# XOPSIS: An Explainable AI Method based on the Order of Preference by Similarity to Ideal Solution

Anika Rahman, *Department of Computer Science and Engineering, BRAC University, 66 Mohakhali, Dhaka - 1212, Bangladesh, anika.rahman@g.bracu.ac.bd*
Md. Golam Rabiul Alam, *Department of Computer Science and Engineering, BRAC University, 66 Mohakhali, Dhaka - 1212, Bangladesh, rabiul.alam@bracu.ac.bd*

***Abstract*—Explainable AI (XAI) techniques are essential for comprehending machine learning model predictions in a variety of fields. In this study, we introduce XOPSIS, an Explainable AI (XAI) method leveraging the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) Score and intended to offer thorough justifications for gradient boosting models. Comparing its performance with established XAI techniques, LIME and SHAP, across diverse datasets, including maternal health records, the Iris dataset, breast cancer dataset, and the Car Acceptability dataset, we highlight significant similarities with LIME in generating explanations. XOPSIS offers comprehensive explanations, consistently identifying influential features to the model's predictions. In addition, by utilizing SHAP values, we acquire a comprehensive comprehension of the model's behavior and the unique contributions of each feature to the predictions. The significance of our proposed approach lies in its ability to enhance interpretability in machine learning models, enabling stakeholders to make informed decisions across various domains. Demonstrating versatility beyond health domains, XOPSIS can be applied to finance, cybersecurity, and customer behavior analysis. Future studies should focus on practical implementation, evaluating effectiveness in real-world scenarios, and exploring integration across industries. Advancing XAI techniques like XOPSIS contributes to responsible AI use, fostering trust and transparency in complex machine learning models.**

***Index Terms*—Explainable AI, XOPSIS, LIME, SHAP, Maternal Health, Interpretability, Transparency.**

## I. Introduction

Explainable AI (XAI) has emerged as a critical component in the field of predictive modeling, aiming to address the growing concern regarding the lack of transparency and interpretability in machine learning models. With the widespread implementation of these models in diverse domains, including healthcare and beyond, understanding the factors that contribute to predictions becomes essential for ensuring accountability and enabling informed decision-making. The lack of transparency and interpretability in machine learning models [1] has been a major hurdle in their widespread adoption across diverse domains. Traditional black-box models often provide accurate predictions but fail to provide meaningful explanations behind their decisions. This lack of interpretability poses challenges for practitioners, researchers, and end-users, who rely on a comprehensive understanding of the factors influencing predictions to make accurate choices [2] and place faith in the outcomes of these models. Furthermore, it raises concerns about potential biases and fairness in decision-making processes, particularly in sensitive domains where the consequences of predictions are far-reaching.

Existing XAI methods such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) have made significant contributions to providing interpretability [3] in machine learning models. LIME approximates the model locally and assigns feature importances based on local perturbations, while SHAP uses Shapley values to assign importances through a game-theoretic approach. However, they have certain limitations in capturing the nuanced decision-making processes of complex models and providing detailed explanations across various domains beyond healthcare.

In this study, we propose XOPSIS, a novel XAI method that revolutionizes the generation of explanations for machine learning model predictions. XOPSIS surpasses the limitations of existing methods by incorporating advanced features such as calculating the highest and lowest possible feature values, performing instance ranking based on the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) score, and considering the average feature values of similar instances. Moreover, XOPSIS introduces the concept of filtering instances based on predicted target, enabling a more tailored and focused explanation generation process.

By incorporating these additional features, XOPSIS delivers comprehensive and granular explanations that capture the intricate relationships and interactions among the features in the data. The calculation of the highest and lowest possible feature values provides insights into the range of influence each feature can have on the prediction outcome. The instance ranking based on TOPSIS score ensures that the most representative and significant instances are considered during the explanation generation process. Furthermore, filtering instances based on predicted target allows XOPSIS to focus on the specific class or label of interest, enhancing the relevance and specificity of the generated explanations. Finally, considering the average feature values of similar instances adds an additional layer of context and accuracy to the explanations, enabling a more nuanced understanding of the contributing factors behind the predictions.

These innovative features set XOPSIS apart from existing

XAI methods, enabling it to generate more insightful and accurate explanations. XOPSIS promotes transparency and interpretability across several domains by giving consumers an in-depth knowledge of how machine learning models make decisions.

To evaluate the effectiveness of XOPSIS, we employ our approach to multiple datasets, including a real-world Maternal Health dataset, the benchmark Iris dataset, the Breast Cancer diagnosis dataset, and the Car Acceptability dataset. Through these applications, XOPSIS demonstrates its practical applicability and performance in generating specific instance prediction explanations. Maternal health is a domain where accurate risk prediction plays a crucial role in facilitating timely intervention and tailored care. By applying XOPSIS in this context, our aim is to enhance the interpretability of machine learning models and gain a comprehensive understanding of the factors influencing maternal health risk, including the benchmark Iris dataset, breast cancer diagnosis, and the Car Acceptability dataset.

### A. Contributions

In a nutshell, the following contributions are made by this study:

- We introduce XOPSIS as an innovative XAI method for generating comprehensive explanations of machine learning model predictions.
- We demonstrate the practical applicability of XOPSIS by leveraging it in various datasets, including a real-world Maternal Health dataset, the benchmark Iris dataset, the benchmark Breast Cancer diagnosis dataset, and the Car Acceptability dataset.
- We assess the performance of XOPSIS in generating specific instance prediction explanations, which exhibit similarities to explanations produced by other established XAI methods.

By integrating XOPSIS into the field of health risk prediction and leveraging XAI techniques, our approach aims to enhance transparency, interpretability, and accountability in machine learning models. The insights provided by XOPSIS empower healthcare providers, researchers, and stakeholders across various domains to make informed decisions, ultimately contributing to improved outcomes and fostering trust in AI technologies.

## II. Background Study

In the evolving landscape of machine learning and artificial intelligence, achieving transparency and interpretability in complex models has become pivotal for their widespread adoption and credibility across various domains. Interpretability methods [4] play a crucial role in demystifying the decision-making processes of these models, allowing stakeholders to understand and trust their outcomes. This section delves into the background study of three significant techniques: LIME, SHAP, and TOPSIS, which contribute to the explainability and insightfulness of machine learning models.

### A. LIME

LIME (Local Interpretable Model-agnostic Explanations) revolutionizes the landscape of interpretability by offering a practical solution [5] to one of the most formidable challenges in machine learning: explaining the predictions of complex models. As machine learning models have advanced in complexity, their decision-making processes have become increasingly difficult to decipher. LIME addresses this issue by providing transparent, instance-specific explanations, making it an indispensable tool for model transparency and accountability. At its core, LIME operates on the principle of approximating a black-box model's behavior around a specific instance of interest. This approach is rooted in the realization that while global explanations might be intricate, local interpretations can be more understandable [6]. To achieve this, LIME generates a dataset of perturbed instances by introducing controlled noise to the original instance's features. These perturbed instances are then used to create a surrogate interpretable model that mimics the behavior of the complex model around the instance in question.

LIME follows a systematic workflow encompassing several pivotal stages. Initially, an instance requiring an explanation is selected. This forms the basis for subsequent analysis. Subsequently, [7] minor alterations are introduced to the features of the chosen instance, thereby creating a diverse dataset through perturbation. Predictions are then derived for these perturbed instances, effectively capturing the output behavior of the underlying black-box model. A surrogate model is subsequently established, trained using the perturbed instances and their corresponding predictions. This surrogate model is deliberately constructed to be both interpretable and locally accurate. Leveraging this surrogate model, the next step involves interpreting the model's decisions, thereby extracting critical insights and feature importances. These insights contextualize the rationale behind the black-box model's specific prediction for the chosen instance. By interpreting the surrogate model's explanations, local insights emerge, unveiling the intricate interplay between features and predictions specific to the selected instance. This comprehensive workflow not only demystifies the predictions of complex models but also empowers users to make informed decisions based on the extracted insights. LIME's true strength lies in its versatility. It's model-agnostic [8], meaning it can be applied to a wide array of machine learning models without requiring knowledge of their internal architectures. This makes LIME an invaluable tool for understanding the decision-making of models like deep neural networks, random forests, support vector machines, and more. This adaptability makes LIME suitable for addressing the opacity of models ranging from image classifiers to natural language processing algorithms.

However, like any technique, LIME has its limitations. The accuracy of the surrogate model heavily depends on the quality and diversity of the perturbed instances. Also, the explanations generated by LIME are localized and may not capture global

model behavior accurately. Nonetheless, LIME's significance is undeniable. It bridges the gap between complex, high-performance models and human understanding, making it a cornerstone of explainable AI. It empowers domain experts, regulators, and end-users with the ability to validate, trust, and even enhance machine learning systems by providing a comprehensible rationale for individual predictions.

### B. SHAP

SHAP, an acronym for SHapley Additive exPlanations, a pioneering technique in the field of Explainable AI (XAI), offers a sophisticated approach to understanding the influence of individual features on the predictions made by complex machine learning models. Anchored in cooperative game theory, SHAP provides a robust framework for attributing contributions to each feature, thereby demystifying the decision-making process of black-box models. At the heart of SHAP lies the concept of Shapley values [9], a central notion in cooperative game theory used to distribute the value of a cooperative endeavor among its participants fairly. SHAP cleverly adapts this concept to machine learning models, where features collaborate to predict an outcome. In this context, SHAP quantifies the average marginal contribution of a feature by considering its impact on predictions across all possible feature combinations.

The SHAP algorithm operates by comparing the model's prediction for a specific instance with a reference prediction, typically the average prediction of the training dataset. This difference is referred [10] to as the Shapley value, representing the contribution of each feature to the variation in prediction. The crux of SHAP's power lies in its ability to disentangle the complex interplay of features, enabling us to discern which features amplify or dampen a particular prediction. In practice, SHAP generates a comprehensive set of explanations, each elucidating the role of a feature in influencing the prediction for a specific instance. These explanations manifest as positive or negative values, signifying whether a feature positively or negatively affects the prediction. Notably, the summation of SHAP values across all features equates to the disparity between the model's prediction for the instance and the reference prediction.

One of SHAP's compelling features is its universality – it is applicable to various model types, including those that are not inherently interpretable. By providing insights into feature contributions, SHAP bridges the gap between complex models and human comprehension. This attribute has far-reaching implications, from understanding the factors driving an individual prediction to identifying bias and fairness concerns within the model's decision-making process. In essence, SHAP stands as a critical tool for fostering transparency, accountability, and interpretability in the realm of machine learning. By attributing contributions to features, SHAP empowers practitioners, researchers, and end-users to trust and comprehend the decisions made by complex models, even in scenarios where the model's internal workings remain elusive.

### C. TOPSIS

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution), a key technique in Multi-Criteria Decision Making (MCDM) [11], addresses the challenge of ranking alternatives when considering multiple conflicting criteria. It operates by measuring the similarity of each alternative to the ideal solution and the dissimilarity to the anti-ideal solution. This method is valuable in scenarios where decisions involve a range of criteria, each with varying degrees of significance. While not exclusively designed for XAI, TOPSIS finds application in model interpretation [12] by facilitating nuanced comparisons between instances and shedding light on their distinctive attributes.

The procedure of TOPSIS, or Technique for Order of Preference by Similarity to Ideal Solution [13], encompasses several sequential steps. The first step involves the normalization of a decision matrix, comprising criterion values for each alternative. This normalization process ensures equitable evaluation across all criteria, preventing undue influence from disproportionately large values. Following this, the weighted normalization stage ensues, wherein each normalized value is multiplied by its corresponding criterion weight. These weights convey the relative significance of the criteria, introducing nuanced effects on the final ranking. Ideal and anti-ideal solutions are then defined for every criterion, with the ideal solution characterized by maximal values for beneficial criteria and minimal values for non-beneficial ones. Conversely, the anti-ideal solution represents [14] the least desirable values. Subsequently, the calculation of distances comes into play, involving the computation of Euclidean distances from both the ideal and anti-ideal solutions for each alternative. These distances provide insights into the relative proximity or divergence of each alternative concerning these reference points in the multidimensional criterion space. Determination of a relative closeness score follows, reflecting each alternative's performance in relation to the criteria. This score is ascertained by evaluating the ratio of the distance to the anti-ideal solution to the summation of distances to both the ideal and anti-ideal solutions. Finally, based on their relative closeness scores, alternatives are ranked, with higher scores signifying superior overall performance and more favorable rankings. The fundamental principle of TOPSIS revolves around the concept of an "ideal solution" and an "anti-ideal solution." The ideal solution represents the characteristics that an alternative should ideally possess to be considered optimal, while the anti-ideal solution embodies the exact opposite. The distance between an alternative and these two reference points forms the crux of the TOPSIS method.

When applied to XAI, TOPSIS can be employed to provide insights into instance-level predictions by assessing the proximity of instances to the ideal and anti-ideal solutions. This approach enables a comprehensive understanding [15] of the relationships between instances and the criteria they are evaluated upon. By quantifying these distances, TOPSIS

can highlight instances that align closely with the ideal solution and those that deviate significantly. Comparing TOPSIS to AHP (Analytic Hierarchy Process), another popular MCDM method [16], reveals distinct characteristics. In AHP, decision-makers assign pairwise comparisons and weights to criteria, resulting in a complex weighting process. Conversely, TOPSIS uses preset weights [17] and focuses on assessing the similarity of alternatives to the ideal and anti-ideal solutions.

The selection of TOPSIS for the XOPSIS XAI method is driven by specific reasons. One key factor is the use of uniform weights for each feature in TOPSIS, simplifying the process compared to AHP, where feature weights are ranked. Furthermore, TOPSIS is well-suited for scenarios where transparency and interpretability [18] are paramount, as it produces clear rankings based on a set of predefined criteria. In contrast, AHP's complex weight assignment may hinder transparency. Additionally, TOPSIS aligns with the goal of XOPSIS to enhance interpretability in machine learning models across various domains. Its structured and comprehensible approach facilitates meaningful explanations, promoting trust and informed decision-making. TOPSIS is particularly useful in scenarios where model predictions are influenced by a multitude of factors, [19] each with varying degrees of importance. The method facilitates the systematic comparison of instances across these diverse criteria, offering a holistic view of how individual attributes contribute to the final decision. This can be especially valuable in fields like healthcare, [20] finance, and risk assessment, where decisions are often influenced by multiple conflicting considerations. One of the defining characteristics of TOPSIS is its flexibility in handling both quantitative and qualitative data. This adaptability is vital when dealing with real-world data [21] that can vary widely in terms of format and meaning. By converting data into a standardized form, TOPSIS enables fair comparisons between different attributes and instances, regardless of their initial nature.

In conclusion, while not a traditional XAI method, TOPSIS complements the interpretability [22] landscape by providing a structured approach to understanding multi-criteria decision-making. By assessing the distance between instances and reference solutions, TOPSIS contributes to uncovering the rationale [23] behind predictions in complex models. Its ability to accommodate diverse data types and facilitate comprehensive comparisons makes it a valuable tool for understanding the intricate relationships within predictive models.

## III. Literature Review

Concerns about the transparency and interpretability of machine learning models for health risk prediction, notably in the area of maternal health, have been raised. Explainable AI (XAI) approaches have come to light as a viable solution to these challenges in response to these worries. Existing XAI methods such as LIME, SHAP, and ELI5 have been widely studied and applied in various domains, including healthcare. However, there is a need for novel XAI methods that can provide specific instance prediction explanations with high granularity. By presenting XOPSIS, an entirely new XAI method that provides thorough insights into how to make choices of machine learning models, this work seeks to narrow this gap.

### A. Explainable AI

Explainable Artificial Intelligence (XAI) techniques, which seek to shed light on the inner workings of machine learning models and make their predictions more transparent and intelligible, have gained popularity in recent years. One such method is the use of the multi-criteria decision-making technique TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution), which is an explainable AI method for ranking and prioritizing various risk factors according to their relative importance in the context of maternal health. When utilized for decision-making in a variety of fields, TOPSIS has proven to be efficient in managing several factors and producing outcomes that are easy to understand. Several studies have applied LIME and SHAP for model interpretation in various domains, including healthcare. These methods are being applied to clarify how feature importance, feature interactions, and overall model behavior relate to predictions made by machine learning models. To to the finest of the information we have, however, there hasn't been much study on the use of TOPSIS as an explainable AI technique for interpreting maternal health risk predictions and explaining specific data point predictions, along with the complementary use of LIME and SHAP for model interpretation.

In order to explain the predictions of any classifier, this paper [24] introduces a technique called LIME (Local Interpretable Model-Agnostic Explanations). The interpretability issue in black-box machine learning models is addressed by LIME by offering local explanations that highlight the key factors influencing each prediction. The method generates simplified, interpretable models around specific instances and quantifies the importance of each feature. By providing clear explanations, this method increases consumers' faith in the predictions by allowing them to see how complex classifiers make decisions.

SHAP (SHapley Additive exPlanations), a standardized method for unraveling model predictions, is presented in the study [25]. Assigning relevance levels to each feature and measuring their contribution to the prediction is done by SHAP using game theory principles. The behavior of the model is explained globally by SHAP by computing Shapley values, which reflect the minimal impact of each feature across various combinations. This approach allows for better understanding of the model's decision-making process and enables insights into individual feature contributions, leading to improved interpretability and trust in machine learning models.

The paper [26] introduces DeepLIFT, a method for learning important features in deep neural networks by propagating activation differences. By comparing the difference in neuron activations when a feature is present versus missing, DeepLIFT calculates the impact of every characteristic to the outcome. This approach allows for fine-grained feature attribution and provides insights into the important features driving the model's predictions. By quantifying feature importance, DeepLIFT enhances interpretability and understanding of deep neural networks, facilitating trust and further analysis of these complex models.

The authors of [31] propose a method for evaluating each feature's importance to a prediction generated by a model, offering insights into the significance of various features for the predictions made by the model. The use of this method in numerous domains is discussed in the study, which also emphasizes its potential to make prediction models more comprehensible and reliable. The paper serves as a valuable contribution to the field of interpretable machine learning and provides insights into the explanation of model predictions.

A method called "Anchors" has been introduced in [32] that provides interpretable explanations for individual predictions, identifying the most important features that influence a model's output. The paper indicates that Anchors performs better than conventional explanation techniques both in terms of accuracy and comprehensibility. It also illustrates the usefulness of Anchors in numerous trials. The study offers a potential method for producing precise and understandable justifications for model predictions, which makes it an important contribution to the subject of interpretable machine learning.

This paper [33] addresses a crucial gap in current Explainable AI (XAI) research by introducing Contextual Importance and Utility (CIU), drawing on principles from Decision Theory. Despite the long-standing history of explainability in AI, XAI emerged only in 2016, often neglecting valuable knowledge from other domains. CIU extends importance and utility concepts to non-linear AI models, offering a universal, model-agnostic foundation for XAI. By bridging the gap between XAI and Decision Theory, CIU enriches the discourse on transparency and interpretability in machine learning. The study underscores the significance of integrating established principles to advance the comprehensiveness of XAI methodologies.

### B. Maternal Health

Maternal health risk prediction has become a critical area of research to improve maternal health outcomes and reduce maternal morbidity and mortality. Machine learning approaches have been used in several research to forecast threats to maternal health, such as gestational diabetes, preeclampsia, premature birth, and other issues. Although the predictive accuracy of these studies has shown encouraging results, the black-box nature of machine learning models has constrained their interpretability, making it difficult to comprehend the underlying causes influencing the risk estimations.

Pre-eclampsia, a potentially dangerous pregnancy condition that involves excessive blood pressure and organ damage, is thoroughly discussed in this study [35]. The paper discusses the pathophysiology, risk factors, diagnosis, and management of pre-eclampsia, including the role of regular antenatal care, monitoring, and timely interventions. It also emphasizes how critical early detection and effective treatment are to preventing the harmful effects of pre-eclampsia on both the mother and the fetus.

This paper [38] provides an overview of the existing research on the importance of prenatal care and the diagnosis of high-risk pregnancies for effective maternal health care. It highlights the use of machine learning and deep learning algorithms in predicting risk levels based on pregnancy risk factors. Additionally, Explainable AI techniques such as LIME and SHAP are explored for providing interpretable explanations. The review emphasizes the significance of early diagnosis and appropriate treatment in reducing maternal mortality and improving maternal and fetal well-being.

A study [39] conducted in 2020 implemented a modified decision tree algorithm for diagnosing high-risk pregnancies. The study comprised six independent variables and one objective variable, and it used data from six hospitals from 2018 to 2020. IoT technology was used to gather the research's data. The results showed that, when predicting the model, the improved decision tree approach showed a 97% gain in accuracy over comparable methods. WEKA and Python software were also utilized in the algorithm's implementation.

### C. Breast Cancer

In their recent work [43] published by MDPI, Naeem and Ali delve into the domain of breast cancer diagnosis through the application of machine learning techniques. The authors explore the utilization of various machine learning methodologies for the purpose of breast cancer detection, emphasizing the potential of these techniques in contributing to accurate and efficient diagnostic procedures. This study contributes to the ongoing efforts in leveraging machine learning advancements for medical diagnosis, particularly in the critical area of breast cancer detection and classification.

In a recent preprint published on arXiv, [44] Zuluaga-Gomez delves into the realm of breast cancer diagnosis through the lens of machine learning techniques. The author's work revolves around exploring the potential of machine learning methodologies for enhancing breast cancer diagnosis accuracy and efficiency. By investigating the application of these techniques to breast cancer data, this study aims to contribute valuable insights into the field of medical diagnosis, particularly focusing on the critical area of breast cancer

detection and classification. The work adds to the growing body of research dedicated to leveraging machine learning advancements for improved medical diagnostic procedures.

### D. Car Acceptability

In the International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), the authors [47] present an article titled "Predicting Overall Car Performance Using Artificial Neural Network." This research article investigates the application of artificial neural networks (ANNs) to predict the overall performance of cars. The authors delve into the realm of machine learning by utilizing ANNs to model and forecast the intricate and multifaceted aspects of car performance. Through this work, Al-Mubayyed, Abu-Nasser, and Abu-Naser contribute to the development of predictive models in the automotive domain, enhancing the understanding and evaluation of car performance characteristics. The study showcases the potential of artificial neural networks in analyzing and forecasting complex automotive parameters, offering insights into the practical utilization of machine learning techniques in the automotive industry.

In their article [48], the authors delve into the realm of predictive modeling for car performance. Published in 2020, the study explores the application of Artificial Neural Networks (ANNs), specifically Jittered Neural Networks (JNNs), to forecast various aspects of car performance. The authors harness the power of machine learning and neural networks to analyze and predict intricate performance parameters of automobiles. By focusing on JNNs, the research aims to enhance the accuracy and efficiency of predictions, contributing to the field of automotive engineering and predictive modeling. Through this work, Al-Mobayed, Al-Madhoun, Al-Shuwaikh, and Abu-Naser offer insights into the potential of neural network-based methodologies in the analysis and prediction of car performance characteristics.

## IV. Methodology

### A. Proposed Explainable AI Method

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is a widely used multi-criteria decision-making (MCDM) method that involves ranking alternatives based on their proximity to an ideal solution, providing insights into the model's prediction behavior by representing how each feature contributes towards the prediction for a specific instance. TOPSIS is widely used in various fields such as finance, marketing, engineering, and environmental management to support decision-making processes when multiple criteria are involved. It provides a simple yet effective approach for decision-makers to compare and rank alternatives based on their performance against multiple criteria simultaneously, making it a valuable tool in decision analysis and decision support systems. The XOPSIS method is executed as per the following steps. The flowchart presented in Figure 1 outlines the key steps, focusing on the initial stages of Data Preprocessing, Classification/Prediction, Topsis score calculation and XOPSIS Explanation Generation.

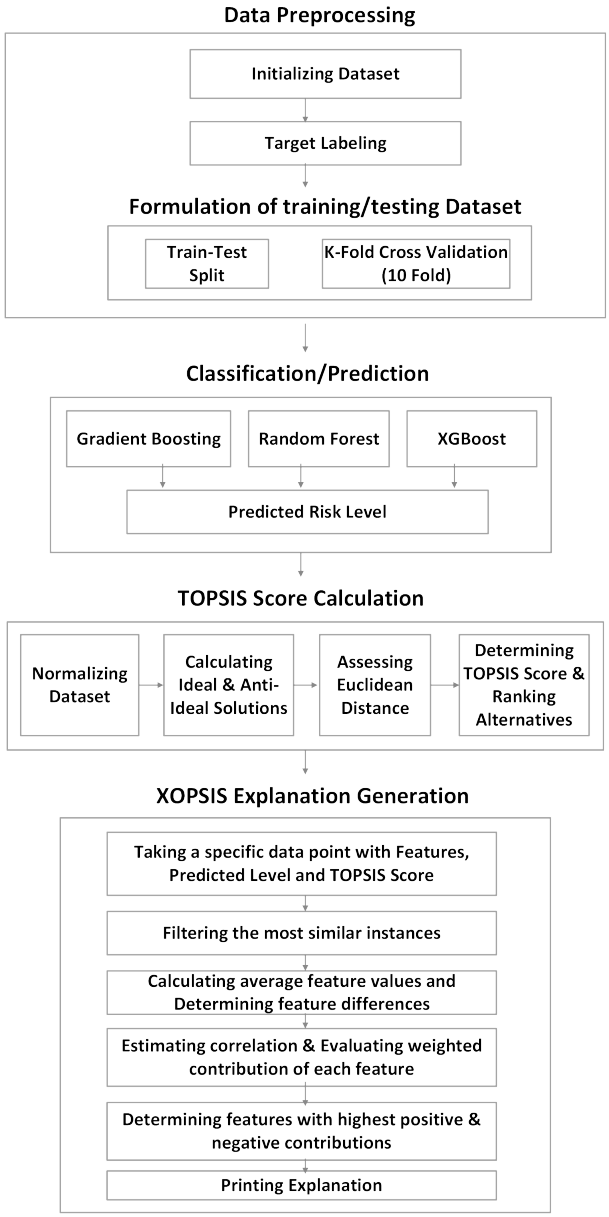



Fig. 1: Top Level Overview of the XOPSIS Method

In the Data Preprocessing step, the flowchart begins by initializing the dataset and performing target labeling to prepare the data for analysis. Subsequently, the formulation of training and testing datasets is conducted using two techniques: train-test split and k-fold cross validation. These techniques ensure the availability of suitable data subsets for model training and evaluation, promoting robustness and generalizability of the subsequent analysis.

Moving to the next step, the flowchart transitions to the classification/prediction phase. Here, the three main algorithms utilized for risk assessment, namely gradient boosting, random forest, and XGBoost, are highlighted. These algorithms are among the 14 employed in the study, chosen for their proven

effectiveness in predictive modeling. By applying these algorithms to the preprocessed data, the flowchart facilitates the generation of predicted risk labels, offering valuable insights into the classification of risk levels in the target population.

**Algorithm 1** XOPSIS Based Explanation Method

**procedure** XOPSISEXPLANATION(Dataset X, Weight vector $w_i$, Data Point)

**1.** Normalize the dataset:

$$X' \leftarrow \frac{X}{\sum X}$$

**2.** Calculate the weighted normalized decision matrix:

$$X'' \leftarrow X' \cdot w_i$$

**3.** Calculate the Euclidean distance to ideal and anti-ideal solutions:

$$D^+ \leftarrow \sqrt{\sum (X'' - A^+)^2}$$

$$D^- \leftarrow \sqrt{\sum (X'' - A^-)^2}$$

**4.** Determine the TOPSIS score for each alternative:

$$\text{TOPSIS Score} \leftarrow \frac{D^-}{D^+ + D^-}$$

**5.** Rank alternatives based on TOPSIS Scores in descending order

**6.** Take a specific Data Point as input and filter the most similar instances

based on TOPSIS Score and predicted risk level

**7.** Calculate average feature values for these instances:

$$\text{Avg. Features} \leftarrow \frac{\sum \text{features of instances}}{\text{number of instances}}$$

**8.** Determine the feature differences:

$$\text{Feature Diff} \leftarrow \text{DataPoint Features} - \text{Avg. Features}$$

**9.** Estimate correlation between each feature and target variable:

$$\text{Corr} \leftarrow \text{CalculateCorrelation}(df, \text{'RiskLevel'})$$

**10.** Evaluate the weighted contribution of each feature:

$$\text{Weighted Contrib} \leftarrow \text{Feature Diff} \cdot \text{Corr}$$

**11.** Detect features with positive and negative contributions:

**if** Weighted Contrib $\geq 0$ **then**

Positive Contrib $\leftarrow$ Pos. Weighted Contrib.

**else**

Negative Contrib $\leftarrow$ Neg. Weighted Contrib.

**end if**

**12.** Print the Explanation: Data Point index, actual risk level, predicted risk level, features with highest positive and negative contributions

**end procedure**

Subsequently, the flowchart progresses to the third step: Topsis score calculation. This step involves the computation of Topsis scores, a multi-criteria decision-making method, which further refines the risk assessment process. The Topsis score calculation enables the evaluation of alternatives based on multiple attributes, assisting in the identification of the most optimal options.

Assuming that, we start with an input dataset consisting of m alternatives and n criteria, represented by a matrix X with dimensions m x n. Each row of X corresponds to an alternative, and each column represents a criterion. Next, we normalize the dataset by dividing each element by the sum of its corresponding column. This normalization step ensures that each criterion is treated equally and comparable across alternatives.

After normalization, we calculate the weighted normalized decision matrix, where each element of the normalized matrix is multiplied by the weight assigned to its corresponding criterion. Then, we determine the ideal and anti-ideal solutions. The ideal solution represents the best possible values for each criterion, while the anti-ideal solution represents the worst possible values. These solutions are derived by finding the maximum and minimum values for each criterion in the dataset, respectively. Using the ideal and anti-ideal solutions, we calculate the Euclidean distance of each alternative to both solutions. The Euclidean distance measures the similarity or dissimilarity between an alternative and the ideal or anti-ideal solution. With the calculated Euclidean distances, we can determine the TOPSIS score for each alternative.

The TOPSIS Score is calculated as the ratio of the distance to the anti-ideal solution divided by the sum of the distances to both the ideal and anti-ideal solutions. Finally, based on the TOPSIS scores, the alternatives are ranked in descending order. Higher TOPSIS scores indicate better rankings, reflecting the alternatives' proximity to the ideal solution and their distance from the anti-ideal solution.

Figure 1 showcases a comprehensive flowchart representing the step-by-step execution of our proposed XOPSIS methodology. This methodology, displayed in this figure, encompasses the sequential steps involved in generating explanations for the predicted risk level of a given data point. By utilizing TOPSIS scores and weighted contributions, the flowchart provides valuable insights into the specific features that contribute positively or negatively to the prediction. After calculating TOPSIS Score, the process takes a specific Data Point as an input, which contains several features along with Ideal solutions, anti-ideal solutions, TOPSIS Score, and Predicted Risk Level.

The next step in the process is to filter out the most similar instances to the given data point based on their TOPSIS Score in descending order and the predicted risk level. This is done to identify instances that are most similar to the given data point and can provide useful insights into predicting the risk level.

Once the most similar instances are identified, the process calculates the average feature values for these instances. This step helps in determining the average feature values for instances that are similar to the given data point. The next step is to determine the feature differences between the given

data point and the average feature values. This step helps in identifying the features that are different between the given data point and the average feature values.

The process then estimates the correlation between each feature and the target variable (Risk Level). This step helps in evaluating the correlation between each feature and the target variable and identifying the features that are most correlated with the target variable. The process then evaluates the weighted contribution of each feature to the prediction by multiplying the feature differences with the correlations. This step helps in determining the contribution of each feature towards predicting the risk level. The next step is to detect features with the highest positive and negative contributions based on their weighted values. This step helps in identifying the features that have the most significant positive and negative impact on predicting the risk level.

Finally, the process prints an "Explanation" that includes the data point index, actual risk level, predicted risk level, and the features with the highest positive and negative contributions indicating the completion of the process.

*1) Algorithm Application on a Small Dataset*

In this section, we will apply the XOPSIS-based explanation algorithm step by step on a sample small dataset. The XOPSIS algorithm is designed to provide explanations for data points by identifying the contributions of individual features towards the predicted outcome. The sample dataset shown in Table I we will be using consists of various health-related measurements, such as age, blood pressure, body temperature, and heart rate, along with the corresponding risk level assigned to each individual.

The goal of our algorithm is to analyze the dataset, calculate the weighted contributions of each feature, and identify the features that have the most positive or negative impact on the risk level prediction. By doing so, we aim to gain insights into which factors are influential in determining the risk level and provide an explanation for the prediction outcome. Now, we will apply the XOPSIS algorithm meticulously, dissecting each step, to unravel the intricate web of feature contributions.

| Serial | Age | Systolic BP | Diastolic BP | BS | Body Temp | Heart Rate | Risk Level |
|---|---|---|---|---|---|---|---|
| 1 | 25 | 130 | 80 | 15 | 98.6 | 86 | high |
| 2 | 30 | 140 | 85 | 12 | 98 | 70 | high |
| 3 | 23 | 130 | 70 | 7.5 | 98 | 78 | mid |
| 4 | 20 | 120 | 75 | 7.3 | 100 | 70 | mid |
| 5 | 50 | 140 | 90 | 15 | 98 | 90 | high |
| 6 | 21 | 90 | 65 | 6.8 | 98 | 76 | low |
| 7 | 22 | 100 | 65 | 7.2 | 98 | 70 | low |
| 8 | 17 | 85 | 60 | 9 | 102 | 86 | mid |
| 9 | 23 | 90 | 60 | 6.4 | 98 | 76 | low |
| 10 | 23 | 120 | 80 | 7 | 98 | 66 | low |

TABLE I: Sample Dataset

After train-test split (80:20) and preprocessing, we obtain the following sample train-set.

After train-test split (80:20) and preprocessing, we obtain the following sample test-set.

*a) Step 01*

In this step, we divide each element by the maximum value of its corresponding column of the sample dataset. For example, to normalize the first element of the "Age" column, we divide 25 by the maximum value of all the values in the "Age" column (25 / 50 = 0.50). Similarly, we perform this calculation for each element in the dataset. After Normalizing, we obtain the dataset as follows.

| Serial | Age | Systolic BP | Diastolic BP | BS | Body Temp | Heart Rate | Risk Level |
|---|---|---|---|---|---|---|---|
| 6 | 21 | 90 | 65 | 6.8 | 98 | 76 | 0 |
| 1 | 25 | 130 | 80 | 15 | 98.6 | 86 | 2 |
| 8 | 17 | 85 | 60 | 9 | 102 | 86 | 1 |
| 3 | 23 | 130 | 70 | 7.5 | 98 | 78 | 1 |
| 10 | 23 | 120 | 80 | 7 | 98 | 66 | 0 |
| 5 | 50 | 140 | 90 | 15 | 98 | 90 | 2 |
| 4 | 20 | 120 | 75 | 7.3 | 100 | 70 | 1 |
| 7 | 22 | 100 | 65 | 7.2 | 98 | 70 | 0 |

TABLE II: Sample Train Dataset

| Serial | Age | Systolic BP | Diastolic BP | BS | Body Temp | Heart Rate | Risk Level |
|---|---|---|---|---|---|---|---|
| 9 | 23 | 90 | 60 | 6.4 | 98 | 76 | 0 |
| 2 | 30 | 140 | 85 | 12 | 98 | 70 | 2 |

TABLE III: Sample Test Dataset

$$X' = \begin{bmatrix} 0.50 & 0.93 & 0.89 & 1.00 & 0.97 & 0.96 \\ 0.60 & 1.00 & 0.94 & 0.80 & 0.96 & 0.78 \\ 0.46 & 0.93 & 0.78 & 0.50 & 0.96 & 0.87 \\ 0.40 & 0.86 & 0.83 & 0.49 & 0.98 & 0.78 \\ 1.00 & 1.00 & 1.00 & 1.00 & 0.96 & 1.00 \\ 0.42 & 0.64 & 0.72 & 0.45 & 0.96 & 0.84 \\ 0.44 & 0.71 & 0.72 & 0.48 & 0.96 & 0.78 \\ 0.34 & 0.61 & 0.67 & 0.60 & 1.00 & 0.96 \\ 0.46 & 0.64 & 0.67 & 0.43 & 0.96 & 0.84 \\ 0.46 & 0.86 & 0.89 & 0.47 & 0.96 & 0.73 \end{bmatrix}$$

*b) Step 02*

After normalizing the dataset in step 1, we proceed to step 2, which involves calculating the weighted normalized decision matrix. In this step, we multiply the normalized dataset by the weight vector using equal weights. Here, we consider a sample weight vector obtained using equal weights: $w_i = [\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}]$. Let us apply this step to the sample dataset. For example, to calculate weighted normalized decision matrix, the first element of the "Age" column, we obtain 0.0874 from Step 01 and then multiply it by 0.167 as we obtain from $w_i$ (0.50*0.167) = 0.083333. Similarly, we perform this calculation for each element in the dataset. Finally, we obtain the dataset as follows.

$$X'' = X' \times w_i = \begin{bmatrix} 0.08 & 0.16 & 0.15 & 0.17 & 0.16 & 0.16 \\ 0.10 & 0.17 & 0.16 & 0.13 & 0.16 & 0.13 \\ 0.08 & 0.16 & 0.13 & 0.08 & 0.16 & 0.14 \\ 0.07 & 0.14 & 0.14 & 0.08 & 0.16 & 0.13 \\ 0.17 & 0.17 & 0.17 & 0.17 & 0.16 & 0.17 \\ 0.07 & 0.11 & 0.12 & 0.08 & 0.16 & 0.14 \\ 0.07 & 0.12 & 0.12 & 0.08 & 0.16 & 0.13 \\ 0.06 & 0.10 & 0.11 & 0.10 & 0.17 & 0.16 \\ 0.08 & 0.11 & 0.11 & 0.07 & 0.16 & 0.14 \\ 0.08 & 0.14 & 0.15 & 0.08 & 0.16 & 0.12 \end{bmatrix}$$

In this step, each element in the normalized dataset $(X')$ is multiplied by its corresponding weight value from the weight vector $(w_i)$. For example, the first element in the weighted normalized decision matrix $(X'')$ is calculated by multiplying the first element of $(X')$ by the first weight value from $w_i$.

*c) Step 03*

To determine the ideal solution and anti-ideal solution, we need to find the maximum and minimum values for each criterion in the dataset. The ideal solution represents the best possible values for each criterion, so the ideal solution is, $A^+$ = [0.166667, 0.166667, 0.166667, 0.166667, 0.166667, 0.166667]. On the other hand, the anti-ideal solution represents the worst possible values, so the anti-ideal solution is, $A^-$ = [0.056667, 0.101190, 0.111111, 0.071111, 0.160131, 0.122222].

Step 3 of the algorithm involves calculating the Euclidean distance to the ideal and anti-ideal solutions for each data point in the dataset. Let's calculate the Euclidean distances for the given dataset using the ideal solution and anti-ideal solution we determined earlier. For each data point, we calculate the Euclidean distance to the ideal solution $(D^+)$ and the Euclidean distance to the anti-ideal solution $(D^-)$.

Data Point 1:

- Features: [0.083333, 0.154762, 0.148148, 0.166667, 0.161111, 0.159259]

- Euclidean distance to the ideal solution (D+):

$$\begin{aligned}
&= \sqrt{(0.083 - 0.167)^2 + (0.155 - 0.167)^2} \\
&\quad + \sqrt{(0.148 - 0.167)^2 + (0.167 - 0.167)^2} \\
&\quad + \sqrt{(0.161 - 0.167))^2 + (0.159 - 0.167)^2} \\
&= \sqrt{0.007056 + 0.000144 + 0.000361} \\
&\quad + \sqrt{0 + 0.000036 + 0.000064} \\
&= \sqrt{0.007661} \approx 0.086688
\end{aligned}$$

- Euclidean distance to the anti-ideal solution (D-):

$$\begin{aligned}
&= \sqrt{(0.083 - 0.057)^2 + (0.155 - 0.101)^2} \\
&\quad + \sqrt{(0.148 - 0.111)^2 + (0.167 - 0.071)^2} \\
&\quad + \sqrt{(0.161 - 0.160))^2 + (0.159 - 0.122)^2} \\
&= \sqrt{0.000676 + 0.002916 + 0.001369} \\
&\quad + \sqrt{0.009216 + 0.000001 + 0.001369} \\
&= \sqrt{0.015547} \approx 0.124323
\end{aligned}$$

Similarly, we calculate the Euclidean distances for the remaining data points in Table IV.

*d) Step 04*

Step 4 of the algorithm involves determining the TOPSIS score for each data point. The TOPSIS score represents the relative closeness of each alternative (data point) to the ideal and anti-ideal solutions. Let's calculate the TOPSIS score for each data point using the Euclidean distances we obtained in Step 3. For each data point, we calculate the TOPSIS score as follows:

$$\text{TOPSIS Score} = \frac{D^-}{D^+ + D^-} \tag{1}$$

| Data Point | D+ | D- |
|---|---|---|
| 1 | 0.086688 | 0.124323 |
| 2 | 0.083999 | 0.110611 |
| 3 | 0.130746 | 0.065238 |
| 4 | 0.141565 | 0.052662 |
| 5 | 0.006536 | 0.174871 |
| 6 | 0.155071 | 0.025722 |
| 7 | 0.148484 | 0.028570 |
| 8 | 0.154832 | 0.047424 |
| 9 | 0.156765 | 0.027899 |
| 10 | 0.137583 | 0.059601 |

TABLE IV: Euclidean distance to the ideal solution and the anti-ideal solution

For example, for Data Point 1, we obtain the TOPSIS Score $= \frac{0.124323}{0.086688+0.124323} \approx 0.58917823$. Similarly, we calculate the TOPSIS Score for the remaining data points as follows in Table V.

| Data Point | D+ | D- | TOPSIS Score |
|---|---|---|---|
| 1 | 0.086688 | 0.124323 | 0.58917823 |
| 2 | 0.083999 | 0.110611 | 0.56837344 |
| 3 | 0.130746 | 0.065238 | 0.33287573 |
| 4 | 0.141565 | 0.052662 | 0.27113549 |
| 5 | 0.006536 | 0.174871 | 0.96397072 |
| 6 | 0.155071 | 0.025722 | 0.1422742 |
| 7 | 0.148484 | 0.028570 | 0.16136506 |
| 8 | 0.154832 | 0.047424 | 0.23447515 |
| 9 | 0.156765 | 0.027899 | 0.15108087 |
| 10 | 0.137583 | 0.059601 | 0.30226133 |

TABLE V: Euclidean distance to the ideal solution and the anti-ideal solution with TOPSIS Score

*e) Step 05*

To apply Step 5 of the algorithm, we need to rank the alternatives (data points) based on their TOPSIS scores in descending order. Let's sort the data points in descending order of their TOPSIS scores and create a table to display the results.

| Rank | Data Point | TOPSIS Score |
|---|---|---|
| 1 | 5 | 0.96397072 |
| 2 | 1 | 0.58917823 |
| 3 | 2 | 0.56837344 |
| 4 | 3 | 0.33287573 |
| 5 | 10 | 0.30226133 |
| 6 | 4 | 0.27113549 |
| 7 | 8 | 0.23447515 |
| 8 | 7 | 0.16136506 |
| 9 | 9 | 0.15108087 |
| 10 | 6 | 0.1422742 |

TABLE VI: TOPSIS Ranking

In Table VI, the data points have been ranked based on their TOPSIS scores, with the highest score being ranked first.

*f) Step 06*

To filter the most similar instances based on TOPSIS Scores and predicted risk level from the test sample dataset for the specific data point (30, 140, 85, 12.0, 98.0, 70, High), we will compare the TOPSIS scores and predicted risk levels of each instance in the dataset. For the TOPSIS scores, we can see from the previous step that the instances are ranked in descending order. Now, we will filter the instances that have the highest TOPSIS scores and the same predicted risk level as the specific data point.

From the Table VII, we can see that instances 5 and 1 have the highest TOPSIS scores (0.949, 0.651 and 0.4999, respectively) and the predicted risk level of "high." Therefore, we will filter instances 5 and 1 as the most similar instances to the specific data point based on TOPSIS scores and predicted risk level.

| Rank | Data Point | Risk Level | TOPSIS Score |
|---|---|---|---|
| 1 | 5 | High | 0.963971 |
| 2 | 1 | High | 0.589178 |

TABLE VII: Filtered Similar Instances for High Predicted Risk Level

In this Table VII, the instances that have the highest TOPSIS scores and match the predicted risk level (High) are Data Point 5 and Data Point 1 respectively.

*g) Step 07*

To calculate the average feature values for the instances that match the predicted risk level, we sum up the feature values of those instances and divide by the number of instances. Here are the average feature values: Average Features for Risk Level High:

- Age: (50 + 25) / 2 = 37.5
- SystolicBP: (140 + 130) / 2 = 135.0
- DiastolicBP: (90 + 80) / 2 = 85.0
- BS: (15 + 15) / 2 = 15.0
- BodyTemp: (98 + 98.6) / 2 = 98.3
- HeartRate: (90 + 86) / 2 = 88.0

*h) Step 08*

Step 8 of the algorithm involves determining the feature differences between the specific data point and the average feature values calculated in step 7. Let's calculate the feature differences for the provided specific data point (30, 140, 85, 12.0, 98.0, 70) and the average feature values:

Feature Differences:

- Age: 30 - 37.5 = -7.5
- SystolicBP: 140 - 135 = 5.0
- DiastolicBP: 85 - 85 = 0.0
- BS: 12 - 15 = -3.0
- BodyTemp: 98 - 98.3 = -0.3
- HeartRate: 70 - 88 = -18.0

*i) Step 09*

Table VIII displays the correlation values between each feature and the Risk Level in the dataset. Positive correlation values indicate a positive relationship with the Risk Level, while negative correlation values indicate a negative relationship. The higher the correlation value (closer to 1 in absolute value), the stronger the correlation with the Risk Level. The features include Age, SystolicBP, DiastolicBP, BS (Blood Sugar), BodyTemp (Body Temperature), and HeartRate.

| Feature | Correlation with Risk Level |
|---|---|
| Age | 0.566361 |
| SystolicBP | 0.738184 |
| DiastolicBP | 0.694773 |
| BS | 0.893124 |
| BodyTemp | 0.119903 |
| HeartRate | 0.540133 |

TABLE VIII: Correlation between Features and Risk Level

*j) Step 10*

In this step, we need to calculate the weighted feature contributions. For this, We will multiply the feature differences by their respective correlations.

Here are the calculations:

- Age: -7.5 * 0.566361 = -4.247705
- SystolicBP: 5.0 * 0.738184 = 3.690919
- DiastolicBP: 0 * 0.694773 = 0.0000
- BS: -3.0 * 0.893124 = -2.679373
- BodyTemp: -0.3 * 0.119903 = -0.035971
- HeartRate: -18.0 * 0.540133 = -9.722396

These values represent the weighted feature contributions.

*k) Step 11*

| Feature | Normalized Weighted Contribution | Contribution Type |
|---|---|---|
| Age | -0.21 | Negative |
| SystolicBP | 0.18 | Positive |
| DiastolicBP | 0.00 | Positive |
| BS | -0.13 | Negative |
| BodyTemp | -0.00 | Negative |
| HeartRate | -0.48 | Negative |

TABLE IX: Weighted Feature Contributions to Risk Level

Table IX displays the weighted contributions of each feature to the Risk Level. The contributions are calculated by multiplying the correlation value of each feature with the Risk Level by the respective weight as shown in the previous step. If the weighted contribution is more than or equal to zero, it is classified as a positive contribution, indicating a positive impact on the Risk Level. Conversely, if the weighted contribution is less then zero, it is classified as a negative contribution, indicating a negative impact on the Risk Level.

*l) Step 12*

Finally, we will print the explanation and plot for this specific data point.

### *B. Maternal Health Risk Prediction and Explanation*

*1) Data Collection*

We collected data from the publicly available dataset "Maternal Health Risk Data" obtained from Kaggle [49]. The dataset contains 1014 rows and includes the following six features: Age, SystolicBP, DiastolicBP, BS, BodyTemp, and HeartRate. These features are potential risk factors associated with maternal health.

- Age: The age of the patient in years
- SystolicBP: Upper value of the blood pressure and the measurement in millimeters of mercury (mmHg)
- DiastolicBP: Lower value of the blood pressure and the measurement in millimeters of mercury (mmHg)
- BS: The blood sugar (glucose) level measurement in millimoles per liter (mmol/L)
- BodyTemp: The human body temperature measurement in degrees Fahrenheit (°F)
- HeartRate: The heart rate measurement in beats per minute (bpm)

The dataset includes a target variable called 'Risk Level' which is categorized into three values: Low, Mid, and High Risk. This variable represents the risk level of maternal health based on the given features.

*2) Dataset Preprocessing*

*a) Checking for Null Values*

The dataset was checked for any missing or null values. It was found that there were no null values present in the dataset, indicating that the dataset is complete in terms of data availability.

*b) Encoding of Categorical Variable*

The target variable, which contains categorical values (Low, Mid, High), was encoded using the ordinal encoder. Ordinal encoding is used to convert categorical variables with ordered categories into numerical representations. In this case, the "Risk Level" categories were encoded as (0, 1, 2) respectively, based on the order of risk levels (Low, Mid, High). Encoding categorical variables is required to convert them into numerical representations that can be utilized as input to machine learning algorithms. Ordinal encoding was used in this case to maintain the order of categories and represent the risk levels as numerical values. Table X shows the distribution of the dataset.

| Risk Level | Frequency | Percentage |
|---|---|---|
| Low Risk | 406 | 40.04 |
| Mid Risk | 336 | 33.14 |
| High Risk | 272 | 26.82 |
| Total | 1014 | 100.0 |

TABLE X: Data Distribution By Risk Level

*c) Feature Scaling*

Standard Scaling (also known as z-score normalization) was applied to the numerical features in the dataset, including Age, Systolic BP, Diastolic BP, Blood Sugar level (BS), Body Temperature, and Heart Rate. Standard scaling transforms the features to have zero mean and unit variance, which helps in bringing all the features to a similar scale, making them comparable and avoiding any dominance of one feature over the others during model training. Feature scaling is essential to normalize the numerical features and bring them to a similar scale, avoiding any potential bias towards features with larger values. Standard scaling helps in centering the features around zero with unit variance, which can improve the model's training and prediction accuracy.

*d) Train-Test Split*

The dataset was divided into training and testing sets in an 80:20 ratio using the train-test split approach. This means that 80% of the dataset was used to train the machine learning model, with the remaining 20% left aside for testing and evaluating the performance of the trained model. This aids in determining the generalization capacity and performance of the model on previously encountered data. The train-test split is critical for evaluating the performance of the model on unknown data and its generalization capacity. We may obtain an unbiased estimate of the performance and identify any potential overfitting issues by evaluating a subset of the dataset.

*e) K-fold Cross-Validation*

This technique divides the dataset into k equally sized folds and uses them for training and testing iteratively in order to assess the performance of a machine learning model. The dataset is divided into k equally sized folds (e.g., k=10, meaning 10 folds). The model is trained on k-1 folds (i.e., k-1 parts of the dataset) and tested on the remaining 1 fold (i.e., the remaining part of the dataset). This process is repeated k times, with each fold being used once as the test set and the remaining k-1 folds used as the training set in each iteration. The performance metrics, such as accuracy, precision, recall, F1 score, etc., are calculated for each fold, and the average performance is reported as the final evaluation of the model.

The goal of utilizing k-fold cross-validation is to achieve a more trustworthy estimate of the performance by leveraging the whole dataset for both training and testing. It aids in decreasing model performance variability due to variable train-test splits and enables a more robust evaluation of the performance of the model.

*f) Hyperparameter Tuning*

Hyperparameters are the parameters of a machine learning model that are not learned during the training process and need to be set manually. Examples of hyperparameters include the learning rate, regularization strength, number of estimators, etc. Hyperparameter tuning is the process of finding the optimal values for these hyperparameters to optimize the performance of the model.

Grid Search CV is a popular hyperparameter tuning technique that exhaustively searches for the best combination of hyperparameter values from a predefined grid of possible values. A grid of hyperparameter values is defined, specifying the possible values for each hyperparameter. The model is trained and evaluated using k-fold cross-validation for each combination of hyperparameter values from the grid. The performance metrics are recorded for each combination of hyperparameter values. The combination of hyperparameter values that results in the best performance is selected as the optimal set of hyperparameter values. The purpose of using Grid Search CV is to systematically search for the optimal hyperparameter values to maximize the outcome of the model. It helps in finding the best hyperparameter values that result in the best model performance on the given dataset, improving the accuracy, robustness, and generalization ability of the model.

*3) Model Training*

In the next step of our methodology, we conducted model training using a total of 14 algorithms, including 13 machine learning algorithms and 1 deep learning algorithm. These algorithms were applied to the preprocessed dataset that underwent encoding, feature scaling, and train-test splitting. The accuracy of all these algorithms is presented in Table XI Furthermore, this section discusses some of these algorithms in detail.

*a) Gradient Boosting*

Gradient boosting is a machine learning ensemble approach which integrates the predictions of numerous base models to produce a more accurate and resilient final model. It operates by fitting weak learners, often decision trees, to the residuals of previous model predictions iteratively. The residuals are the difference between the actual target values and the predicted values from the prior model, and the next weak learner is

trained to detect patterns in the residuals that the previous models missed.

The algorithm starts with an initial prediction for each data point and then computes the negative gradient of the loss function with respect to these initial predictions. This gradient represents the direction in which the model's predictions need to be adjusted to minimize the loss function. The next weak learner, typically a decision tree, is trained to predict the negative gradient, which is added to the initial predictions to update the overall model. The algorithm continues this process for a specified number of iterations or until a predefined stopping criterion is met. Each iteration of gradient boosting adjusts the predictions of the previous models by adding new predictions from the weak learner, with the learning rate controlling the magnitude of the update. The final prediction of the gradient boosting model is obtained by summing the predictions from all the weak learners, weighted by the learning rate.

Mathematically, the update formula for the gradient boosting algorithm can be represented as follows:

For iteration m = 1 to M:

1. Compute the negative gradient of the loss function with respect to the current predictions:

$$rim = -\partial L(yi, y^i)/\partial y^i \tag{2}$$

2. Train a weak learner, typically a decision tree, to predict the negative gradient:

$$hm(xi) = WeakLearner(X, rim) \tag{3}$$

3. Update the current predictions with the predictions from the weak learner, weighted by the learning rate:

$$y^i = y^i + \alpha hm(xi) \tag{4}$$

4. Repeat steps 1-3 for M iterations or until a predefined stopping criterion is met.

Here,

• $L(yi, y^i)$ is the loss function that measures the error between the actual target values yi and the current predictions $y^i$.

• rim is the negative gradient of the loss function with respect to the current predictions for the i-th data point at iteration m.

• hm(xi) is the prediction from the weak learner (e.g., decision tree) at iteration m for the i-th data point.

• $\alpha$ is the learning rate, a hyperparameter that controls the magnitude of the update at each iteration.

• M is the total number of iterations or boosting rounds.

The gradient boosting algorithm iteratively improves the model predictions by fitting weak learners to the residuals and updating the predictions based on the negative gradient of the loss function. This process continues until a stopping criterion is met, resulting in an ensemble model that can capture complex patterns in the data and achieve high predictive accuracy.

*b) Random Forest*

Random Forest is an ensemble method that combines multiple decision trees to form a robust and accurate predictive model. It uses a combination of decision trees to overcome the limitations of individual trees, such as overfitting and bias. Random Forest uses a bagging technique, where each tree is trained on a randomly sampled subset of the training data with replacement. This helps to reduce the risk of overfitting and improve the generalization performance of the model. Random Forest provides interpretable outputs in the form of decision trees, which can be visualized and easily understood. This allows for model interpretation and explanation, making it suitable for applications where model transparency is important. Random Forest provides a feature importance score, which indicates the relative importance of each feature in the model's decision-making process. This can be used for feature selection, interpretation, and model explanation. The functioning of Random Forest is as follows.

1. Initialize the model with the number of decision trees N and the maximum depth d for each tree.

2. For each decision tree t=1,2,. . . ,N:

a. Randomly select a subset of the training data with replacement, typically called a "bootstrap" sample. Let Dt denote the bootstrap sample.

b. Train a decision tree Tt on the bootstrap sample Dt with a maximum depth of d. The decision tree is trained using a feature subset that is randomly selected at each split. Let Tt(x) denote the prediction of the t-th decision tree for the input data point x.

3. Predict the class label of a new data point xx using the majority vote of the predictions from all the decision trees: Prediction(x)=mode(T1(x),T2(x),. . . ,TN(x)) where the mode function returns the most frequent prediction among all the decision trees.

In this notation, Dt represents the bootstrap sample used to train the t-th decision tree, and Tt(x) represents the prediction of the t-th decision tree for the input data point x.

The Random Forest algorithm uses a collection of decision trees, each trained on a randomly selected subset of the training data with replacement. This introduces randomness into the model and helps to reduce overfitting. The majority vote of the individual tree predictions is used as the final prediction for a new data point, making Random Forest a powerful and robust ensemble learning algorithm for classification tasks. Overall, Random Forest is a powerful and flexible algorithm that offers several useful features beyond its mathematical aspects, making it widely used in various machine learning tasks and applications.

*c) XGBoost*

XGBoost, short for Extreme Gradient Boosting, is a popular gradient boosting algorithm that offers several additional features. XGBoost provides built-in support for L1 (Lasso) and L2 (Ridge) regularization techniques, which help to prevent overfitting and improve model generalization

performance. Regularization can be controlled through hyperparameters, allowing for fine-tuning of the model. XGBoost supports early stopping, where the model training can be automatically stopped if no improvement in model performance is observed on a validation set over a certain number of iterations. This helps to prevent overfitting and reduces training time. XGBoost provides a feature importance score, which indicates the relative importance of each feature in the model's decision-making process. This can be used for feature selection, interpretation, and model explanation. Mathematically, the XGBoost algorithm can be described as follows.

1. Given a training dataset with input features denoted as X and corresponding target labels denoted as y.

2. Initialize the model with a constant prediction $\hat{y}$ for all instances in the training set, usually set to the mean of the target labels.

3. For t=1,2,. . . ,T, where T is the number of boosting iterations:

a. Compute the negative gradient of the loss function with respect to the predicted values, denoted as

$$gt = -\frac{\partial loss(y, \hat{y})}{\partial \hat{y}} \tag{5}$$

where $loss(y, \hat{y})$ is the chosen loss function.

b. Fit a weak learner (e.g., decision tree) to the negative gradient gt with respect to the input features X, and obtain a prediction denoted as ht(X).

c. Update the model by adding the prediction ht(X) scaled by a learning rate $\eta$, denoted as $\hat{y} \leftarrow \hat{y} + \eta ht(X)$.

4. Once the desired number of boosting iterations T is reached, the final model $\hat{y}$ is obtained.

The XGBoost algorithm uses a gradient boosting framework to sequentially train decision trees based on the gradients and Hessians of the loss function, and combines their predictions with a learning rate to update the predicted values. The sigmoid function is used to convert the final predicted values into class probabilities for classification tasks. XGBoost is known for its accuracy, efficiency, and scalability, making it a popular choice for many machine learning tasks. Overall, XGBoost combines the power of gradient boosting with additional features for efficient and effective model training, making it a widely used and popular algorithm in machine learning competitions and real-world applications.

### C. IRIS Species Classification and Explanation

#### 1) Dataset Description

We chose the popular "IRIS Dataset" [50] as a part of our experimental evaluation to assess the performance and effectiveness of our XOPSIS algorithm. This dataset offers a suitable testing ground for our algorithm's capabilities in accurately classifying iris flowers into their respective species. By applying XOPSIS to the IRIS dataset, we aim to validate the algorithm's performance and demonstrate its potential for generating comprehensive explanations and insights in a well-established and widely used dataset.

The IRIS dataset is a well-known benchmark dataset in the field of machine learning and consists of samples from three different species of iris flowers: Iris-setosa, Iris-versicolor, and Iris-virginica. It serves as an excellent example for classification tasks.

The dataset comprises four features, namely sepal length, sepal width, petal length, and petal width. These features provide measurements in centimeters and represent different aspects of the iris flowers' morphology. The sepal length denotes the length of the iris flower's sepal, while the sepal width represents its width. Similarly, the petal length and petal width indicate the length and width of the iris flower's petals, respectively.

The target variable in the IRIS dataset is the "Species" column, which classifies each sample into one of the three iris species mentioned earlier: Iris-setosa, Iris-versicolor, and Iris-virginica. This categorical variable enables the use of supervised learning techniques for classification tasks.

#### 2) Dataset Preprocessing

To ensure the reliability and consistency of the data, a series of preprocessing steps were applied to the IRIS dataset. The following subsection describes each step in detail.

##### a) Checking for Null Values

Prior to analysis, the dataset was examined for any missing or null values. Fortunately, no null values were found, indicating that the dataset is complete and no imputation or data filling was required. This ensures the integrity and reliability of the data used for the subsequent analysis.

##### b) Encoding of Categorical Variable

Since the target variable "Species" contains categorical values (Iris-setosa, Iris-versicolor, Iris-virginica), an ordinal encoder was employed to convert these categorical labels into numerical representations. The ordinal encoding method was chosen to maintain the order of the categories, with Iris-setosa, Iris-versicolor, and Iris-virginica encoded as 0, 1, and 2, respectively. This transformation enables the utilization of categorical variables as input to machine learning algorithms, facilitating the classification task.

##### c) Feature Scaling

To bring the numerical features of the dataset onto a similar scale and avoid any dominance of one feature over others during model training, standard scaling (z-score normalization) was applied to the following features: sepal length, sepal width, petal length, petal width. Standard scaling transforms the features to have zero mean and unit variance, enhancing the comparability and training process of the model. This normalization step ensures that all features contribute equally and prevents bias towards features with larger values.

##### d) Train-Test Split

To evaluate the performance of the XOPSIS algorithm on the IRIS dataset, the dataset was divided into training and testing sets using an 80:20 ratio. The training set, comprising 80% of the dataset, was used to train the machine learning model, while the remaining 20% served as an independent test set for evaluating the model's performance. This partitioning enables the assessment of the model's generalization capacity and its ability to accurately classify iris flowers into their respective species.

*e) K-fold Cross-Validation*

In order to obtain a more reliable estimate of the model's performance and minimize the impact of variable train-test splits, k-fold cross-validation was employed. The dataset was divided into k equally sized folds, where each fold was used as the test set once, while the remaining k-1 folds were used for training. This process was repeated k times, with performance metrics such as accuracy, precision, recall, and F1 score calculated for each fold. The average performance across all folds was reported as the final evaluation of the XOPSIS algorithm on the IRIS dataset. This technique ensures a robust evaluation of the model's performance and reduces the variability associated with different train-test splits.

*f) Hyperparameter Tuning*

Hyperparameters play a crucial role in determining the performance of a machine learning model. Grid Search CV, a popular hyperparameter tuning technique, was employed to identify the optimal set of hyperparameter values for the XOPSIS algorithm. A predefined grid of hyperparameter values was specified, and the model was trained and evaluated using k-fold cross-validation for each combination of hyperparameters. The performance metrics were recorded, and the combination of hyperparameter values yielding the best performance was selected as the optimal set. This rigorous optimization process enhances the accuracy, robustness, and generalization ability of the XOPSIS algorithm on the IRIS dataset.

By following these dataset preprocessing steps, we ensure the reliability and suitability of the IRIS dataset for evaluating the performance and effectiveness of the XOPSIS algorithm.

*3) Model Training*

In the next step of our methodology, we conducted model training using three different algorithms: Gradient Boosting, Random Forest, and XGBoost. These algorithms were applied to the preprocessed dataset that underwent encoding, feature scaling, and train-test splitting. The accuracy of all these algorithms is presented in Table XIII.

It is important to note that while we previously provided detailed descriptions of these algorithms in an earlier section, this subsection focuses on mentioning the application of these algorithms without repeating the detailed model descriptions. Instead, we refer readers back to the earlier section for comprehensive explanations of each algorithm.

The results and performance metrics of these three algorithms applied to the Iris dataset are presented in Table XIII for reference.

### *D. Breast Cancer Prediction and Explanation*

*1) Dataset Description*

The dataset [51] used in this study is a well-known benchmark dataset commonly used for training and evaluating machine learning algorithms. It comprises biopsy features of 569 breast masses classified as malignant (cancer) or benign (not cancer).

The features in the dataset were extracted computationally from digital images of fine needle aspirate biopsy slides. These features correspond to various properties of cell nuclei, including size, shape, and regularity. Specifically, the dataset provides the mean, standard error, and worst values of 10 nuclear parameters, resulting in a total of 30 features.

Each feature in the dataset is associated with a specific description:

- radius_mean: Mean radius of the tumor cells
- texture_mean: Mean texture of the tumor cells
- perimeter_mean: Mean perimeter of the tumor cells
- area_mean: Mean area of the tumor cells
- smoothness_mean: Mean smoothness of the tumor cells
- compactness_mean: Mean compactness of the tumor cells
- concavity_mean: Mean concavity of the tumor cells
- concave_points_mean: Mean number of concave portions of the contour of the tumor cells
- symmetry_mean: Mean symmetry of the tumor cells
- fractal_dimension_mean: Mean "coastline approximation" of the tumor cells
- radius_se: Standard error of the radius of the tumor cells
- texture_se: Standard error of the texture of the tumor cells
- perimeter_se: Standard error of the perimeter of the tumor cells
- area_se: Standard error of the area of the tumor cells
- smoothness_se: Standard error of the smoothness of the tumor cells
- compactness_se: Standard error of the compactness of the tumor cells
- concavity_se: Standard error of the concavity of the tumor cells
- concave_points_se: Standard error of the number of concave portions of the contour of the tumor cells
- symmetry_se: Standard error of the symmetry of the tumor cells
- fractal_dimension_se: Standard error of the "coastline approximation" of the tumor cells
- radius_worst: Worst (largest) radius of the tumor cells
- texture_worst: Worst (most severe) texture of the tumor cells
- perimeter_worst: Worst (largest) perimeter of the tumor cells
- area_worst: Worst (largest) area of the tumor cells
- smoothness_worst: Worst (most severe) smoothness of the tumor cells
- compactness_worst: Worst (most severe) compactness of the tumor cells
- concavity_worst: Worst (most severe) concavity of the tumor cells
- concave_points_worst: Worst (most severe) number of concave portions of the contour of the tumor cells
- symmetry_worst: Worst (most severe) symmetry of the tumor cells
- fractal_dimension_worst: Worst (most severe) "coastline approximation" of the tumor cells

The target variable in the dataset indicates whether a mass is malignant ("M") or Benign ("B"). The predictors consist of a matrix with the mean, standard error, and worst values of 10 nuclear measurements, resulting in a total of 30 features per biopsy.

These measurements include factors such as nucleus radius,

texture, perimeter, area, smoothness, compactness, concavity, number of concave portions, symmetry, and fractal dimension.

Overall, this dataset provides a comprehensive set of features derived from biopsy images to facilitate the development and evaluation of machine learning models for breast cancer classification.

*2) Dataset Preprocessing*

To ensure the reliability and consistency of the data, a series of preprocessing steps were applied to the breast cancer Wisconsin dataset. The following subsection describes each step in detail.

*a) Checking for Null Values*

Prior to analysis, the dataset was examined for any missing or null values. The column "Unnamed: 32" was found to have null values, and therefore it was removed from the dataset. Additionally, the column "id" was also removed from the dataset, although it did not contain any null values. By removing these columns, we ensure that the dataset is clean and free from missing values, maintaining the integrity and reliability of the data for subsequent analysis.

*b) Encoding of Categorical Variable*

Since the target variable "diagnosis" contains categorical values ("M" for malignant and "B" for benign), an ordinal encoder was employed to convert these categorical labels into numerical representations. In the ordinal encoding process, the label "M" was encoded as 0, indicating malignant, and the label "B" was encoded as 1, indicating benign. This transformation allows the utilization of the categorical variable as input to machine learning algorithms, facilitating the classification task while preserving the inherent order of the categories.

*c) Feature Scaling*

To bring the numerical features of the dataset onto a similar scale and avoid any dominance of one feature over others during model training, standard scaling (z-score normalization) was applied to the 30 features. Standard scaling transforms the features to have a zero mean and unit variance, enhancing the comparability and training process of the model. This normalization step ensures that all features contribute equally and prevents bias towards features with larger values.

*d) Train-Test Split*

To evaluate the performance of the XOPSIS algorithm on the breast cancer Wisconsin dataset, the dataset was divided into training and testing sets using an 80:20 ratio. The training set, comprising 80% of the dataset, was used to train the machine learning model, while the remaining 20% served as an independent test set for evaluating the model's performance. This partitioning enables the assessment of the model's generalization capacity and its ability to accurately classify breast masses as malignant or benign.

*e) K-fold Cross-Validation*

In order to obtain a more reliable estimate of the model's performance and minimize the impact of variable train-test splits, k-fold cross-validation was employed. The dataset was divided into k equally sized folds, where each fold was used as the test set once, while the remaining k-1 folds were used for training. This process was repeated k times, with performance metrics such as accuracy, precision, recall, and F1 score calculated for each fold. The average performance across all folds was reported as the final evaluation of the XOPSIS algorithm on the breast cancer Wisconsin dataset. This technique ensures a robust evaluation of the model's performance and reduces the variability associated with different train-test splits.

*f) Hyperparameter Tuning*

Hyperparameters play a crucial role in determining the performance of a machine learning model. Grid Search CV, a popular hyperparameter tuning technique, was employed to identify the optimal set of hyperparameter values for the XOPSIS algorithm. A predefined grid of hyperparameter values was specified, and the model was trained and evaluated using k-fold cross-validation for each combination of hyperparameters. The performance metrics were recorded, and the combination of hyperparameter values yielding the best performance was selected as the optimal set. This rigorous optimization process enhances the accuracy, robustness, and generalization ability of the XOPSIS algorithm on the breast cancer Wisconsin dataset.

By following these dataset preprocessing steps, including the removal of the "Unnamed: 32" column and ordinal encoding of the "diagnosis" variable, we ensure the reliability and suitability of the breast cancer Wisconsin dataset for evaluating the performance and effectiveness of the XOPSIS algorithm.

*3) Model Training*

In the next step of our methodology, model training was performed using three different algorithms: Gradient Boosting, Random Forest, and XGBoost. These algorithms were applied to the preprocessed breast cancer Wisconsin dataset, which had undergone encoding, feature scaling, and train-test splitting. The accuracy of each algorithm is presented in Table XV.

It is important to note that while detailed descriptions of these algorithms were provided earlier in the paper, this subsection focuses on highlighting their application without repeating the comprehensive model explanations. Readers are referred back to the earlier section for a thorough understanding of each algorithm.

The results and performance metrics of these three algorithms applied to the Breast Cancer Wisconsin Dataset are presented in Table XV for easy reference.

### *E. Car Acceptability Prediction and Explanation*

*1) Data Collection*

The Car Acceptability dataset [52] used in this research was collected from Kaggle. It is based on a simple hierarchical decision model originally developed for the demonstration of the DEX expert system for decision making (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990). The Car Acceptability dataset used in this research consists of 1729 unique rows and contains a target variable, Car_Acceptability, along with six feature variables. The goal of the analysis is to predict the car acceptability based on the values of these features. The dataset evaluates cars according to the following features:

- Buying_Price: Categorical Data [vhigh, high, med, low]
- Maintenance_Price: Categorical Data [vhigh, high, med, low]

- No_of_Doors: Categorical Data [2, 3, 4, 5more]
- Person_Capacity: Categorical Data [2, 4, more]
- Size_of_Luggage: Categorical Data [small, med, big]
- Safety: Categorical Data [low, med, high]

The Car_Acceptability variable serves as the target variable and is also categorical, with the following values: [unacc, acc, good, vgood]. The objective is to develop a model using XOPSIS that accurately predicts the car acceptability based on the given features.

The Car Acceptability Classification Database is derived from this decision model and is available under the CC-BY-NC-SA 4.0 license for non-commercial usage. The dataset is provided by the UCI Machine Learning Repository, a widely recognized and reliable source for machine learning datasets.

This dataset offers a valuable opportunity to evaluate the effectiveness of XOPSIS in the classification of car acceptability based on various features. By applying XOPSIS to this dataset, we aim to gain insights into the decision-making process of the model and generate comprehensive explanations for the predictions. The analysis of this dataset will contribute to the understanding of XOPSIS's performance in a new domain and its potential for providing interpretable insights in the field of car acceptability classification.

*2) Dataset Preprocessing*

To ensure the reliability and consistency of the Car Acceptability dataset, several preprocessing steps were applied. The following subsection outlines each step in detail.

*a) Checking for Null Values*

The dataset was first examined for any missing or null values. No missing values were found, ensuring that the dataset is complete and does not require imputation or further handling of missing data.

*b) Encoding of Categorical Variables*

As the dataset contains categorical variables, including the target variable "Car_Acceptability" and the feature variables, an encoding process was performed to transform these categorical labels into numerical representations. Specifically, an ordinal encoder was used to assign numerical values to each category. The encoding mappings are as follows:

- Buying_Price: [vhigh = 3, high = 2, med= 1, low = 0]
- Maintenance_Price: [vhigh = 3, high = 2, med= 1, low = 0]
- No_of_Doors: [2 = 0, 3 = 1, 4 = 2, 5more = 3]
- Person_Capacity: [2 = 0, 4 = 1, more = 2]
- Size_of_Luggage: [small = 0, med = 1, big = 2]
- Safety: [low = 0, med = 1, high = 2]
- Car_Acceptability (Target): [unacc=0, acc=1, good=2, vgood=3]

This encoding process enables the utilization of the categorical variables as input to machine learning algorithms, allowing for effective classification tasks while preserving the inherent order of the categories.

*c) Feature Scaling*

To ensure that the numerical features are on a similar scale and avoid dominance of any particular feature during model training, feature scaling was performed. Specifically, standard scaling (z-score normalization) was applied to the feature variables. This transformation centers the data by subtracting the mean and scales it by dividing by the standard deviation, resulting in features with a mean of 0 and a standard deviation of 1. This normalization step enhances the comparability and training process of the model, ensuring that each feature contributes equally to the model's predictions.

*d) Train-Test Split*

To evaluate the performance of the XOPSIS algorithm on the Car Acceptability dataset, the dataset was split into training and testing sets using an 80:20 ratio. The training set, comprising 80% of the dataset, was used to train the machine learning model, while the remaining 20% served as an independent test set for evaluating the model's performance. This division enables the assessment of the model's generalization capacity and its ability to accurately classify car acceptability.

*e) K-fold Cross-Validation*

To obtain a more reliable estimate of the model's performance and reduce the impact of variable train-test splits, 10-fold cross-validation was employed. The Car Acceptability dataset was divided into 10 equally sized folds. The model was trained and evaluated using each fold as the test set once, while the remaining nine folds were used for training. This process was repeated 10 times, with performance metrics calculated for each fold using the k-fold cross-validation method.

*f) Hyperparameter Tuning*

Hyperparameters play a crucial role in determining the performance of a machine learning model. To optimize the XOPSIS algorithm's performance on the Car Acceptability dataset, hyperparameter tuning was performed using Grid Search CV. A predefined grid of hyperparameter values was specified, and the model was trained and evaluated using k-fold cross-validation for each combination of hyperparameters. The performance metrics were recorded, and the combination of hyperparameter values yielding the best performance was selected as the optimal set. This rigorous optimization process enhances the accuracy, robustness, and generalization ability of the XOPSIS algorithm on the Car Acceptability dataset.

By following these dataset preprocessing steps, including the encoding of categorical variables, feature scaling, and k-fold cross-validation, we ensure the reliability and suitability of the Car Acceptability dataset for evaluating the performance and effectiveness of the XOPSIS algorithm.

*3) Model Training*

In the model training phase, we employed three different algorithms: Gradient Boosting, Random Forest, and XGBoost. These algorithms were trained on the preprocessed Car Acceptability dataset, which had undergone preprocessing steps including encoding of categorical variables, feature scaling, and train-test splitting.

Each algorithm was trained on the training set of the dataset, using the labeled data to learn the underlying patterns and relationships between the features and the target variable. The goal of model training is to optimize the algorithm's parameters and adjust the model's internal mechanisms to make accurate predictions on unseen data.

During the training process, the algorithms iteratively adjusted their internal parameters based on the provided features and their corresponding target labels. This iterative optimiza-

tion process aimed to minimize the discrepancy between the predicted outputs and the true labels in the training set.

By training multiple algorithms on the preprocessed Car Acceptability dataset, we aimed to evaluate their performance and effectiveness in capturing the underlying patterns and making accurate predictions. The trained models will be further evaluated and compared using various performance metrics in the subsequent analysis.

## V. Result Analysis

This section of the study encompasses three key subsections such as Accuracy, Feature Importance Plot, and Interpretability through XAI Methods. In the Accuracy subsection, we present and elaborate on the accuracy plots, providing comprehensive details on the performance of the developed models. Moving on to the Feature Importance Plot subsection, we showcase three different plots, namely the feature importance ranking using SHAP summary plot with Random Forest, feature importance ranking using SHAP summary plot with XGBoost, and feature importance ranking using TOPSIS methods with Gradient Boosting. Lastly, in the Interpretability through XAI Methods subsection, we demonstrate the explanations generated by XOPSIS, LIME, and SHAP for specific instances, offering multiple insightful cases to illustrate the interpretability of our proposed approach and compare it with existing methods.

### A. Experimental Setup

The experiments were conducted using the Google Colab platform, which provided a robust and scalable environment for coding and analysis. The computational resources offered by Google Colab were leveraged to efficiently execute the code and algorithms. The experiments were performed on a computer with the following specifications:

- Processor: Intel Core i5-8265U CPU @ 1.60GHz, 1800Mhz, 4 Core(s), 8 Logical Processor(s)
- RAM: 8 GB DDR4
- GPU: Intel(R) UHD Graphics 620
- Storage: ST1000LM035-1RK172 (1 TB)
- Operating System: Windows (Microsoft Windows 11 Pro)

Python, a widely used programming language in machine learning research, was employed for coding. The primary libraries utilized include scikit-learn for machine learning implementations, matplotlib for data visualization, and numpy for numerical computations. The experimentation process involved the following key steps:

1. Data Preprocessing: The datasets were loaded, and preprocessing steps such as feature scaling and splitting into training and testing sets were performed.

2. Model Training: The machine learning models, including Gradient Boosting and XGBoost, were trained on the training data using optimized hyperparameters.

3. Prediction and Evaluation: The trained models were used to make predictions on the testing data. Accuracy, confusion matrix, sensitivity, specificity, and ROC curves were computed for model evaluation.

4. Explainability Techniques: LIME, SHAP, and TOPSIS-based XAI methods (XOPSIS) were applied to interpret the model predictions. Explanations were generated for specific instances, highlighting feature importance and decision rationales.

5. Analysis and Interpretation: The results obtained from different models and explainability techniques were analyzed and compared to gain insights into the models' performance and interpretability.

The experimental setup ensured consistency and reproducibility of the results, allowing for an in-depth analysis of the models' behavior and the effectiveness of the XAI techniques.

### B. Result Analysis on Maternal Health Dataset

#### 1) Accuracy

As demonstrated in Table XI, the train-test split technique outperforms the k-fold cross-validation method in terms of accuracy for all 14 algorithms. The resulting accuracy scores are depicted in a bar plot in Figure 2.

| Serial No | Algorithms | Accuracy | |
|---|---|---|---|
| | | Train-Test Split | K-Fold Cross Validation |
| 01 | KNN | 89.163% | 85.389% |
| 02 | Random Forest | 90.148% | 84.402% |
| 03 | Gradient Boosting | 90.640% | 84.990% |
| 04 | XGBoost | 89.655% | 81.345% |
| 05 | CatBoost | 87.192% | 81.839% |
| 06 | Bagging Classifier | 89.163% | 82.434% |
| 07 | AdaBoost | 75.862% | 62.713% |
| 08 | Decision Tree | 88.670% | 83.023% |
| 09 | Extra Tree Classifier | 89.655% | 83.032% |
| 10 | Logistic Regression | 72.906% | 64.793% |
| 11 | Support Vector Classifier | 84.729% | 82.338% |
| 12 | Gaussian Naive Bayes | 71.921% | 64.577% |
| 13 | Voting Classifier | 89.655% | 84.011% |
| 14 | Multi Layer Perceptron | 83.744% | 77.708% |

TABLE XI: Accuracy of 14 Algorithms

From Table XI, it is evident that the accuracy varies significantly across different algorithms. The top-performing algorithms based on the train-test split technique are Random Forest, Gradient Boosting, and XGBOOST, while the least accurate algorithms are Logistic Regression, Gaussian Naive Bayes, and AdaBoost. Furthermore, the k-fold cross-validation technique generally produces lower accuracy scores compared to the train-test split technique, likely due to the use of multiple subsets of data for training and testing, which can result in a more generalized model but with slightly lower accuracy.

The bar plot in Figure 2 visually represents the accuracy scores for each algorithm using the train-test split technique, revealing that the top-performing algorithms have accuracy scores above 90%, while the least accurate algorithms have scores below 75%. Notably, the Gradient Boosting algorithm

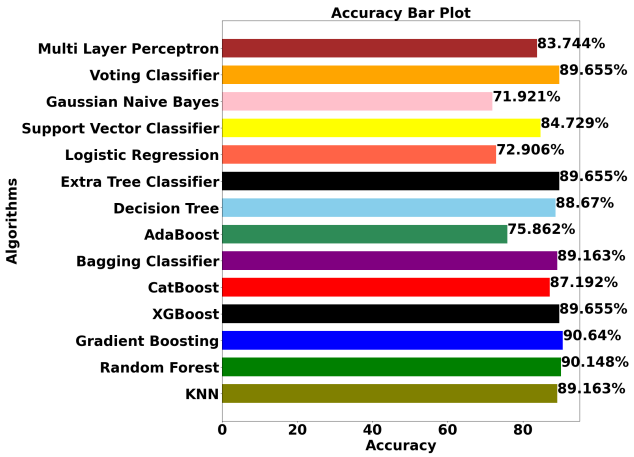



Fig. 2: Train-Test Split Method Accuracy of 14 Algorithms

achieves the highest accuracy score of 90.64% among all the tested algorithms.

*2) Confusion Matrix*

The confusion matrix presented in Figure 3 is a 3x3 matrix that represents the classification results for three risk levels: low risk, mid risk, and high risk. The matrix provides a comprehensive overview of the performance of the model by displaying the true labels on the left side and the predicted labels at the bottom.

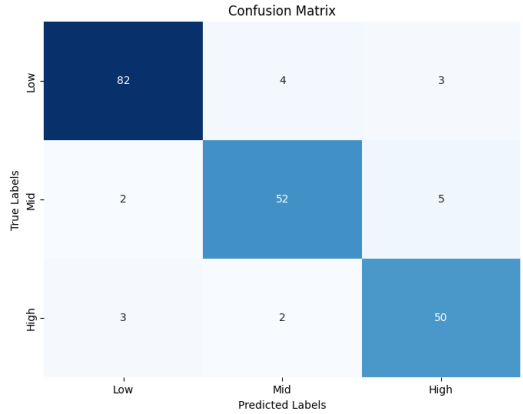



Fig. 3: Confusion Matrix of Gradient Boosting Model

In the first row of the matrix, it is observed that out of the instances labeled as low risk, 82 were correctly predicted as low risk, while 4 instances were misclassified as mid risk and 3 instances as high risk. Moving to the second row, which represents instances labeled as mid risk, the model correctly predicted 52 instances as mid risk, misclassified 2 instances as low risk, and 5 instances as high risk. Lastly, in the third row representing high-risk instances, the model correctly classified 50 instances as high risk, misclassified 3 instances as low risk, and 2 instances as mid risk.

This confusion matrix provides a detailed breakdown of the classification results, enabling a deeper analysis of the model's performance for different risk levels. By examining the matrix, it becomes evident that the model exhibits a higher accuracy in predicting low-risk and mid-risk instances compared to high-risk instances. The presented confusion matrix serves as a valuable tool for evaluating the performance of the classification model and provides insights into the effectiveness of the risk level prediction.

*3) Sensitivity, Specificity, Precision and F1-Score*

We focus on evaluating the performance of the Gradient Boosting model, which we selected as our final model based on its highest accuracy score obtained earlier. We assess its effectiveness by analyzing several important performance metrics, including precision, recall, F1-score, sensitivity, and specificity. These metrics provide valuable insights into the ability of the model to accurately classify instances and its performance across different classes. Table XII shows the values of performance evaluation metrics.

| | Sensitivity | Specificity | Precision | F1-Score |
|---|---|---|---|---|
| Low Risk | 0.92 | 0.96 | 0.94 | 0.93 |
| Mid Risk | 0.88 | 0.96 | 0.90 | 0.89 |
| High Risk | 0.91 | 0.95 | 0.86 | 0.88 |

TABLE XII: Performance Evaluation Metrics

The plot showed in Figure 4 illustrates the performance metrics for a multiclass classification task across three classes: Low, Mid, and High Risk. The x-axis represents the classes, with Low on the left side, Mid in the middle, and High on the right side. The y-axis represents the scores of the performance metrics. The plot includes four performance metrics: Sensitivity, Specificity, Precision, and F1-Score. These metrics provide valuable insights into the model's performance in correctly classifying instances and its overall effectiveness across different classes.

Sensitivity, also known as the true positive rate, measures the proportion of correctly classified positive instances for each class. A higher sensitivity indicates that the model is effective at identifying instances belonging to a specific class.

Specificity, also known as the true negative rate, measures the proportion of correctly classified negative instances for each class. A higher specificity indicates that the model is effective at correctly classifying instances not belonging to a specific class.

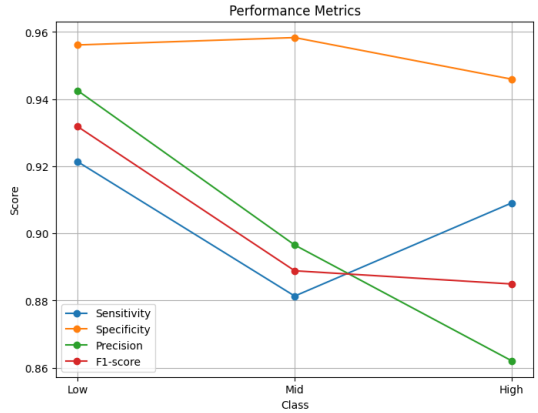



Fig. 4: Comparison of Sensitivity, Specificity, Precision, and F1-Score Across Three Risk Levels

Precision measures the proportion of correctly classified instances for each class among all instances predicted to belong to that class. A higher precision indicates that the model has a low rate of falsely classifying instances into a specific class.

The F1-Score is a harmonic mean of precision and recall (which is equivalent to sensitivity in this case). It provides a balanced measure of the model's performance by considering both precision and recall. A higher F1-Score indicates a better trade-off between precision and recall for each class.

By plotting these performance metrics together, we can observe the relative performance of the model across the Low, Mid, and High classes. We can see that the model performs consistently well across the classes, with generally high scores for Sensitivity, Specificity, Precision, and F1-Score. However, there are some variations in the scores across the classes, indicating potential differences in the model's performance in correctly classifying instances among the different classes.

*4) Interpretability Through XAI Methods*

*a) High Risk Explanations*

**Case 01**

From Figure 5, it appears to be a summary of a prediction result for a specific data point. It starts by showing the actual risk level and the predicted risk level, both of which are 2.0 indicating a high risk level.

```
------------------ TOPSIS Explanation ------------------
Data point index: 954
------------------ Data Point Features ------------------
Age             25.00
SystolicBP     140.00
DiastolicBP    100.00
BS               7.01
BodyTemp        98.00
HeartRate       80.00
Name: 954, dtype: float64
Actual risk level: 2.0
Predicted Risk Level:  2.0
Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • DiastolicBP
        • SystolicBP
- The features that contribute negatively to the prediction are:
        • Age
        • BS
        • HeartRate
        • BodyTemp
```

Fig. 5: Explanation of high risk level by XOPSIS (Case 01)

Next, it mentions that the explanation is based on topsis score, which is calculated by comparing the similarity of feature values between the given data point and similar instances. The explanation then proceeds to mention which features contribute positively and negatively to the prediction. According to the explanation, the features that have a positive impact on the prediction are Diastolic BP and Systolic BP. On the other hand, the features that have a negative impact on the prediction are Age, BS, HeartRate and Body Temperature.

Overall, the explanation provides insight into how different features are influencing the prediction of high risk level for the given data point, with some features contributing positively and others negatively to the prediction.

The plot showed in Figure 6, represents normalized feature contributions on the x-axis and feature names on the y-axis.

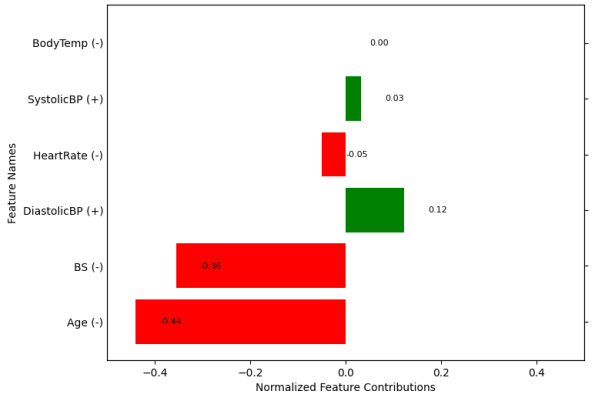



Fig. 6: Normalized Feature Contibutions of high risk level by XOPSIS (Case 01)

Feature contributions are represented as numerical values, with positive contributions in green and negative contributions in red. The feature contributions are as follows: Age (-0.44), BS (-0.36), BodyTemp (0.0), HeartRate (-0.05), SystolicBP (0.03), and DiastolicBP (0.12).

Figure 7 illustrates that the probabilities predicted by the model for different risk categories, such as "low risk," "mid risk," and "high risk." In this case, the model predicts a probability of 1.0 for "high risk," and 0.0 for both "low risk" and "mid risk."

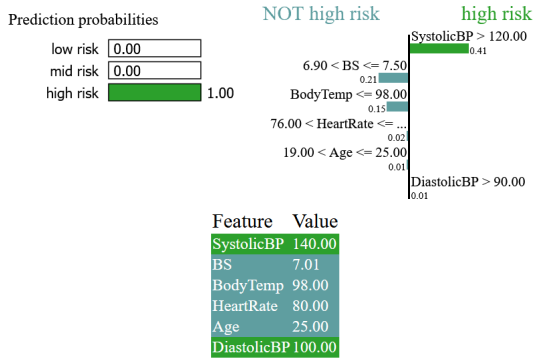



Fig. 7: Explanation of high risk level by LIME (Case 01)

The plot generated by LIME visually presents the features that contribute to the predicted risk category. It is divided into two sections, with "high risk" on the right side and "not high risk" on the left side. On the right side of the plot, LIME identifies the features that positively contribute to the prediction of "high risk." These features are SystolicBP $>$ 120 with a weight of 0.41 and DiastolicBP $>$ 90 with a weight of 0.01. The weights indicate the relative importance of these features in determining the "high risk" prediction. On the left side of the plot, LIME identifies the features that negatively contribute to the prediction of "not high risk" These features are 6.90 $<$ BS $<=$ 7.50 with a weight of 0.21, BodyTemp $<=$ 90 with a weight of 0.15, 76 $<$ HeartRate with a weight of 0.02 and 19 $<$ Age $<=$ 25 with a weight of 0.01. Similar to the features for "high risk", these features are shown with

their corresponding weights.

The LIME output also displays the actual values of the features for the specific data point that was explained. These feature values provide context and reference to better understand how the model arrived at its prediction.

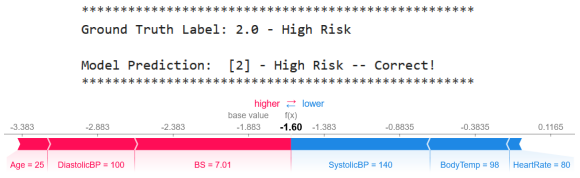



Fig. 8: Explanation of high risk level by SHAP (Case 01)

The plot in Figure 8, starts by showing the Ground Truth Label, which is the actual label or class of the data point, and the Model Prediction, which is the predicted label or class by the machine learning model. In your case, it shows "Ground Truth Label: 2.0" or "High Risk" and "Model Prediction: 2.0 - High Risk - Correct!" indicating that the data point belongs to the "High Risk" class and the model has made a correct prediction.

The red colored features in the plot, such as Age=25, DiastolicBP=100, and BS=7.01, are driving up the predicted probability of belonging to the "high risk" class. These features have a positive impact on the prediction, meaning that higher values of these features are associated with a higher predicted probability of high risk. The blue colored features in the plot, such as SystolicBP=140, BodyTemp=98, and HeartRate=80, are driving down the predicted probability of belonging to the "high risk" class. These features have a negative impact on the prediction, meaning that higher values of these features are associated with a lower predicted probability of high risk.

The "higher$-$ $><$ $-$lower" notation indicates the direction of the impact of the features on the prediction. "Higher" in red color indicates that higher values of the red features are associated with a higher predicted probability of high risk, while "lower" in blue color indicates that lower values of the blue features are associated with a lower predicted probability of high risk. In this case, higher values of the features associated with red color are driving up the predicted probability of the "high risk" class, while blue color are driving down the predicted probability of the "high risk" class. The value of -1.60 that is displayed in bold text in the middle of the red and blue colored features represents the contribution of the combined set of features associated with the red and blue data points on the predicted probability of belonging to the "high risk" class. It is the aggregated impact of all the features considered together, taking into account their respective SHAP values.

Based on the explanations provided by LIME, SHAP, and XOPSIS for the same data point, it can be concluded that features such as Diastolic BP and Systolic BP contribute positively to the prediction of high risk, indicating that higher values of these features are associated with a higher predicted probability of high risk. Conversely, features such as Age, BS, HeartRate and Body Temperature contribute negatively to the prediction of high risk, meaning that higher values of these features are associated with a lower predicted probability of high risk. Notably, LIME and XOPSIS show consistent results in terms of the direction of feature contributions, where both methods highlight the same features as positive or negative contributors to the prediction. SHAP explanation provides specific feature values that drive up or down the predicted probability of high risk. Moreover, XOPSIS explanation emphasizes the similarity of feature values between the given data point and similar instances in the calculation of the TOPSIS score. Overall, these explanations offer valuable insights into how different features impact the prediction of high risk and can aid in a better understanding of the behavior of the model.

*b) Mid Risk Explanations*

**Case 02**

The features of the given data point are displayed in Figure 9. The actual and predicted risk level of the data point is shown, with a value of 1.0. This indicates that the model has predicted a "Mid Risk" level for the given data point. The features that positively contribute to the prediction are HeartRate and BodyTemp, while the features that negatively contribute to the prediction are SystolicBP, DiastolicBP, Age, and BS.

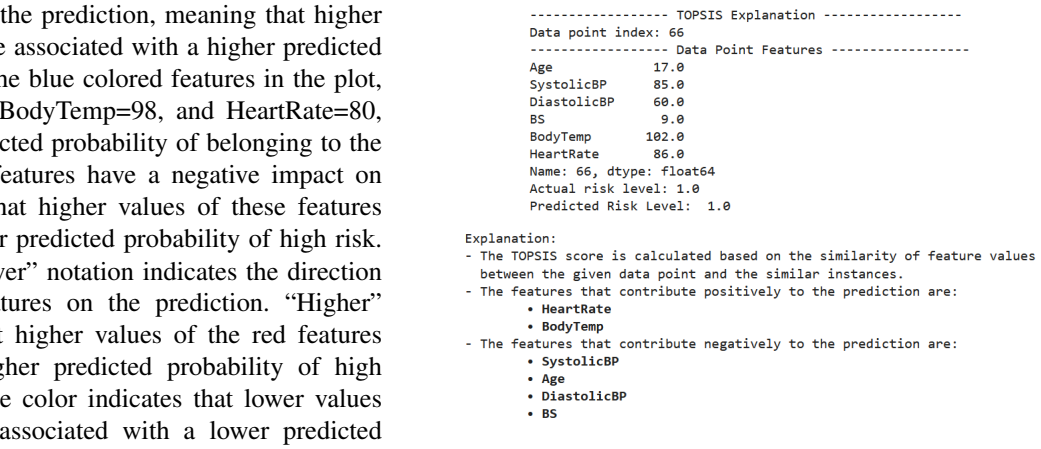



Fig. 9: Explanation of mid risk level by XOPSIS (Case 02)

A plot is shown in Figure 10, with normalized feature contributions. Positive contributions are displayed in green color, while negative contributions are displayed in red color. The normalized contributions for each feature, displayed in parentheses next to the feature name, are as follows: BodyTemp (0.02), BS (-0.04), HeartRate (0.07), DiastolicBP (-0.23), Age (-0.24), and SystolicBP (-0.41). These values indicate the strength and directionality of the contributions of each feature towards the predicted "Mid Risk" level for the given data point. Positive contributions suggest that higher values of those features positively contribute to the "Mid Risk" prediction, while negative contributions suggest that lower values of those features negatively contribute to the "Mid Risk" prediction.

In Figure 11, the initial section displays the predicted probabilities for each risk category. In this case, the predicted

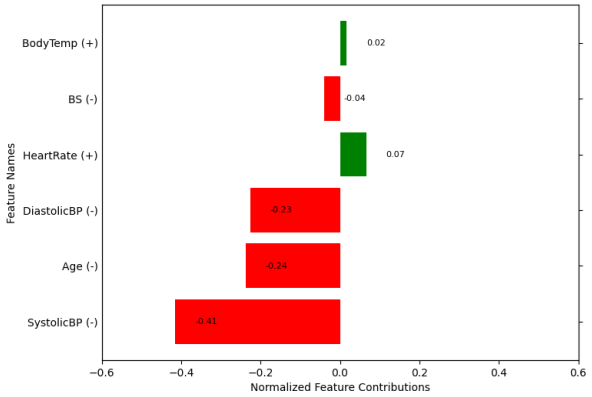



Fig. 10: Normalized Feature Contibutions of mid risk level by XOPSIS (Case 02)

probability for “Mid Risk” is 1.0, while the predicted probabilities for “Low Risk” and “High Risk” are both 0.0. This suggests that the model’s prediction for this data point is a “Mid Risk” classification with high confidence.

The plot is divided into two sections, with “Mid Risk” on the right side and “Not Mid Risk” on the left side. Each section represents the features that are contributing to the respective classification. On the right side of the plot, the features that are positively contributing to the prediction of “Mid Risk” are displayed. These features include “BodyTemp” greater than 98.0 with a contribution weight of 0.11, and “HeartRate” greater than 80.0 with a contribution weight of 0.01. These higher values of “BodyTemp” and “HeartRate” are influencing the model to predict “Mid Risk” for this data point. On the left side of the plot, the features that are negatively contributing to the prediction of “Mid Risk” are displayed. These features include “BS greater than 8.0 with a contribution weight of 0.15, “Age” less than or equal to 19.0 with a contribution weight of 0.11, “SystolicBP” less than or equal to 100 with a contribution weight of 0.05, and “DiastolicBP” less than or equal to 65 with a contribution weight of 0.01. These lower values of “BS”, “Age”, “SystolicBP”, and “DiastolicBP” are influencing the model to predict “Not Mid Risk” for this data point.

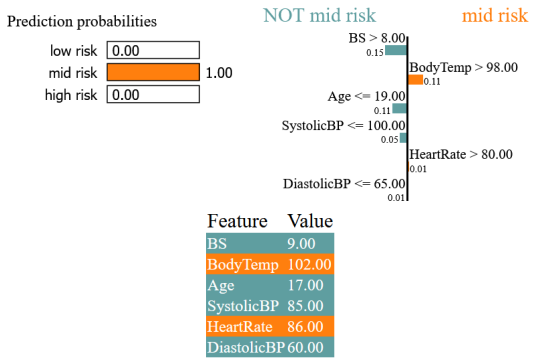



Fig. 11: Explanation of mid risk level by LIME (Case 02)

Finally, the LIME output also provides the feature values for the specific data point being explained. These feature values are likely the values of the input features of the data point for which the explanation is being generated.

Figure 12 displays the SHAP force plot which provides an explanation of the prediction for the data point being analyzed, which is labeled as “Mid Risk” with a correct model prediction. Features such as “BodyTemp=102”, “DiastolicBP=60”, and “HeartRate=86” are shown with red arrows pointing to the right, indicating that higher values of these features have a positive contribution towards the prediction of “Mid Risk”. The corresponding SHAP values (-1.377, -0.8773, -0.3773) quantify the magnitude of the negative contribution for each feature. Features such as “SystolicBP=85”, “Age=17”, and “BS=9” are shown with blue arrows pointing to the left, indicating that lower values of these features have a negative contribution towards the prediction of “Mid Risk”. The corresponding SHAP values (0.1227, 0.6227, 1.123) quantify the magnitude of the negative contribution for each feature. The base value of 0.1227, written in bold letters, represents the expected prediction value without considering any specific feature contributions. The value of -0.12, written in the middle of the red and blue colored features, represents the final prediction value after considering the contributions of all the features together.

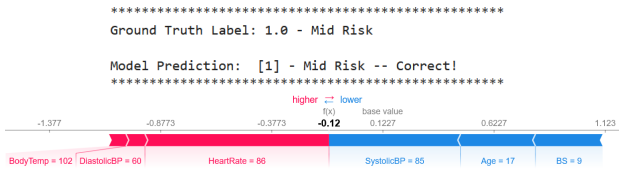



Fig. 12: Explanation of mid risk level by SHAP (Case 02)

Overall, the SHAP force plot helps in understanding the contribution of each feature towards the prediction of “Mid Risk” for the specific data point, with red indicating positive contributions and blue indicating negative contributions. The magnitude of the contributions is quantified by the SHAP values, and the plot provides a visual representation of the feature contributions, aiding in the interpretability of the prediction of the model.

Finally, all three explanations indicate that the predicted risk level for the given data point is “Mid Risk”. They identify similar positively contributing features, such as BodyTemp and HeartRate, as significant for predicting “Mid Risk” in the given data point. They also highlight similar negatively contributing features, such as SystolicBP, DiastolicBP, Age, and BS, as influential in predicting “Not Mid Risk” for the given data point. One key difference between the explanations provided by LIME, SHAP, and XOPSIS is that SHAP identifies DiastolicBP as a positive contributing feature for predicting “Mid Risk” in the given data point, whereas LIME and TOPSIS do not.

LIME provides probability-based explanations and importance scores for positively and negatively contributing features. SHAP uses a force plot to represent the magnitude and direction of contributions for each feature, with red indicating positive and blue indicating negative contributions. XOPSIS

provides normalized contributions for each feature, with positive contributions shown in green and negative contributions shown in red.

The explanations provided by LIME, SHAP, and XOPSIS are aligned in terms of identifying similar positively and negatively contributing features for predicting "Mid Risk" in the given data point. However, they differ in terms of the representation and visualization of the contributions. LIME provides probability-based explanations, SHAP uses a force plot, and XOPSIS provides normalized contributions. The choice of explanation method may depend on the specific use case, context, and user preference.

*c) Low Risk Explanation*

**Case 03**

In the given XOPSIS explanation provided in Figure 13, the data point features are shown initially, followed by the actual risk level and predicted risk level, which are both 0.0. Then, the explanation mentions that the TOPSIS score is calculated based on the similarity of feature values between the given data point and similar instances.

```
------------------ TOPSIS Explanation ------------------
Data point index: 30
------------------ Data Point Features ------------------
Age             20.0
SystolicBP     100.0
DiastolicBP     90.0
BS               7.1
BodyTemp        98.0
HeartRate       88.0
Name: 30, dtype: float64
Actual risk level: 0.0
Predicted Risk Level:  0.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • SystolicBP
        • Age
        • BS
        • BodyTemp
- The features that contribute negatively to the prediction are:
        • DiastolicBP
        • HeartRate
```

Fig. 13: Explanation of low risk level by XOPSIS (Case 03)

The explanation further specifies that there are features that contribute positively to the prediction, which are SystolicBP, Age, BS, and BodyTemp. These positive contributions are shown in green color in the plot in Figure 14. Additionally, there are features that contribute negatively to the prediction, which are DiastolicBP and HeartRate. These negative contributions are shown in red color in the plot. It is important to note that in the context of the given XOPSIS explanation, green color indicates positive contributions, while red color indicates negative contributions. This color scheme is used to visually represent the impact of each feature on the prediction of the risk level.

Figure 15 shows the predicted probabilities for each risk level as (low risk 1.0, mid risk 0.0, high risk 0.0), indicating that the model has predicted the data point to be in the low risk category with a probability of 1.0. The plot is divided into two sections, the right side showing the features that contribute to the prediction of low risk, and the left side showing the features that contribute to the prediction of not being in the low risk category.

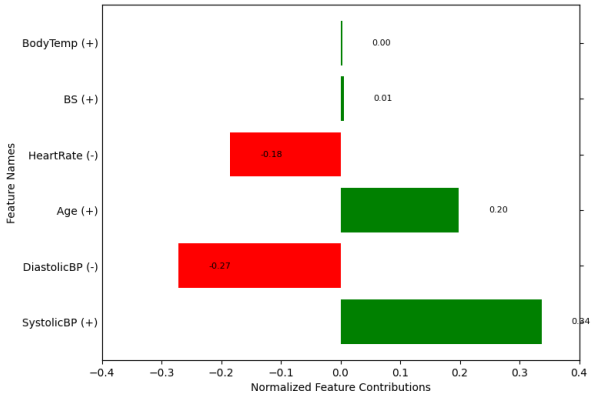



Fig. 14: Normalized Feature Contibutions of low risk level by XOPSIS (Case 03)

The features that contribute positively to the prediction of low risk are: "BS" (Blood Sugar) with a value range of 6.90 to 7.50, with a contribution weight of 0.39; "BodyTemp" (Body Temperature) with a value less than or equal to 98.0, with a contribution weight of 0.25; "SystolicBP" (Systolic Blood Pressure) with a value less than or equal to 100.0, with a contribution weight of 0.16; and "Age" with a value range of 19 to 27, with a contribution weight of 0.09. On the other hand, the features that contribute negatively to the prediction of being in the low risk category are: "HeartRate" (Heart Rate) with a value greater than 80, with a contribution weight of 0.06; and "DiastolicBP" (Diastolic Blood Pressure) with a value greater than 80, with a contribution weight of 0.01. The actual feature values of the data point are explicitly mentioned in this figure, and they are used in the calculation of the contribution weights for each feature in determining the predicted risk level.

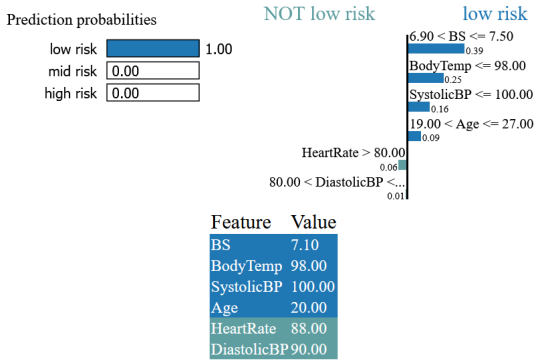



Fig. 15: Explanation of low risk level by LIME (Case 03)

In summary, the LIME output indicates that the model has predicted the data point to be in the low risk category based on the contributions of features such as BS, BodyTemp, SystolicBP, and Age, while features such as HeartRate and DiastolicBP have lesser contributions to the prediction of not being in the low risk category.

The SHAP force plot shown in Figure 16, appears to be for a classification model with a ground truth label of 0.0, indicating a low risk prediction, and a model prediction of 0,

which is also classified as low risk and is correct.

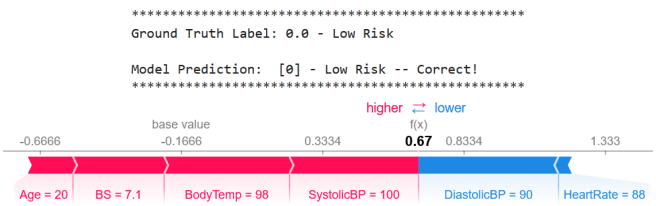



Fig. 16: Explanation of low risk level by SHAP (Case 03)

The plot uses red and blue colors to represent the direction and magnitude of feature contributions. Features with positive contributions are shown in red with the label 'higher− >', while features with negative contributions are shown in blue with the label '< −lower'. The features and their corresponding contributions shown on the plot are as follows: Age = 20, BS=7.1, BodyTemp=98, SystolicBP=100 in red color, indicating a positive contribution towards the model prediction and an increase the predicted risk level. DiastolicBP = 90, HeartRate = 88 are shown in blue color, indicating a negative contribution towards the model prediction and a reduction in the predicted risk level. The base value is -0.1666, which serves as a reference point for the contributions. The value "f(x) 0.67" in bold letters and in the middle of the red and blue color features represents the final predicted risk level based on the accumulated contributions of the features.

Overall, the SHAP force plot provides a visual representation of how each feature contributes towards the prediction of the model of low risk, with red indicating positive contributions and blue indicating negative contributions. The magnitude of the contributions is shown with values, and the base value provides a reference point.

In conclusion, all three explanations provided by LIME, SHAP, and XOPSIS for the same data point share similarities in terms of identifying the features that contribute positively (BS, BodyTemp, SystolicBP, and Age ) and negatively (HeartRate and DiastolicBP) to the prediction of low risk, using visual representations to convey the contributions, providing information about actual and predicted risk levels, explaining the calculation methodology or scoring system, and mentioning specific feature values used in the calculation. These explanations provide valuable insights into the factors that influence the prediction of risk level for the given data point, and can aid in interpreting the predictions of the model and building trust in the decision-making process of the model.

### C. Result Analysis on Iris Dataset

#### 1) Accuracy

The accuracy of the trained models on the Iris dataset was evaluated using two different techniques: the train-test split and k-fold cross-validation with 10 folds. The accuracy scores for three algorithms, namely Gradient Boosting, XGBoost, and Random Forest, are presented in Table XIII.

From the accuracy scores in Table XIII, we observe that all three algorithms achieve high accuracy on the Iris dataset. The

| Serial No | Algorithms | Accuracy | |
|---|---|---|---|
| | | Train-Test Split | K-Fold Cross Validation |
| 01 | Gradient Boosting | 96.667% | 95.333% |
| 02 | XGBoost | 93.333% | 95.299% |
| 03 | Random Forest | 90.000% | 94.667% |

TABLE XIII: Accuracy of 3 Algorithms

Gradient Boosting algorithm achieves the highest accuracy of 96.667% using the train-test split technique, while XGBoost and Random Forest achieve accuracies of 93.333% and 90.000%, respectively. Comparing the train-test split accuracy with the 10-fold cross-validation accuracy, we notice a slight decrease in accuracy for all algorithms when using cross-validation. This difference may be attributed to the use of multiple subsets of data for training and testing in cross-validation, leading to a more generalized model but with slightly lower accuracy.

Additionally, Figure 17 illustrates the train-test split accuracy scores for the three algorithms. The x-axis represents the algorithm names (Gradient Boosting, XGBoost, and Random Forest), while the y-axis represents the accuracy. Each bar in the plot represents the accuracy score achieved by the corresponding algorithm.

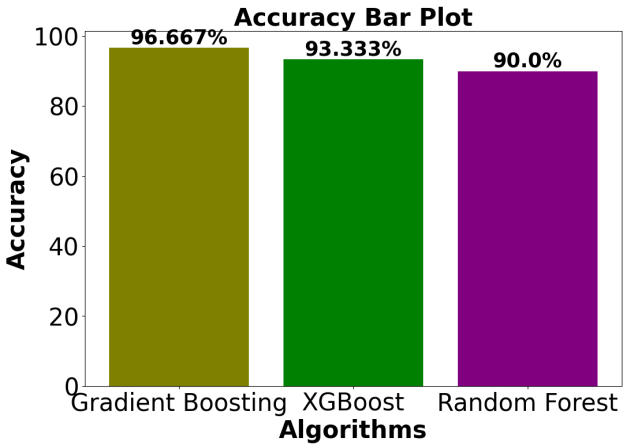



Fig. 17: Train-Test Split Accuracy for Different Algorithms on Iris Dataset

The bar plot visually demonstrates the variation in accuracy across different algorithms. It is evident that Gradient Boosting achieves the highest accuracy score of 96.667%, followed by XGBoost with an accuracy score of 93.333%. Random Forest achieves an accuracy score of 90.000%. This bar plot provides a clear comparison of the accuracy achieved by each algorithm, highlighting the superior performance of Gradient Boosting on the Iris dataset. These accuracy results demonstrate the effectiveness of the Gradient Boosting, XGBoost, and Random Forest algorithms in accurately classifying the Iris dataset.

#### 2) Confusion Matrix

The Figure 18 presents the confusion matrix for the classification of Iris dataset using the trained model (Gradient Boost-

ing). The true labels are displayed vertically, representing Iris-Setosa, Iris-Versicolor, and Iris-Virginica from top to bottom. The predicted labels are shown horizontally. The values in the cells indicate the count of samples classified into each class.

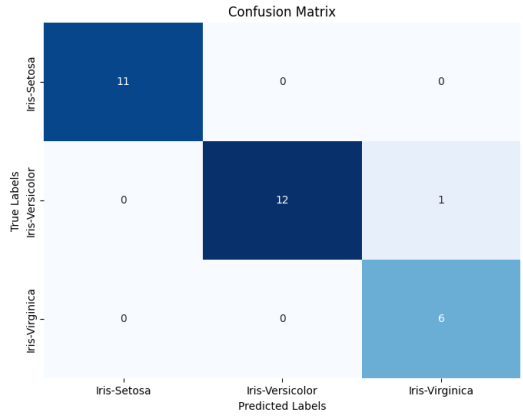



Fig. 18: Confusion Matrix for Iris Dataset

The values in the confusion matrix indicate the counts of samples that belong to each class. Specifically, the values in the first row represent the samples that are truly labeled as Iris-Setosa. In this case, there are 11 samples correctly classified as Iris-Setosa, while there are no misclassifications for this class. Moving to the second row, the values represent the samples that are truly labeled as Iris-Versicolor. Out of the 13 samples of this class, 12 are correctly classified as Iris-Versicolor, while 1 sample is misclassified. Finally, the third row represents the samples that are truly labeled as Iris-Virginica. All 6 samples of this class are correctly classified.

By analyzing the confusion matrix, we can gain insights into the performance of the classification model. In this case, the model shows high accuracy in predicting the Iris-Setosa and Iris-Virginica classes, but there is a single misclassification in the Iris-Versicolor class.

*3) Sensitivity, Specificity, Precision and F1-Score*

| | Sensitivity | Specificity | Precision | F1-Score |
|---|---|---|---|---|
| Iris Setosa | 1 | 1 | 1 | 1 |
| Iris Versicolor | 0.92 | 1 | 1 | 0.96 |
| Iris Virginica | 1 | 0.96 | 0.86 | 0.92 |

TABLE XIV: Performance Evaluation Metrics on Iris Dataset

The Table XIV presents the performance evaluation metrics, including sensitivity, specificity, precision, and F1-score, for the classification of the Iris dataset using the trained model. The metrics are calculated for each class: Iris Setosa, Iris Versicolor, and Iris Virginica.

For Iris Setosa, the model achieves perfect sensitivity, specificity, precision, and F1-score, with a score of 1 for each metric. This indicates that all samples of Iris Setosa are correctly classified. For Iris Versicolor, the model demonstrates a sensitivity of 0.92, meaning that 92% of the Iris Versicolor samples are accurately identified. The model also achieves perfect specificity, precision, and an F1-score of 0.96, indicating high performance in correctly classifying

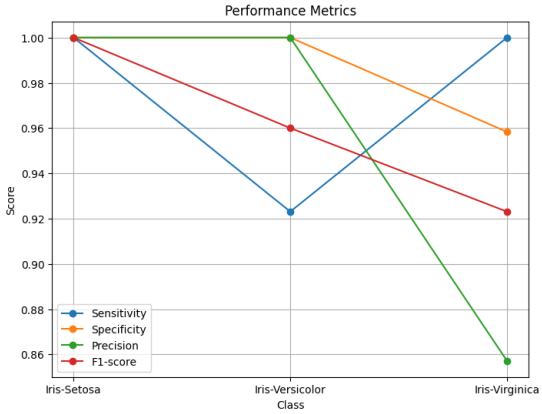



Fig. 19: Performance Evaluation Metrics on Iris Dataset

Iris Versicolor. For Iris Virginica, the model achieves perfect sensitivity, correctly identifying all samples of Iris Virginica. It demonstrates a specificity of 0.96, indicating a high ability to distinguish Iris Virginica from other classes. However, the precision for Iris Virginica is 0.86, implying that there may be some misclassifications. The F1-score for Iris Virginica is 0.92, reflecting a balance between precision and recall.

Furthermore, Figure 19 displays a plot representing the performance evaluation metrics for each class. The x-axis denotes the three labels: Iris Setosa, Iris Versicolor, and Iris Virginica, while the y-axis represents the score. This plot allows for a visual comparison of the sensitivity, specificity, precision, and F1-score across the different classes of the Iris dataset.

*4) Interpretability Through XAI Methods*

In this subsection, we provide explanations for the predictions made by the Gradient Boosting model on the Iris dataset using three XAI (Explainable Artificial Intelligence) methods: XOPSIS, LIME, and SHAP. These methods aim to shed light on the important features that contribute to the model's decision-making process.

*a) Iris Setosa Explanations*

**Case 01**

In Figure 20, we present the XOPSIS explanation for Case 01 in the Iris dataset. The figure starts by displaying the actual feature values of the data point under consideration. It provides insights into the specific values of the features, such as sepal length, sepal width, petal length, and petal width.

Furthermore, the figure reveals that the actual species of the data point is Iris Setosa, while the model predicted the species as 0, which also corresponds to Iris Setosa. The XOPSIS explanation highlights the contributing features to the model's prediction. It identifies positive contributing features, which in this case are petal length and petal width. These features positively influenced the model's decision towards predicting the data point as Iris Setosa. On the other hand, the XOPSIS explanation also identifies negative contributing features, which are sepal length and sepal width. These features had a negative impact on the model's decision, suggesting that they were not significant in classifying the data point as Iris Setosa.

In Figure 21, we present the normalized feature contri-

```
------------------ TOPSIS Explanation ------------------
Data point index: 44
------------------ Data Point Features ------------------
sepal_length    5.1
sepal_width     3.8
petal_length    1.9
petal_width     0.4
Name: 44, dtype: float64
Actual Species: 0.0
Predicted Species:  0.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • petal_length
        • petal_width
- The features that contribute negatively to the prediction are:
        • sepal_length
        • sepal_width
```

Fig. 20: Explanation of Iris-Setosa level by XOPSIS (Case 01)

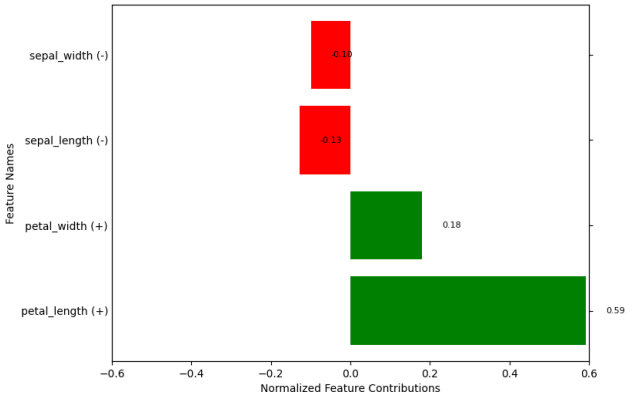



Fig. 21: Normalized Feature Contibutions of Iris-Setosa level by XOPSIS (Case 01)

butions by XOPSIS for Case 01 in the Iris dataset. The figure illustrates the relative importance and direction of each feature's contribution to the model's prediction. The normalized feature contributions are visualized using a color scheme, where green represents positive contributions and red represents negative contributions. The intensity of the color indicates the magnitude of the contribution. These normalized feature contributions provide insights into the relative importance of each feature in determining the model's prediction for Case 01. The positive contributions of petal width and petal length suggest that higher values of these features favor the prediction of Iris Setosa. Conversely, the negative contributions of sepal width and sepal length imply that lower values of these features contribute towards the prediction of Iris Setosa.

By providing such explanations, XOPSIS enhances our understanding of how individual features influence the model's decision-making process for this specific data point in the Iris dataset.

In Figure 22, we present the LIME explanation for Case 01 in the Iris dataset. LIME provides an interpretable understanding of the model's prediction by highlighting the important features and their contributions.

The left section of the figure displays the prediction probabilities for each class. In this case, the prediction probability for Iris-Setosa is 1.0, indicating a high certainty in the model's prediction for this class. The probabilities for the remaining classes (Not Iris-Setosa) are all 0, suggesting a clear distinction

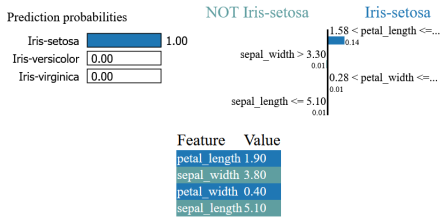



Fig. 22: Explanation of Iris-Setosa level by LIME (Case 01)

in the model's prediction. In the right section, LIME highlights the features that contribute to the prediction of Iris-Setosa. Specifically, it indicates that a petal length greater than 1.58 and a petal width greater than 0.28 favor the prediction of Iris-Setosa. Conversely, it suggests that a sepal width greater than 3.30 and a sepal length less than or equal to 5.10 contribute to the prediction of Not Iris-Setosa. The bottom section of the figure displays the actual feature values for Case 01. This provides a context for understanding how the feature values align with the LIME explanation. By comparing the actual feature values with the highlighted features in the middle section, we can observe the consistency or discrepancy between the two.

Through the LIME explanation, we gain insights into the specific feature values that drive the model's prediction for Case 01. This helps us understand the decision-making process of the model and provides interpretability to the predictions made for the Iris dataset.

Figure 23 presents the SHAP (SHapley Additive exPlanations) explanation for Case 01 in the Iris dataset. SHAP provides a unified framework for interpreting the predictions of machine learning models by assigning importance values to each feature.

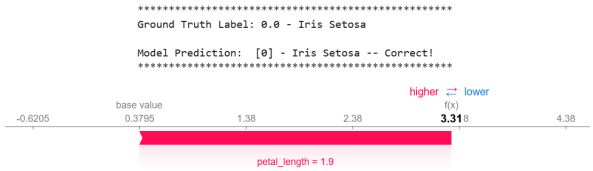



Fig. 23: Explanation of Iris-Setosa level by SHAP (Case 01)

The top section of the figure displays the ground truth label and the model's prediction. In this case, the ground truth label is Iris-Setosa, and the model's prediction is also Iris-Setosa (predicted label: 0). The bottom section of the figure showcases the SHAP force plot. The plot reveals the contribution of each feature to the model's prediction for the given data point. The base value is indicated as 0.3759, representing the expected model output for the dataset.

For Case 01, the plot emphasizes the contribution of the Petal Length feature. The feature value of 1.9 is highlighted, and its impact on the prediction is depicted as a horizontal line connecting the base value to the feature value. The plot indicates that a higher value of Petal Length leads to a higher prediction value, while a lower value results in a lower prediction value. The color scheme employed in the plot conveys the impact of the feature on the prediction. The

region from the base value to the feature value (0.3759 to 3.31) is shown in red, indicating a positive influence, while the region below the base value is shown in blue, suggesting a negative influence. The force plot also provides additional information on the contribution of other features. However, in this particular case, the plot only includes Petal Length. This suggests that Petal Length is the primary driver behind the model's prediction for Case 01. By utilizing the SHAP force plot, we gain insights into the relative importance and influence of individual features on the model's predictions. This enhances our understanding of the decision-making process of the model for the Iris dataset and facilitates the interpretation of its predictions for specific data points.

In summary, these explanations reveal commonalities in identifying important features such as Petal Length and Petal Width contributing positively to the prediction of Iris-Setosa. They also highlight the influence of Sepal Length and Sepal Width, which have negative contributions to the prediction. While XOPSIS provides feature-level contributions, LIME offers local approximations, and SHAP assigns importance values. Together, these explanations provide a comprehensive understanding of how the model reaches its prediction for Case 01 in the Iris dataset.

*b) Iris Versicolor Explanations*

**Case 02**

```
------------------ TOPSIS Explanation ------------------
Data point index: 56
------------------ Data Point Features ------------------
sepal_length    6.3
sepal_width     3.3
petal_length    4.7
petal_width     1.6
Name: 56, dtype: float64
Actual risk level: 1.0
Predicted Risk Level:  1.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • petal_width
        • petal_length
        • sepal_length
- The features that contribute negatively to the prediction are:
        • sepal_width
```

Fig. 24: Explanation of Iris-Versicolor level by XOPSIS (Case 02)

In Case 02, the XOPSIS explanation provides insights into the prediction for the Iris Versicolor species. The explanation begins by displaying the actual feature values of the data point in Figure 24. It then reveals that the actual species is 1 (Iris Versicolor) and the predicted species is also 1, indicating a correct prediction. The XOPSIS explanation further highlights the positively contributed features, which include Petal length, Petal width, and Sepal length. On the other hand, Sepal width is identified as a negatively contributed feature.

Figure 25 displays the normalized feature contributions, where the positive contributions (Petal length: 0.28, Petal width: 0.30, Sepal length: 0.11) are depicted in green color, and the negative contribution (Sepal width: -0.31) is shown in red color. Overall, the XOPSIS explanation for Case 02 emphasizes the importance of features such as Petal length, Petal width, and Sepal length in predicting the Iris Versicolor species.

The LIME explanation from Figure 26 provides insights into the prediction of the Iris Versicolor species for a specific data point.

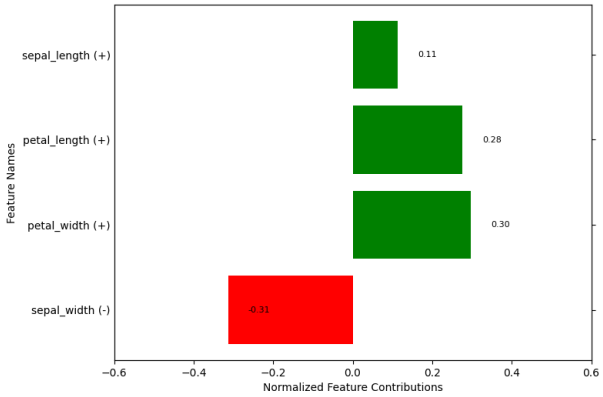



Fig. 25: Normalized Feature Contibutions of Iris-Versicolor level by XOPSIS (Case 02)

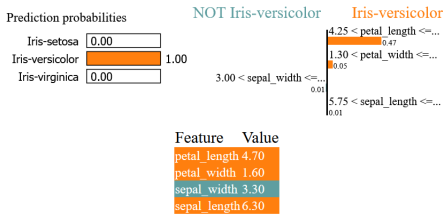



Fig. 26: Explanation of Iris-Versicolor level by LIME (Case 02)

The explanation starts by displaying the prediction probabilities, indicating that the predicted probability for Iris Versicolor is 1, while the probabilities for other species are 0. The LIME explanation further highlights the specific feature ranges that contribute to the prediction of Iris Versicolor. It states that for Iris Versicolor, the feature values of Petal length are greater than 4.25, Petal width is greater than 1.30, and Sepal length is greater than 5.75. Additionally, it indicates that for a prediction of Not Iris Versicolor, the feature value of Sepal width is greater than 3.

In the LIME explanation, these details are presented numerically, with corresponding importance weights assigned to each feature. For Iris Versicolor, Petal length has an importance weight of 0.47, Petal width has an importance weight of 0.05, Sepal length has an importance weight of 0.01, and Sepal width has an importance weight of 0.01. Finally, the actual feature values of the data point are shown, providing a clear understanding of the values associated with each feature. Overall, the LIME explanation for Case 02 highlights the specific feature ranges and their importance in predicting the Iris Versicolor species. It emphasizes that higher values of Petal length, Petal width, and Sepal length contribute to the prediction of Iris Versicolor, while a higher value of Sepal width is associated with a prediction of Not Iris Versicolor.

The SHAP explanation for Case 02 provides insights into the prediction of the Iris Versicolor species for a specific data point.

In Figure 27, the explanation begins by displaying the ground truth label and model prediction, indicating that the true label is 1 (Iris Versicolor) and the model predicts the

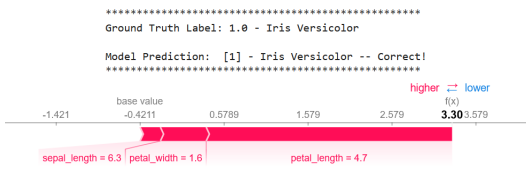



Fig. 27: Explanation of Iris-Versicolor level by SHAP (Case 02)

same. The SHAP explanation further illustrates the contribution of different features through a force plot. The base value is -0.4211, and the force plot shows the progression of the f(x) value. On top of the plot, it includes the notation "higher− >< −lower," where "higher− >" is highlighted in red, and "< −lower" is highlighted in blue.

In the force plot, three features are shown in red color, indicating their positive contributions to the prediction of Iris Versicolor. These features are Sepal length (from -0.4211 to -0.1211), Petal width (from -0.1211 to 0.5789), and Petal length (from 0.5789 to 3.30). The red portion representing the contribution of Sepal length is relatively smaller compared to the contribution of Petal width. By examining the force plot, it becomes apparent that higher values of Sepal length, Petal width, and Petal length positively contribute to the prediction of Iris Versicolor. Overall, the SHAP explanation for Case 02 highlights the contribution of specific features and their ranges in predicting the Iris Versicolor species. It emphasizes that higher values of Sepal length, Petal width, and Petal length have a positive impact on the prediction, providing a clearer understanding of the importance of these features in the model's decision-making process.

In summary, these three explanations consistently emphasize the importance of features such as Petal length, Petal width, and Sepal length in determining the prediction of Iris Versicolor in Case 02. The features Sepal width are shown to have negative contributions in XOPSIS, LIME and SHAP explanations. The similarity among the explanations provides confidence in the significance of these features for the model's decision-making process.

*c) Iris-Virginica Explanations*

**Case 03**

In the XOPSIS explanation for Case 03, the actual feature values of the data point are displayed in the Figure 28. It is revealed that the actual species is Iris-Virginica and the predicted species is also Iris-Virginica (2). The explanation highlights the positively contributed features, which include Petal length, Petal width, Sepal length, and Sepal width.

In Figure 29, the normalized feature contributions are shown. Petal width has a contribution of 0.06, Petal length has a contribution of 0.52, Sepal length has a contribution of 0.32, and Sepal width has a contribution of 0.10. Positive contributions are depicted in green color, while negative contributions are represented in red color.

Overall, the XOPSIS explanation for Case 03 suggests that Petal length, Petal width, Sepal length, and Sepal width play important roles in predicting the class of Iris-Virginica.

In Figure 30, for the same datapoint, the prediction probabilities are displayed. It shows that the prediction for Iris-Virginica is 1.0, indicating high confidence in the predicted class, while the probabilities for the other classes are all 0. The explanation further highlights the specific threshold values for the features that contribute to the prediction of Iris-Virginica. It reveals that the Petal length should be greater than 5.10 (contribution: 0.64), the Petal width should be greater than 1.80 (contribution: 0.61), the Sepal length should be greater than 6.40 (contribution: 0.02), and the Sepal width should be less than or equal to 2.80 (contribution: 0.01) for the classification of Iris-Virginica. All these features positively contribute to the prediction of Iris-Virginica, indicating that higher values of Petal length, Petal width, Sepal length, and lower values of Sepal width are indicative of the Iris-Virginica class.




Fig. 28: Explanation of Iris-Virginica level by XOPSIS (Case 03)

Fig. 29: Normalized Feature Contibutions of Iris-Virginica level by XOPSIS (Case 03)

Fig. 30: Explanation of Iris-Virginica level by LIME (Case 03)

Lastly, the actual feature values for the datapoint are presented, allowing for a direct comparison with the LIME

explanations. Overall, the LIME explanation for Case 03 provides insights into the threshold values of key features that contribute to the prediction of Iris-Virginica, emphasizing the importance of Petal length, Petal width, Sepal length, and Sepal width in determining the classification outcome.

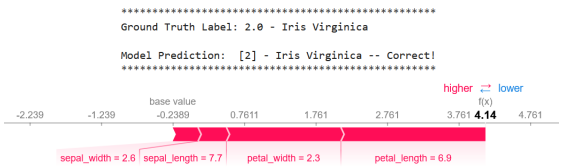



Fig. 31: Explanation of Iris-Virginica level by SHAP (Case 03)

The Shap explanation in Figure 31, for the given datapoint indicates that the ground truth label and model prediction are both Iris-Virginica. The corresponding force plot displays a base value of -0.2389 and f(x) value at 4.14. The plot is annotated with "higher− >" in red color and "< −lower" in blue color. Examining the contributions, we observe that sepal width has the lowest red part, suggesting a relatively smaller positive impact on the prediction. Sepal length shows a lower red part, indicating a slightly stronger positive contribution. Moving further, petal width exhibits a larger red part, implying a more significant positive influence. Finally, petal length displays the largest red part, signifying the highest positive contribution among the features considered in this explanation.

In summary, the three explanations (XOPSIS, LIME, SHAP) consistently highlight the importance of all four features (petal length, petal width, sepal length, and sepal width) in predicting Iris-Virginica in Case 03. Each explanation identifies these features as positively contributing to the prediction, emphasizing their significance in different degrees. This consistency underscores the robustness of these features in determining the classification of Iris-Virginica.

### *D. Result Analysis on Breast Cancer Wisconsin Dataset*

#### *1) Accuracy*

The results of the accuracy analysis for the breast cancer Wisconsin dataset using three different algorithms, namely Gradient Boosting, XGBoost, and Random Forest, are presented in Table XV. The accuracy values are provided for both the train-test split and k-fold cross-validation scenarios.

| Serial No | Algorithms | Accuracy | |
|---|---|---|---|
| | | Train-Test Split | K-Fold Cross Validation |
| 01 | Gradient Boosting | 94.737% | 92.278% |
| 02 | XGBoost | 92.982% | 93.333% |
| 03 | Random Forest | 90.000% | 91.917% |

TABLE XV: Accuracy of 3 Algorithms

For Gradient Boosting, the train-test split accuracy is reported as 94.737%, indicating that the model achieved a high level of accuracy when evaluated on the independent test set. Similarly, the k-fold cross-validation accuracy for Gradient Boosting is reported as 92.278%, which demonstrates the model's consistent performance across different folds.

XGBoost also yielded promising results, with a train-test split accuracy of 93.860%. This indicates the model's ability to accurately classify breast masses into malignant or benign categories. The k-fold cross-validation accuracy for XGBoost is reported as 93.333%, suggesting that the model's performance remains stable across different train-test splits.

Random Forest achieved a train-test split accuracy of 92.982%, showing its effectiveness in accurately predicting the breast cancer outcomes. The k-fold cross-validation accuracy for Random Forest is reported as 91.917%, indicating the model's reliability and generalization capability across multiple folds.

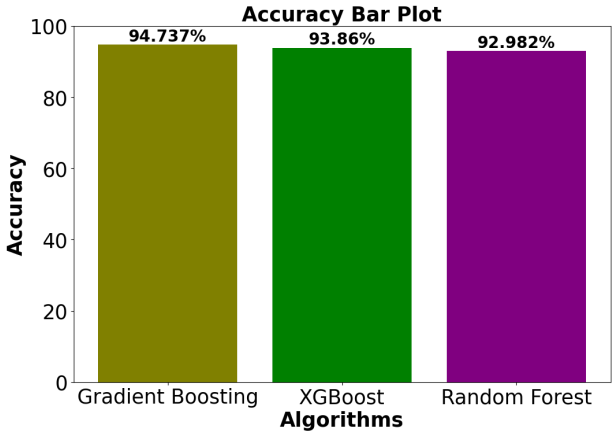



Fig. 32: Train-Test Split Accuracy for Different Algorithms on Breast Cancer Dataset

To visually represent the accuracy of these algorithms, a bar plot was generated in Figure 32. The x-axis of the bar plot represents the names of the three algorithms (Gradient Boosting, XGBoost, and Random Forest), while the y-axis represents the corresponding accuracy values. This plot provides a quick and intuitive comparison of the accuracy performance among the algorithms, further supporting the findings presented in Table XV.

Overall, the accuracy analysis demonstrates the effectiveness of Gradient Boosting, XGBoost, and Random Forest in accurately classifying breast masses and highlights their potential for assisting in breast cancer diagnosis and decision-making processes.

#### *2) Confusion Matrix*

The confusion matrix, presented in Figure 33, provides a detailed analysis of the model's performance by comparing the true labels and predicted labels for the breast cancer Wisconsin dataset. The matrix is organized vertically based on the true labels and horizontally based on the predicted labels.

In the first row of the confusion matrix, the label "M" represents malignant breast masses. The value (38,4) indicates that out of the total 42 instances of malignant masses, the model correctly predicted 38 cases as malignant (true positives), while 4 cases were incorrectly classified as benign (false negatives).

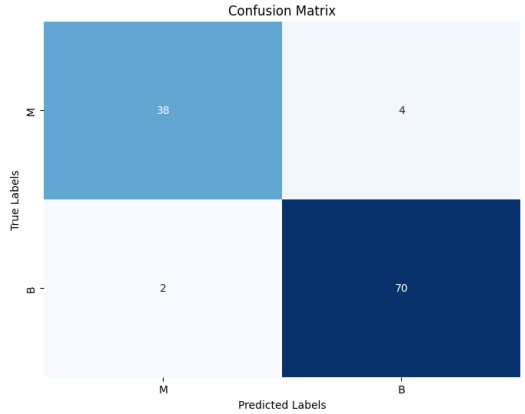



Fig. 33: Confusion Matrix for Breast Cancer Dataset

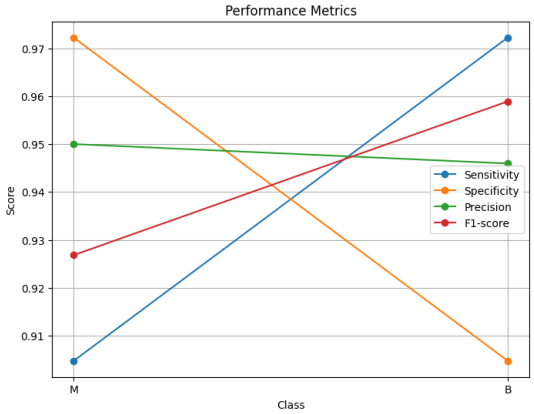



Fig. 34: Performance Evaluation Metrics on Breast Cancer Dataset

In the second row, the label "B" represents benign breast masses. The value (2, 70) shows that out of the total 72 instances of benign masses, the model correctly predicted 70 cases as benign (true negatives), while 2 cases were incorrectly classified as malignant (false positives).

The confusion matrix provides valuable insights into the model's performance in terms of correctly identifying malignant and benign breast masses. It highlights the trade-offs between false positives and false negatives, which are crucial considerations in breast cancer diagnosis. The model's ability to accurately classify malignant masses (high true positive rate) and benign masses (high true negative rate) is essential for minimizing misdiagnosis and ensuring appropriate medical interventions.

The presented confusion matrix assists in evaluating the model's performance metrics such as accuracy, precision, recall, and F1 score, which provide a comprehensive understanding of the model's effectiveness in breast cancer classification. These metrics help gauge the model's ability to correctly identify both malignant and benign cases and serve as important benchmarks for assessing the model's performance in real-world clinical settings.

*3) Sensitivity, Specificity, Precision and F1-Score*

| | Sensitivity | Specificity | Precision | F1-Score |
|---|---|---|---|---|
| M | 0.905 | 0.972 | 0.95 | 0.93 |
| B | 0.972 | 0.905 | 0.95 | 0.96 |

TABLE XVI: Performance Evaluation Metrics on Breast Cancer Dataset

The subsection on Sensitivity, Specificity, Precision, and F1-Score presents a comprehensive evaluation of the model's performance on the breast cancer Wisconsin dataset. The performance metrics are summarized in Table XVI and visualized in a plot. This table provides an overview of the metrics for both malignant (M) and benign (B) classes.

For the malignant class (M), the model achieved a sensitivity (true positive rate) of 0.905, indicating that it correctly identified 90.5% of the actual malignant cases. The specificity (true negative rate) for the malignant class is 0.972, indicating that the model correctly classified 97.2% of the actual benign cases as benign. The precision score, which represents the proportion of correctly classified positive predictions, is 0.95, indicating that 95% of the instances predicted as malignant were indeed malignant. The F1-score, which considers both precision and recall, is 0.93, providing an overall measure of the model's performance for the malignant class.

For the benign class (B), the model achieved a sensitivity of 0.972, indicating that it correctly identified 97.2% of the actual benign cases. The specificity for the benign class is 0.905, indicating that the model correctly classified 90.5% of the actual malignant cases as malignant. The precision score for the benign class is 0.95, indicating that 95% of the instances predicted as benign were indeed benign. The F1-score for the benign class is 0.96, providing an overall measure of the model's performance for the benign class.

These metrics play a crucial role in evaluating the model's performance and assessing its effectiveness in breast cancer classification. Sensitivity and specificity provide insights into the model's ability to correctly identify positive and negative cases, respectively. Precision represents the proportion of true positive predictions among all positive predictions, while the F1-score combines precision and recall into a single metric that balances the trade-off between them.

To visually depict these performance metrics, a plot is presented in Figure 34. The plot shows the class labels (malignant and benign) on the x-axis and the scores on the y-axis. Four performance metrics (sensitivity, specificity, precision, and F1-score) are displayed for both classes. This plot provides a concise and intuitive representation of the model's performance across different evaluation metrics, allowing for easy comparison and interpretation.

*4) Interpretability Through XAI Methods*

In this subsection, we provide explanations for the predictions made by the Gradient Boosting model on the Breast Cancer dataset using three XAI (Explainable Artificial Intelligence) methods: XOPSIS, LIME, and SHAP. These methods aim to shed light on the important features that contribute to the model's decision-making process.

*a) Malignant Explanations*

**Case 01**

The XOPSIS analysis was performed on a specific data point to provide insights into its diagnosis prediction. The analysis includes an examination of the actual feature values, the actual and predicted diagnosis, as well as the contribution of each feature to the prediction.

```
------------------ TOPSIS Explanation ------------------
Data point index: 70
------------------ Data Point Features ------------------
radius_mean                 18.940000
texture_mean                21.310000
perimeter_mean             123.600000
area_mean                 1130.000000
smoothness_mean              0.090090
compactness_mean             0.102900
concavity_mean               0.108000
concave points_mean          0.079510
symmetry_mean                0.158200
fractal_dimension_mean       0.054610
radius_se                    0.788800
texture_se                   0.797500
perimeter_se                 5.486000
area_se                     96.050000
smoothness_se                0.004444
compactness_se               0.016520
concavity_se                 0.022690
concave points_se            0.013700
symmetry_se                  0.013860
fractal_dimension_se         0.001698
radius_worst                24.860000
texture_worst               26.580000
perimeter_worst            165.900000
area_worst                1866.000000
smoothness_worst             0.119300
compactness_worst            0.233600
concavity_worst              0.268700
concave points_worst         0.178900
symmetry_worst               0.255100
fractal_dimension_worst      0.065890
Name: 70, dtype: float64
Actual Diagnosis: 0.0
Predicted Diagnosis:  0.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • area_mean
        • area_worst
        • area_se
        • perimeter_mean
        • perimeter_worst
        • texture_worst
        • radius_mean
        • perimeter_se
        • texture_mean
        • concavity_worst
        • compactness_worst
        • concavity_mean
        • compactness_mean
        • radius_se
        • concave points_worst
        • symmetry_worst
        • concave points_mean
        • symmetry_mean
        • smoothness_worst
        • fractal_dimension_worst
        • concavity_se
        • compactness_se
        • smoothness_mean
        • concave points_se
        • fractal_dimension_se
- The features that contribute negatively to the prediction are:
        • radius_worst
        • texture_se
        • smoothness_se
        • fractal_dimension_mean
        • symmetry_se
```

Fig. 35: Explanation of Malignant level by XOPSIS (Case 01)

Figure 35 displays the actual feature values of the data point, providing a visual representation of its characteristics. The analysis reveals that the actual diagnosis of the data point is 'M' (malignant). The predicted diagnosis is also 'M', indicating that the model correctly identified the data point as malignant.

The XOPSIS analysis further delves into the features that positively and negatively contribute to the prediction. The positive contributions are the features that support the prediction of malignancy, while the negative contributions are the features that suggest a benign diagnosis.

The following features positively contribute to the prediction: area_mean, area_worst, area_se, perimeter_mean, perimeter_worst, texture_worst, radius_mean, perimeter_se, texture_mean, concavity_worst, compactness_worst, concavity_mean, compactness_mean, radius_se, concave_points_worst, symmetry_worst, concave_points_mean, symmetry_mean, smoothness_worst, fractal_dimension_worst, concavity_se, compactness_se, smoothness_mean, concave_points_se, fractal_dimension_se. The following features negatively contribute to the prediction: radius_worst, texture_se, smoothness_se, fractal_dimension_mean, symmetry_se.

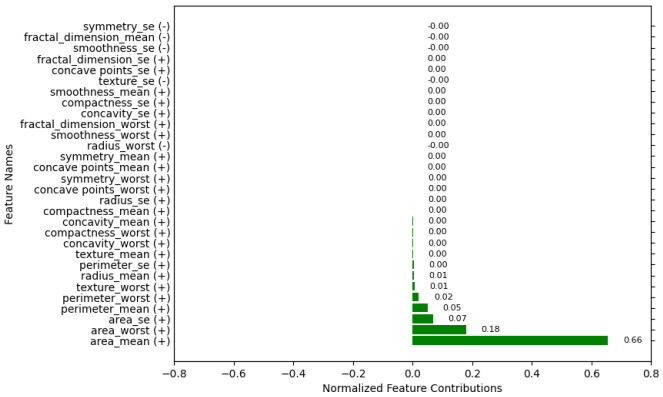



Fig. 36: Normalized Feature Contibutions of Malignant level by XOPSIS (Case 01)

Figure 36 depicts the normalized feature contributions, with positive contributions shown in green and negative contributions shown in red. It is observed that most of the normalized feature contributions are 0, indicating that they have minimal impact on the prediction for this specific data point.

The XOPSIS analysis provides valuable insights into the importance of different features in predicting the diagnosis of the data point. By understanding the contribution of each feature, we can gain a better understanding of the factors influencing the model's prediction and make informed decisions based on this analysis.

Figure 37 represents the LIME explanation by showing the prediction probabilities for the two classes: M (malignant) with a probability of 1.0 and B (benign) with a probability of 0.0.

Next, a plot is displayed, divided into two sections: M (left side) and B (right side). In the M section, the following features are highlighted:

- The concave points feature has a value greater than 0.01 and contributes to the prediction as 0.08.
- The smoothness worst feature has a value greater than 0.11 and contributes as 0.02.
- The symmetry worst feature has a value greater than 0.25 and contributes as 0.02.
- The area mean feature has a value larger than 767.60 and contributes as 0.01.
- The perimeter mean feature has a value larger than 1 and contributes as 0.01.
- The smoothness mean feature has a value greater than 0.09 and contributes as 0.01.
- The texture worst feature has a value greater than 25.22 and contributes as 0.01.
- The concavity worst feature has a value greater than 0.23 and contributes as 0.00.
- Other features like fractal dimension worst, concavity se, and compactness se also contribute as 0.00.

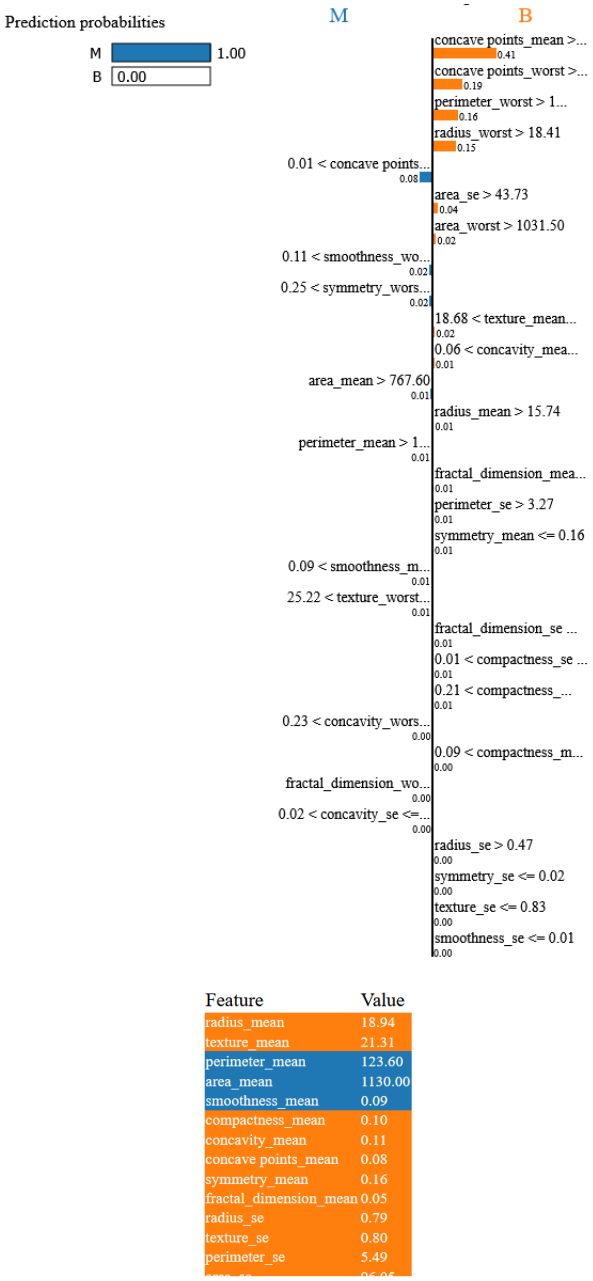



Fig. 37: Explanation of Malignant level by LIME (Case 01)

In the B section, the following features are highlighted:

- The concave points mean feature has a value greater than 0.41 and contributes to the prediction as 0.19.

- The concave points worst feature has a value greater than 0.19 and contributes as 0.16.

- The perimeter worst feature has a value larger than 1 and contributes as 0.15.

- The radius worst feature has a value larger than 18.41 and contributes as 0.15.

- Other features like area se, area worst, texture mean, concavity mean, radius mean, and fractal dimension mean also contribute to the prediction with varying values.

Finally, the actual feature values of the data point are displayed, providing a comprehensive view of the attributes associated with the prediction.

This LIME explanation highlights the specific features and their corresponding contributions that influence the prediction of the given data point, providing insights into the decision-making process of the model.

The SHAP explanation for the given data point in Figure 38 starts by indicating the Ground Truth label, which is 0, and the Model prediction, which is also 0, indicating a correct prediction.


Fig. 38: Explanation of Malignant level by SHAP (Case 01)

Next, a force plot is displayed, representing the SHAP values. The base value is shown as 0.9932, and the f(x) value is -7.92. The force plot is divided into two sections: "$higher->$" shown in red color and "$<-lower$" shown in blue color.

In the "$higher->$" section, the following features are highlighted as having a positive impact on the prediction: - Compactness se - Compactness mean - Fractal dimension mean - Symmetry mean

In the "$<-lower$" section, the following features are highlighted as having a negative impact on the prediction: - Concave points worst - Concave points mean - Area se - Area worst - Texture worst - Concavity worst - Radius worst

The SHAP explanation provides insights into the contribution of each feature to the prediction outcome. The features in the "$higher->$" section positively influence the prediction, while those in the "$<-lower$" section have a negative influence.

By analyzing the SHAP values, we can understand how each feature contributes to the final prediction and gain a deeper understanding of the model's decision-making process for the given data point.

*b) Benign Explanations*

**Case 02**

In Figure 39, the actual feature values of the data point are presented. It is then revealed that the actual and predicted diagnosis for this data point is 1.0, indicating a benign classification. The explanation for this prediction is provided as follows:

- The TOPSIS score is calculated based on the similarity of feature values between the given data point and similar instances.

- The features that positively contribute to the prediction are listed, including area_se, perimeter_se, texture_mean, concavity_worst, compactness_worst, concavity_mean, radius_se, concave_points_worst, compactness_mean, concavity_se, concave_points_mean, compactness_se, fractal_dimension_worst, perimeter_mean, concave_points_se, symmetry_mean, texture_se, and fractal_dimension_se.

- On the other hand, there are features that negatively contribute to the prediction, including area_worst, area_mean, texture_worst, radius_worst, radius_mean, perimeter_worst, symmetry_worst, smoothness_worst, smoothness_mean, fractal_dimension_mean, smoothness_se, and symmetry_se.

```
------------------ TOPSIS Explanation ------------------
Data point index: 394
------------------ Data Point Features ------------------
radius_mean                12.100000
texture_mean               17.720000
perimeter_mean             78.070000
area_mean                 446.200000
smoothness_mean             0.102900
compactness_mean            0.097580
concavity_mean              0.047830
concave points_mean         0.033260
symmetry_mean               0.193700
fractal_dimension_mean      0.061610
radius_se                   0.284100
texture_se                  1.652000
perimeter_se                1.869000
area_se                    22.220000
smoothness_se               0.008146
compactness_se              0.016310
concavity_se                0.018430
concave points_se           0.007513
symmetry_se                 0.020150
fractal_dimension_se        0.001798
radius_worst               13.560000
texture_worst              25.800000
perimeter_worst            88.330000
area_worst                559.500000
smoothness_worst            0.143200
compactness_worst           0.177300
concavity_worst             0.160300
concave points_worst        0.062660
symmetry_worst              0.304900
fractal_dimension_worst     0.070810
Name: 394, dtype: float64
Actual Diagnosis: 1.0
Predicted Diagnosis:  1.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • area_se
        • perimeter_se
        • texture_mean
        • concavity_worst
        • compactness_worst
        • concavity_mean
        • radius_se
        • concave points_worst
        • compactness_mean
        • concavity_se
        • concave points_mean
        • compactness_se
        • fractal_dimension_worst
        • perimeter_mean
        • concave points_se
        • symmetry_mean
        • texture_se
        • fractal_dimension_se
- The features that contribute negatively to the prediction are:
        • area_worst
        • area_mean
        • texture_worst
        • radius_worst
        • radius_mean
        • perimeter_worst
        • symmetry_worst
        • smoothness_worst
        • smoothness_mean
        • fractal_dimension_mean
        • smoothness_se
        • symmetry_se
```

Fig. 39: Explanation of Benign level by XOPSIS (Case 02)

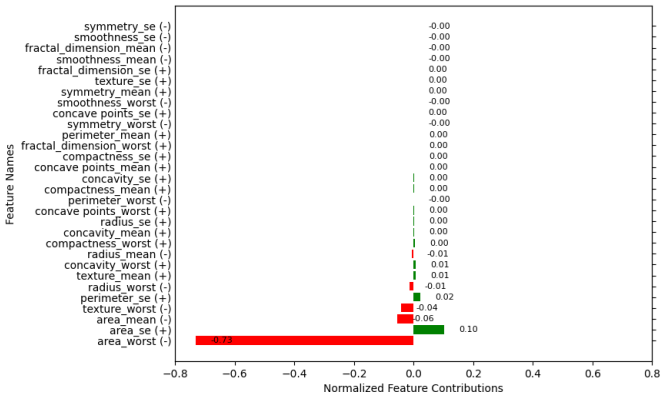
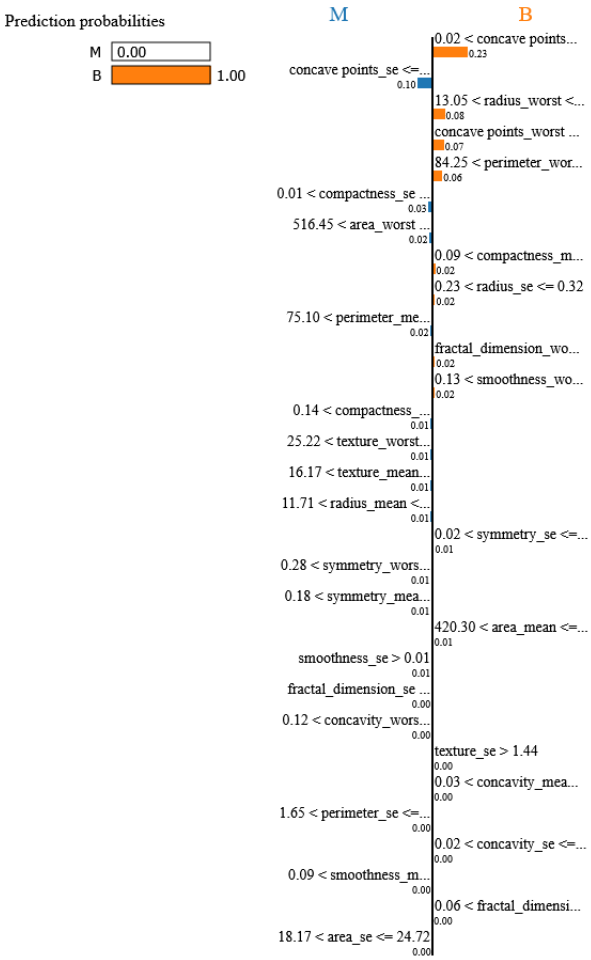



Fig. 40: Normalized Feature Contibutions of Benign level by XOPSIS (Case 02)

Figure 40 presents the normalized feature contributions, where positive contributions are depicted in green, and negative contributions are shown in red. It is observed that most of the normalized feature contributions are close to 0.

This XOPSIS explanation sheds light on the factors influencing the prediction of a benign diagnosis for the given data point, providing valuable insights into the decision-making process.

The LIME explanation for the given data point starts by presenting the prediction probabilities in Figure 41, where the model predicts the probability of M (malignant) as 0.0 and the probability of B (benign) as 1.0.




| Feature | Value |
|---|---|
| radius_mean | 12.10 |
| texture_mean | 17.72 |
| perimeter_mean | 78.07 |
| area_mean | 446.20 |
| smoothness_mean | 0.10 |
| compactness_mean | 0.10 |
| concavity_mean | 0.05 |
| concave points_mean | 0.03 |
| symmetry_mean | 0.19 |
| fractal_dimension_mean | 0.06 |
| radius_se | 0.28 |
| texture_se | 1.65 |
| perimeter_se | 1.87 |

Fig. 41: Explanation of Benign level by LIME (Case 02)

A plot is then displayed, with the left side representing the

explanation for the M class and the right side representing the explanation for the B class. In the M class (left side), the following observations are made:

- concave_points_se is less than or equal to a certain threshold (0.10). - compactness_se is greater than a specific value (0.01). - area_worst is greater than a certain threshold (516.45). - perimeter_mean falls within a specific range (75.10). - compactness_mean and texture_worst satisfy certain conditions. - Other features such as texture_mean, radius_mean, symmetry_worst, symmetry_mean, smoothness_se, fractal_dimension_se, concavity_worst, perimeter_se, smoothness_mean, and area_se also contribute to the explanation.

In the B class (right side), the following observations are made: - concave_points_mean falls within a certain range (0.02). - radius_worst falls within a specific range. - concave_points_worst is present. - perimeter_worst falls within a specific range (84.25). - compactness_mean, radius_se, fractal_dimension_worst, smoothness_worst, symmetry_se, area_mean, texture_se, concavity_mean, concavity_se, and fractal_dimension fall within certain thresholds.

Finally, the actual feature values of the data point are shown, completing the LIME explanation.

This LIME explanation provides insights into the important features and their values that contribute to the model's prediction for the given data point, offering transparency and interpretability for the decision-making process.

Figure 42 displays the SHAP explanation also mentions the ground truth label and the model's prediction, which is 1.0 (correct). This indicates that the model accurately classified the data point as the predicted class.

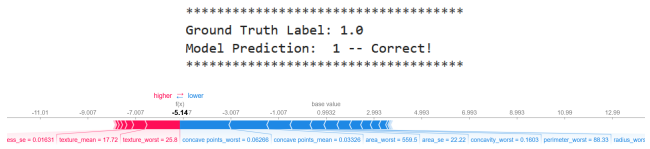



Fig. 42: Explanation of Benign level by SHAP (Case 02)

The SHAP explanation for the given data point begins by showing the base value, which is 0.9932, and the value of f(x), which is observed to be -5.14. On top of the plot, there is a label that reads "$higher- >< -lower$". The text "$higher- >$" is displayed in red color, while "$< -lower$" is shown in blue color.

In the "$higher- >$" part of the explanation, the following features are highlighted: texture_worst, texture_mean, symmetry_se, compactness_mean, fractal_dimension_mean.

These features indicate higher values that contribute positively to the prediction for the given data point.

In the "$< \ -lower$" part of the explanation, the following features are highlighted: concave_points_worst, concave_points_mean, area_worst, area_se, concavity_worst, perimeter_worst.

These features indicate lower values that contribute negatively to the prediction for the given data point.

The SHAP explanation provides insights into the individual feature contributions and their impact on the model's prediction. It helps to understand which features have a positive or negative effect on the predicted outcome and contributes to the interpretability of the model's decision-making process.

### *E. Result Analysis on Car_Acceptability Dataset*

#### *1) Accuracy*

The accuracy of the trained models was evaluated using two different approaches: train-test split and k-fold cross-validation. The results are summarized in Table XVII.

| Serial No | Algorithms | Accuracy | |
|---|---|---|---|
| | | Train-Test Split | K-Fold Cross Validation |
| 01 | Gradient Boosting | 98.266% | 87.447% |
| 02 | XGBoost | 97.688% | 88.021% |
| 03 | Random Forest | 95.954% | 87.276% |

TABLE XVII: Accuracy of 3 Algorithms

From the train-test split analysis, the Gradient Boosting algorithm achieved an accuracy of 98.266%, followed by XGBoost with an accuracy of 97.688%, and Random Forest with an accuracy of 95.954%. These accuracies represent the models' performance in predicting car acceptability on the test set. It is worth noting that all three algorithms performed remarkably well, demonstrating their ability to capture the underlying patterns and make accurate predictions.

Furthermore, the k-fold cross-validation accuracy was evaluated for each algorithm. The results indicated that the Gradient Boosting algorithm achieved an accuracy of 87.447%, followed by XGBoost with an accuracy of 88.021%, and Random Forest with an accuracy of 87.276%. These accuracies reflect the models' performance in predicting car acceptability across different folds of the dataset. The consistency of accuracies across the folds demonstrates the robustness of the models in generalizing to unseen data.

To provide a visual comparison of the train-test split accuracy, we created a barplot (Figure 43) showcasing the performance of each algorithm. The x-axis represents the different algorithms (Gradient Boosting, XGBoost, and Random Forest), while the y-axis represents the accuracy. The barplot clearly illustrates the superior performance of the Gradient Boosting algorithm, followed closely by XGBoost, and then Random Forest.

The high accuracies obtained by the trained models in both the train-test split and k-fold cross-validation analyses indicate their effectiveness in predicting car acceptability based on the provided features. The superior performance of the Gradient Boosting and XGBoost algorithms highlights their potential for accurate car acceptability classification, while Random Forest also exhibits competitive performance.

In summary, the accuracy analysis of the trained models showcases their ability to accurately predict car acceptability. The Gradient Boosting and XGBoost algorithms demonstrate superior performance, while Random Forest also exhibits commendable accuracy. These results highlight the potential of

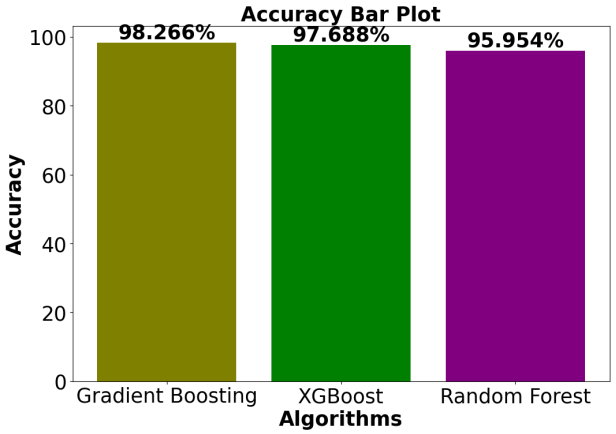



Fig. 43: Train-Test Split Accuracy for Different Algorithms on Car Acceptability Dataset

these algorithms for practical applications in car acceptability classification tasks.

*2) Confusion Matrix*

The confusion matrix provides detailed insights into the performance of the trained models in predicting car acceptability. The matrix represents the relationship between the true labels and the predicted labels. In our analysis, the confusion matrix is presented in Figure 44.

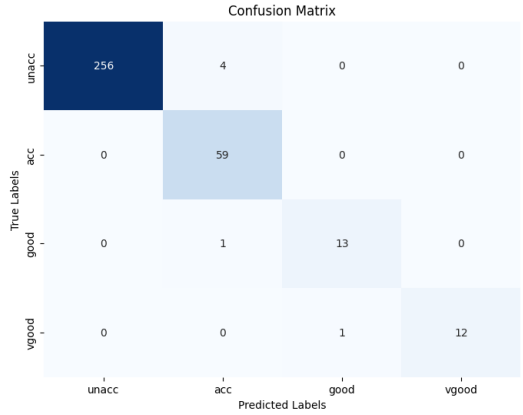



Fig. 44: Confusion Matrix for Car Acceptability Dataset

In the confusion matrix, the rows represent the true labels, while the columns represent the predicted labels. The four car acceptability categories, namely "unacc" (unacceptable), "acc" (acceptable), "good", and "vgood" (very good), are arranged in the same order both vertically and horizontally.

The values within the matrix indicate the number of instances that fall into each category. For example, in the first row, the model correctly predicted 256 instances as "unacc" (unacceptable) while misclassifying 4 instances as "acc" (acceptable). Similarly, in the second row, all instances were correctly classified as "acc", resulting in 59 accurate predictions. The remaining rows follow a similar pattern, reflecting the model's performance for each car acceptability category.

The confusion matrix provides a comprehensive overview of the model's performance, allowing us to assess its accuracy and identify any potential misclassifications. The diagonal elements represent the true positive predictions, indicating instances that were correctly classified. Off-diagonal elements represent misclassifications or false positive predictions.

By analyzing the confusion matrix, we can evaluate the model's performance for each car acceptability category and identify any imbalances or discrepancies in the predictions. These insights can guide further improvements in the model or aid in making informed decisions based on the specific requirements of car acceptability classification.

In summary, the confusion matrix presents a detailed breakdown of the model's predictions for car acceptability. It allows us to assess the accuracy and performance of the trained models across different car acceptability categories, providing valuable insights for further analysis and evaluation.

*3) Sensitivity, Specificity, Precision and F1-Score*

To comprehensively evaluate the performance of the trained models on the car acceptability dataset, we calculated several performance metrics, including sensitivity, specificity, precision, and F1-score. Table XVIII presents the performance evaluation metrics for each car acceptability class.

| | Sensitivity | Specificity | Precision | F1-Score |
|---|---|---|---|---|
| unacc | 0.985 | 1.0 | 1.0 | 0.99 |
| acc | 1.0 | 0.983 | 0.92 | 0.96 |
| good | 0.929 | 0.997 | 0.93 | 0.93 |
| vgood | 0.923 | 1.0 | 1.0 | 0.96 |

TABLE XVIII: Performance Evaluation Metrics on Car Acceptability Dataset

The sensitivity metric measures the ability of the model to correctly identify positive instances, while specificity measures the model's ability to correctly identify negative instances. Precision indicates the proportion of correctly predicted positive instances out of all instances predicted as positive. The F1-score provides a balanced measure of precision and recall, considering both the false positive and false negative rates.

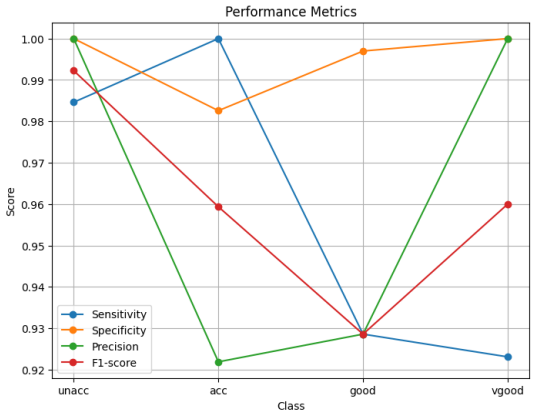



Fig. 45: Performance Evaluation Metrics on Car Acceptability Dataset

In Figure 45, we visualize the performance metrics for each car acceptability class on the same plot, allowing for easy

comparison. The x-axis represents the car acceptability classes, namely “unacc”, “acc”, “good”, and “vgood”, while the y-axis represents the performance scores for sensitivity, specificity, precision, and F1-score. Each class is represented by a line in the plot, enabling a visual assessment of the variations in the metrics across different car acceptability categories.

By examining the performance metrics and the corresponding plot, we can gain insights into the strengths and weaknesses of the models in classifying car acceptability. The high sensitivity values across all classes indicate the models’ ability to correctly identify positive instances. The specificity values demonstrate the models’ capacity to accurately identify negative instances. The precision scores indicate the models’ effectiveness in correctly predicting positive instances, while the F1-scores provide a balanced evaluation of precision and recall.

In summary, the sensitivity, specificity, precision, and F1-score metrics provide a comprehensive evaluation of the models’ performance on the car acceptability dataset. The visual representation of these metrics in the plot allows for easy comparison and interpretation of the models’ capabilities in classifying different car acceptability categories. These performance metrics contribute to a deeper understanding of the models’ performance and can guide further improvements or decision-making processes in the context of car acceptability classification.

*4) Interpretability Through XAI Methods*

In this subsection, we provide explanations for the predictions made by the Gradient Boosting model on the Car Acceptability dataset using three XAI (Explainable Artificial Intelligence) methods: XOPSIS, LIME, and SHAP. These methods aim to shed light on the important features that contribute to the model’s decision-making process.

*a) Unacc Explanation*

**Case 01**

In the XOPSIS explanation for a specific data point, Figure 46 presents the actual feature values. It showcases the features of the data point and provides insights into the actual values recorded. Additionally, it displays the predicted Acceptability Level, which in this case is 0.0 (Unacc).

```
------------------ TOPSIS Explanation ------------------
Data point index: 344
------------------ Data Point Features ------------------
Buying_Price         3.0
Maintenance_Price    0.0
No_of_Doors          0.0
Person_Capacity      2.0
Size_of_Luggage      0.0
Safety               2.0
Name: 344, dtype: float64
Actual Acceptability Level: 0.0
Predicted Acceptability Level:  0.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • Size_of_Luggage
        • No_of_Doors
        • Buying_Price
- The features that contribute negatively to the prediction are:
        • Maintenance_Price
        • Safety
        • Person_Capacity
```

Fig. 46: Explanation of Unacc level by XOPSIS (Case 01)

The explanation proceeds by introducing the TOPSIS score, which is calculated based on the similarity of feature values between the given data point and similar instances. It highlights the features that contribute positively to the prediction, including Size_of_Luggage, No_of_Doors, and Buying_Price. These features have a favorable impact on the prediction outcome, suggesting that higher values or specific characteristics in these features increase the likelihood of the car being classified as “Unacc” (Unacceptable).

Conversely, the explanation identifies the features that contribute negatively to the prediction. These features, namely Maintenance_Price, Safety, and Person_Capacity, have a detrimental effect on the prediction outcome. Lower values or certain characteristics in these features are associated with a higher probability of the car being classified as “Unacc.”

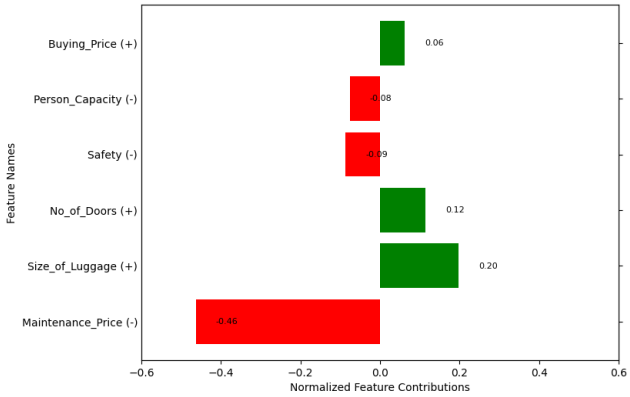



Fig. 47: Normalized Feature Contibutions of Unacc level by XOPSIS (Case 01)

Figure 47 illustrates the plot of normalized feature contributions. Each feature’s contribution to the prediction is represented by a bar in the plot. The positive contributions, denoted in green, indicate the features’ positive influence on the prediction, while the negative contributions, shown in red, represent the features’ negative impact.

According to the plot, Buying_Price has a positive contribution (+0.06), suggesting that higher values or specific characteristics in this feature contribute favorably to the prediction. Similarly, No_of_Doors and Size_of_Luggage also have positive contributions, indicating their importance in the prediction. On the other hand, Maintenance_Price has a notable negative contribution (-0.46), suggesting that lower values or certain characteristics in this feature have a strong negative influence on the prediction outcome. Additionally, Safety and Person_Capacity also exhibit negative contributions (-0.09 and -0.08, respectively), indicating their adverse effects on the prediction.

The visualization of the normalized feature contributions provides a comprehensive understanding of how each feature contributes to the prediction outcome. The distinct colors (green and red) aid in easily identifying the positive and negative contributions, enabling users to interpret the impact of individual features on the car acceptability classification.

In Figure 48, the LIME explanation for the given data point starts by displaying the prediction probabilities for

each class. In this case, the prediction probabilities are as follows: unacc=1.0, acc=0.00, good=0.00, vgood=0.0. This indicates a high confidence in classifying the data point as "unacc" (Unacceptable) and negligible probabilities for the other classes.

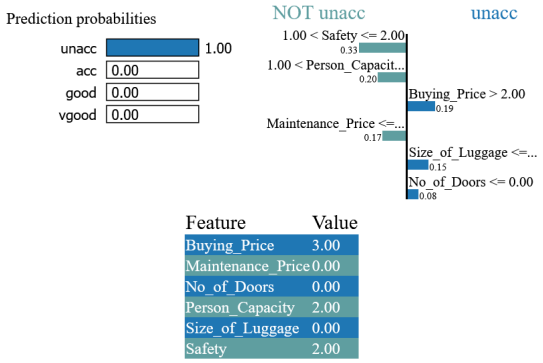



Fig. 48: Explanation of unacc level by LIME (Case 01)

The explanation proceeds with a plot that is divided into two sections. On the right side of the plot, it focuses on the features that contribute to the classification of "unacc," while on the left side, it explores the features that indicate a deviation from the "unacc" class.

On the right side of the plot, it is evident that the feature Buying_Price has a positive contribution (0.19), indicating that a value greater than 2.00 contributes to the classification of "unacc." Similarly, the features Size_of_Luggage (0.15) and No_of_Doors (0.08) also have positive contributions, suggesting that certain threshold values or higher values contribute to the prediction of "unacc" class.

On the left side of the plot, representing the features associated with "Not unacc" classes, all features exhibit negative contributions. The feature Safety has a positive contribution (0.33) when the value falls within the range of 1.00 to 2.00. Likewise, the feature Person_Capacity has a positive contribution (0.20) when the value exceeds 1.00. Furthermore, the feature Maintenance_Price demonstrates a negative contribution (0.17) when the value is below a certain threshold, contributing to the classification of "Not unacc" categories.

The explanation concludes by presenting the actual feature values of the data point. These values provide insight into the specific characteristics of the data point that led to its classification as "unacc" and the contribution of each feature to this prediction.

The provided SHAP explanation in Figure 49 indicates that the ground truth label for the given data point is 0.0, which corresponds to the "unacc" class. The model's prediction for this data point is also 0, which aligns with the ground truth label. Thus, the model prediction is correct in classifying the data point as "unacc."

The SHAP explanation for the given datapoint reveals the following:

In the accompanying force plot, the base value is 4.371, and the value of f(x) is seen as 7.03. The force plot is divided into two sections, with "higher− >" displayed in red color and "< −lower" displayed in blue color.

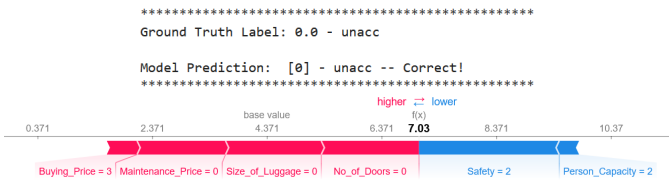



Fig. 49: Explanation of unacc level by SHAP (Case 01)

In the red-colored "higher− >" section, the features Buying_Price=3.0, Maintenance_Price=0.0, Size_of_luggage=0, and No_of_doors=0 contribute positively to the prediction. These features indicate that higher values for Buying_Price, Maintenance_Price, and Size_of_luggage, as well as having No_of_doors equal to 0, contribute to the classification of "unacc."

In the blue-colored "< −lower" section, the features Safety=2.0 and Person_Capacity=2.0 contribute negatively to the prediction. This suggests that lower values for Safety and Person_Capacity are associated with the classification of "unacc."

Overall, the SHAP explanation provides insights into the feature contributions that lead to the prediction of "unacc" for the given data point.

In summary, the explanations provided by XOPSIS, LIME, and SHAP for the specific data point consistently support the classification of the data point as "unacc." XOPSIS and LIME highlight the positive contributions of features such as Buying_Price, Size_of_Luggage, and No_of_Doors, indicating their influence in favor of the "unacc" class. It is evident that both XOPSIS and LIME provide identical feature contributions for the data point, reinforcing the classification of "unacc." This consistency in feature importance across XOPSIS and LIME strengthens the interpretability and trustworthiness of the model's prediction for the car's acceptability. The alignment of explanations between XOPSIS and LIME underscores the agreement in identifying the key features influencing the classification decision, further enhancing our understanding of the model's decision-making process. SHAP further reinforces these findings, with Buying_Price, Maintenance_Price, and Size_of_Luggage exhibiting positive contributions, and Safety and Person_Capacity demonstrating negative contributions.

These consistent explanations from multiple XAI methods strengthen our confidence in the classification of the data point as "unacc" and provide valuable insights into the features that contribute to this prediction. Such interpretability aids in understanding the decision-making process of the model and enhances transparency in the classification of car acceptability.

*b) Acc Explanation*

**Case 02**

In the XOPSIS explanation for a new data point, the Figure 50 presents the actual feature values. It then displays the actual and predicted Acceptability Level, which in this case is predicted as 1.0 (Acc). The explanation provided by XOPSIS includes the following details. The features that contribute positively to the predictio of "Acc" are Buying_Price, Maintenance_Price, Safety, and No_of_Doors. Their positive

contributions indicate that higher values or specific ranges of these features favor the classification of "Acc."

Conversely, the XOPSIS analysis identifies Size_of_Luggage and Person_Capacity as features that negatively contribute to the prediction. This means that lower values or certain thresholds of these features are associated with the classification of "Acc."

```
------------------ TOPSIS Explanation ------------------
Data point index: 1172
------------------ Data Point Features ------------------
Buying_Price         1.0
Maintenance_Price    1.0
No_of_Doors          3.0
Person_Capacity      1.0
Size_of_Luggage      0.0
Safety               2.0
Name: 1172, dtype: float64
Actual Acceptability Level: 1.0
Predicted Acceptability Level:  1.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • Buying_Price
        • Maintenance_Price
        • Safety
        • No_of_Doors
- The features that contribute negatively to the prediction are:
        • Size_of_Luggage
        • Person_Capacity
```

Fig. 50: Explanation of Acc level by XOPSIS (Case 02)

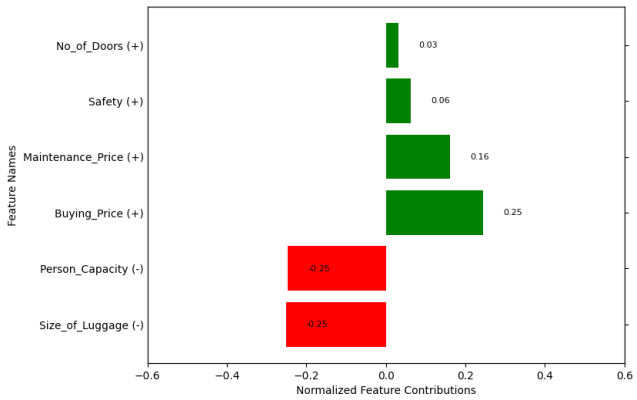



Fig. 51: Normalized Feature Contibutions of Acc level by XOPSIS (Case 02)

Figure 51 illustrates the normalized feature contributions. Positive contributions are represented in green, while negative contributions are shown in red. The normalized feature contributions for the data point are as follows: No_of_Doors (+0.03), Safety (+0.06), Maintenance_Price (+0.16), Buying_Price (+0.25), Person_Capacity (-0.25), and Size_of_Luggage (-0.25). These values indicate the degree to which each feature influences the classification of the data point.

By uncovering these specific feature contributions, XOPSIS provides valuable insights into the decision-making process of the model, shedding light on the factors that drive the prediction of "Acc." This information enhances our understanding of the underlying dynamics of the dataset and reinforces the interpretability of the XOPSIS algorithm.

In Figure 52, the LIME explanation for the same data point begins by displaying the prediction probabilities: unacc=0.00, acc=1.00, good=0.00, vgood=0.00. This indicates that the model predicts the data point to belong to the "acc" (Acceptability Level: 1.0) class with a probability of 1.00.

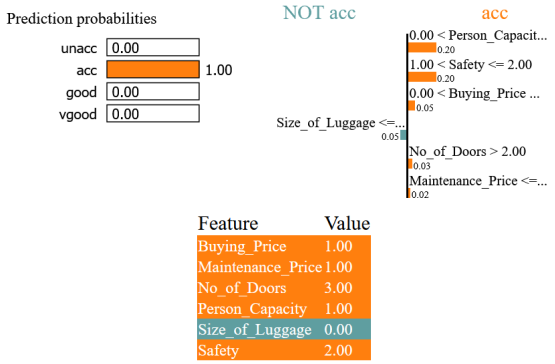



Fig. 52: Explanation of acc level by LIME (Case 02)

Next, the LIME explanation presents a plot that showcases the feature contributions. On the right side of the plot, it displays the features associated with the "acc" class. These features include Person_Capacity (0.20), Safety (0.20), Buying_Price (0.05), No_of_Doors (0.03), and Maintenance_Price (0.02). These positive contributions suggest that higher values or specific ranges of these features support the classification of "acc."

On the left side of the plot, it represents the features associated with the "Not acc" classes. In this case, it shows that the Size_of_Luggage feature has a contribution of 0.05. This negative contribution suggests that lower values or certain thresholds of Size_of_Luggage favor the classification of "Not acc."

Lastly, the LIME explanation provides the actual feature values of the data point, providing a comprehensive understanding of the specific characteristics that contribute to the model's prediction.

Figure 53 displays the SHAP explanation for the same data point begins by indicating the ground truth label as 1.0 - acc (Acceptability Level: 1.0) and the model's correct prediction of [1] - acc.

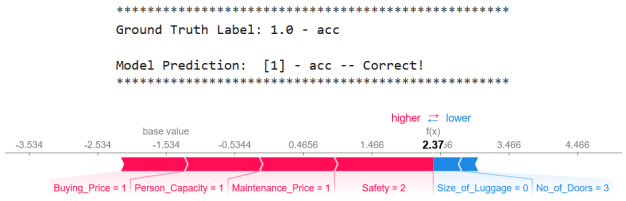



Fig. 53: Explanation of acc level by SHAP (Case 02)

Next, it presents a force plot where the base value is -1.534. The f(x) value is observed at 2.37, and on top of the plot, the annotation "higher− >< −lower" is displayed. The text "higher− >" is highlighted in red, while "< −lower" is highlighted in blue.

In the plot, the features Buying_Price, Person_Capacity, Maintenance_Price, and Safety are displayed in red, representing their positive contributions (higher− >). These features contribute positively to the classification of "acc," indicating

that higher values or specific ranges of these features support the prediction of "acc."

On the other hand, the features Size_of_Luggage and No_of_Doors are shown in blue, representing their negative contributions ($<$ $-$lower). These features have a negative impact on the prediction of "acc," suggesting that lower values or specific thresholds of these features favor the classification of "Not acc."

By visualizing the feature contributions in the force plot, the SHAP explanation provides insights into how each feature influences the model's prediction for the given data point.

Analyzing the explanations provided by XOPSIS, LIME, and SHAP for the same data point, it is evident that all three methods consistently highlight the features that contribute to the prediction of "acc" (Acceptability Level: 1.0). According to XOPSIS, Buying_Price, Maintenance_Price, Safety, and No_of_Doors have positive contributions, indicating that higher values or specific ranges of these features support the classification of "acc." On the other hand, Size_of_Luggage and Person_Capacity are identified as features with negative contributions, suggesting that lower values or specific thresholds of these features favor the classification of "Not acc" while LIME shows only Size_of_Luggage as negative contributing feature.

Similarly, the SHAP explanation aligns with XOPSIS and LIME, showcasing the positive contributions of Buying_Price, Person_Capacity, Maintenance_Price, and Safety. Additionally, SHAP identifies Size_of_Luggage and No_of_Doors as features with negative contributions. This consistency across the three explanations enhances our understanding of the influential features and reinforces the model's prediction for the acceptability level of the car.

The agreement in feature importance between XOPSIS, LIME, and SHAP provides a robust interpretation of the model's decision-making process. It instills confidence in the identified features and their impact on the classification outcome, ultimately increasing the transparency and interpretability of the model's predictions for the given data point.

*c) Good Explanation*

**Case 03**

```
------------------ TOPSIS Explanation ------------------
Data point index: 1667
------------------ Data Point Features ------------------
Buying_Price        0.0
Maintenance_Price   0.0
No_of_Doors         1.0
Person_Capacity     2.0
Size_of_Luggage     0.0
Safety              2.0
Name: 1667, dtype: float64
Actual Acceptability Level: 2.0
Predicted Acceptability Level:  2.0

Explanation:
- The TOPSIS score is calculated based on the similarity of feature values
  between the given data point and the similar instances.
- The features that contribute positively to the prediction are:
        • Safety
        • Buying_Price
        • Maintenance_Price
        • Person_Capacity
- The features that contribute negatively to the prediction are:
        • Size_of_Luggage
        • No_of_Doors
```

Fig. 54: Explanation of Good level by XOPSIS (Case 03)

The XOPSIS explanation for another new data point consistently identifies the features that contribute to the prediction of "Good" (Acceptability Level: 2.0) in Figure 54. According to XOPSIS, the features Safety, Buying_Price, Maintenance_Price, and Person_Capacity have positive contributions, while Size_of_Luggage and No_of_Doors have negative contributions. This insight provides a deeper understanding of the factors influencing the model's decision and highlights the importance of safety, pricing, maintenance, and passenger capacity in determining the acceptability level of a car. The normalized feature contributions further reinforce the significance of these features in Figure 55, with Safety exhibiting the highest positive contribution, followed by Buying_Price and Maintenance_Price. Conversely, Size_of_Luggage has the highest negative contribution, indicating its impact on the prediction of "Good."

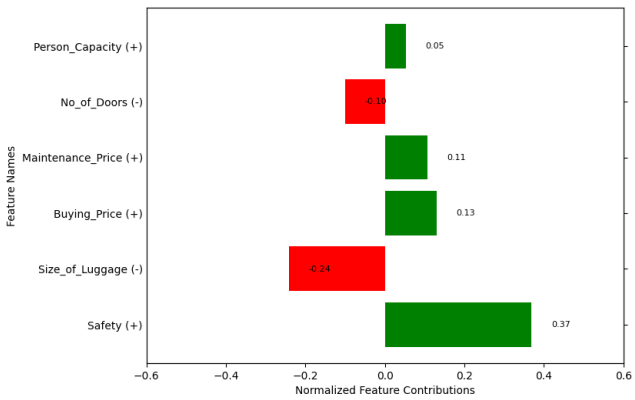



Fig. 55: Normalized Feature Contibutions of Good level by XOPSIS (Case 03)

The LIME explanation in Figure 56 for the same data point reveals the feature contributions that support the prediction of "Good" (Acceptability Level: 2.0). LIME assigns probabilities to each class, and in this case, the model predicts a high probability of "Good" while assigning negligible probabilities to other classes. The LIME plot illustrates that certain feature conditions on the right side, such as Buying_Price $<=$ 0.00, Maintenance_Price $<=$ ..., 1.00 $<$ Person_Capacity ..., and 1.00 $<$ Safety $<=$ 2.00, contribute to the classification of "Good." Conversely, features on the left side, such as Size_of_Luggage $<=$ ..., and 0.00 $<$ No_of_Doors ..., are associated with "Not good" categories. The actual feature values of the data point are also provided for reference.

Figure 57 represents the SHAP explanation for the data point reveals that the ground truth label and model prediction both correctly identify it as "good." The force plot demonstrates the feature contributions, with a base value of -4.879 and a f(x) value of -5.76. The plot is divided into two sections, denoted by "higher$-$ $>$" in red and "$<$ $-$lower" in blue. In the red section, the features Buying_Price, Size_of_luggage, and Safety exhibit positive contributions. On the other hand, the blue section shows negative contributions from the features Person_Capacity, Maintenance_Price, and No_of_doors.

After reviewing the XOPSIS, LIME, and SHAP explanations for the same data point, we can observe a consistent pattern in feature importance. All three methods highlight Safety,

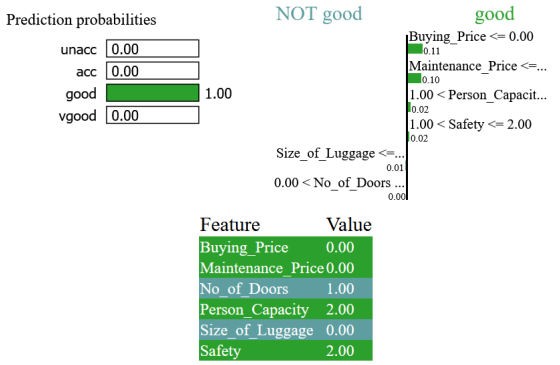



Fig. 56: Explanation of good level by LIME (Case 03)

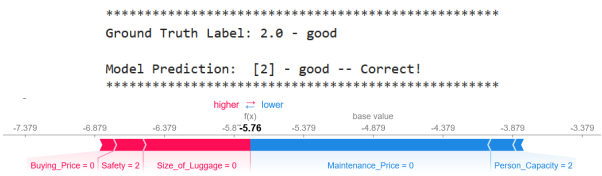



Fig. 57: Explanation of good level by SHAP (Case 03)

and Buying_Price as positive contributors and No_of_doors as negative contributors to the prediction of "good" acceptability. However, there is a discrepancy in the treatment of Size_of_Luggage. While XOPSIS and LIME consider it as a negative contributor, SHAP does not highlight it as a significant factor. This inconsistency suggests that Size_of_Luggage may have varying degrees of impact on the model's decision across different explanation methods. Nonetheless, the consensus on the positive contributions of Safety, and Buying_Price and negative contributions of No_of_doors enhances our understanding of the key features driving the prediction of "good" acceptability.

*d) Vgood Explanation*

**Case 04**

In the XOPSIS explanation for a new data point, the analysis begins by displaying the actual feature values. Figure 58 then reveals the actual and predicted acceptability level, indicating a prediction of "Vgood" (very good). The explanation continues by identifying the features that contribute positively and negatively to the prediction.

The features that positively contribute to the prediction of "Vgood" are Buying_Price, Person_Capacity, and No_of_Doors. This suggests that higher values or specific thresholds for these features align with a higher likelihood of the car being classified as "Vgood." On the other hand, the features Maintenance_Price, Size_of_Luggage, and Safety demonstrate negative contributions to the prediction. Lower values or specific thresholds for these features are associated with a higher likelihood of the car being classified as "Vgood."

Figure 59 presents the normalized feature contributions, visually represented by green and red colors. Safety exhibits a negligible contribution (close to 0), indicating that it does not strongly influence the prediction. No_of_Doors has a neutral contribution (close to 0), implying that its value does not significantly impact the prediction. Size_of_Luggage demon-

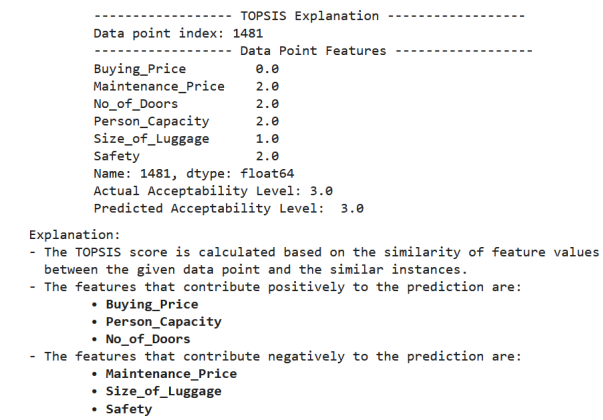



Fig. 58: Explanation of Vgood level by XOPSIS (Case 04)

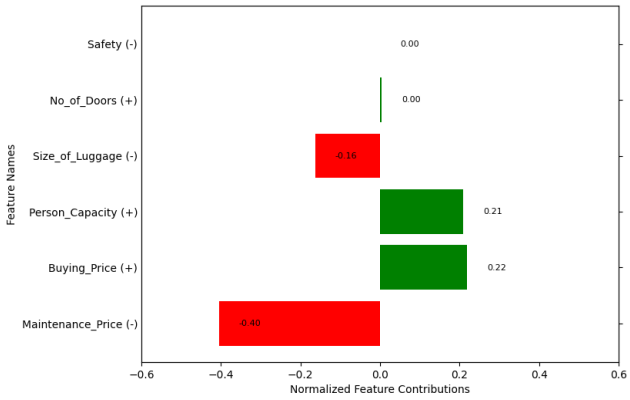



Fig. 59: Normalized Feature Contibutions of Vgood level by XOPSIS (Case 04)

strates a negative contribution (-0.16), suggesting that values lower than a certain threshold contribute to the classification of "Vgood." Conversely, Person_Capacity and Buying_Price show positive contributions (0.21 and 0.22, respectively), indicating that higher values or specific thresholds for these features contribute to the prediction. Lastly, Maintenance_Price exhibits a substantial negative contribution (-0.40), suggesting that lower values or specific thresholds for this feature strongly influence the classification of "Vgood".

Figure 60 represents the LIME explanation for the same data point begins by displaying the prediction probabilities, indicating a high confidence prediction of "Vgood" (very good) with a probability of 1.0. The explanation continues with a plot that visually represents the contributions of different features to the prediction.

On the right side of the plot, which corresponds to the "Vgood" class, the features that positively contribute to the prediction are Safety, Buying_Price, Person_Capacity, and No_of_Doors. This suggests that specific thresholds or higher values for these features are indicative of the car being classified as "Vgood." On the left side of the plot, representing the "Not vgood" class, the feature Maintenance_Price demonstrates a negative contribution, implying that lower values or specific thresholds for this feature are associated with a lower likelihood of the car being classified as "Vgood." Additionally,

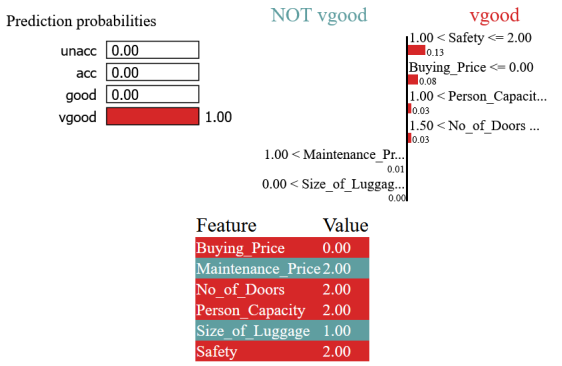



Fig. 60: Explanation of vgood level by LIME (Case 04)

Size_of_Luggage exhibits a negligible contribution, indicating that its value does not significantly influence the prediction.

The LIME explanation concludes by presenting the actual feature values of the data point, allowing for a comprehensive understanding of the specific characteristics that led to the prediction of "Vgood" for this particular instance.

Figure 61 displays the SHAP explanation for the same data point begins by stating the Ground Truth Label and the Model Prediction, both of which correctly identify the class as "Vgood."

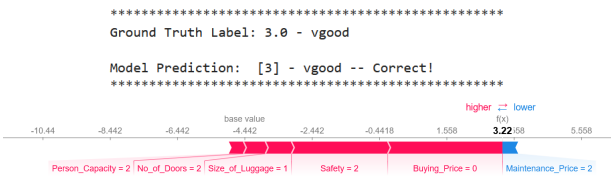



Fig. 61: Explanation of vgood level by SHAP (Case 04)

The explanation further presents a force plot that visualizes the contributions of different features to the prediction. The base value is indicated as -4.442, and the plot shows that the overall contribution leads to a value of 3.22. The plot is divided into two sections, with "higher− >" denoted in red color and "< −lower" denoted in blue color.

On the "higher− >" side, features such as Person_Capacity, Buying_Price, Size_of_Luggage, and Safety demonstrate positive contributions, suggesting that higher values or specific thresholds for these features positively impact the classification as "Vgood." On the "< −lower" side, the feature Maintenance_Price exhibits a negative contribution, indicating that lower values contribute to the classification as "Vgood."

By visualizing the feature contributions in the force plot, the SHAP explanation provides insights into how each feature influences the prediction of "Vgood" for the given data point.

Upon analyzing the explanations provided by XOPSIS, LIME, and SHAP for the same data point, it becomes evident that all three methods consistently identify the key features contributing to the prediction of "Vgood" as the acceptability level for the car.

According to XOPSIS, the features Buying_Price, Person_Capacity, and No_of_Doors positively contribute to the classification, while Maintenance_Price, Size_of_Luggage, and Safety have negative contributions.

Similarly, LIME highlights the importance of Safety, Buying_Price, and Person_Capacity as positive contributors to the prediction of "Vgood." On the other hand, Size_of_Luggage and Maintenance_Price are identified as negative contributors.

In line with the other methods, SHAP indicates that Person_Capacity, Buying_Price, Size_of_Luggage, Safety and No_of_Doors positively impact the classification of "Vgood,". Maintenance_Price is identified as a negative contributor.

Overall, the consistency in the identified features across XOPSIS, LIME, and SHAP explanations reinforces the significance of Buying_Price, Person_Capacity, and Size_of_Luggage as influential factors in predicting the "Vgood" acceptability level for the given data point. The alignment in feature importance enhances our understanding of the model's decision-making process and strengthens the reliability and interpretability of the predictions.

## VI. Conclusion

In conclusion, this research presents XOPSIS, a novel explainable AI (XAI) method, and evaluates its performance across multiple datasets, including maternal health, breast cancer, the benchmark Iris dataset, and the Car Acceptability dataset. XOPSIS demonstrates its effectiveness in providing interpretable insights into predictions, particularly in maternal health risk prediction, iris species prediction, and car acceptability prediction.

Our analysis using XOPSIS consistently highlighted the importance of specific features in predicting maternal health risks and breast cancer diagnoses. These findings were in line with previous research and demonstrated the ability of XOPSIS to uncover meaningful insights. The positive and negative feature contributions identified by XOPSIS provided valuable information about the factors influencing the predictions.

Furthermore, our research revealed the convergence of results between XOPSIS and established explainable AI methods such as LIME and SHAP. In most cases, we observed exact similar explanations in terms of the positive and negative contributing features for XOPSIS and LIME. SHAP also showed similar findings, providing additional support to our findings. This convergence not only validated the reliability and interpretability of XOPSIS but also emphasized the consistency in the importance of features across different explainable AI techniques.

While the evaluation primarily focused on maternal health and breast cancer datasets, the successful application of XOPSIS to the benchmark Iris dataset and the Car Acceptability dataset suggests its potential for broader applicability across diverse domains. This versatility and effectiveness of XOPSIS as an XAI method underscore its value in providing interpretable insights.

The availability of limited data for the maternal health dataset may have influenced the generalizability of our findings. Future studies should consider larger and more diverse datasets to validate and enhance the robustness of XOPSIS in different healthcare contexts.

Our research has significant implications for maternal health care and the advancement of the field of explainable AI. By providing interpretable insights into maternal health risk prediction, XOPSIS can aid healthcare professionals in making informed decisions and implementing targeted interventions. The consistent and robust findings from our study, particularly the similarities observed with LIME, underscore the importance of utilizing explainable AI methods in healthcare research.

In summary, our research introduces and evaluates the novel explainable AI method, XOPSIS, showcasing its effectiveness in providing interpretable insights into predictions across multiple datasets. The convergence of results with established methods, such as LIME, and its potential for broader applicability highlight the effectiveness and versatility of XOPSIS in providing interpretable insights. These findings contribute to the advancement of the field of explainable AI and its application in various domains, including maternal health care, breast cancer diagnosis, and beyond.

## REFERENCES


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

[2] Guidotti, Riccardo, et al. "A survey of methods for explaining black box models." ACM computing surveys (CSUR) 51.5 (2018): 1-42.

[3] Lundberg, Scott M., Gabriel G. Erion, and Su-In Lee. "Consistent individualized feature attribution for tree ensembles." arXiv preprint arXiv:1802.03888 (2018).

[4] Craven, Mark, and Jude Shavlik. "Extracting tree-structured representations of trained networks." Advances in neural information processing systems 8 (1995).

[5] Baehrens, David, et al. "How to explain individual classification decisions." The Journal of Machine Learning Research 11 (2010): 1803-1831.

[6] Chen, Jianbo, et al. "Learning to explain: An information-theoretic perspective on model interpretation." International conference on machine learning. PMLR, 2018.

[7] Zafar, Muhammad Rehman, and Naimul Khan. "Deterministic local interpretable model-agnostic explanations for stable explainability." Machine Learning and Knowledge Extraction 3.3 (2021): 525-541.

[8] Zhao, Xingyu, et al. "Baylime: Bayesian local interpretable model-agnostic explanations." Uncertainty in artificial intelligence. PMLR, 2021.

[9] Nohara, Yasunobu, et al. "Explanation of machine learning models using improved shapley additive explanation." Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 2019.

[10] Hastie, Trevor, et al. The elements of statistical learning: data mining, inference, and prediction. Vol. 2. New York: springer, 2009.

[11] Yeh, Chung-Hsing. "A problem-based selection of multi-attribute decision-making methods." International Transactions in Operational Research 9.2 (2002): 169-181.

[12] Zeng, Shouzhen, Jianping Chen, and Xingsen Li. "A hybrid method for Pythagorean fuzzy multiple-criteria decision making." International Journal of Information Technology Decision Making 15.02 (2016): 403-422.

[13] Zhang, Huiyuan, Guiwu Wei, and Cun Wei. "TOPSIS method for spherical fuzzy MAGDM based on cumulative prospect theory and combined weights and its application to residential location." Journal of Intelligent Fuzzy Systems 42.3 (2022): 1367-1380.

[14] Zavadskas, Edmundas Kazimieras, and Zenonas Turskis. "Multiple criteria decision making (MCDM) methods in economics: an overview." Technological and economic development of economy 17.2 (2011): 397-427.

[15] Černevičienė, Jurgita, and Audrius Kabašinskas. "Review of multi-criteria decision-making methods in finance using explainable artificial intelligence." Frontiers in artificial intelligence 5 (2022): 827584.

[16] Galankashi, Masoud Rahiminezhad, et al. "Hospital Selection Problem: An Integrated Analytic Hierarchy Process (AHP) and Fuzzy-TOPSIS Approach." Proceedings of the 12th Annual International Conference on Industrial Engineering and Operations Management (IEOM). 2022.

[17] Olson, David L. "Comparison of weights in TOPSIS models." Mathematical and Computer Modelling 40.7-8 (2004): 721-727.

[18] Behzadian, Majid, et al. "A state-of the-art survey of TOPSIS applications." Expert Systems with applications 39.17 (2012): 13051-13069.

[19] Lai, Young-Jou, Ting-Yun Liu, and Ching-Lai Hwang. "Topsis for MODM." European journal of operational research 76.3 (1994): 486-500.

[20] Stević, Željko, et al. "Sustainable supplier selection in healthcare industries using a new MCDM method: Measurement of alternatives and ranking according to COmpromise solution (MARCOS)." Computers industrial engineering 140 (2020): 106231.

[21] Liu, Shulin. "Research on the teaching quality evaluation of physical education with intuitionistic fuzzy TOPSIS method." Journal of Intelligent Fuzzy Systems 40.5 (2021): 9227-9236.

[22] Nădăban, Sorin, Simona Dzitac, and Ioan Dzitac. "Fuzzy TOPSIS: a general view." Procedia computer science 91 (2016): 823-831.

[23] Seçme, Neşe Yalçın, Ali Bayrakdaroğlu, and Cengiz Kahraman. "Fuzzy performance evaluation in Turkish banking sector using analytic hierarchy process and TOPSIS." Expert systems with applications 36.9 (2009): 11699-11709.

[24] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.

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

[26] Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences." International conference on machine learning. PMLR, 2017.

[27] Lundberg, Scott M., et al. "From local explanations to global understanding with explainable AI for trees." Nature machine intelligence 2.1 (2020): 56-67.

[28] Friedman, Jerome H., and Bogdan E. Popescu. "Predictive learning via rule ensembles." The annals of applied statistics (2008): 916-954.

[29] Lundberg, Scott M., and Su-In Lee. "Consistent feature attribution for tree ensembles." arXiv preprint arXiv:1706.06060 (2017).

[30] Doshi-Velez, Finale, and Been Kim. "Towards a rigorous science of interpretable machine learning." arXiv preprint arXiv:1702.08608 (2017).

[31] Štrumbelj, Erik, and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions." Knowledge and information systems 41 (2014): 647-665.

[32] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations." Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018.

[33] Främling, Kary. "Decision theory meets explainable ai." International workshop on explainable, transparent autonomous agents and multi-agent systems. Cham: Springer International Publishing, 2020.

[34] Rasmussen, Svein, et al. "Predicting preeclampsia in the second pregnancy from low birth weight in the first pregnancy." Obstetrics Gynecology 96.5 (2000): 696-700.

[35] Sibai, Baha, Gus Dekker, and Michael Kupferminc. "Pre-eclampsia." The Lancet 365.9461 (2005): 785-799.

[36] Ananth, Cande V., Katherine M. Keyes, and Ronald J. Wapner. "Pre-eclampsia rates in the United States, 1980-2010: age-period-cohort analysis." Bmj 347 (2013).

[37] Lisonkova, Sarka, and K. S. Joseph. "Incidence of preeclampsia: risk factors and outcomes associated with early-versus late-onset disease." American journal of obstetrics and gynecology 209.6 (2013): 544-e1.

[38] A. Rahman and M. G. Rabiul Alam, "Explainable AI based Maternal Health Risk Prediction using Machine Learning and Deep Learning," 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2023, pp. 0013-0018, doi: 10.1109/AIIoT58121.2023.10174540.

[39] Ahmed, Marzia, and Mohammod Abul Kashem. "IoT based risk level prediction model for maternal health care in the context of Bangladesh." 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE, 2020.

[40] Caruana, Rich, et al. "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission." Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015.

[41] Swain, Madhusmita, et al. "An approach for iris plant classification using neural network." International Journal on Soft Computing 3.1 (2012): 79.

[42] Pachipala, Yellamma, et al. "Iris Flower Classification by using Random Forest in AWS." 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2022.

[43] NAEEM, SAMREEN, and AQIB ALI. "Breast Cancer Diagnosis Using Machine Learning Techniques." (2022).

[44] Zuluaga-Gomez, Juan. "Breast Cancer Diagnosis Using Machine Learning Techniques." arXiv preprint arXiv:2305.02482 (2023).

[45] Uzun Ozsahin, Dilber, et al. "The Systematic Review of Artificial Intelligence Applications in Breast Cancer Diagnosis." Diagnostics 13.1 (2022): 45.
[46] Akay, Mehmet Fatih. "Support vector machines combined with feature selection for breast cancer diagnosis." Expert systems with applications 36.2 (2009): 3240-3247.
[47] Al-Mubayyed, Osama M., Bassem S. Abu-Nasser, and Samy S. Abu-Naser. "Predicting Overall Car Performance Using Artificial Neural Network." (2019).
[48] Al-Mobayed, Awni Ahmed, et al. "Artificial Neural Network for Predicting Car Performance Using JNN." (2020).
[49] Dataset Available From: Maternal Health Risk Dataset.
[50] Dataset Available From: IRIS Species Dataset.
[51] Dataset Available From: Breast Cancer Wisconsin (Diagnostic) Dataset.
[52] Dataset Available From: Car Acceptability Classification Dataset.

**Anika Rahman** Biography text here.

**Md. Golam Rabiul Alam** Biography text here.