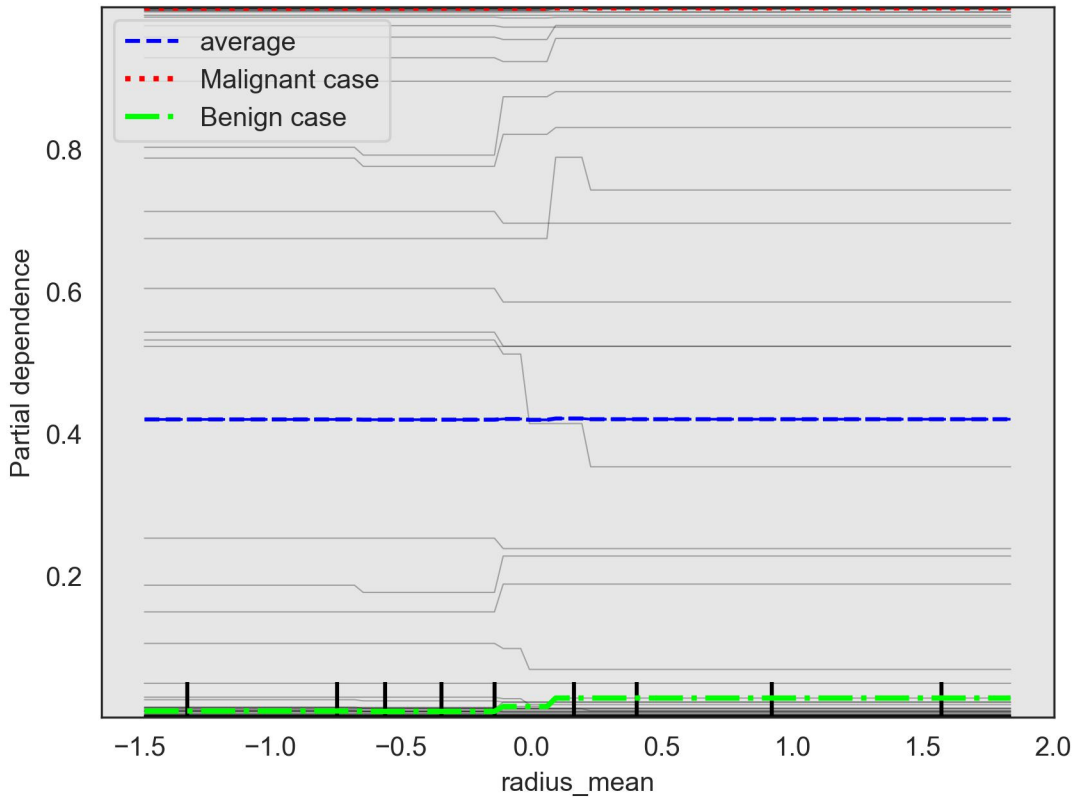

Figure A.9: ICE for radius mean, XGBoost.

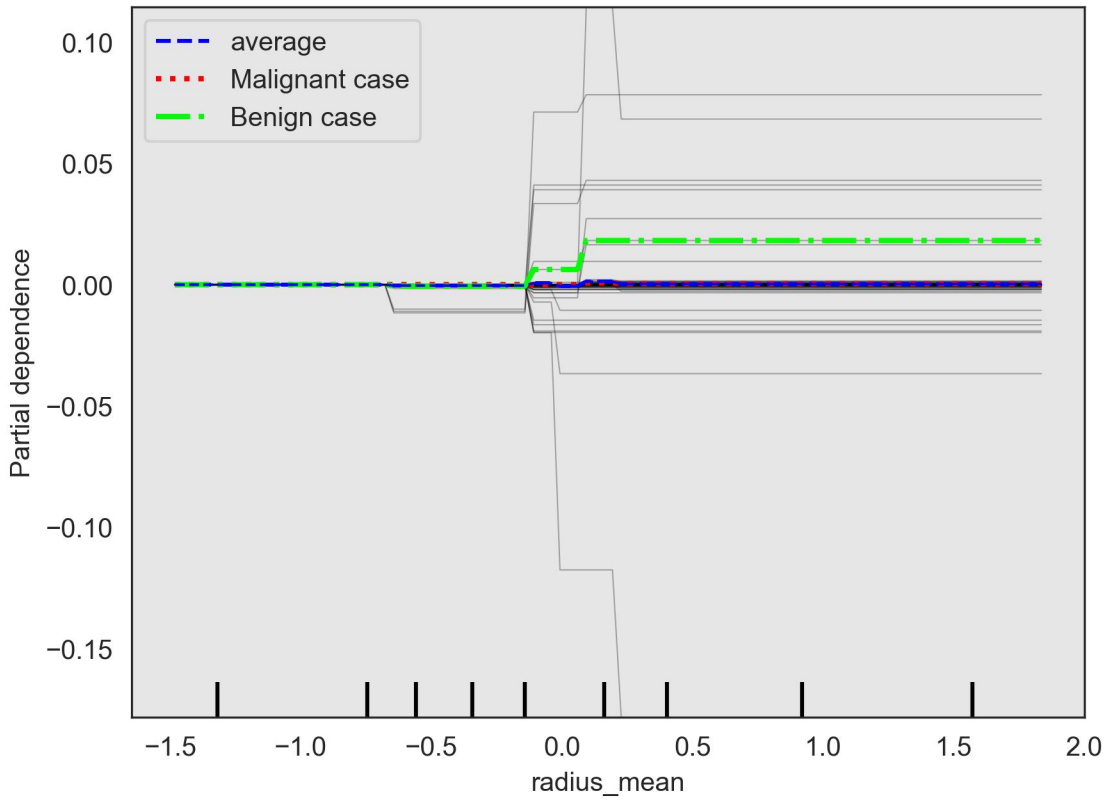

Figure A.10: C-ICE for radius mean, XGBoost.

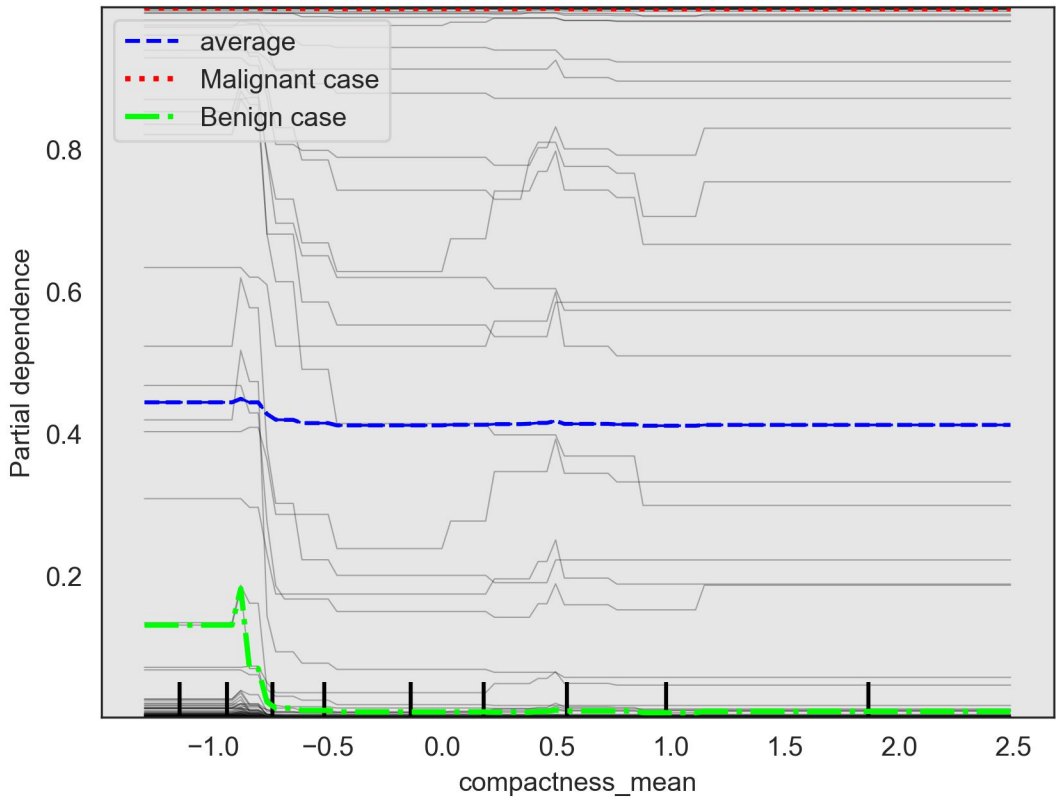

Figure A.11: ICE for compactness mean, XGBoost.

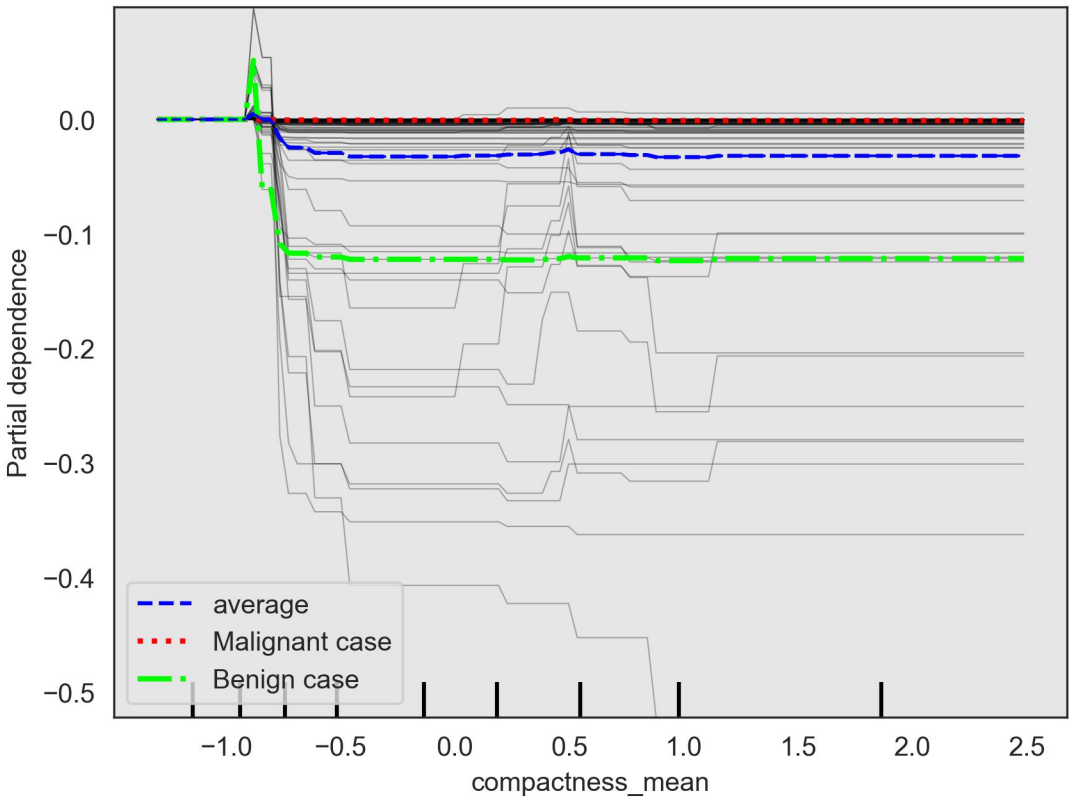

Figure A.12: C-ICE for compactness mean, XGBoost.

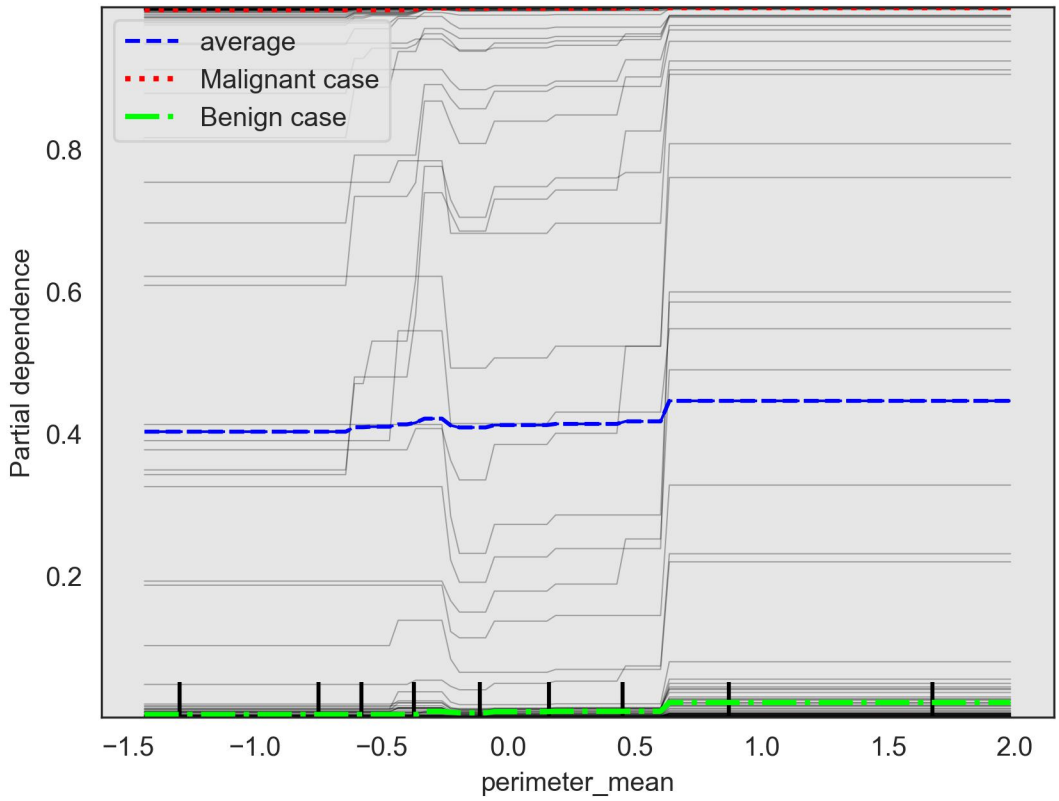

Figure A.13: ICE for perimeter mean, XGBoost.

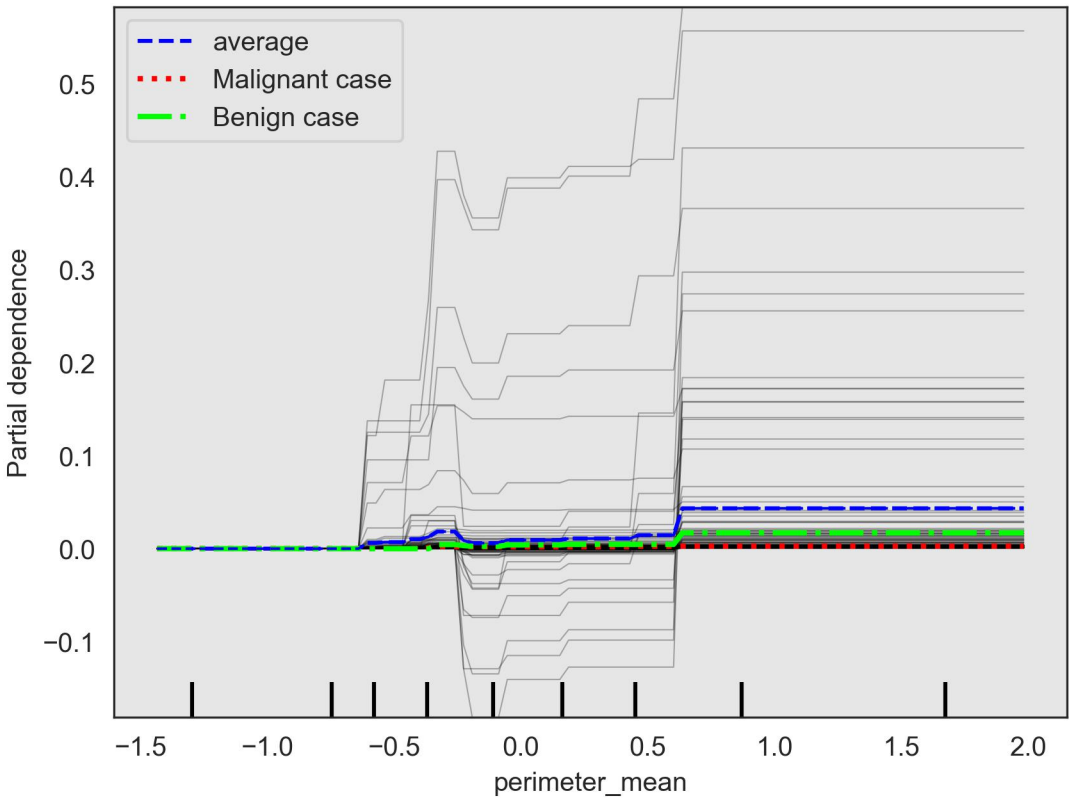

Figure A.14: C-ICE for perimeter mean, XGBoost.

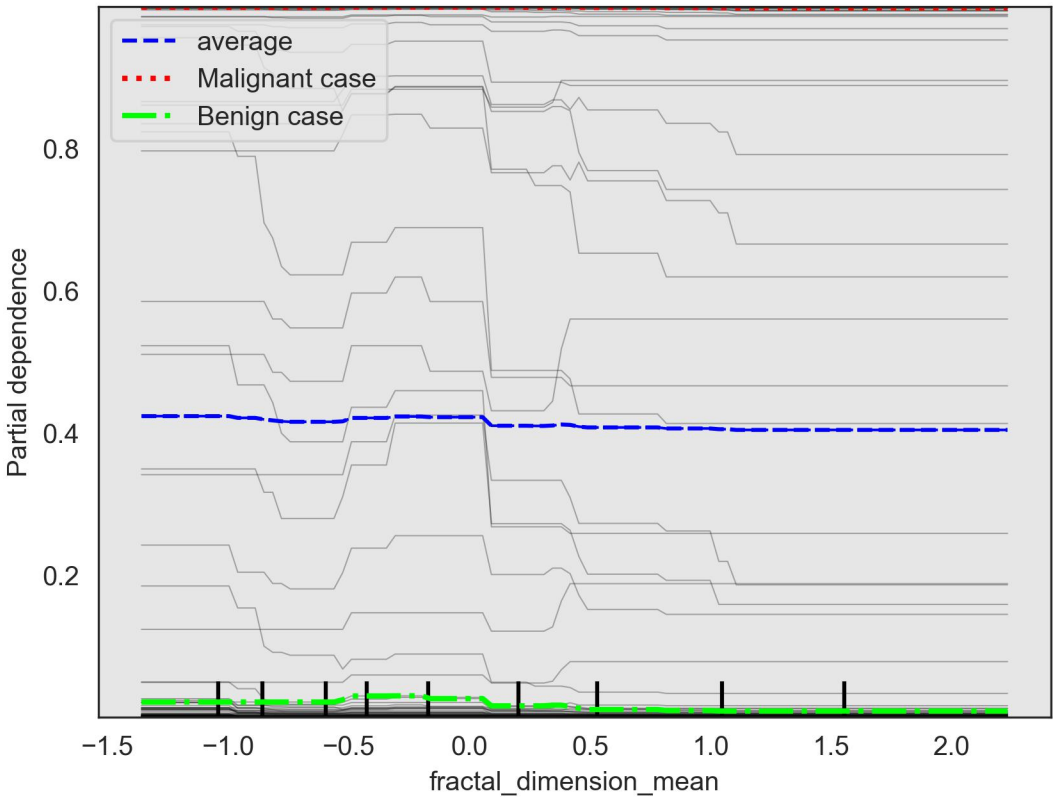

Figure A.15: ICE for fractal dimension mean, XGBoost.

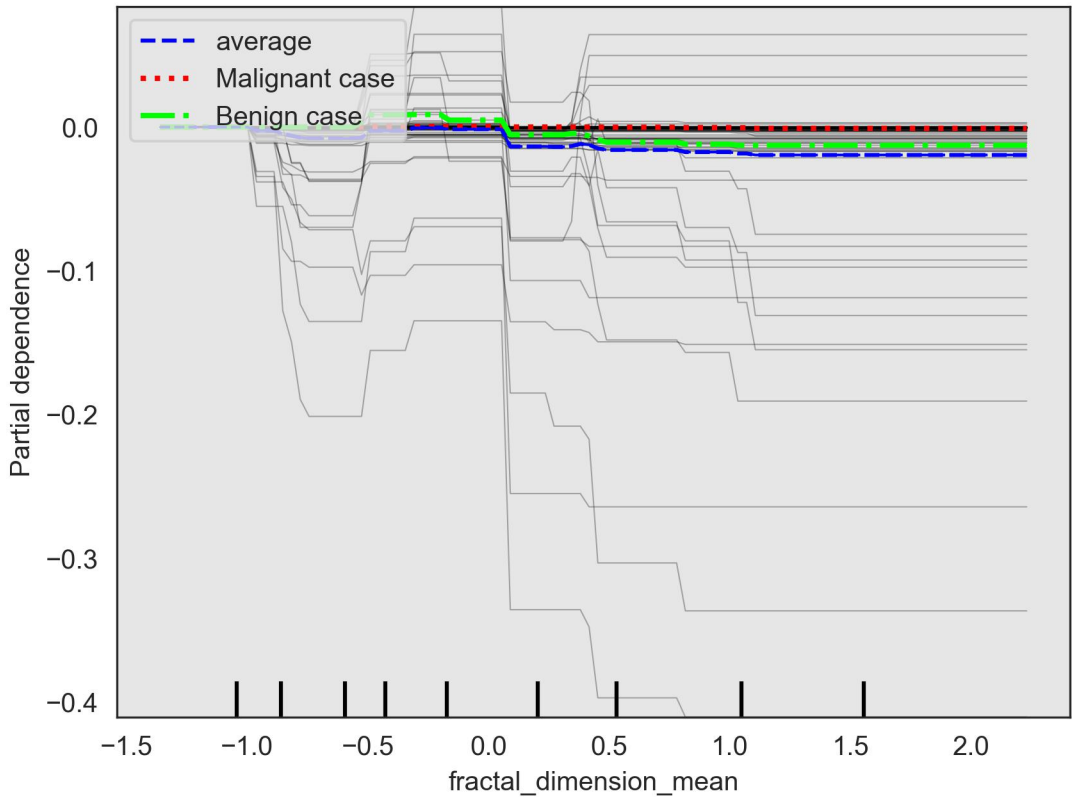

Figure A.16: C-ICE for fractal dimension mean, XGBoost.

Appendix B

LIME Using Logistic Regression, Benign Case

Benign Case LIME Summary
Logistic Regression as Local Interpretable Model , Kernel Width = 2.3717 (Default)

Blackbox Prediction	0
Local Prediction	0
Blackbox Accuracy	96%
Local Accuracy	93%
LIME Neighbourhood Accuracy	85%
LIME R^2 Score	44%

Feature	Value	Logistic Regression Coefficient
Texture	-0.22	2.22
Area	-0.45	1.97
Concave Points	0.11	1.70
Concavity	-0.50	0.85
Smoothness	1.84	0.59
Perimeter	-0.30	0.52
Symmetry	-0.37	0.15
Radius	-0.35	0.00
Fractal Dimension	1.45	-0.18
Compactness	1.24	-0.21

Table B.1: XGBoost Black Box - Benign Case LIME Summary, Kernel Width=2.3717

Benign Case LIME Summary
Logistic Regression as Local Interpretable Model , Kernel Width = 0.9

Blackbox Prediction	0
Local Prediction	0
Blackbox Accuracy	96%
Local Accuracy	89%
LIME Neighbourhood Accuracy	83%
LIME R^2 Score	60%

Feature	Value	Logistic Regression Coefficient
Texture	-0.22	3.49
Area	-0.45	3.11
Concave Points	0.11	2.69
Concavity	-0.50	1.54
Perimeter	-0.30	1.11
Symmetry	-0.37	0.68
Smoothness	1.84	0.41
Radius	-0.35	0.29
Compactness	1.24	0.01
Fractal Dimension	1.45	-0.87

Table B.2: XGBoost Black Box - Benign Case LIME Summary, Kernel Width=0.9

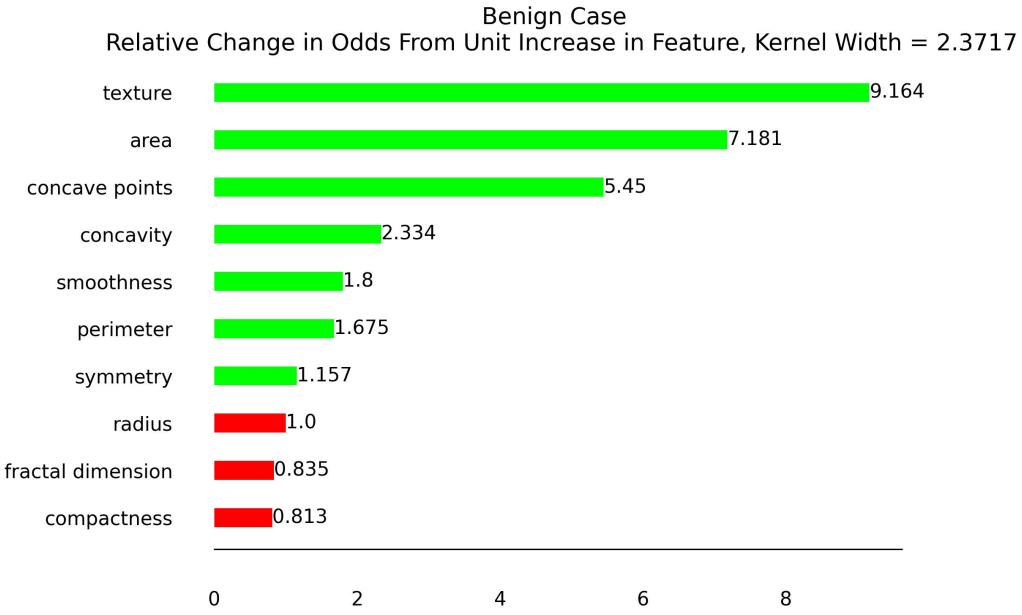

Figure B.1: XGBoost Black Box - Benign Case, Relative Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 2.3717

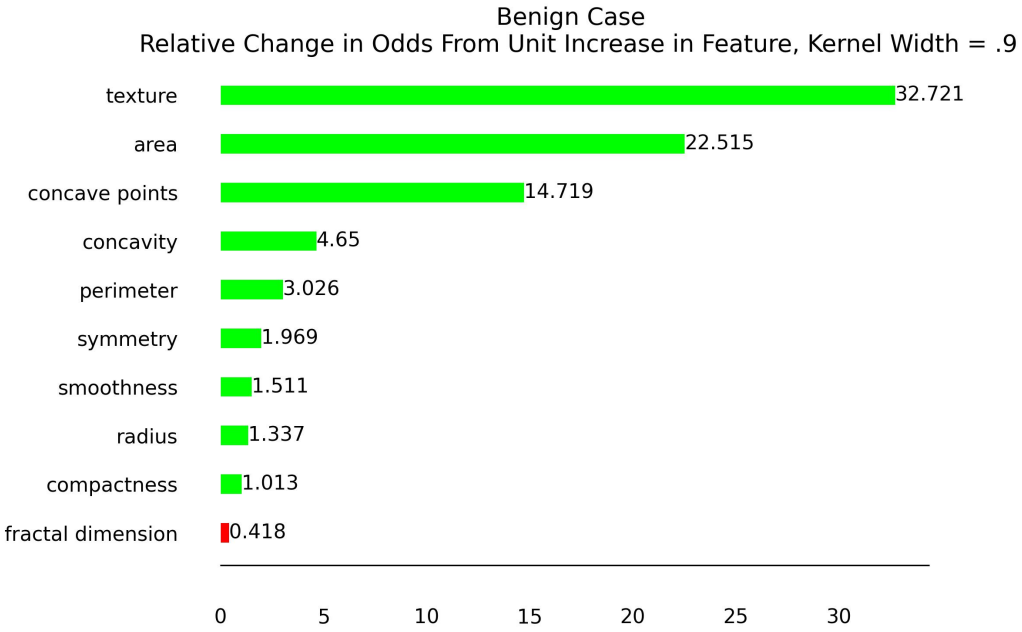

Figure B.2: XGBoost Black Box - Benign Case, Relative Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 0.9

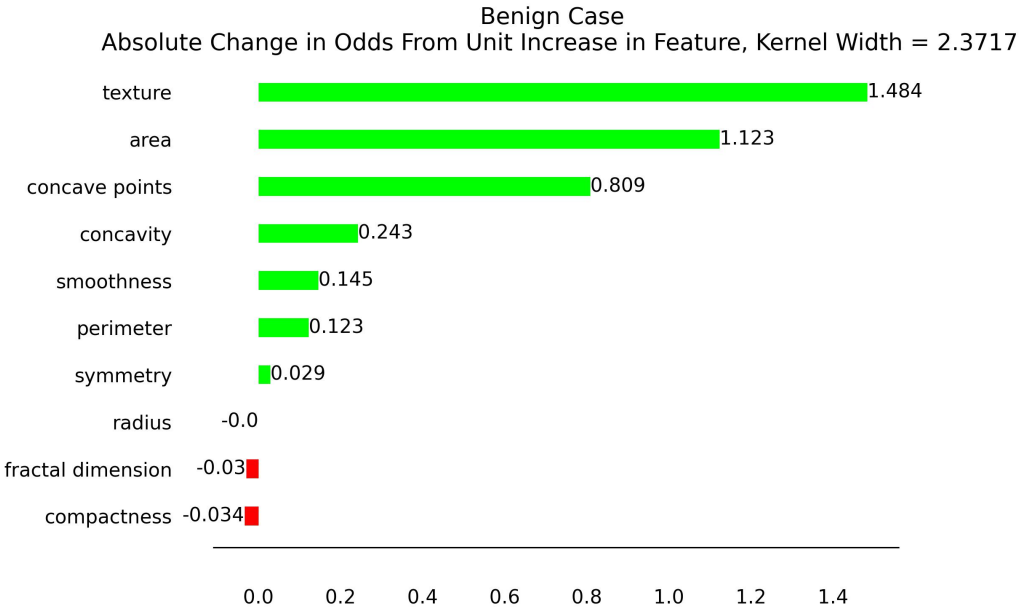

Figure B.3: XGBoost Black Box - Benign Case, Absolute Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 2.3717

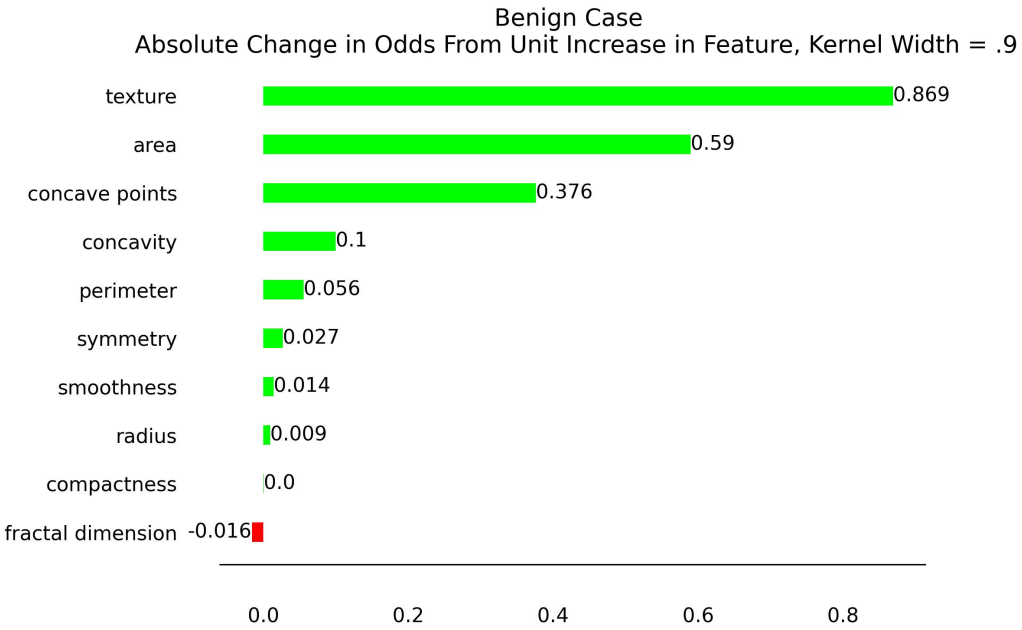

Figure B.4: XGBoost Black Box - Benign Case, Absolute Change of Odds Ratio from Isolated Unit Increase In Feature Value, Kernel Width = 0.9

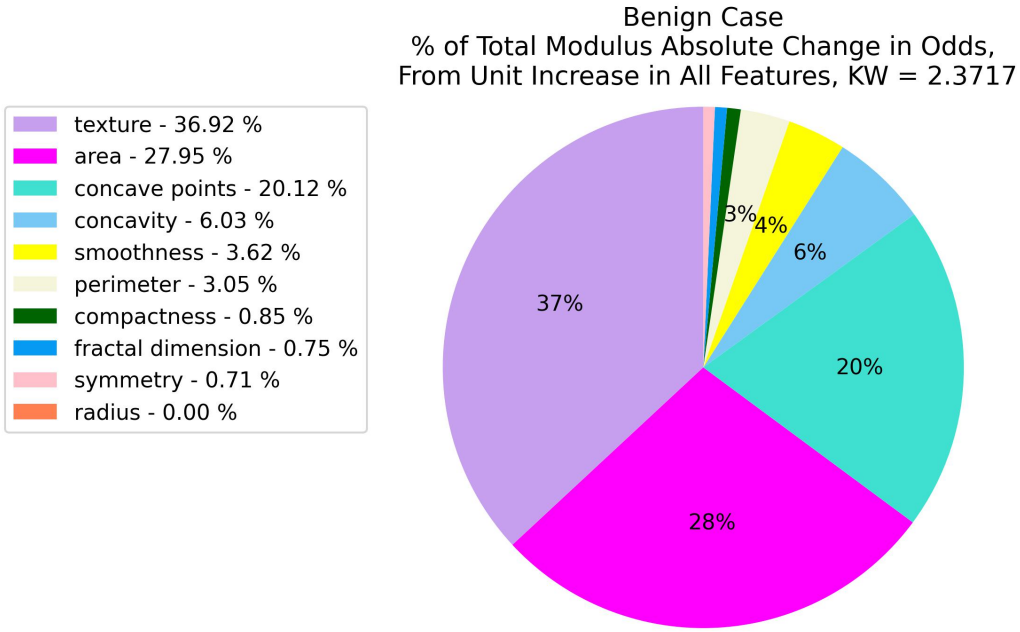

Figure B.5: XGBoost Black Box - Benign Case, % Percent of Total Modulus (Absolute Value) Absolute Change in Odds from Isolated Unit Increase For All Features, Kernel Width = 2.3717

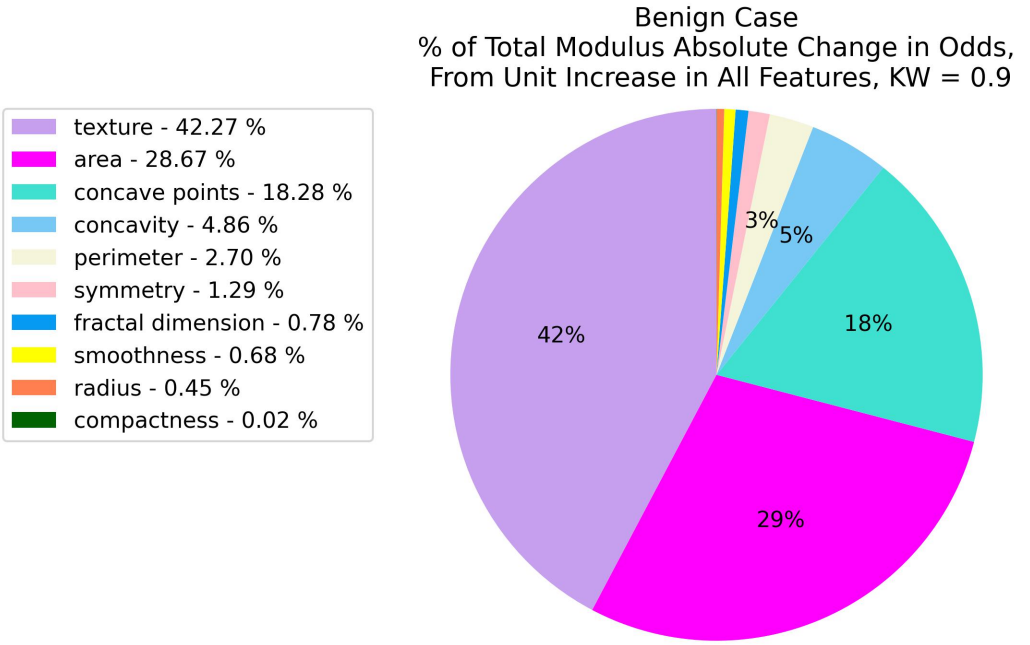

Figure B.6: XGBoost Black Box - Benign Case, % Percent of Total Modulus (Absolute Value) Absolute Change in Odds from Isolated Unit Increase For All Features, Kernel Width = 0.9

Appendix C

Kernel SHAP, Benign Case

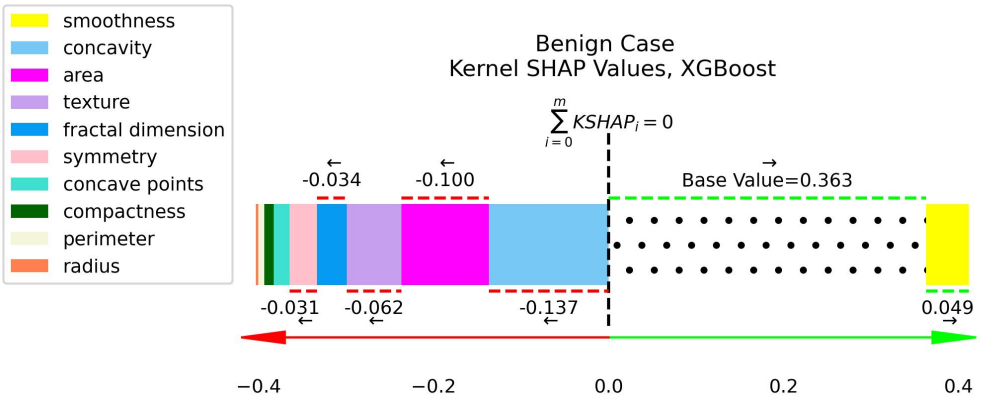

Figure C.1: XGBoost Black Box - Exact Kernel SHAP values - Benign Case, Kernel SHAP Additive Illustration

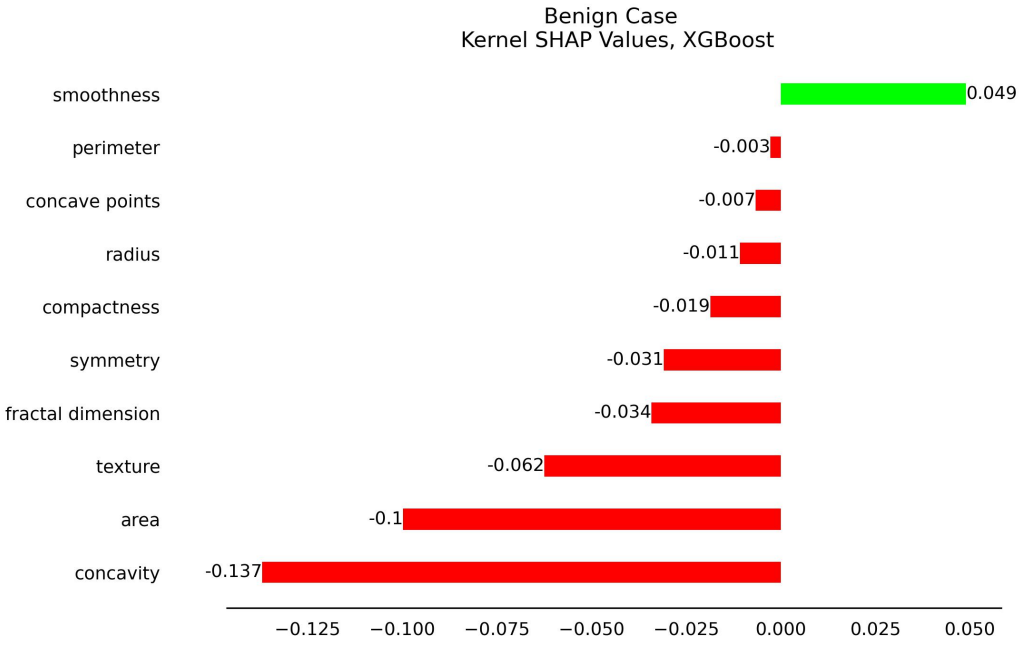

Figure C.2: XGBoost Black Box - Exact Kernel SHAP values - Benign Case, Kernel SHAP Bars Illustration

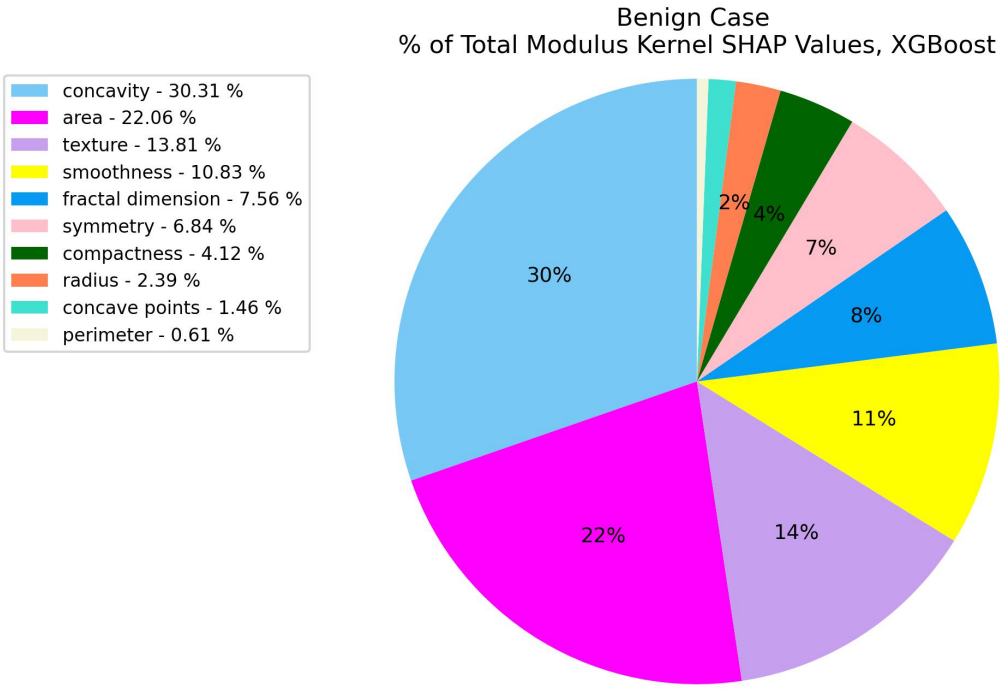

Figure C.3: XGBoost Black Box - % of Total Modulus Kernel SHAP Values - Benign Case

Appendix D

Kernel SHAP Approximations, Benign Case

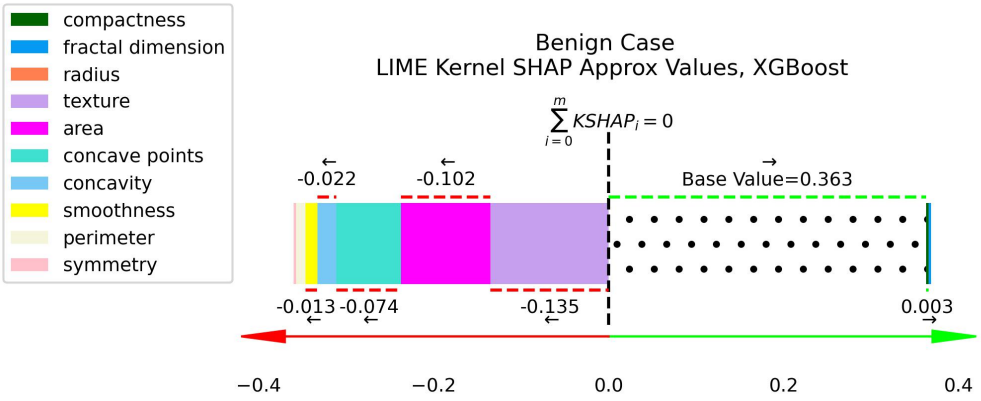

Figure D.1: XGBoost Black Box - LIME Inferred Kernel SHAP Approximation - Benign Case, Additive Illustration, Kernel Width = 2.3717

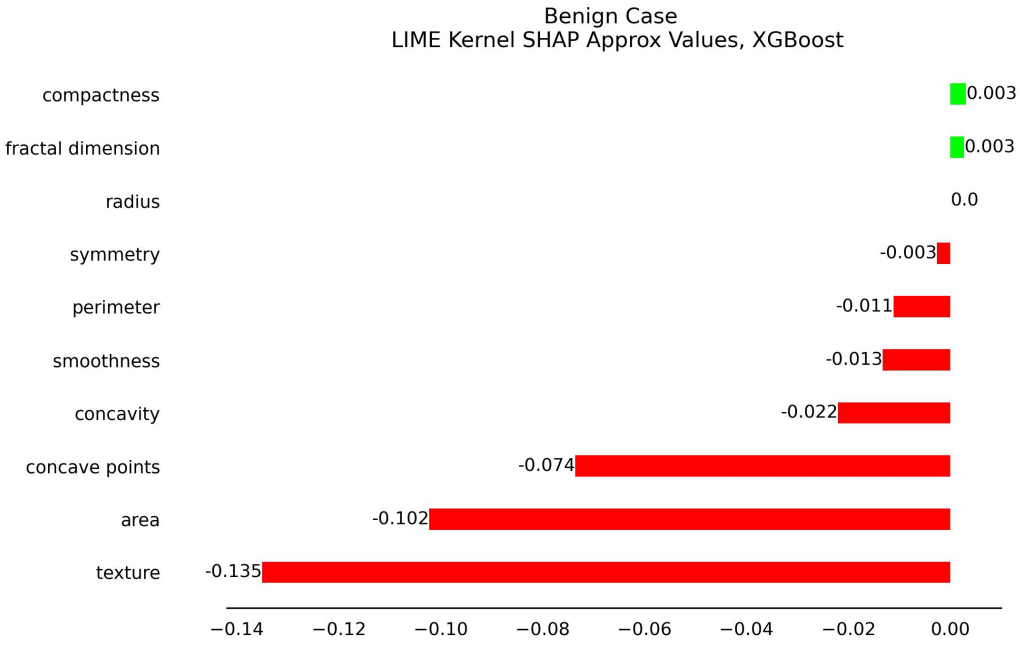

Figure D.2: XGBoost Black Box - LIME Inferred Kernel SHAP Approximation - Benign Case, Bars Illustration, Kernel Width = 2.3717

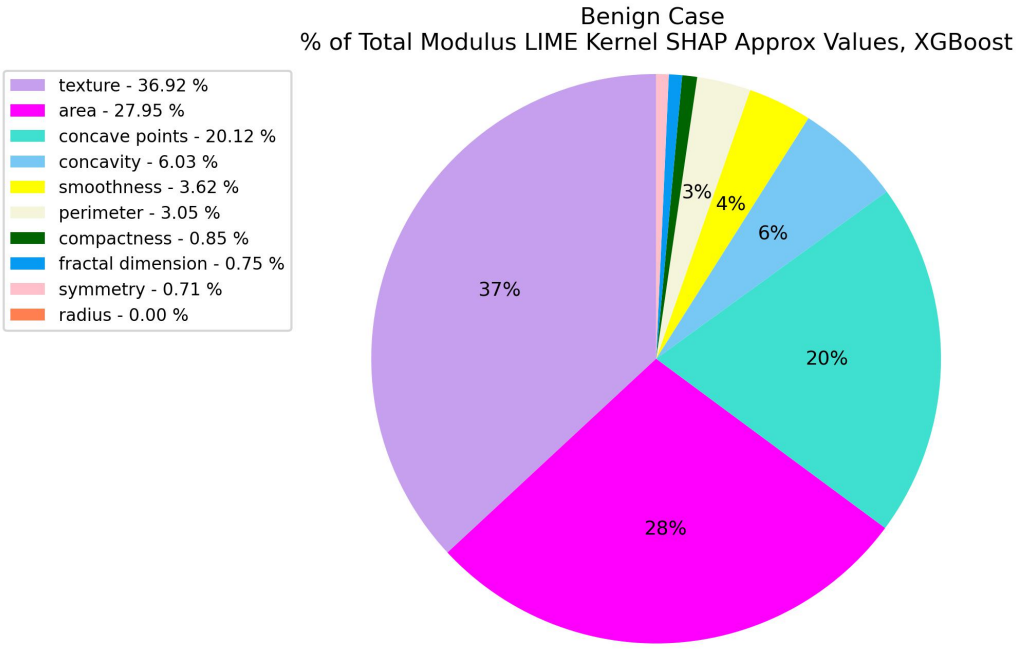

Figure D.3: XGBoost Black Box - % of Total Modulus Approx Kernel SHAP Values using LIME Inferred Kernel SHAP Approximation - Benign Case

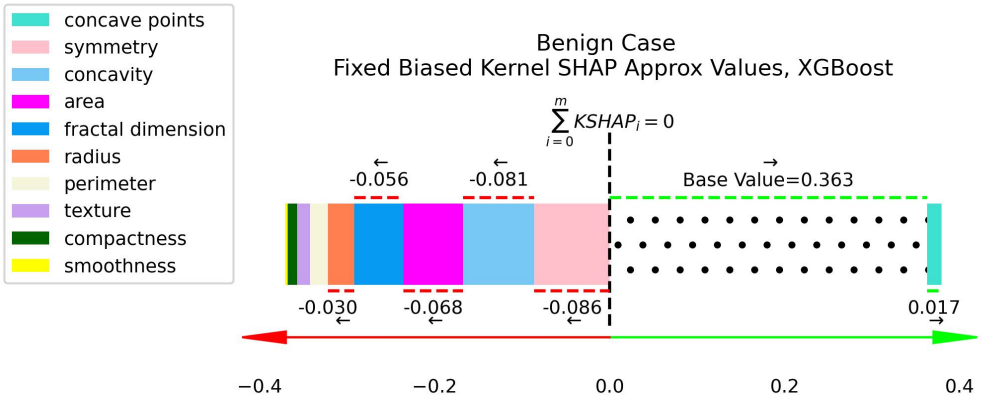

Figure D.4: XGBoost Black Box - Fixed Biased Kernel SHAP Approximation - Benign Case, Additive Illustration

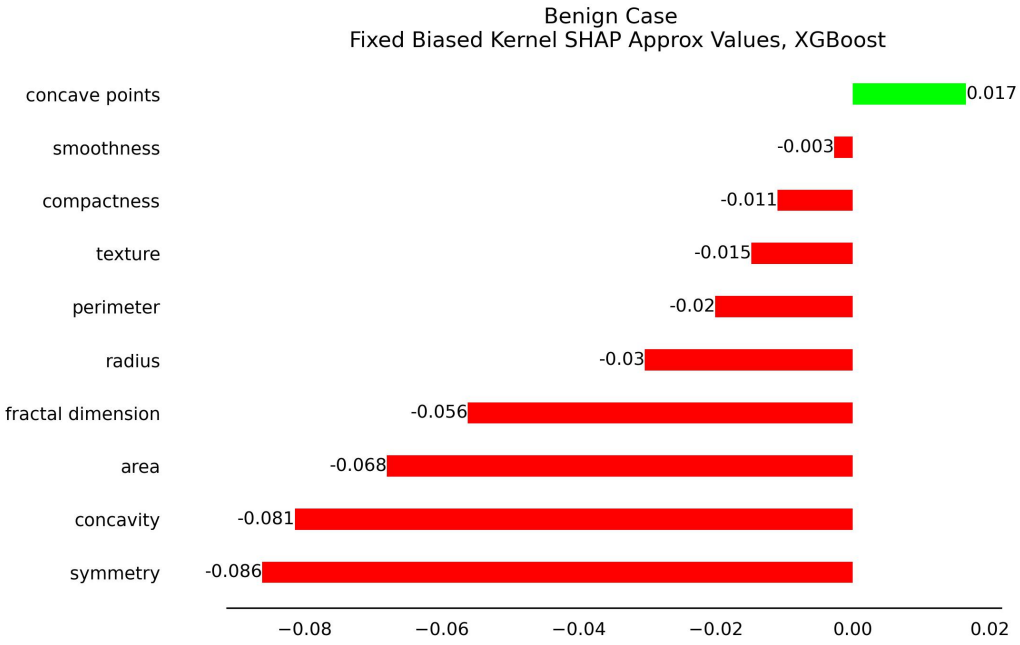

Figure D.5: XGBoost Black Box -Fixed Biased Kernel SHAP Approximation - Benign Case, Bars Illustration

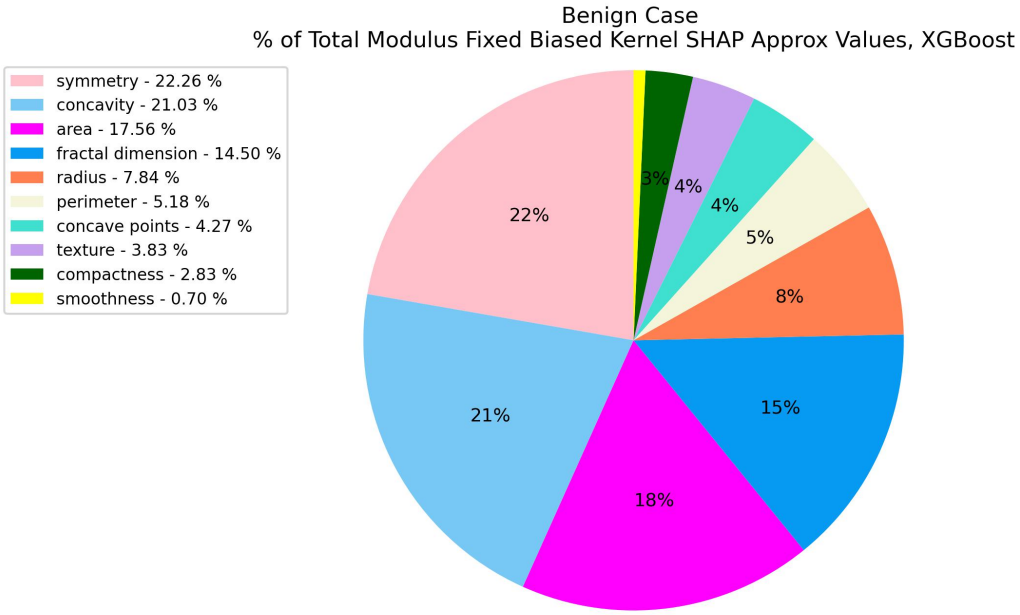

Figure D.6: XGBoost Black Box - % of Total Modulus Approx Kernel SHAP Values using Fixed Biased Kernel SHAP - Benign Case

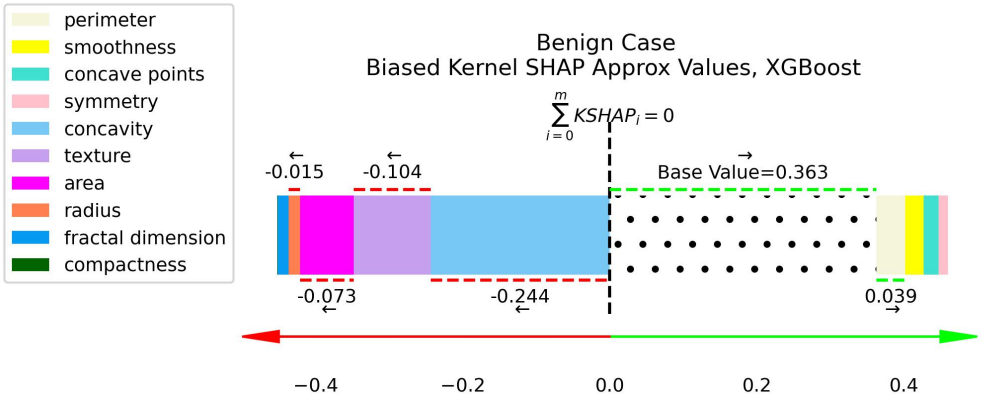

Figure D.7: XGBoost Black Box - Biased Kernel SHAP Approximation - Benign Case, Additive Illustration

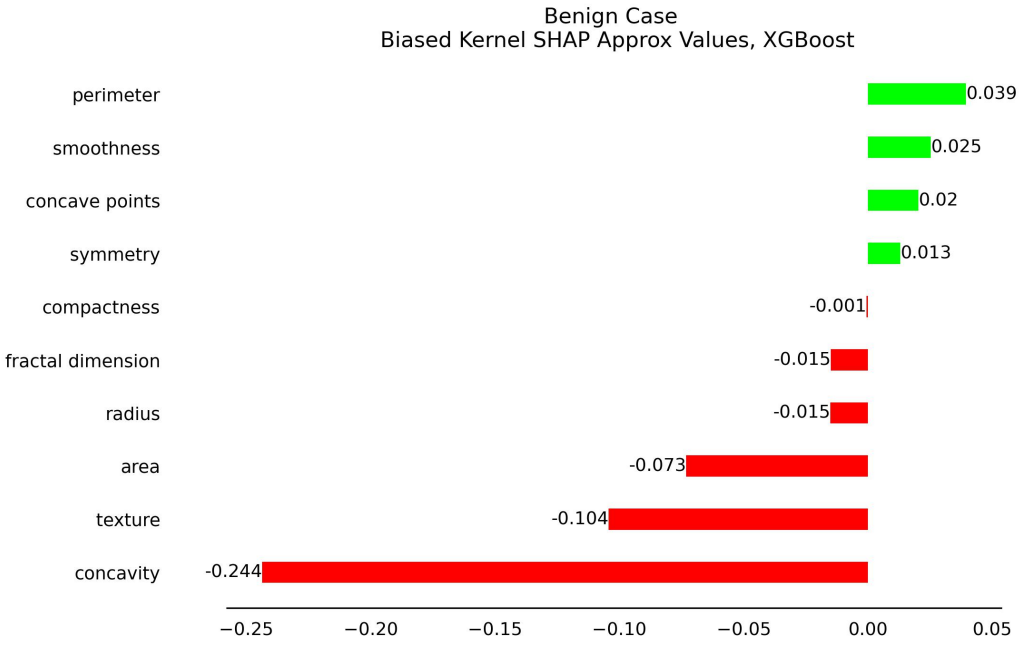

Figure D.8: XGBoost Black Box - Biased Kernel SHAP Approximation - Benign Case, Bars Illustration

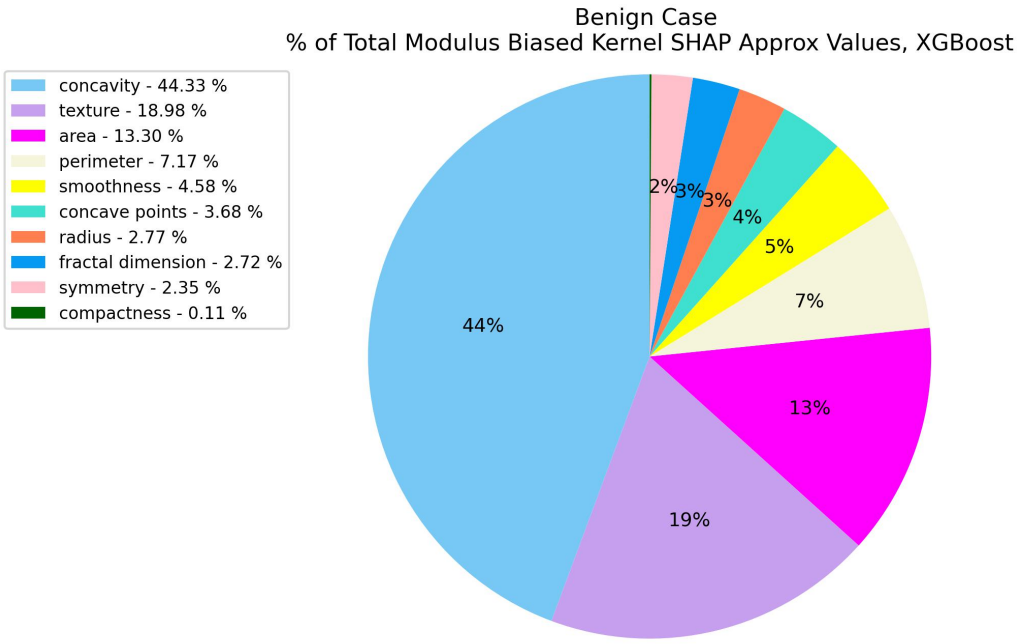

Figure D.9: XGBoost Black Box - % of Total Modulus Approx Kernel SHAP Values using Biased Kernel SHAP - Benign Case

Curriculum Vitae

Name:	Heather Hartley
Post-Secondary Education and Degrees:	McGill University Montreal, QB 2011 - 2016 B.Comm. Toronto Metropolitan University Toronto, ON 2018 - 2020 Certificate, Computer Programming Applications University of Western Ontario London, ON 2020 - 2022 M.Sc. Computer Science, Artificial Intelligence
Related Work Experience:	Teaching Assistant The University of Western Ontario 2020 Software Engineer Evertz Microsystems 2020 - 2022