Saint Petersburg State University

Department of mathematical game theory and statistical decisions

Zhang Peichi

Master's thesis

Research on Diabetes Prediction Model Based on Machine Learning from the Perspective of Cooperative Game Theory

Specialization 01.04.02

Applied Mathematics and Informatics

Master's Program Game Theory and Operations Research

Research advisor,
L.A.Petrosyan
Dean of the Faculty of Applied Mathematics
- Control Processes
Head of the Department of Mathematical Theory of Games and Statistical Solutions

Reviewer,
Nikitina Natalia Nikolaevna

Saint Petersburg

2024

Contents

1 Abstract **3**

2 Introduction **4**

3 Diabetes data set **7**

3.1 Description of the dataset 7

3.2 Data preprocessing 8

4 Medical detection model **10**

4.1 One-Class SVM algorithm 10

4.2 Prediction Model Results and Analysis 14

4.3 Experimental Results and Analysis of the Prediction Model 14

4.3.1 Evaluation Metrics 14

4.3.2 Experimental Results 16

5 Medical detection model interpretability methods **18**

5.1 SHAP algorithm 18

5.1.1 Shapley values in cooperative games 18

5.1.2 SHAP algorithm in machine learning 19

5.1.3 Feature function of SHAP algorithm 20

5.1.4 Compute SHAP values. 23

5.2 τ algorithm 25

5.2.1 τ-values in cooperative games 25

5.2.2 τ-values algorithm 26

6 Diabetes detection model interpretation methods **28**

6.1 SHAP model and result analysis 28

6.1.1 Visualization of Predictions 28
6.1.2 SHAP Feature Importance 29
6.1.3 SHAP Summary Plot 31
6.2 τ-values model and result analysis 33
6.2.1 τ-values Feature Importance 33
6.3 Model Comparison 34

7 Predictive model based on XGBoost **36**
7.1 XGBoost 36
7.2 XGBoost prediction model 39
7.2.1 A prediction model based on Shapley 39
7.2.2 A prediction model based on τ-values 41
7.2.3 Interpretable Model Comparison 42

8 Conclusion **44**

References **46**

9 appendix **50**

1 Abstract

In the field of artificial intelligence, the interpretability of models has always been a focal point for researchers and engineers. With the widespread application of machine learning models across various domains, understanding the decision-making process of models has become an important topic. This paper utilizes machine learning to establish a medical detection model and conducts interpretability research on this model. The main algorithms used are the SHAP algorithm and the τ-algorithm, exploring the performance of different cooperative game methods in interpretability research under the same model and data. Furthermore, the medical detection model is reconstructed using Shapley values and τ-values based on the XGBoost model, analyzing and comparing the strengths and weaknesses of the two different methods. Based on this analysis, improvements are made to the model to enhance the credibility of the prediction results. The τ-algorithm used in this paper is novel in the fields of machine learning and detection models.

2 Introduction

Explainable Artificial Intelligence (XAI) is an AI system built on the principles of Artificial Intelligence (AI) that is understandable and trusted by users. It has vast potential applications in the field of medical diagnostics[5]. Utilizing artificial intelligence technology to perform deep learning on medical data and establish medical detection models assists doctors in clinical diagnosis and treatment. This demonstrates how AI empowers the medical industry, helping to improve the efficiency of doctors' work, alleviate work pressure, and mitigate the current shortage of medical resources. Currently, numerous scholars are devoted to researching how to utilize artificial intelligence technology for medical diagnosis. However, due to the "black box" nature of most AI systems, their behavior is opaque, and doctors and patients cannot understand their decision-making mechanisms. This lack of transparency reduces the credibility of diagnostic results and hinders the application of AI in clinical practice. Therefore, possessing predictive and explanatory capabilities is an inevitable direction for the development of medical detection models.

In machine learning, the interpretability of predictive models, as the most common kind of models, is crucial, determining the degree of trust in the model. shapley value in cooperative game is a commonly used method to explain the contribution degree of model features[1]. However, in the case of high-dimensional data, shapley huge calculation amount is time-consuming and labor-intensive, and the conventional dimensionality reduction method may lead to some deviation from the expected results. Use the τ-value in the cooperative game to replace the shapley value in the model. The calculation amount is much smaller than the shapley value, and

it can perfectly show the influence of different characteristics on the results of the prediction model. This paper investigates three aspects related to explanatory medical diagnostic models:

- Firstly, it is necessary to establish a medical detection model, which is trained on medical data using machine learning algorithms. The core algorithm used in this model is the One-Class SVM algorithm, and the medical data utilized is diabetes instance data. This model provides a basic framework for medical detection and highlights the necessity for model interpretation. Additionally, it offers essential model and data support for the establishment of an explanatory model.

- By using the Shapley method to construct the SHAP algorithm, this study investigates the contributions of features to the model and computes the feature rankings that influence the prediction results.This process is called feature importance analysis, and the result feature importance diagram and feature summary diagram can be obtained. According to the feature importance diagram, we can intuitively observe each feature in the prediction process and the impact of the feature on the overall prediction result, and arrange the feature in descending order, which can be more clear and direct observation. The feature summary diagram shows the influence and distribution of the most total prediction results of each feature instance, which is convenient for us to observe the positive or negative promoting effect of the feature on the result. Although the SHAP algorithm has strong interpretability and can be used to explain the results of prediction models, medical data typically involve high-dimensional data with multiple features. The computational complexity of the SHAP

algorithm increases exponentially with the increase of features, resulting in significant time and computational resource consumption during the calculation process. Moreover, it may fail to compute results when there are too many data features, limiting its application in medical detection. This paper employs the τ-algorithm to further explore the interpretability of medical detection models. Based on cooperative game theory, the τ-value method is used to calculate contributions. Compared to the SHAP method, it performs better on high-dimensional data. In the prediction model, all features are regarded as coalition N, different features form sub-coalition S, $S \in N$, and different prediction results can be obtained. The prediction results of features composed by the sub-coalition and the prediction results of coalition N can be calculated to obtain the income v(S) of each coalition. For v(S) we can Compute the upper bound vector and lower bound vector for each feature, and Calculate these two values to get the τ value of a feature.

- The research on the interpretability of medical models through the two aforementioned approaches has been completed. In the final stage, we evaluate the SHAP algorithm and the τ-algorithm, comparing their performance in predicting models. Both Shapley values and τ-values can reflect the contributions of different features to the prediction results. By re-establishing the prediction model based on the XGBoost model and incorporating instances for calculation of accuracy, we can compare the advantages and disadvantages of these two algorithms.

3 Diabetes data set

3.1 Description of the dataset

In this article, the data set for predictive model training needs to be prepared in advance. The selected data set is from the open source data set on the Internet, the diabetes data set. The dataset contains 9 features and 768 samples, among which the features with 0 or 1 label are diagnostic results. A total of 268 samples with positive diabetes diagnosis (label 1) and 500 samples with negative diabetes diagnosis (label 0) are included. Based on these data, the model can be trained and a prediction model can be established.

Attributes	Attribute Description
Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure	Diastolic blood pressure (mm Hg)
SkinThickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	2-Hour serum insulin (mu U/ml)
DiabetesPedigreeFunction	Body mass index (weight in kg/(height in m)^2)
Age	Age (years)
Outcome	Class variable (0 or 1) class value 1 is interpreted as "tested positive for diabetes"

Table 3.1: The attributes of the dataset

Variables of the study were controlled as much as possible in this dataset, and all sample instances were from women of Pima Indian descent at least 21 years of age who were medically diagnosed with diabetes, with eight medical predictors and one outcome.

3.2 Data preprocessing

To train a model using the diabetes dataset, we first need to preprocess the data. We start by detecting outliers in the dataset to observe any abnormal patterns.

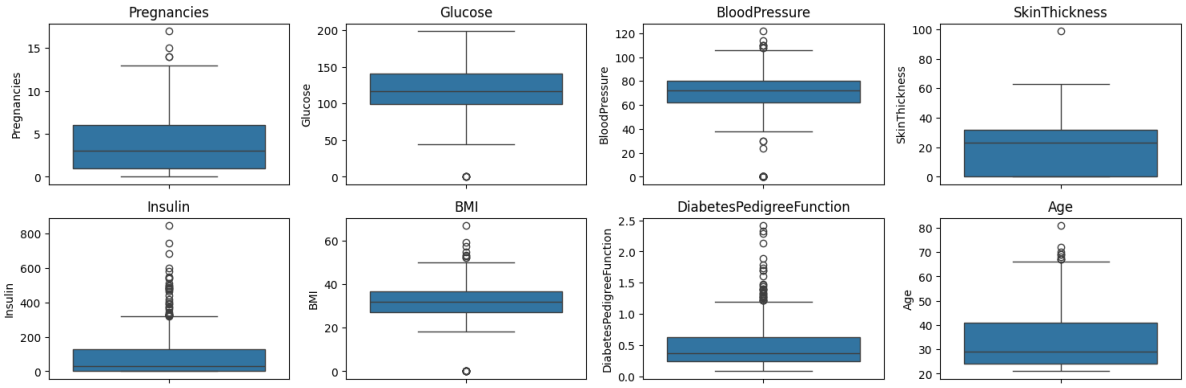


Figure 3.1: Data outlier detection

Through Figure 3.1, it can be observed that the attributes Insulin, BloodPressure, BMI, DiabetesPedigreeFunction, and Age in the dataset have a large number of outliers. However, in medical disease detection, outliers are often used as diagnostic criteria and do not need to be processed. However, it is observed that there are a large number of values in the data set that appear as outliers and are equal to 0. For all instances, these attributes should not be 0, so they need to be processed for missing values. In order to make the experiment more rigorous, we classified the data according to the prediction result label (0 or 1) of the data, and then obtained the upper and lower limits of the distribution of the feature data according to the box diagram. Generate random numbers to fill in missing values of data for missing value processing.

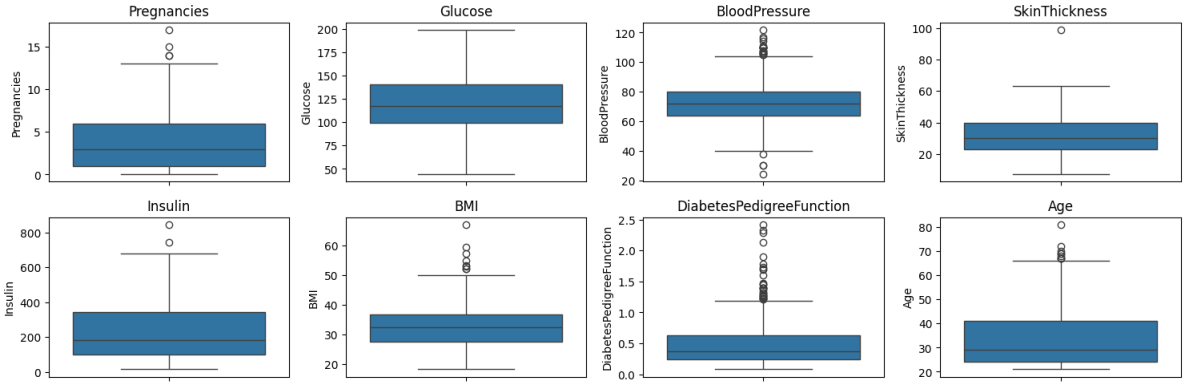


Figure 3.2: The dataset after handling missing values

By observing Figure 3.2, there are still a large number of outliers in the attributes. However, taking Age as an example, an excessively high age does not necessarily indicate data error, so we do not need to process these "numeric" values. Instead, we performed missing value processing for all those values that were 0, making the dataset more accurate.

4 Medical detection model

4.1 One-Class SVM algorithm

There are many commonly used anomaly detection algorithms, such as Isolation Forest, Z-score, and LOF. However, due to the large number of anomalies in the diabetes dataset and the difficulty in labeling them, using these algorithms to train a model makes it hard to ensure accuracy. In the medical field, the One-Class SVM algorithm has many advantages. One-Class SVM is an unsupervised learning algorithm that does not require labeled anomalous data for training. Since anomalous data is often difficult to obtain or incomplete, unsupervised learning is particularly useful.

One-Class SVM can be applied to various types of data, including high-dimensional and nonlinear data. In medical detection, the complexity and diversity of medical data make this flexibility especially important. One-Class SVM is specifically designed for anomaly detection, effectively identifying anomalies in medical data. This helps medical professionals detect potential health issues or diseases early. Compared to other machine learning algorithms, the One-Class SVM model is relatively simple, easy to understand, and interpret. This allows medical professionals to better understand the model's decision-making process and make adjustments and optimizations as needed. Therefore, choosing to use the One-Class SVM algorithm to establish a medical detection model is a suitable approach.

The working principle of One-Class SVM is as follows:

- Data mapping: Normal data is mapped to the three-dimensional feature space, so that normal data points can be distributed in the sphere of three-dimensional space, and the result is obtained by judging whether the data is in the sphere, which is normal, not abnormal.

This sphere serves as a distinguishing surface called a hyperplane.

- Finding the Optimal Hyperplane: By maximizing the margin between the hyperplane and the normal data, find an optimal separating hyperplane that keeps the abnormal points as far away from it as possible. This means the decision boundary should be as far away from the normal data points as possible.

- Anomaly Detection: For new data points, determine whether they are anomalies by calculating their distance from the hyperplane. Data points with larger distances are more likely to be anomalies.

One Class SVM also belongs to the category of Support Vector Machines (SVM). The main problem in detecting anomalies is finding the separating hyperplane and identifying the support vectors. We use the Support Vector Data Description (SVDD) algorithm, which is similar to the One Class SVM algorithm. It treats all non-anomalous samples as the positive class and uses a hypersphere instead of a hyperplane for separation. This algorithm obtains a spherical boundary around the data in the feature space and aims to minimize the volume of this hypersphere, thereby minimizing the impact of anomalous data points.

In the ONE-class SVM algorithm, suppose to generate a hypersphere, the center origin is o, the hypersphere radius is $r(r>)$ 0, The volume $V(r)$ of the hypersphere is required to be minimized in order to be more accurate in the prediction process, and the origin can be obtained by a linear combination expressed as a support vector. The ONE-class SVM algorithm is similar to the traditional vector machine, the distance from the mapped feature data point to the origin needs to be less than the radius. The introduction of non-negative relaxation variable ζ_i allows classification

errors within a certain range and assigns a penalty factor C. Therefore, the optimization problem for this problem is:

$$\underbrace{\min}_{\mathrm{r,o}} V(r) + C\sum_{i=1}^{m}\zeta_i$$

$$\|x_i - o\|_2 \leq r + \xi_i, i = 1, 2, 3...m$$

$$\xi_i \geq 0, i = 1, 2, ...m$$

By solving the Lagrange duality problem, we can identify a new data point z Whether z is inside the hypersphere. If the distance from z to the center of the circle is less than or equal to the radius r, then it is not an outlier. If it is outside the hypersphere, it is considered an outlier.

The kernel functions commonly used in One-Class SVM algorithm are Linear kernel function, polynomial kernel function, Gaussian radial basis (RBF) kernel function and *Sigmoid* kernel function, this paper adopts Gaussian radial basis kernel function:

$$K(x, y) = \exp\{-\frac{\|x - y\|^2}{\sigma^2}\}$$

The parameters of One-Class SVM determine the learning ability and generalization ability of the algorithm. For One-Class SVM with RBF kernel function, its parameters include penalty parameter C and kernel parameter σ. The penalty parameter C is the compromise between structural risk and sample error, and the larger the value, the smaller the allowable error; The nuclear parameter σ is related to the range and width of the input space of the learning sample. The larger the input space of the sample, the larger the value of σ.

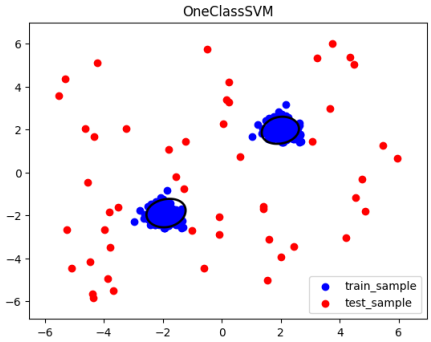


Figure 4.1: Example Diagram: Anomaly Detection Using One-Class SVM

This example demonstrates how to use OneClassSVM for anomaly detection. First, training samples X_{train} are generated using randomly created data. Then, a OneClassSVM model clf is created and trained. Next, test samples X_{test} are generated, and the trained model is used to predict anomalies in the samples. Finally, by plotting the training samples, test samples, and anomaly boundaries, the results of the anomaly detection are illustrated.

4.2 Prediction Model Results and Analysis

During the model training phase, the dataset is first divided into a training set and a test set in a ratio of 8:2. The organized data is shown in the table below.

Diagnostic Results	Training Set	Testing Set	Total
Diseased	214	54	268
Not Diseased	400	100	500

Table 4.1: Distribution of the Data Set for the Prediction Model

4.3 Experimental Results and Analysis of the Prediction Model

4.3.1 Evaluation Metrics

The model trained based on the One-Class SVM algorithm is an anomaly detection model, which is used as a prediction model in this study. To evaluate the performance of the prediction model, this study selected confusion matrix, accuracy, ROC curve, and AUC as evaluation metrics[4].

The main purpose of the confusion matrix is to prevent misleading results due to an unbalanced distribution of the sample data set. For example, if the samples in the data set 95% are positive and 5% are negative, then the model that predicts all the samples are positive will achieve 95% accuracy, but the model's recognition rate for negative samples will be 0. In order to avoid the model can only predict the results of the positive results, the confusion matrix of the model is established to explain the accuracy of the model. The confusion matrix shows TP(true positive), FN(false negative), FP(false positive), and TN(true negative). Accuracy (Acc), precision (P), recall (R), and F1 scores can be calculated using values from the confusion matrix.

	Actual Positive	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Table 4.2: Confusion Matrix

Then the accuracy can be expressed as:

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

Precision can be expressed as the formula:

$$P = \frac{TP}{TP + FP} \tag{4.2}$$

ROC curve and AUC value can avoid the situation where the above metrics cannot guarantee objectivity due to imbalanced samples. The ROC curve is plotted with FPR (False Positive Rate/specificity) on the x-axis and TPR (True Positive Rate/sensitivity) on the y-axis, considering the model classification results under different classification thresholds. Since the shape of the ROC curve is not easy to quantify and compare, the AUC value, which represents the area under the ROC curve, is used as an indicator that can be intuitively compared. The TPR expression is consistent with the recall rate, just expressed in a different form. FPR can be expressed as formula (4.2) and formula (4.3).

$$FPR = \frac{FP}{TN + FP} \tag{4.3}$$

FPR represents the model's false positive rate, while TPR represents the coverage rate of the predictions. The higher the TPR and the lower the FPR, the better the model's performance. This is reflected on the ROC curve as a steeper curve that is closer to the upper left corner, and a larger AUC value.

4.3.2 Experimental Results

Figure 4.2 illustrates the confusion matrix of the One-Class SVM model on the test set.

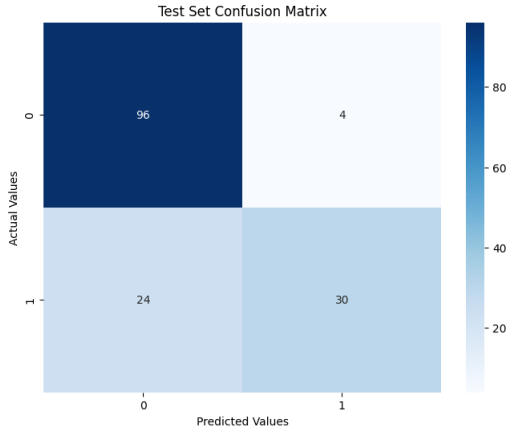


Figure 4.2: The confusion matrix of the predicted results.

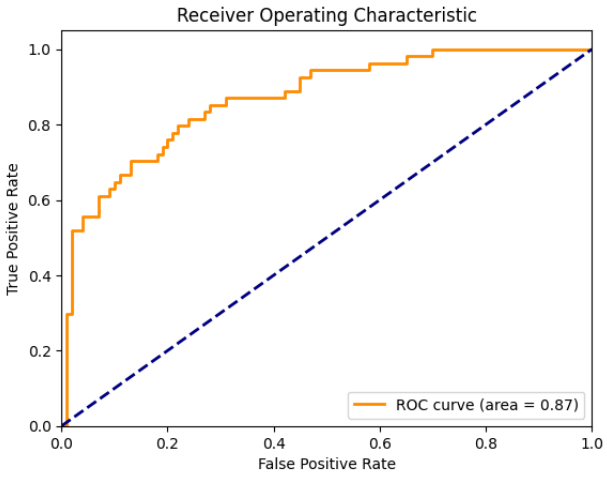


Figure 4.3: The experimental ROC curve graph.

Analyze the confusion matrix and ROC curve above veals: First of all, by looking at the confusion matrix, we can see that the diabetes prediction model can predict the outcome of each instance to be positive or negative, and the prediction function is perfect. Can be calculated The prediction model was 82% accurate. For a predictive model, this Accuracy is predictive, higher than most medical predictive model models, and can be experimented with as a target model for interpreting models. The validity of a class of support vector machine algorithm is verified To the prediction model. In addition, a class of support vector machine algorithm is simple and easy to understand, convenient for users and doctors Understanding enhances the interpretability of the model.

Upon analyzing the ROC curve, it is observed that the coupling degree is high, and the model's AUC value is close to 1. (AUC is defined as the area under the ROC curve, with a value not exceeding 1. However, the closer it is to 1, the better the model's predictive ability.) This also suggests that using the One-Class SVM algorithm to train the predictive model is beneficial, both for predictive accuracy and improving interpretability in subsequent analyses.

5 Medical detection model interpretability methods

5.1 SHAP algorithm

5.1.1 Shapley values in cooperative games

The main algorithm of SHAP model is shapley value, which is an algorithm in the field of cooperative game. It was proposed by shapley in 1953[13]. This is a way to distribute revenue based on player contributions. Different players form different alliances to get different benefits, and each player's contribution can be obtained by calculating the benefits of these alliances. shapley's formula is given in 5.1:

$$\varphi_i = \sum_{S|i \in S \subseteq N} \frac{(|S|-1)!(|N|-1)!}{|N|!}[v(S) - v(S \setminus \{i\})] \tag{5.1}$$

Here, φ_i represents the Shapley value for player i, reflecting the player's contribution to the total payoff of the coalition. In a prediction model, this can be interpreted as the influence of attribute i on the prediction outcome. N is the set of all players, which in the context of a prediction model can be understood as the set of all attributes. $S \in N$ is a coalition of players, which in a prediction model can be interpreted as different subsets of attributes. $V(S)$ is the characteristic function of coalition S, reflecting the total payoff when the players form coalition S. In a prediction model, this can be interpreted as the prediction result obtained by the prediction model when using the coalition of attributes S.

Before obtaining the Shapley value for each feature, it is necessary to calculate $V(S)$ for each $S \in N$ coalition. This is also a crucial step in calculating the Shapley value, which we will explain in the following section.

In interpretable AI, we use Shapley values to measure the contribution of each player (feature). By using Shapley values, we can obtain the contribution of each feature to the prediction results of a specific solution (local interpretation) as well as the overall system, the global prediction results (global interpretation).

When interpreting machine learning predictions using Shapley values, Total expenditure represents what the prediction model predicts for a single sample. In the data set, the player is the eigenvalue of the corresponding feature of each sample, and the payoff is the difference between the predicted result of the sub-coalition of the sample feature and the average predicted result in all cases, and we will get the alliance payoff of the sub-coalition of the sample feature.

5.1.2 SHAP algorithm in machine learning

SHAP algorithm is a method for interpreting a model based on the Shapley value in game theory. It is a post-hoc explanation framework that can compute the importance value (Shapley value) for each feature variable in each sample, achieving the effect of explanation. Shapley value was originally used to solve the problem of allocating contributions to total income among participants in cooperative games. The SHAP algorithm considers the contribution of each feature value as a "fair" distribution, ensuring that each feature value contributes its fair share to the model output.

Coalitional Contribution represents the influence of each feature on the model prediction. For the SHAP algorithm, the general steps for computing marginal contributions are as follows:

- For each sample to be explained, determine the model's baseline prediction, which is typically the average prediction value of the model for the entire training set.
- Replace each feature in the sample to be explained with a virtual value, such as setting it to the mean value, and then predict the model again.
- Calculate the difference between the model prediction values after replacing each feature and the baseline prediction value, which represents the marginal contribution of that feature.

In this way, the influence of each feature on the model prediction can be determined. In the SHAP algorithm, marginal contributions are used to compute the Shapley values of each feature, thereby obtaining the relative importance of features.

5.1.3 Feature function of SHAP algorithm

The key to SHAP is understanding how each feature affects predictions.It's easy to calculate the contribution of each feature in linear models. Here is an example of a data instance's prediction from a linear model:

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \tag{5.2}$$

Where x is the instance for which we want to compute contributions, each $x_i (i = 1, \cdots, n)$ is a feature value of the instance, and β_i is the weight corresponding to feature i.

The contribution of the i-th feature to the prediction $\hat{f}(x)$ is defined as:

$$\phi_i = \beta_i x_i - E(\beta_i X_i) = \beta_i x_i - \beta_i E(X_i) \tag{5.3}$$

Where $E(\beta_i X_i)$ is the average causal effect of feature j, the contribution is the difference between the feature effect and the average effect. Now that we know the contribution of each feature to the prediction, if we sum up the contributions of all features of an instance, the result is as follows:

$$E(\beta_j X_j) = \sum_{i=1}^{n} \beta_j x_{ij} p(x_{ij}) \tag{5.4}$$

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} \phi_i(\hat{f}) = \sum_{i=1}^{n} (\beta_i x_i - E(\beta_i X_i)) \\ &= (\beta_0 + \textstyle\sum_{i=1}^{n} \beta_i X_i) - (\beta_0 + \sum_{i=1}^{n} E(\beta_i X_i)) \\ &= \hat{f}(x) - E(\hat{f}(X)) \end{aligned} \tag{5.5}$$

For equation (5.4), β_j is the coefficient (weight) of feature j, x_{ij} is the value of feature j for the i-th sample, $p(x_{ij})$ is the probability (weight) of the value x_{ij} for feature j, and n is the number of samples.

That means the sum of contributions from data point x equals the prediction minus the average prediction. Since we don't have similar weights in other non-linear models (like ensemble models), we need a different solution to obtain the feature contributions of individual predictions of machine learning models, using Shapley values from cooperative game theory.

The Shapley value of each feature is the contribution of that feature to the prediction, obtained by summing the weighted sum of contributions across all possible combinations of feature values:

$$\phi_i(\nu al) = \sum_{S \subseteq \{x_1,\ldots,x_n\} \setminus \{x_i\}} \frac{(|S|-1)!(|n|-1)!}{|n|!} (\nu al(S \bigcup \{x_i\}) - \nu al(S))$$

Where S is the feature subcoalition of different features in the prediction model (cooperative game colation), and x is The vector represents the eigenvalue of the sample to be interpreted, where N is the sample size. Where, $valx(S)$ is the prediction result of the colation on the sub-colation S. By differentiating the feature x that is not included in the sub-colation, it can be marginalized, so that the prediction colation features have a more significant impact on the prediction result.

$$\nu al_x(S) = \int \hat{f}(x_1, \cdots, x_n) dN_{x \notin S} - E_X(\hat{f}(X)) \tag{5.6}$$

In fact, multiple integrations are performed for each feature not included. For example, if a machine learning model uses four features $x1, x2, x3$ and $x4$, we estimate the prediction of coalition S composed of feature values $x1$ and $x3$.

$$\nu al_x(S) = \nu al_x(\{x_1, x_3\}) = \int_R \int_R \hat{f}(x_1, X_2, x_3, X_4) dN_{X_2, X_4} - E_X(\hat{f}(X))$$

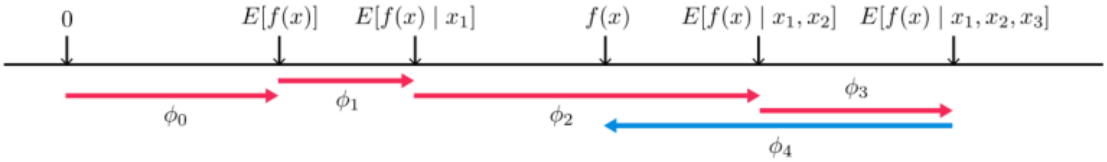
Feature value is the numerical or categorical value of an instance feature; Shapley value is the contribution of a feature to the prediction; value function is the expenditure function of the coalition.

The feature value (feature value) is the numerical value or category value of the instance feature; Shapley value is the contribution of features to prediction. The value function is the expenditure function of the feature values of the coalition.

The Shapley value thus calculated is the only attribution method that satisfies the four attributes of Efficiency, Symmetry, Dummy, and Additivity,These four attributes can be collectively referred to as the definition of the shapley value.

5.1.4 Compute SHAP values.

To compute SHAP values, we define $f_x(S) = E[f(x)|x_s]$, where S is the subset of possible input features (coalitions mentioned in Shapley values), and $E[f(x)|x_s]$ is the conditional expectation value of the subset S of input features (*val* function mentioned in Shapley values). The following figure explains how we get the prediction from $E[f(x)]$.



Through the above figure, it can be seen that SHAP values assign the attribution value of each feature to the expected change in model prediction when that feature is adjusted[2], explaining the model f prediction for sample $\{x_1 = a_1, x_2 = a_2, x_3 = a_3, x_4 = a_4\}$ as the sum of the influences of each feature's conditional expectation . This figure only displays the case of single sorting. So, the explanation process of the above figure is:

- When S is an empty set, $\phi_0 = f_x(\emptyset) = E[f(x)]$,$E[f(x)]$ is the expectation of the model prediction, which can be approximated by the average of the model predictions on the training samples.
- Feature x_i is sequentially added to S, $\phi_1 = f_x(\{x_1\}) - f_x(\emptyset) = E[f(x)|x_1] - E[f(x)]$ is the difference between the model prediction expectation when $\{x_1 = a_1\}$ is included and the model prediction expectation.
- Then, when feature is sequentially added to S, $\phi_2 = f_x(\{x_1, x_2\}) - f_x(\{x_1\}) = E[f(x)|x_1, x_2] - E[f(x)|x_1]$,That is, the model prediction expectation when $\{x_1 = a_1, x_2 = a_2\}$ is included minus the model prediction expectation when $\{x_1 = a_1\}$ is included.

- ·····

- Until the last feature x_4 is added sequentially to S,$\phi_4 = f_x(\{x_1, x_2, x_3, x_4\}) - f_x(\{x_1, x_2, x_3\}) = E[f(x)|x_1, x_2, x_3, x_4] - E[f(x)|x_1, x_2, x_3]$, That is, the model prediction expectation when $\{x_1 = a_1, x_2 = a_2, x_3 = a_3, x_4 = a_4\}$ is included minus the model prediction expectation when $\{x_1 = a_1, x_2 = a_2, x_3 = a_3\}$ is included,At this point, f is the prediction value under the single sorting of four features, which is actually the prediction value of sample x.

However, in practice, when the model is nonlinear or the input features are not independent, SHAP values should calculate the weighted average of all possible feature orderings. SHAP combines these conditional expectations with the classic Shapley values from game theory into the attribution values of each feature ϕ_i, calculated according to the following formula.

$$\phi_i = \sum_{S \subseteq \{x_1, \cdots, x_n\} \setminus \{x_i\}} \frac{(\left|S\right| - 1)!(\left|n\right| - 1)!}{\left|n\right|!} (f_x(S \bigcup \{x_i\}) - f_x(S))$$

Where $\{x_1, \cdots, x_p\}$ is the set of all input features, p is the number of all input features, $\{x_1, \cdots, x_p\} \setminus \{x_j\}$ is the set of all possible subsets of input features excluding x_j, and $f_x(S)$ is the prediction of feature subset S. It can be seen that this formula is the same as the definition introduced earlier for Shapley values.

5.2 τ algorithm

5.2.1 τ-values in cooperative games

τ-value is also a method derived from cooperative game theory, introduced by Tijs in 1981 for imputations. The τ-value is essentially a coordinating value between the value vector for upper games and the value vector for lower games.

Let $v \in G$, for each $i \in N$and each $S \in 2^N, i \in S$, the marginal contribution of player i to the coalition S is $v(S) - v(S\backslash i)$. for each $i \in N$, $b_i^v = v(N) - v(N\backslash\{i\})$ represents the marginal contribution of player i to Major League N, which is the utopia payoff that player i can get in Major league. If player i wants to get more payment from major League, Then N people in other innings exclude people in innings i from major league N. So b_i^v is the players i in the countermeasures that can obtain the upper bound of the payment in $b^v = (b_1^v, \cdots, b_n^v) \in \mathbb{R}^N$ called countermeasure v value on the vector (upper vector).

Let $i \in N$,$S \in 2^N$ and $i \in S$, when every player in coalition S except i is paid Utopia, the remainder of player i is defined as follows.

Theorem 5.1. *Let $S \in 2^N\backslash\varnothing$,$i \in S$,coalition S player i ,The remaining $R(S,i)$ defined as*

$$R(S,i) = v(s) - \sum_{j \in S\backslash i} M_j(v).$$

It means that for the alliance value $v(S)$, if all players in alliance S except player i get the maximum payment value they can expect, then player i can get this residual value.

For each $i \in N$, the i-th component of the lower vector a^v, a_i^v be defined as

$$a_i^v = \max_{S:i \in S} R(S, i).$$

In major League N, an innings player i is justified in demanding the minimum payment he gets a_i^v, Let's call it the minimum reasonable payment of player i. It means that after every player in alliance s except i receives the maximum payout Utopia payment, player i is guaranteed to get as much of the remaining a_i^v as possible.

Theorem 5.2. *The definition of the τ-value $\tau(v)$ is:*

$$\tau(v) = \alpha a^v + (1 - \alpha) b^v,$$

Where $\alpha \in [0, 1]$ is uniquely determined by $\sum_{i \in N} \tau_i(v) = v(N)$.

5.2.2 τ-values algorithm

To improve model efficiency and reduce computational complexity, logistic regression algorithm is utilized to fit the model when computing τ-values. This approach saves a considerable amount of time and computational resources compared to training the SHAP model using data. For comparison, we simultaneously calculate coalition contributions using the same method as the SHAP model and then compute τ-values again, making the results more direct and convincing.

The general steps for calculating τ-values using logistic regression are as follows:

- Prepare data: Prepare a dataset containing features and target variables.

- Fit the model: Fit the dataset using a logistic regression model.
- Compute baseline value: Compute the average prediction value of the model for the entire training set as the baseline value.
- Compute marginal contributions: For each sample, replace the value of each feature with its mean value across the entire training set, and then re-predict the model. Calculate the difference between the model prediction value and the baseline value, which represents the marginal contribution of each feature.
- Compute τ-values: Calculate the τ-value for each feature using marginal contributions.
- Interpret results: Analyze the τ-values to understand the impact of each feature on the model predictions.

6 Diabetes detection model interpretation methods

6.1 SHAP model and result analysis

6.1.1 Visualization of Predictions

First, we use SHAP to explain the prediction of a single sample, where each feature value contributes to increasing or decreasing the prediction force. Predictions start from a baseline, which is the average of all predictions, and each Shapley value is an arrow indicating an increase (positive) or decrease (negative) in prediction.

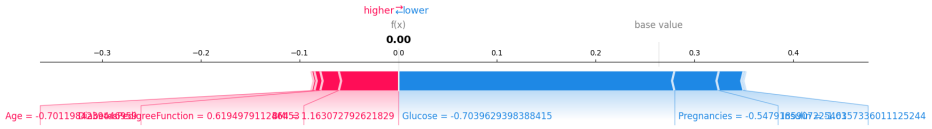


Figure 6.1: Visualization of individual samples (force diagram).

The SHAP values interpret the diabetes prediction probability for this particular instance as follows: the woman's prediction probability is 0, indicating a negative diabetes test result. BMI is to the left of the baseline 0 (in red), suggesting that for this woman, BMI increases the risk of diabetes. Glucose is to the right of the baseline (in blue), indicating a decreasing effect. Therefore, controlling weight may reduce the risk of diabetes for this woman.

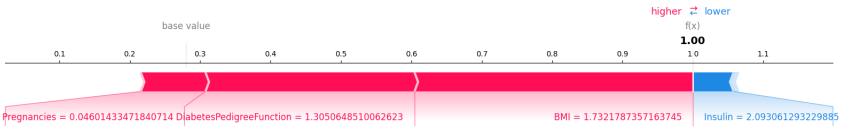


Figure 6.2: Visualization of individual samples (force diagram).

For the instance depicted in Figure 6.2, the woman's predicted probability is 1, indicating a positive prediction for diabetes. BMI and Diabete-

sPedigreeFunction are on the left side of the baseline of 1 (shown in red), suggesting that for this woman, BMI and DiabetesPedigreeFunction reduce the risk of diabetes. Insulin is on the right side of the baseline (shown in blue), indicating its positive impact, meaning that for this woman, Insulin is the main contributing factor to diabetes.

Through these two example graphs, it can be observed that SHAP can predict results based on individual instances and display the magnitude of influence of each physiological indicator on the prediction outcome. When the red portion exceeds the blue portion, the prediction outcome is negative, and vice versa. By interpreting each instance with SHAP, the impact of each indicator on the prediction outcome can be thoroughly explained, thereby facilitating the interpretation of the prediction process of the model.

6.1.2 SHAP Feature Importance

The value of Shapley reflects the influence of features on prediction results. The larger the value of shapley, the greater the influence and the more important the features are. The importance of each feature can be intuitively observed by sorting the value of shapley. Therefore, the average of the absolute shapley values of each eigenvalue can be calculated to obtain the SHAP feature importance graph.

Next, we sort the feature importance in descending order and plot them. The following figure shows the feature importance obtained through SHAP for predicting diabetes using a model trained based on the One-Class SVM algorithm.

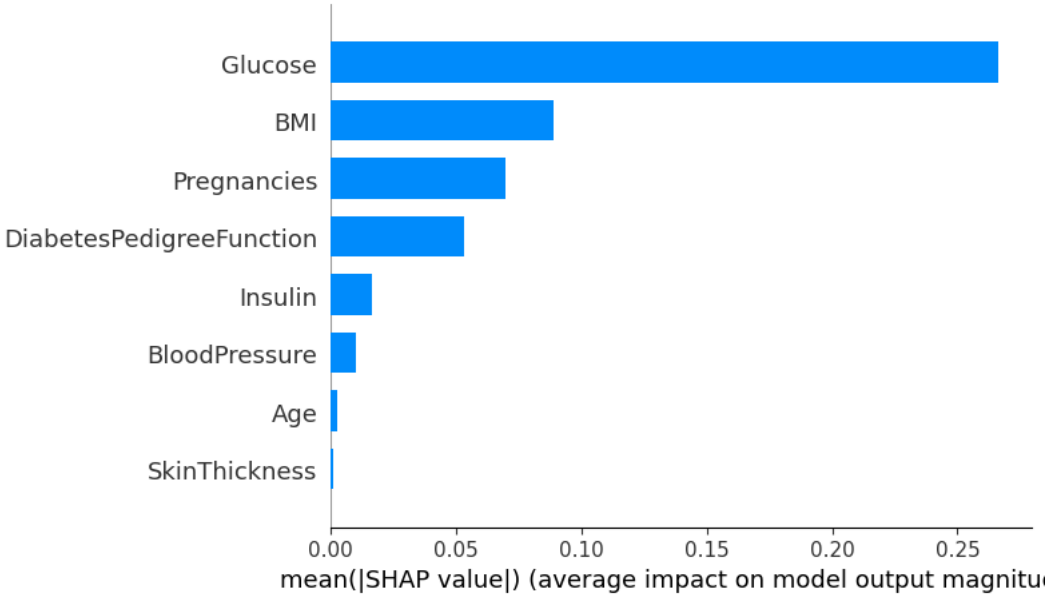


Figure 6.3: The graph showing the feature importance sorted in descending order.

Measured by the average Shapley absolute value, Glucose emerges as the most important feature, altering the predicted diabetes probability by an average of 25 percentage points. This indicates that Glucose is the most influential feature for determining diabetes across all samples and is a vital physiological indicator we must pay attention to. Additionally, the feature importance plot elucidates the weight of different features in the overall prediction process and explains the contribution of each feature to the prediction outcome.

While the feature importance plot is informative, it lacks additional details beyond importance. To gain more comprehensive insights, we will now examine the summary plot.

6.1.3 SHAP Summary Plot

The summary plot combines feature importance with the shapley distribution for each instance[3]. Each point on the graph represents a feature and a sample Shapley value, where the Y-axis position is determined by the importance of each feature, arranged in descending order. The position on the X-axis is determined by the shapley of the feature corresponding to each instance, and the color indicates the influence of the feature value on the prediction from low to high, with red being the highest.

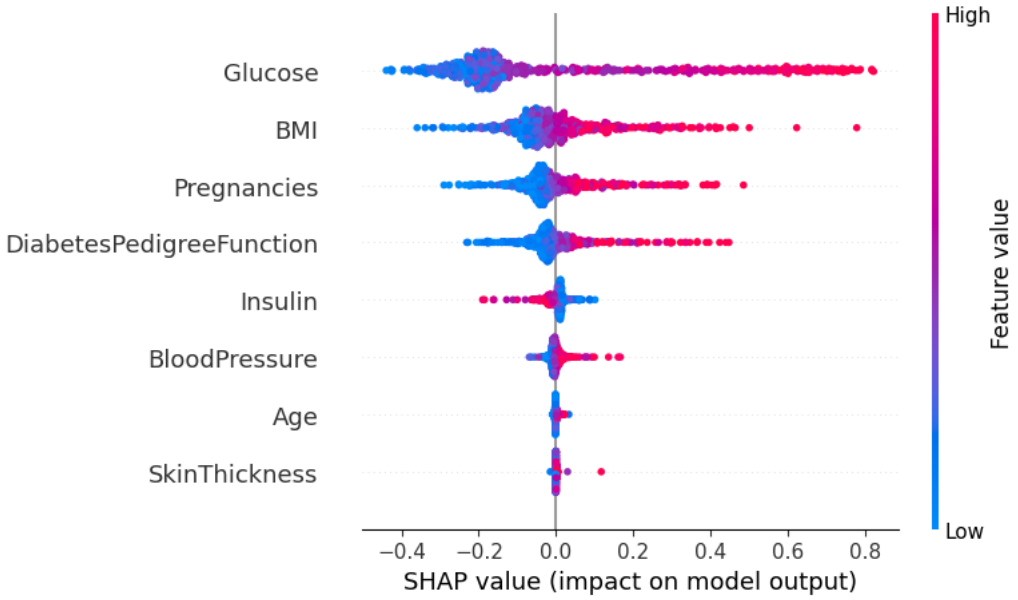


Figure 6.4: SHAP Summary Plot

The SHAP summary plot illustrates that lower levels of Glucose are associated with a lower risk of diabetes, while higher levels of Glucose correspond to a higher risk. For all features except insulin, higher values are associated with a higher risk of diabetes, whereas lower levels of insulin indicate an increased likelihood of diabetes. However, it's important to note that these effects only describe the model's behavior and may not imply causation in the real world.

In the summary plot, we first observe the relationship between feature values and their predictive impact, followed by the distribution of all instances in prediction. Through the summary plot, we can provide further explanations for the model, enhancing its credibility. The intuitive data and plots demonstrate that the model's predictions are scientifically rigorous, and the entire prediction process involves continuous learning and training. When the model's accuracy is further improved through training or new algorithms, it becomes more applicable in practical scenarios.

6.2 τ-values model and result analysis

The key to the τ-value algorithm also lies in the feature contributions $V(S)$, and the feature contribution algorithm of the τ-value algorithm is the same as that of the Shapley value. We can directly model it in the same way and train a τ-value model.

6.2.1 τ-values Feature Importance

To more intuitively compare the differences in explaining the direction of predictive models between SHAP and t-models, feature importance plots are generated by measuring the average absolute t-value.

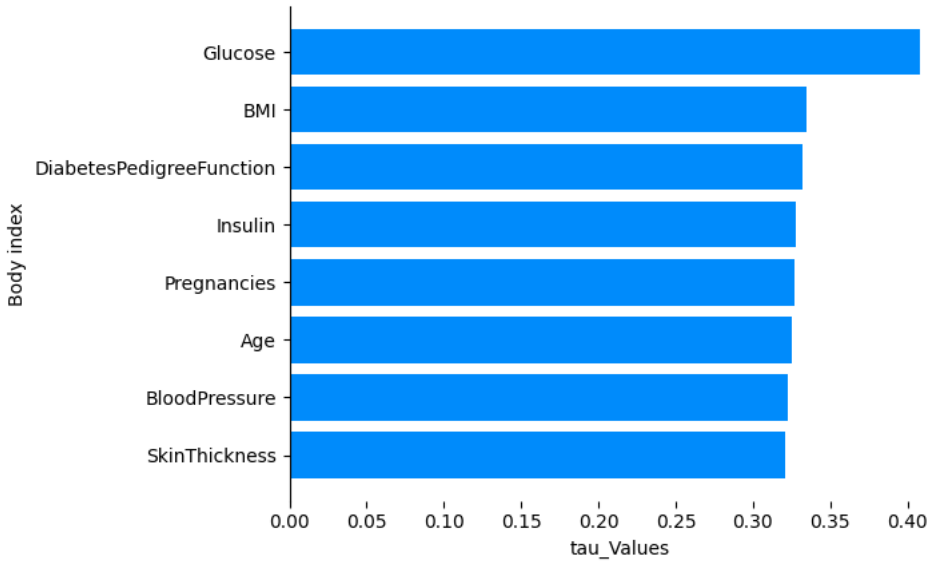


Figure 6.5: τ-values Feature Importance Plot

By observing the t-value feature importance plot, it can be noted that Glucose is the most important feature, with an average absolute change of 40 percentage points in the predicted probability of diabetes. For all samples, Glucose is the feature that has the greatest impact on whether or not diabetes is present. BMI and other features also have significant effects on the prediction results, with changes ranging from 30% to 35%, but only Glucose shows a significant numerical effect. Additionally, the

feature importance plot explains the weights of different features in the overall prediction process and elucidates the contribution of each feature to the prediction results.

6.3 Model Comparison

In this paper, we employed two model interpretation methods, namely SHAP and t-values. Compared to t-values, SHAP has been widely used in the field of interpretable AI, offering diverse and more reliable interpretation techniques. However, t-values, also being cooperative game-based methods, have their own advantages in model interpretation. Let's compare the feature importance plots of both methods.

SHAP Feature Importance	τ-values Feature Importance
Glucose	Glucose
BMI	BMI
Pregnancies	DiabetesPedigreeFunction
DiabetesPedigreeFunction	Insulin
Insulin	Pregnancies
BloodPressure	Age
Age	BloodPressure
SkinThickness	SkinThickness

Table 6.1: Feature Importance Comparison Table in Descending Order

Through the sorting of feature importance, it can be observed that both methods yield similar results. The two features that have the greatest impact on the prediction results are Glucose and BMI, and their importance rankings are also the same. Similarly, the middle three features in the rankings are Pregnancies, DiabetesPedigreeFunction, and Insulin, and the last three features are similar as well. Therefore, both of these explanatory

methods can be used, each with its own advantages.

The results from SHAP are more significant, showing a significant difference in the impact of different features on the prediction results, providing intuitive understanding of feature importance information. Moreover, it can generate instance-level plots to make explanations more convincing. However, SHAP's disadvantages are also quite apparent, with high computational complexity and exponential increase in calculation time and resources with the addition of features, which may not perform well in high-dimensional data.

On the other hand, t-values also provide a clear understanding of the importance ranking of each feature in predicting the results. Apart from the most obvious feature Glucose, the remaining features are not significant. However, its feature importance ranking still provides interpretability and reliability. The biggest advantage of t-values is the reduction in computational time, saving computational resources.

Each method has its own strengths and weaknesses. To further compare the interpretability of these two methods, a new model can be built to predict and compare instances again.

7 Predictive model based on XGBoost

7.1 XGBoost

Although both the SHAP and τ-value models perform well, for a more intuitive comparison of the two models, we can achieve it through re-modeling. By interpreting the models, we can obtain Shapley values based on the SHAP algorithm and τ-values based on the τ-algorithm. Since both of these values can reflect the impact of features on the overall predictive model, they can be used as feature weights to reconstruct the medical detection model. The superiority of the two models can be analyzed to draw conclusions about the interpretability and credibility of the two algorithms. In the case of known feature weights, the simplest approach is to build a linear prediction model. However, due to the dataset's nonlinear distribution, we choose to train the prediction model based on XGBoost.

The XGBoost model (eXtreme Gradient Boosting) is a gradient boosting framework developed by Tianqi Chen in 2014 and widely used in the field of machine learning. The core idea of XGBoost is to iteratively train multiple weak learners and combine them to achieve powerful predictive capabilities. It improves and optimizes upon the gradient boosting algorithm, featuring efficiency, flexibility, and scalability.

XGBoost implements parallel computing and distributed computing, hence excelling in handling large-scale datasets. It enhances training and prediction speed through optimized algorithms and data structures. Employing the Gradient Boosting Decision Tree algorithm, it iteratively trains weak classifiers and combines them into a strong classifier. Such ensemble learning methods typically exhibit high accuracy.

The basic idea is to construct a regression tree model in each round

of iteration and update the model based on the residuals, ultimately combining multiple weak classifiers into a strong classifier. Specifically, the objective function of XGBoost involves adding a regularization term to the loss function, and the model is trained by optimizing this objective function. The main formulas include the loss function, model prediction, and residuals:

Loss Function:

$$\text{Loss} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

$L(y_i, \hat{y}_i)$ is the loss function, which measures the difference between the predicted value $\hat{y}_i$ and the true value y_i , $\Omega(f_k)$ is the regularization term, which controls the complexity of the model.

Model Prediction:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{K} f_k(x_i)$$

$\hat{y}_i^{(t)}$is the predicted value after t rounds of iteration, and $f_k(x_i)$ is the predicted value of the k-th regression tree model for the sample x_i.

XGBoost trains the model by optimizing this objective function, gradually approaching the optimal solution, thus achieving efficient prediction and generalization capabilities. The general steps to build an XGBoost model are as follows:

Prepare the Dataset:

- Split the dataset into feature set (X) and target variable (y).Ensure there are no missing values or outliers in the dataset, and perform necessary data preprocessing (such as feature scaling, encoding categorical variables, etc.).

Create DMatrix:

- Use the DMatrix class provided by XGBoost to convert the feature set and target variable into the internal data structure of XGBoost, to enhance the efficiency of training and prediction.

Set Parameters:

- Define the parameters of the XGBoost model, including the objective function, evaluation metrics, tree depth, learning rate, etc. You can adjust them based on the characteristics of the problem and the scale of the dataset.

Train the Model:

- Train the XGBoost model using the xgb.train() function. Provide the DMatrix object, parameters, and the number of training iterations as inputs. During training, the model will iteratively build multiple decision tree models based on the specified parameters and objective function, continuously optimizing the predictive performance.

- Compute τ-values: Calculate the τ-value for each feature using marginal contributions.

Evaluate the Model:

- Evaluate the performance of the model using a test dataset. You can use various evaluation metrics such as accuracy, precision, recall, F1 score, etc., to understand the model's performance.

7.2 XGBoost prediction model

7.2.1 A prediction model based on Shapley

After establishing the corresponding medical detection model using Shapley values, it is necessary to evaluate the model. If the model accuracy is too low, it cannot be used as a reference for comparison. Here, we still use confusion matrices, precision, and ROC curves for analysis.

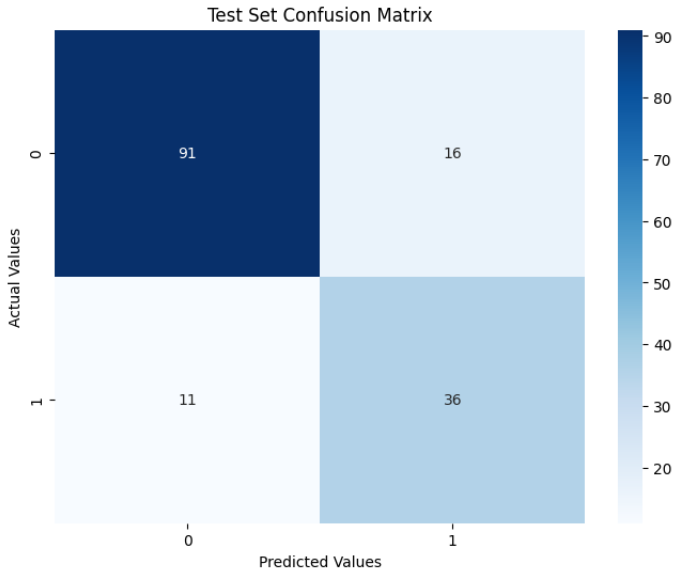


Figure 7.1: Confusion matrix of the XGBoost medical detection model based on Shapley values.

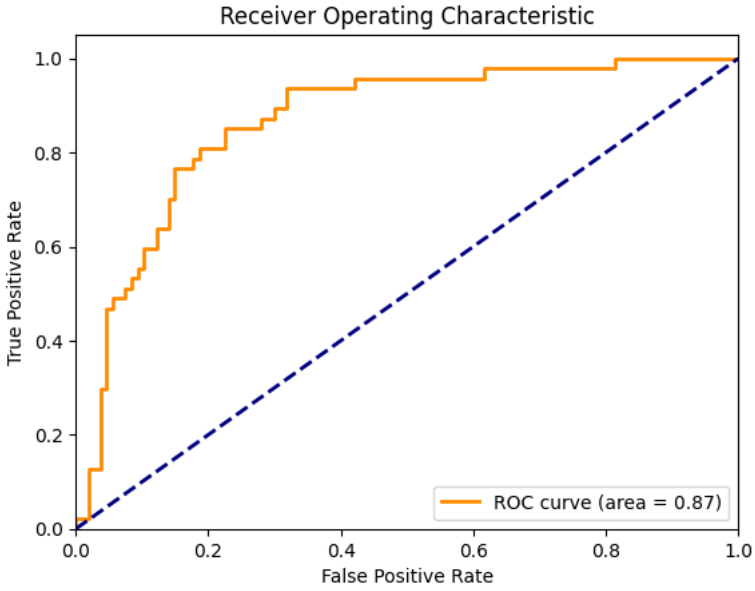


Figure 7.2: ROC curve of the XGBoost medical detection model based on Shapley values.

Through Figure 7.1, it can be inferred that the accuracy of this model is 82.5%, which is higher than the medical detection model established based on the ONE-Class SVM algorithm. Therefore, there is no problem in using the SHAP model for interpretable AI research. Moreover, its higher predictive accuracy also implies a high level of credibility when explaining "black box" models using SHAP. By observing the ROC curve, it is also noted that the AUC is 0.87, close to one. When the AUC value is within the range of 0.5 to 1, the predictive model has predictive value, indicating that this predictive model has great research value. Therefore, the medical detection model established through Shapley can also be explored as a new prediction method. If the computational complexity of the SHAP method can be reduced, the SHAP model will have great development and research space in both the interpretable and predictive fields.

7.2.2 A prediction model based on τ-values

Plot the confusion matrix and ROC curve of the XGBoost medical detection model based on τ-values, investigate the interpretability of τ-values, evaluate the predictive model, and study whether it has predictive value.

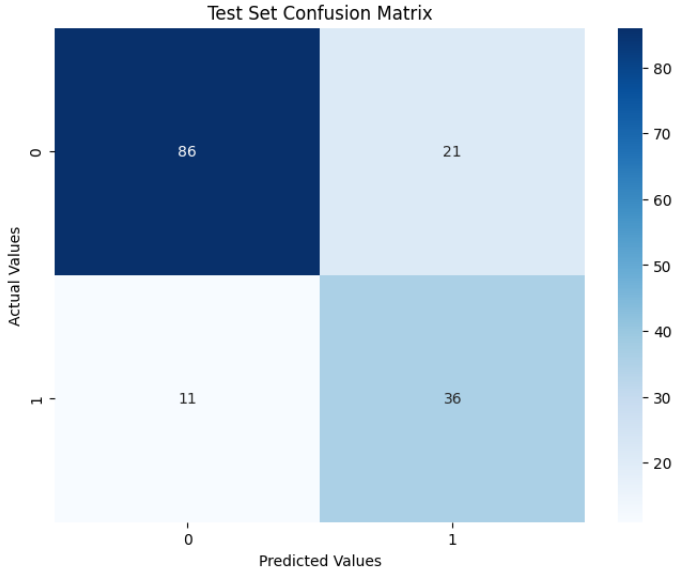


Figure 7.3: Confusion matrix of the XGBoost medical detection model based on τ- values.

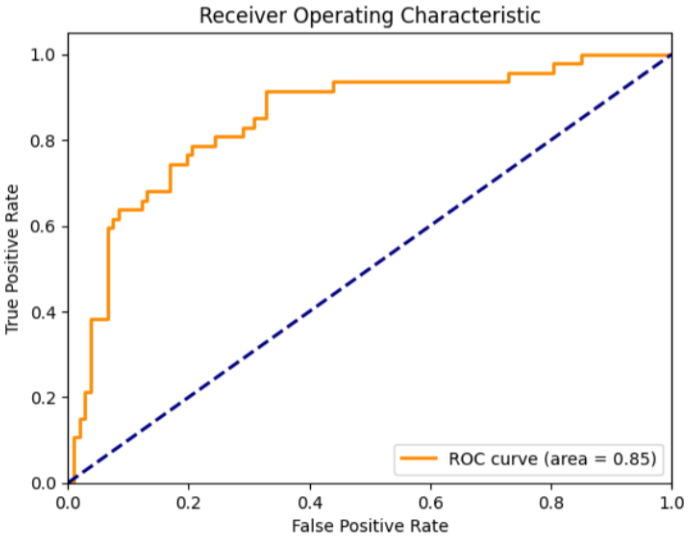


Figure 7.4: ROC curve of the XGBoost medical detection model based on τ-values.

Through the analysis of Figures 7.3 and 7.4, it is evident that although the τ-value method does not yield significant numerical results in feature importance research, the model established based on τ-values still achieves high accuracy. With an accuracy of 80%, which is close to the accuracy of detection models established based on ONE-class SVM algorithm and SHAP algorithm. Furthermore, its ROC curve has an AUC of 0.85, demonstrating its authority in both interpretable research and medical detection research. This also proves the feasibility of using the τ-value method for interpretable research.

7.2.3 Interpretable Model Comparison

The two predictive models based on XGBoost effectively compare the interpretability and predictive capabilities of the SHAP and τ-value models. In terms of interpretability, both methods exhibit strong explanatory power and reliability. The SHAP method is superior to the τ-value method in interpretability, but this does not necessarily mean that the SHAP method is inherently better than the τ-value method. The SHAP method has become very mature through years of development and research, capable of interpreting both single samples and overall models with good data performance.

However, the τ-value method also shows impressive performance in interpretability. Its conclusions on feature importance are remarkably similar to those derived from the SHAP method. Both methods can conduct interpretability studies, and the τ-value method performs better in terms of training time and handling high-dimensional data. Using the same predictive model, the time difference in calculating SHAP values and τ-values was 30-fold, with SHAP calculations consuming substantial time and com-

putational resources. In contrast, the τ-value method saves significant time while ensuring accuracy.

Moreover, the predictive models built on these two algorithms hold substantial research value. Using feature weights to construct the XGBoost model, both achieved an accuracy rate of over 80%, with AUC values greater than 0.85. These predictive values surpass some mainstream predictive models in the machine learning field. Overall, both τ-values and SHAP values can reliably explain medical diagnostic models and can serve as the foundation for building predictive models with high accuracy.

8 Conclusion

This paper primarily focuses on the study of diabetes prediction models based on machine learning from the perspective of cooperative game theory. The challenge in medical prediction models lies in their interpretability. Most machine learning-based prediction models are "black-box" models, making it difficult to explain their results in a way that convinces doctors and patients of their reliability. To address this issue, it is essential to develop a medical prediction model that can diagnose diabetes and analyze the diagnostic process, providing explanations for the results.

The process begins with the creation of a diabetes prediction model based on ONE-CLASS SVM. It is crucial to analyze and preprocess the dataset to avoid any adverse effects on the prediction results. Following this, the model is trained, prediction results are generated, and the model is evaluated. After the predictions are made, the results are interpreted to explain the feasibility of the model, thereby increasing the confidence of doctors and patients in the model's predictions.

The explanation process is primarily divided into two steps. Firstly, Shapley values and τ-values are computed based on the predictive model from the perspective of cooperative game theory. Both of these values can reflect the importance of features. After obtaining specific values, statistical graphs regarding instances and features are plotted to intuitively observe the influence of different features on the overall prediction results. The feature importance can then explain where the results of the "black-box" predictive model come from.

Shapley values are a commonly used method for studying interpretability, but τ-values are being used for the first time. To investigate

whether τ-values can serve as a method for interpretability research, it is necessary to study their feasibility as an explanatory tool. Comparing the feature importance obtained by τ-values with that obtained by the SHAP method reveals similar conclusions, indicating that τ-values can be used as a method for researching interpretable AI. However, their performance in feature importance values is not significant.

I reconstructed medical detection models using τ-values and SHAP values based on XGBoost. Both of these detection models exhibit high accuracy and predictive value, with similar results. Therefore, τ-values can be fully applied to research on interpretable AI, and they can also be used to construct medical detection models along with SHAP values.

References

1. Zou J , Xu F , Zhang Y ,et al.High-Dimensional Explainable AI for Cancer Detection[J]. 2021.

2. Lundberg S , Lee S I .A Unified Approach to Interpreting Model Predictions[J]. 2017.DOI:10.48550/arXiv.1705.07874.

3. Lundberg S M , Erion G G , Lee S I .Consistent Individualized Feature Attribution for Tree Ensembles[J]. 2018.DOI:10.48550/arXiv.1802.03888.

4. HUANG Yuteng, Pei Xubin, Kong Libo, Li Bo, Yin Jie. Research on Power outlier User detection algorithm based on Ant colony Algorithm improved One-Class SVM [J]. Automation and instrumentation, 2019 (5) : 111-114. The DOI: 10.14016 / j.carol carroll nki. 1001-9227.2019.05.111.

5. Huang Meihuaju. Thyroid nodules can explain the AI diagnosis system [D]. The research and implementation of donghua university, 2021. The DOI: 10.27012 /, dc nki. Gdhuu. 2021.001151.

6. ZHONG K T. Research and application of explainable AI diagnostic model for breast tumor [D]. Donghua university, 2021. DOI: 10.27012 /, dc nki. Gdhuu. 2021.000439.

7. Jing Jie, WANG Beilei, LIU Shanrong. Interpretable application of artificial intelligence in disease diagnosis and treatment [J]. Laboratory Medicine, 2019,36(09):976-980. (in Chinese)

8. Yuan Weilin, Luo Junren, Lu Lina, et al. Smart game against method: game theory and reinforcement learning perspective analysis [J]. Jour-

nal of computer science, 2022, 49 (8) : 14. DOI: 10.11896 / JSJKX. 220200174.

9. Lenatti M, Carlevaro A, Keshavjee K, Guergachi A, Paglialonga A, Mongelli M. Characterization of Type 2 Diabetes Using Counterfactuals and Explainable AI. Stud Health Technol Inform. 2022 May 25;294:98-103. doi: 10.3233/SHTI220404. PMID: 35612024.

10. Castro J, Gomez D, Tejada J. Polynomial calculation of the Shapley value based on sampling[J]. Computers and Operations Research, 2009, 36(5): 1726-1730.

11. Cho YR, Kang M. Interpretable machine learning in bioinformatics. Methods. 2020 Jul 1;179:1-2. doi: 10.1016/j.ymeth.2020.05.024. Epub 2020 May 30. PMID: 32479800.

12. Adadi A, Berrada M. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)[J]. IEEE Access, 2018, 6: 52138-52160.

13. Shapley L S. A value for n-person games[J]. Contributions to the Theory of Games, 1953,2(28): 307-317.

14. Balakrishnama S, Ganapathiraju A. Linear discriminant analysis-a brief tutorial[C]//Institute for Signal and information Processing. 1998, 18(1998): 1-8.

15. Hou N, Li M, He L, Xie B, Wang L, Zhang R, Yu Y, Sun X, Pan Z, Wang K. Predicting 30-days mortality for MIMIC-III patients with sepsis-3: a machine learning approach using XGboost. J Transl Med. 2020 Dec 7;18(1):462. doi: 10.1186/s12967-020-02620-5. PMID: 33287854; PMCID: PMC7720497.

16. Li J, Liu S, Hu Y, Zhu L, Mao Y, Liu J. Predicting Mortality in Intensive Care Unit Patients With Heart Failure Using an Interpretable Machine Learning Model: Retrospective Cohort Study. J Med Internet Res. 2022 Aug 9;24(8):e38082. doi: 10.2196/38082. PMID: 35943767; PMCID: PMC9399880.

17. Li Y, Xu Y, Ma Z, Ye Y, Gao L, Sun Y. An XGBoost-based model for assessment of aortic stiffness from wrist photoplethysmogram. Comput Methods Programs Biomed. 2022 Nov;226:107128. doi: 10.1016/j.cmpb.2022.107128. Epub 2022 Sep 13. PMID: 36150230.

18. van der Velden BHM, Kuijf HJ, Gilhuijs KGA, Viergever MA. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. Med Image Anal. 2022 Jul;79:102470. doi: 10.1016/j.media.2022.102470. Epub 2022 May 4. PMID: 35576821.

19. Tosun AB, Pullara F, Becich MJ, Taylor DL, Fine JL, Chennubhotla SC. Explainable AI (xAI) for Anatomic Pathology. Adv Anat Pathol. 2020 Jul;27(4):241-250. doi: 10.1097/PAP.0000000000000264. PMID: 32541594.

20. Martini ML, Neifert SN, Oermann EK, Gilligan JT, Rothrock RJ, Yuk FJ, Gal JS, Nistal DA, Caridi JM. Application of Cooperative Game Theory Principles to Interpret Machine Learning Models of Nonhome Discharge Following Spine Surgery. Spine (Phila Pa 1976). 2021 Jun 15;46(12):803-812. doi: 10.1097/BRS.0000000000003910. PMID: 33394980.

21. Zignoli A. Machine Learning Models for the Automatic Detection of Exercise Thresholds in Cardiopulmonary Exercising Tests: From

Regression to Generation to Explanation. Sensors (Basel). 2023 Jan 11;23(2):826. doi: 10.3390/s23020826. PMID: 36679622; PMCID: PMC9867502.

22. Sun J, Yu H, Zhong G, Dong J, Zhang S, Yu H. Random Shapley Forests: Cooperative Game-Based Random Forests With Consistency. IEEE Trans Cybern. 2022 Jan;52(1):205-214. doi: 10.1109/TCYB.2020.2972956. Epub 2022 Jan 11. PMID: 32203041.

9 appendix

```
#$\tau$-values model
# Defining the characteristic function v(S)
def feature_function(S, X_train, y_train):
    model = LogisticRegression()
    X_train_sub = X_train[list(S)]
    model.fit(X_train_sub, y_train)
    y_pred = model.predict(X_train_sub)
    return accuracy_score(y_train, y_pred)

# Compute b_i^v
def calculate_b(v,N,X_train,y_train):
    b_v = {}
    v_N = v(N,X_train,y_train)
    for i in N:
        S =N - set([i])
        b_v[i] = v_N - v(S,X_train,y_train)
    return b_v


# Compute a_i^v
def calculate_a(v, N, X_train, y_train):
    a_v = {}
    for i in tqdm(N):
        max_r = 0
        for size in range(1,len(N)):
            for S in combinations(N,size):
                if i in S:
                    S = set(S)
                    R_S_i = v(S,X_train,y_train) - sum(v(S - set([j]),
                                                         X_train,y_train)
                                                         for j in S if j
                                                         != i)

                    max_r = max(max_r, R_S_i)
        a_v[i] = max_r
    return a_v
# Compute $a^v$ , $b^v$ and $tau^v$
N = set(x.columns)
```

```
a_v = calculate_a(feature_function, N, X_train, y_train)
b_v = calculate_b(feature_function, N, X_train, y_train)
alpha = 0.5
tau_v = {i: alpha * a_v[i] + (1 - alpha) * b_v[i] for i in N}

#$\tau$-values Feature Importance
import matplotlib.pyplot as plt
categories = ['Age','Pregnancies','BMI', 'Insulin', 'SkinThickness', '
                                    Glucose' ,'DiabetesPedigreeFunction'
                                    ,'BloodPressure']
values = [0.3249185667752443, 0.3265472312703583,0.3346905537459283,0.
                                    3273615635179153, 0.
                                    32084690553745926,0.
                                    40798045602605865,
          0.3322475570032573,0.32247557003257327]

sorted_indices = sorted(range(len(values)), key=lambda k: values[k],
                                    reverse=True)
sorted_categories = [categories[i] for i in sorted_indices]
sorted_values = [values[i] for i in sorted_indices]

plt.barh(sorted_categories[::-1], sorted_values[::-1], color='#008BFB')


plt.xlabel('tau_Values')
plt.ylabel('Body index')

plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
plt.show()
```

```
tau_v

{'Age': 0.3249185667752443,
 'Insulin': 0.3273615635179153,
 'BMI': 0.3346905537459283,
 'BloodPressure': 0.32247557003257327,
 'Pregnancies': 0.3232899022801302 7,
 'DiabetesPedigreeFunction': 0.3322475570032573,
 'Glucose': 0.4079804560260586 5,
 'SkinThickness': 0.32084690553745926}
```

Figure 9.1: The τ-value for each feature.

```
#SHAP Model
from IPython.display import display, Javascript
display(Javascript('initjs()'))
!pip install shap
import shap
shap.initjs()
#Calculate the shapley value
df_X=pd.DataFrame(x_train,columns=x.columns)
import matplotlib
#Visualization of Predictions
shap.force_plot(explainer.expected_value, shap_values[0, :], df_X.iloc[0
                                        , :], matplotlib=matplotlib)
import matplotlib
#Single sample shapley diagram
shap.force_plot(explainer.expected_value, shap_values[1, :], df_X.iloc[1
                                        , :], matplotlib=matplotlib)
#SHAP Feature Importance
shap.summary_plot(shap_values,df_X)
#SHAP Summary Plot
shap.summary_plot(shap_values,df_X, plot_type='bar')

# Average the SHAP values for each feature
average_shap_values = np.mean(np.abs(shap_values), axis=0)
# Outputs the weights for each feature
for i, shap_value in enumerate(average_shap_values):
    print(f"Feature {i+1}: {shap_value}")
```

```
Feature 1: 0.06994363082796384 ='Pregnancies
Feature 2: 0.2662947604850174 = 'Glucose
Feature 3: 0.010021105440552909 ='DiabetesPedigreeFunction
Feature 4: 0.001031291730264706 ='SkinThickness
Feature 5: 0.016494402618393283 ='Insulin
Feature 6: 0.08909462977755472 ='BMI
Feature 7: 0.05314129896035272 ='BloodPressure
Feature 8: 0.002488412154429764 ='Age
```

Figure 9.2: The shapley value for each feature.