INNO-XAI

Sensitivity Analysis of Predicting Safe Discharge Using MIMIC-IV Dataset

Naam : Hussin Almoustafa

Studentnummer : 1776495

UNIVERSITY
OF APPLIED
SCIENCES
UTRECHT

April 21, 2023

Abstract

In this study, we perform a sensitivity analysis on a classification model predicting the safe discharge of patients using the MIMIC-IV dataset. We consider a set of 30 features, including patient demographics, medical history, vital signs, and surgical information. We apply various complex sensitivity analysis methods, such as variance-based sensitivity analysis, Morris method, derivative-based sensitivity analysis, regression-based sensitivity analysis, and Random Forest-based sensitivity analysis. Our goal is to identify the most influential features for predicting safe discharge and understand the impact of each feature on the model's performance. This information can help in improving the model's accuracy and generalizability.

1 Introduction

Hospital discharge planning is a crucial aspect of patient care and resource management. Predicting safe discharge can help optimize hospital resources, reduce the length of stay, and improve patient outcomes. In this study, we perform a sensitivity analysis on a classification model that predicts safe discharge using the MIMIC-IV dataset. The dataset contains a variety of patient information, such as demographics, medical history, vital signs, and surgical information.

We apply various complex sensitivity analysis methods to identify the most influential features in predicting safe discharge and understand the impact of each feature on the model's performance. This information can help in improving the model's accuracy and generalizability.

we also investigate two functionalities for analyzing and modifying the behavior of a machine learning model: feature modification by the user and counterfactual analysis. The first functionality allows users to turn off variables (i.e., set to NA), change variable values, and adjust variable importance (out of scope for now). The second functionality focuses on counterfactual analysis, where the system indicates which values should change to obtain a different prediction. Both functionalities are equally important, and a potential extension of this work could include the study of variable importance.

2 Methods and Formulas

In this study, we use the following complex sensitivity analysis methods:

a) Variance-based sensitivity analysis (Sobol Indices)

b) Morris method (Elementary Effects)

c) Derivative-based sensitivity analysis (Gradient-based methods)

d) Regression-based sensitivity analysis

e) Random Forest-based sensitivity analysis (Permutation Importance)

Here is a brief overview of each method and the corresponding formulas:

2.1 Variance-based sensitivity analysis (Sobol Indices)

Variance-based sensitivity analysis measures the importance of each input feature by decomposing the variance of the model output into contributions from individual input features and their interactions. Sobol Indices are the most commonly used measures of variance-based sensitivity analysis and can be computed using Monte Carlo sampling. The first-order Sobol Index measures the proportion of variance in the model output explained by the variation in each individual input feature, while the total-order Sobol Index measures the proportion of variance in the model output explained by the variation in each individual input feature and its interactions with all other input features.

$$S_i = \frac{\mathrm{Var}_{X_i}[E_{\sim i}(Y|X_i)]}{\mathrm{Var}(Y)}$$

$$T_i = 1 - \frac{\mathrm{Var}_{\sim i}[E_{\sim i}(Y|X_i)]}{\mathrm{Var}(Y)}$$

where:

- S_i is the first-order Sobol Index of the i-th input feature
- T_i is the total-order Sobol Index of the i-th input feature
- X_i is the i-th input feature
- Y is the model output
- $E_{\sim i}(Y|X_i)$ is the expected value of Y when X_i is fixed and all other input features are varied
- $\mathrm{Var}_{X_i}[E_{\sim i}(Y|X_i)]$ is the variance of $E_{\sim i}(Y|X_i)$ over X_i
- $\mathrm{Var}_{\sim i}[E_{\sim i}(Y|X_i)]$ is the variance of $E_{\sim i}(Y|X_i)$ over all input features except X_i
- $\mathrm{Var}(Y)$ is the variance of the model output

To apply this method to the MIMIC-IV dataset, you would:

a) Generate a set of input vectors using Monte Carlo sampling.

b) Evaluate the model output for each input vector.

c) Compute the first-order and total-order Sobol Indices for each input feature using the formulas above.

d) Rank the input features by their total-order Sobol Indices to identify the most important features for predicting safe discharge.

Note that this method assumes that the model is well-defined and does not account for non-linear interactions between input features. Additionally, it requires a large number of Monte Carlo samples to achieve accurate results.

2.2 Morris method (Elementary Effects)

Morris method is a global sensitivity analysis technique that estimates the effects of input features on the model output by perturbing one input feature at a time while keeping the other features fixed. The sensitivity of the model output to each input feature is quantified by the Elementary Effects, which is the difference in the output caused by a small perturbation in the input feature divided by the perturbation size. Morris method involves randomly sampling a set of input vectors with varying levels of perturbations for each input feature. The distribution of Elementary Effects can be used to rank the input features by their relative importance.

$$EE_i = \frac{f(X^{(i)+\Delta_i e_i}) - f(X^{(i)})}{\Delta_i}$$

where:

- EE_i is the Elementary Effect of the i-th input feature
- f is the model output
- $X^{(i)}$ is the input vector with the i-th feature unperturbed
- $X^{(i)+\Delta_i e_i}$ is the input vector with the i-th feature perturbed by a small amount Δ_i in the direction of the unit vector e_i

To apply this method to the MIMIC-IV dataset, you would:

a) Generate a set of input vectors by perturbing each input feature with different levels of perturbations.

b) Evaluate the model output for each input vector.

c) Compute the Elementary Effects for each input feature using the formula above.

d) Compute the mean and variance of the Elementary Effects for each input feature across all input vectors.

e) Rank the input features by their mean Elementary Effects to identify the most important features for predicting safe discharge.

Note that this method assumes linearity between the input features and the model output, and may not be appropriate for non-linear models.

2.3 Derivative-based sensitivity analysis (Gradient-based methods)

Derivative-based sensitivity analysis measures the sensitivity of the model output with respect to each input feature by computing the gradient of the output with respect to the inputs. The gradient can be computed using various methods such as finite differences, analytical gradients, or automatic differentiation. The absolute value of the gradient indicates the magnitude of the sensitivity of the output to the corresponding input variable.

$$\text{Gradient} = \left|\frac{\partial Y}{\partial X}\right|$$

where:

- Y is the model output
- X is the input feature

To apply this method to the MIMIC-IV dataset, you would:

a) Train your classification model using the MIMIC-IV dataset and obtain the predicted output (e.g., probability of safe discharge) for each observation.

b) Compute the gradient of the output with respect to each input feature using one of the gradient-based methods.

c) Evaluate the magnitude of the gradient to assess the sensitivity of the model output to each input variable. A higher absolute value of the gradient indicates greater sensitivity to the corresponding input variable.

Keep in mind that this method assumes differentiability of the model and may not hold for non-differentiable models. Additionally, the method may be sensitive to noise and small perturbations in the input features.

2.4 Regression-based sensitivity analysis

Fit a linear regression model between the input features (X) and the model output (Y). The regression coefficients (β) can be used to estimate the sensitivity of the output to each input variable. The standardized regression coefficients can be used to compare the relative importance of the features. The higher the absolute value of the standardized coefficient, the more sensitive the output is to the corresponding input variable.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

where:

- Y is the model output
- X_i is the input feature i
- β_i is the regression coefficient for input feature i
- ϵ is the residual error

To apply this method to the MIMIC-IV dataset, you would:

a) Train your classification model using the MIMIC-IV dataset and obtain the predicted output (e.g., probability of safe discharge) for each observation.

b) Fit a linear regression model using the input features (X) and the predicted output (Y) from your classification model.

c) Calculate the standardized regression coefficients by dividing the coefficients (β) by the standard deviation of the corresponding input feature. This allows for a comparison of the relative importance of each feature.

d) Evaluate the magnitude of the standardized coefficients to assess the sensitivity of the model output to each input variable. A higher absolute value of the standardized coefficient indicates greater sensitivity to the corresponding input variable.

Keep in mind that this method assumes a linear relationship between the input features and the model output, which may not hold for more complex, non-linear models.

2.5 Random Forest-based sensitivity analysis (Permutation Importance)

Random Forest-based sensitivity analysis measures the importance of a feature by permuting its values and measuring the decrease in the model's performance. Permutation Importance is calculated as the difference in performance (e.g., accuracy, R-squared) between the original model and a version of the model with the feature's values permuted. The greater the decrease in performance, the more important the feature is.

$$\text{Permutation Importance} = \text{Performance}_{\text{original}} - \text{Performance}_{\text{permuted}}$$

To apply this method to the MIMIC-IV dataset, you would:

a) Train a Random Forest model using the MIMIC-IV dataset and obtain the predicted output (e.g., probability of safe discharge) for each observation.

b) Permute the values of a feature and obtain the new predicted output for each observation.

c) Calculate the decrease in performance by comparing the original predicted output to the new predicted output using a performance metric such as accuracy or R-squared.

d) Repeat steps 2-3 for all features in the dataset to obtain a Permutation Importance score for each feature.

e) Rank the features by their Permutation Importance scores to identify the most important features for predicting safe discharge.

3 Features

The following features are considered in this study:

a) afd_na_ok

b) afdeling

c) anesthesie

d) ASA

e) BMI

f) dagen_pre_ok

g) datum_DESIRE_voorspelling

h) diagnose

i) duur_ok

j) duur_ok_gpl

k) geboortedatum

l) geslacht

m) herkomst

n) interv_ab

o) interv_ok

p) interv_rad

q) kamer

r) leeftijd

s) minimaal_invasief

t) naam

u) opname_nummer

v) outcome_observed

w) patient_id

x) prediction

y) prediction_binary

z) spoed

) start_operatie

) type_ok

) unieke_med

) vit_af, vit_hr, vit_sat, vit_temp

) zkh_opn_start

4 Conclusion

By applying the sensitivity analysis methods to the classification model trained on the MIMIC-IV dataset, we can gain insights into the relationship between the input features and the model's ability to predict safe discharge. This information can help in refining the model and improving its accuracy, ultimately leading to better patient care and more efficient hospital resource allocation.

Counterfactuals in Machine Learning: Sensitivity Analysis and Feature Modification

5 Feature Modification

When modifying features, it is crucial to ensure that the input values are valid. For categorical variables, only values existing in the database are allowed. For numerical variables, the input value should lie within the minimum and maximum range of the variable in the database.

6 Counterfactual Analysis

Counterfactuals are hypothetical instances that help explain the implications of a scenario: ”if not x, then not y.” In the context of machine learning, counterfactual instances are created by artificially changing the features of a training example to alter the model’s prediction, thereby aiding in model interpretation.

The method proposed by Wachter et al. [**?**] is used to create counterfactuals. This method is model-agnostic and works with any scikit-learn estimator that supports the *predict* (and ideally *predict_proba*) method. The loss function to minimize is given by:

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x') \tag{1}$$

Here, $\hat{f}(x')$ denotes the model prediction for the counterfactual x', and y' is the desired prediction specified by the user. The hyperparameter λ balances the importance of the first term, which minimizes the squared difference between the model prediction for the counterfactual and the desired prediction, and the second term, which calculates the distance $d(x, x')$ between a given instance x and a generated counterfactual x'. The distance function is defined as:

$$d(x, x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{\text{MAD}_j} \tag{2}$$

The Median Absolute Deviation (MAD) is given by:

$$\text{MAD}j = \text{median}i \in 1, \ldots, n \left(|x_{i,j} - \text{median}l \in 1, \ldots, n(xl, j)|\right) \tag{3}$$

The general procedure for using the create_counterfactual function is:

a) Select an instance to explain and specify its desired prediction y'.

b) Choose a value for the hyperparameter λ.

c) Optimize the loss L using the create_counterfactual function.

d) Optionally, increase λ until a user-definedt hreshold ϵ is reached, i.e.,

$$\text{while } |\hat{f}(x') - y'| > \epsilon \; : \tag{4}$$

- increase λ

7 Conclusion

This research focuses on two functionalities in machine learning: user-driven feature modification and counterfactual analysis. Feature modification allows users to turn off variables, change variable values, and potentially adjust variable importance. Counterfactual analysis, on the other hand, uses the method proposed by Wachter et al. [**?**] to create hypothetical instances that help interpret the model's behavior. By combining these two functionalities, we can provide users with greater control over and understanding of the machine learning models they use.

References

a) Tarantola, S. (2005). ”Random Sampling of Model Input.” In: Encyclopedia of Statistics in Quality and Reliability. Edited by: R. A. Johnson, and D. W. Wichern. Wiley.

b) Saltelli, A., Chan, K., and Scott, E. M. (2000). ”Sensitivity Analysis.” Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, pp. 247-257.

c) Breiman, L. (2001). ”Random Forests.” Machine Learning, vol. 45, pp. 5-32.

d) Sobol, I. M. (2001). ”Global Sensitivity Indices for Nonlinear Mathematical Models and their Monte Carlo Estimates.” Mathematics and Computers in Simulation, vol. 55, pp. 271-280.

e) Morris, M. D. (1991). ”Factorial Sampling Plans for Preliminary Computational Experiments.” Technometrics, vol. 33, pp. 161-174.

f) Sobol, I. M. (2001). ”Global Sensitivity Indices for Nonlinear Mathematical Models and their Monte Carlo Estimates.” Mathematics and Computers in Simulation, vol. 55, pp. 271-280.

g) wachter2017 Wachter, S., Mittelstadt, B., Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law Technology*, 31(2), 841-887.

h) molnar Molnar, C. (2020). Interpretable Machine Learning. Retrieved from `https://christophm.github.io/interpretable-ml-book/`