HAW HAMBURG

MASTER THESIS
Sabrina Göllner

VERIFAI
Evaluating the Responsibility of AI-Systems

Faculty of Engineering and Computer Science
Department of Computer Science

HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Sabrina Göllner

VERIFAI
Evaluating the Responsibility of AI-Systems

Master thesis submitted for examination in Master's degree
in the study course *Master of Science Informatik*
at the Department of Computer Science
at the Faculty of Engineering and Computer Science
at the Hamburg University of Applied Sciences

Supervisor: Prof. Dr.-Ing. Marina Tropmann-Frick
Supervisor: Prof. Dr. Olaf Zukunft
Submitted on: 25th July 2023

Sabrina Göllner

Title of Thesis

VERIFAI
Evaluating the Responsibility of AI-Systems

Keywords

Artificial Intelligence, Responsible AI, Privacy-preserving AI, Explainable AI, Ethical AI, Trustworthy AI, Machine Learning, Deep Learning, Evaluation, Verification

Abstract

The rapid progress and extensive integration of artificial intelligence (AI) systems across diverse sectors have heightened concerns regarding their security, explainability, privacy, and ethics. Moreover, AI is becoming increasingly ingrained in daily life, leading to discussions about the responsibility of AI-technologies. Ensuring Responsible AI (RAI) practices is crucial to maintain trust in these systems and mitigating potential negative consequences.

In response to the growing demand for RAI, this thesis presents a novel approach to assessing Responsible AI by combining insights from a systematic literature review with a practical evaluation framework. The thesis provides a concise overview of the key aspects of Responsible AI and highlights the findings from the literature review.

Furthermore, the thesis introduces a set of evaluation metrics specifically designed for the current state of the art, using different model types and data from the healthcare domain. The framework supports the evaluation of natural language processing, computer vision, and tabular data models for classification tasks.

Additionally, the thesis extensively demonstrates VERIFAI, an implementation of the framework, which serves as a comprehensive tool for assessing the responsibility of AI systems. The overall objective of this research is to make a meaningful contribution to the Responsible AI discourse, providing researchers and practitioners with a valuable resource to enhance the overall responsibility of their AI systems. The thesis concludes by discussing future directions to enhance and further extend the framework.

Kurzzusammenfassung

Der rasche Fortschritt und die umfassende Integration von Systemen der künstlichen Intelligenz (KI) in verschiedenen Sektoren haben die Bedenken hinsichtlich ihrer Sicherheit, Erklärbarkeit, ihres Datenschutzes und Ethik verstärkt. Darüber hinaus wird die KI immer stärker in das tägliche Leben integriert, was zu Diskussionen über die Verantwortung von KI-Technologien führt. Die Gewährleistung verantwortungsvoller KI-Praktiken (Responsible AI, RAI) ist von entscheidender Bedeutung, um das Vertrauen in diese Systeme aufrechtzuerhalten und mögliche negative Folgen abzumildern.

Als Reaktion auf die wachsende Nachfrage nach RAI wird in dieser Arbeit ein neuartiger Ansatz zur Bewertung verantwortungsvoller KI vorgestellt, der Erkenntnisse aus einer systematischen Literaturrecherche mit einem praktischen Framework kombiniert. Die Arbeit gibt einen Überblick über die Schlüsselaspekte von RAI und hebt die Ergebnisse der Literaturrecherche hervor.

Darüber hinaus stellt die Arbeit eine Reihe von Bewertungsmetriken vor, die speziell für den aktuellen Stand der Technik entwickelt wurden, wobei verschiedene Modelltypen, die mit Daten aus dem Gesundheitsbereich trainiert wurden, verwendet werden. Das Framework unterstützt die Evaluierung von Modellen zur Verarbeitung natürlicher Sprache, Bildverabeitung und tabellarische Modelle für Klassifizierungsaufgaben.

Darüber hinaus wird in dieser Arbeit VERIFAI, eine Implementierung des Frameworks, demonstriert, das als umfassendes Werkzeug zur Bewertung verantwortungsvoller KI-Systeme dient. Das übergeordnete Ziel dieser Arbeit ist es, einen sinnvollen Beitrag zum Diskurs über verantwortungsvolle KI zu leisten, indem Forschern und Praktikern eine wertvolle Ressource zur Verfügung gestellt wird, um ihre KI-Systeme zu verbessern. Die Arbeit schließt mit einem Ausblick auf zukünftige Entwicklungen, um das Framework zu verbessern und weiter auszubauen.

Contents

List of Figures vii

List of Tables x

1 Introduction 1

2 Technical Background: Responsible AI 4

2.1 Metrics for RAI . . . 9

2.1.1 Metrics for Fairness . . . 9

2.1.1.1 Model Performance . . . 9

2.1.1.2 Group Fairness . . . 11

2.1.1.3 Individual Fairness . . . 13

2.1.1.4 Data Metrics . . . 14

2.1.2 Metrics for Privacy . . . 15

2.1.2.1 Membership Inference . . . 15

2.1.3 Metrics for Security . . . 18

2.1.3.1 Adversarial Attacks . . . 18

2.1.4 Metrics for Explainability . . . 26

2.1.4.1 Explainable AI-Techniques . . . 26

2.2 Toolkits . . . 33

3 Methodology 36

3.1 VERIFAI -Framework . . . 37

3.1.1 Healthcare Scenario . . . 37

3.1.2 Data Sources . . . 38

3.1.3 Models . . . 39

3.1.4 Selection of Metrics . . . 41

3.1.4.1 Evaluation of Tabular Models . . . 42

3.1.4.2 Evaluation of Computer Vision Models . . . 44

3.1.4.3 Evaluation of NLP Models 46
3.1.4.4 Responsibility Score 48
3.1.5 Implementation Details 49

4 Results **55**
4.1 Implementation Results and Use Case Insights 55
4.1.1 Results of the Tabular Model Evaluation 58
4.1.2 Results of the Computer Vision Model Evaluation 70
4.1.3 Results of the NLP Model Evaluation 83
4.2 Technical Challenges 95

5 Conclusion & Future Work **97**
Declaration of Autorship 101

Bibliography **102**

List of Figures

2.1 Interdependence of Technical and Ethical Pillars in the RAI Framework . 7
2.2 Membership inference attack in the black-box setting, (Shokri et al. [2017]) 16
2.3 Training the attack model on the inputs and outputs of the shadow models, (Shokri et al. [2017]) 16
2.4 Adversarial Attack 19
2.5 Visual illustration of a ZOA Attack, (Chen et al. [2017]) 20
2.6 Integrated Gradients Example: IG Attribution Mask and Original + IG Mask Overlay (Image from Tensorflow [2023]) 27
2.7 Visualization of each token's attributions to the prediction 28

3.1 RAI Evaluation Metrics Overview (high-level) 41
3.2 VERIFAI Component Diagram (high level) 50
3.3 VERIFAI Flowchart (high level) 52
3.4 Wireframe of the User Interface 53

4.1 VERIFAI-Lifecycle 56
4.2 Select Use Case and Task 57
4.3 Select use case: Heart Disease 58
4.4 Data visualization of the Heart Disease dataset 59
4.5 Test settings for the assessment of the tabular model 59
4.6 Tabular Model Fairness Evaluation (full-page screenshot) 60
4.7 Tabular Model Fairness Evaluation (results details) 61
4.8 Tabular Model Fairness Evaluation (details) 61
4.9 Tabular Model Privacy Leakage Evaluation (full-page screenshot) 62
4.10 Tabular Model Privacy Leakage Evaluation (results details) 63
4.11 Tabular Model Privacy Leakage Evaluation (details) 63
4.12 Tabular Model Robustness Evaluation (full-page screenshot) 64
4.13 Tabular Model Robustness Evaluation (results) 65
4.14 Tabular Model Robustness Evaluation (details) 65

4.15 Tabular Model Explainability Evaluation using LIME explainer (full-page screenshot) . . . 66
4.16 Tabular Model Explainability Evaluation using LIME explainer (results) . 67
4.17 Tabular Model Explainability Evaluation using LIME explainer (truncated) 67
4.18 Tabular Model Explainability Evaluation using LIME explainer (faithfulness metric) . . . 68
4.19 Tabular Model Explainability Evaluation using LIME explainer (monotonicity metric) . . . 68
4.20 Tabular Model Responsibility Evaluation . . . 69
4.21 Select use case: image data . . . 70
4.22 Data Visualization of the Skin Cancer dataset . . . 71
4.23 Test settings: image data (full-page-screenshot) . . . 72
4.24 Computer Vision Model Fairness Evaluation (full-page-screenshot) . . . 73
4.25 Computer Vision Model Fairness Evaluation (results) . . . 73
4.26 Computer Vision Model Fairness Evaluation (confusion matrix) . . . 74
4.27 Computer Vision Model Fairness Evaluation (F1-Score) . . . 74
4.28 Computer Vision Model Privacy Leakage Evaluation (full-page-screenshot) 75
4.29 Computer Vision Model Privacy Leakage Evaluation (results) . . . 76
4.30 Computer Vision Model Privacy Leakage (AUC Score) . . . 76
4.31 Computer Vision Model Privacy Leakage (Signal Histogram of loss values) 77
4.32 Computer Vision Model Privacy Leakage (Confusion Matrix) . . . 77
4.33 Computer Vision Model Adversarial Robustness Evaluation (full-page-screenshot) . . . 78
4.34 Computer Vision Model Adversarial Robustness Evaluation (results) . . . 79
4.35 Adversarial Attacks for measuring Adversarial Robustness of the Computer Vision Model (tested epsilons = [0.0, 0.0003,0.003,0.03,0.3,1.0]) . . . 79
4.36 Adversarial Attacks through image perturbations with increasing perturbation rates (epsilon) . . . 80
4.37 Computer Vision Model Explainability Evaluation (full-page-screenshot) . 80
4.38 Computer Vision Model Explainability Evaluation (results) . . . 81
4.39 Computer Vision Model Explainability Evaluation Metrics . . . 81
4.40 Visualized explanations using the Integrated Gradients Explainer . . . 82
4.41 Computer Vision Model Responsibility Evaluation . . . 83
4.42 Use Case: Medical Reviews . . . 83
4.43 Data analysis on text data (full-page-screenshot) . . . 84
4.44 NLP Model Fairness Evaluation . . . 85

4.45 NLP Model Fairness Evaluation Results 86
4.46 Membership Inference Attack Results (details) 86
4.47 Model privacy on image data (full-page-screenshot) 87
4.48 NLP Model Privacy Leakage Evaluation (results) 88
4.49 NLP Model Membership Inference Attack Results (details) 88
4.50 NLP Model Membership Inference Attack AUC Score 89
4.51 NLP Model Adversarial Attack Robustness (details) 89
4.52 NLP Model Adversarial Attack Robustness (full-page-screenshot) 90
4.53 Best Adversarial Attack Results (TextFooler) 90
4.54 Adversarial Attack Results Comparison (details) 91
4.55 Adversarial examples using TextFooler (single example) 92
4.56 NLP Model Explainability Evaluation 92
4.57 NLP Model Explainability Evaluation 93
4.58 NLP Model Explainability Evaluation using Transformers Interpret 93
4.59 NLP Model Explainability Evaluation using Transformers Interpret 93
4.60 NLP Model Responsibility Score 94

List of Tables

3.1 Metrics for the Tabular Model 42
3.2 Metrics for the Computer Vision Model 44
3.3 Metrics for the NLP Model 46

1 Introduction

In recent years, significant advancements in the field of Artificial Intelligence (AI) have transformed the way industries and organizations operate. Breakthroughs in Machine Learning (ML) and Deep Learning (DL) techniques have enabled AI systems to perform remarkably in tasks for example in computer vision and natural language processing (Russell and Norvig [2020]).

These developments have led to the widespread adoption of AI in various sectors, including healthcare, finance, and transportation. Moreover, AI is becoming increasingly ingrained in daily life, leading to discussions about the roles of technologies like ChatGPT, especially using GPT-4 (OpenAI [2023]), as artificial generators of text, code, and more. Therefore concerns about the security, explainability, privacy, and ethics of AI systems have emerged, prompting researchers to explore methods of evaluating and ensuring responsible AI practices.

As AI systems continue to evolve, it is essential to develop metrics for measuring both discriminative models and generative models. To effectively assess the performance of various models in different scenarios and use cases, an approach for a unified framework is needed.

Therefore we have created 'VERIFAI' (e**V**aluating th**E** **R**esponsib**I**lity o**F** **AI**-systems), which builds on top of our previous work (Göllner and Tropmann-Frick [2023], Brumen et al. [2023]) and provides a comprehensive assessment of AI systems in terms of their responsibility and performance across various dimensions. By leveraging this framework, researchers and practitioners can better understand the strengths and weaknesses of their AI systems and make informed decisions to improve their overall responsibility level.

Scope of work

In this thesis, we focus on evaluating discriminative models for classification problems. Therefore the selected metrics in this scope are dedicated to this task.

Research Questions

In this work, we investigated the following research questions:

RQ1: What constitues *responsible* AI?

To address this, we first provide a definition for *Responsible AI* based on a structured literature review, identifying the key facets that compose it. Then we delve into the most important findings of each aspect.

RQ2: What are the most appropriate metrics for assessing the aspects of Responsible AI?

To answer this research question, we leverage the insights derived from the literature review, focusing on the most critical aspects identified. We then aim to identify and employ metrics to effectively evaluate these aspects on a trained model.

RQ3: To what extent are the identified metric settings applicable to various types of AI models trained on diverse datasets, such as images, text, and tabular data?

We plan to test the applicability of the identified metrics across different model architectures and datasets. While we anticipate the necessity for diverse metrics depending on the model type and training data, our objective is to establish a universally comparable evaluation process.

RQ4: How can we assess the aspects using the metrics on different model types within an application framework?

We then intend to design a suitable application architecture, incorporating use cases into an overarching scenario to demonstrate the practicality of the proposed framework for evaluating AI responsibility along a defined pipeline.

Thesis Outline

In Chapter 2, we present the results of an extensive literature review to define *Responsible AI* and identify its key aspects including ethics, security, privacy, and explainability. We also present the technical background for each of the metrics used in the implementation.

Chapter 3 describes the research methodology, including the methods for selecting appropriate metrics for the different models and data types based on compatibility and reliability, and the application architecture.

In Chapter 4, we present the results of applying our selected metrics to a variety of AI models and datasets. We discuss the outcomes of our pipeline-guided application and how it effectively evaluated the responsibility aspects. We also highlight the universality and comparability of our selected metrics across various AI models and datasets. We also present technical challenges that occurred during implementation.

The final chapter concludes the thesis, summarizing the key findings and contributions. It also presents the limitations of the current study and provides suggestions for future research directions.

2 Technical Background: Responsible AI

In recent years, a lot of research has been done to further improve artificial intelligence, which is already used in many areas of life and in industry. There is also a lot of discussion in EU politics about trust in the context of AI, and the EU has also recently produced several publications on this topic. First and foremost, AI should be a responsible technology, as it can not only do good for humanity, but unfortunately, it can also be a lethal weapon. Therefore, international regulation is necessary. On the other hand, a framework needs to be created to help companies develop their AI in accordance with regulations. Research should help both legislators and machine learning practitioners prepare for what comes next and what areas they should focus on.

In this section, we will discuss the insights gained from our previous work by incorporating the findings from our previous papers on Responsible AI aspects (Brumen et al. [2023]), the structured literature review aiming also for giving a concise definition (Göllner et al. [2023]). The section will also delve into how the aspects contribute to our understanding of Responsible AI. This will provide a foundation for the subsequent sections, where we delve into the metrics and the implementation of the VERIFAI framework and answer the first research question RQ1.

Definition

In previous work, we aimed to clarify the term Responsible AI (RAI for short) with a concise definition:

> "RAI is **human-centered** and ensures users' **trust** through **ethical** ways of decision making. The decision-making must be fair, accountable, not biased, with good intentions, non-discriminating, and consistent with societal laws and norms. Responsible AI ensures, that automated decisions are **explainable** to users while always preserving users **privacy** through a **secure** implementation."

Pillars of RAI

We have also elucidated the various aspects that constitute RAI, which is essential for developing an RAI framework. Our findings indicate that RAI necessitates a human-centered approach. Furthermore, the concept entails the incorporation of AI methodologies that emphasize ethical considerations, explainability of models, as well as privacy, security, and trustworthiness.

In the following, we summarize these aspects as they have been defined in our paper through the literature review:

Trustworthiness In the literature, trustworthiness is often connected to the way the user perceives the system's reliability. To achieve this, AI systems must prioritize data protection, provide accurate predictions under uncertainty, and offer transparent, explainable reasoning to users. Additionally, these systems should be usable and accessible, act reliably "as intended" in their applications, and be perceived as fair and useful. By focusing on these key aspects, developers can create RAI solutions that foster user trust and deliver value across a wide range of sectors, benefiting both users and society as a whole.

Ethics Among the key requirements for ethical AI, fairness stands out as the most critical aspect according to the literature. Ensuring AI systems are non-biased and non-discriminating in all aspects of their operation is crucial in fostering trust and acceptance. Alongside fairness, accountability is essential, with AI systems justifying their decisions and actions transparently. Sustainability is another vital requirement, with AI systems designed to consider long-term consequences and align with Sustainable Development Goals. Lastly, compliance with robust laws and regulations guarantees that AI systems operate within legal and ethical boundaries.

Privacy Privacy and security techniques play a vital role in ensuring RAI systems, particularly when handling sensitive data. Compliance with regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), the Children's Online Privacy Protection Act (COPPA), and the General Data Protection Regulation (GDPR), is crucial to protect user data, and emerging technologies like Federated Learning can help in this regard. Additionally, implementing proper organizational processes is essential in

complementing these techniques, ensuring robust data protection. Privacy and security measures should be employed based on the tasks executed on the data and specific user transactions.

Privacy is a critical concern in the age of AI, as machine learning models can inadvertently reveal sensitive information about users or expose them to reconstruction attacks and membership inference attacks. Employing hybrid Privacy-Preserving Machine Learning (PPML) approaches allows for optimal trade-offs between ML task performance and privacy overhead. Utilizing techniques that minimize communication and computational costs is particularly important in distributed approaches, enhancing efficiency and scalability.

Security On the other hand, security threats in the branch of machine learning must also be addressed. These threats include stealing the model or sensitive information from the user, poisoning attacks, which involve manipulating the training data to compromise the model's performance, and adversarial attacks, where adversaries create malicious input samples designed to deceive the model and cause incorrect predictions. The field of AI security is rapidly evolving, with researchers developing new methods and defenses to counter these threats.

Despite being separate aspects, privacy and security are strongly interdependent in the context of AI systems. Ensuring privacy helps to mitigate security threats, while robust security measures protect sensitive data and user privacy. By focusing on these key requirements, developers can build AI solutions that prioritize privacy and security while delivering reliable performance.

Explainability Explainable AI has emerged as a critical aspect of developing RAI systems, ensuring that users can understand and interact effectively with these complex technologies. Central to explainable AI is a human-centered approach, where user interaction plays a crucial role in shaping the design and functionality of the system. To achieve this, explanations must be tailored to the user's needs and target group, ensuring that they are relevant and accessible to diverse audiences. An intuitive user interface and experience are also essential components of explainable AI, as they facilitate comprehension and engagement with the system. This can be achieved by presenting the results in a visually understandable language that resonates with users, allowing them to grasp the system's workings and rationale with ease. Explainability is not only a functional

requirement but also a measure of the system's performance in terms of its ability to communicate its decision-making process effectively. This non-functional requirement highlights the importance of understanding the AI system's inner workings as an integral part of its overall efficacy. Finally, the impact of explanations on the decision-making process must be considered, as explainable AI systems should aim to enhance users' ability to make informed choices based on the provided explanations.

Human-centeredness Human-centeredness is a fundamental aspect of RAI, emphasizing the need to consider user interaction and understanding when designing AI systems. This approach places the human user at the center of the AI experience, ensuring that the technology is not only efficient but also accessible and comprehensible to its users. One essential concept in human-centered AI is the Human-in-the-loop (HITL) which involves incorporating human input and feedback throughout the AI system's development and decision-making processes. This approach ensures that AI technologies are not solely reliant on algorithms but also benefit from human knowledge, experience, and intuition. By involving humans in these processes, AI systems can be better aligned with human values, expectations, and ethical considerations.

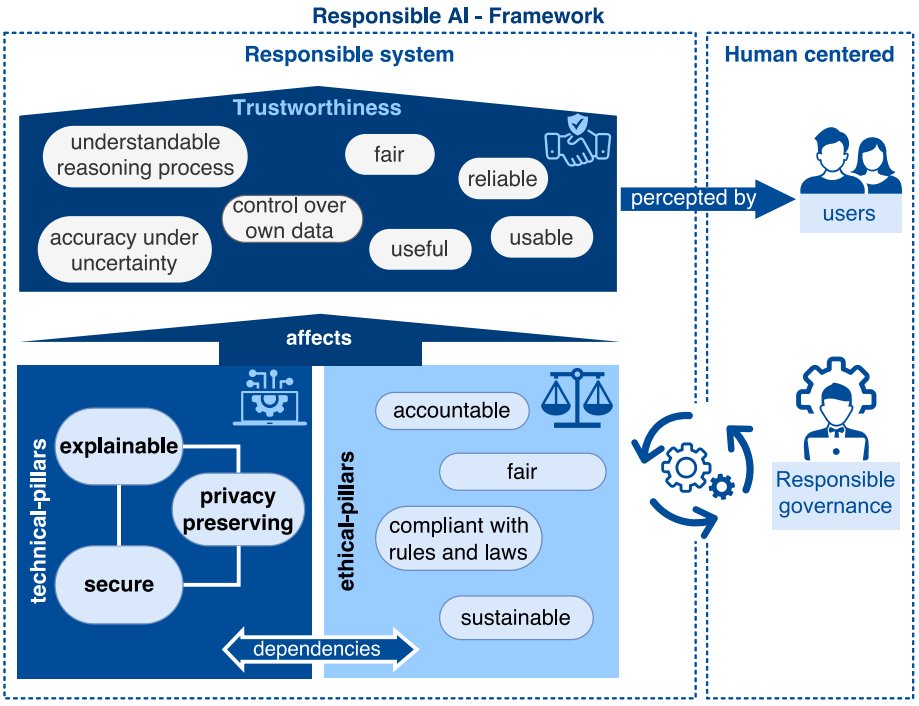


Figure 2.1: Interdependence of Technical and Ethical Pillars in the RAI Framework

Figure 2.1 highlights the interdependence between ethical and technical requirements in RAI, with trust being the users' perception of AI systems. Ethical pillars such as accountability, fairness, sustainability, and compliance are essential for meeting ethical requirements. Explainability methods must also respect privacy and security, as they are interconnected. RAI involves both system-side and developer-side considerations, with the latter continuously monitoring and maintaining the system using special metrics. Human-centered AI and the HITL- approach plays crucial roles by including human expertise and perspective. As a dynamic and interdisciplinary process, RAI requires attention and care throughout the entire system lifecycle.

2.1 Metrics for RAI

Based on the knowledge gained from the SLR, we conducted further research on finding metrics for properly measuring the degree of responsibility based on the aspects mentioned above. In this section, we will present each of the metrics used for the model evaluation in the implementation.

2.1.1 Metrics for Fairness

As fairness plays the most important role in the field of ethical AI (see chapter 2), we focus on this topic and its corresponding evaluation metrics. In the following, we define and explain each of the metrics occurring in our implementation. The metrics are grouped into *Model Performance*, *Group Fairness*, *Individual Fairness*, and *Data Metrics*.

2.1.1.1 Model Performance

This section will shortly define all the necessary model performance metrics that occur in our implementation.

Basic Terms from the confusion matrix True Positive (TP): The number of instances where the model correctly predicts the positive class. True Negative (TN): The number of instances where the model correctly predicts the negative class. False Positive (FP): The number of instances where the model incorrectly predicts the positive class. False Negative (FN): The number of instances where the model incorrectly predicts the negative class.

True Negative Rate (TNR) can also be called *specificity*. Mathematically, the TNR can be defined using the terms from the confusion matrix: $\text{TNR} = \frac{\text{TN}}{\text{TN+FP}}$

True Positive Rate (TPR) can also be referred to as *sensitivity* or *recall*. is the proportion of true positive predictions out of all actual positive instances. Mathematically, the TPR can be defined using the terms from the confusion matrix: $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$

Accuracy is a common metric used to evaluate the performance of classification models. It is defined as the proportion of correctly classified instances out of the total number of instances in the dataset. Mathematically, the accuracy can be defined using the terms from the confusion matrix: Accuracy = $\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$

Balanced Accuracy is a metric used to evaluate the performance of classification models in scenarios where there is an imbalance in the distribution of classes in the dataset. It is particularly useful for assessing the fairness of machine learning models when dealing with minority classes or groups. Balanced accuracy is defined as the average of the TPR and the TNR and can therefore be calculated using the following formula: Balanced Accuracy = $\frac{\text{TPR+TNR}}{2}$

In the context of fair machine learning, using balanced accuracy helps to ensure that the model is evaluated in a way that is less biased towards the majority class. This is important because standard accuracy can be misleading, as it can indicate good performance even if the model is only predicting the majority class correctly and neglecting the minority class. Balanced accuracy, on the other hand, gives equal importance to the performance on both classes, making it a more suitable metric for measuring fairness in classification tasks (Brodersen et al. [2010]).

Precision is the proportion of true positive predictions out of all positive predictions made by the model. $Precision = \frac{TP}{(TP+FP)}$

F1-Score is a metric used to evaluate the performance of classification models, especially in cases where there is an imbalance in the distribution of classes. It is a harmonic mean of precision and recall. F1 Score = $\frac{2\times(\frac{\text{TP}}{\text{TP+FP}}\times\frac{\text{TP}}{\text{TP+FN}})}{\frac{\text{TP}}{\text{TP+FP}}+\frac{\text{TP}}{\text{TP+FN}}}$

Area Under the Curve (AUC) is a performance metric for evaluating the effectiveness of binary classification classifiers, specifically focusing on the Receiver Operating Characteristic (ROC) curve. AUC provides a single value that represents the classifier's ability to discriminate between positive and negative classes across all possible thresholds. AUC ranges from 0 to 1, with 1 indicating the perfect classification and 0.5 representing a random classifier.

Mathematically, AUC is calculated as the integral of the ROC curve, which is a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) for varying decision thresholds. In discrete cases, AUC can be estimated using the trapezoidal rule:

$$\text{AUC} \approx \sum_{i=1}^{N} \frac{(FPR(i) - FPR(i-1))(TPR(i) + TPR(i-1))}{2}$$

where i ranges from 1 to N (number of points on the ROC curve), and FPR(i) and TPR(i) are the False Positive Rate and True Positive Rate, respectively, at the i-th threshold.

2.1.1.2 Group Fairness

Group Fairness, which compares the statistical similarities of predictions relative to known and discrete protected groupings (e.g. Gender, Age, or Ethnicity). The metrics help to ensure that the classifier is equally accurate for both groups, which is important when assessing fairness in high-stakes decision-making scenarios. (see Allen et al. [2020])

Statistical Parity Difference (SPD) is a fairness metric used to evaluate whether a classifier treats different groups, such as privileged and unprivileged groups, equally. Mathematically, the Statistical Parity Difference (SPD) can be defined as:

$$\text{SPD} = P(\hat{Y} = 1|A = \text{privileged}) - P(\hat{Y} = 1|A = \text{unprivileged})$$

$\hat{Y}$ represents the predicted outcome, and A denotes the group membership (privileged or unprivileged). $P(\hat{Y} = 1|A = \text{privileged})$ is the probability of a positive outcome for the privileged group, whereas $P(\hat{Y} = 1|D = \text{unprivileged})$ is the probability of a positive outcome for the unprivileged group. An SPD value of 0 indicates perfect statistical parity, meaning that the classifier treats both groups equally. Positive values mean that the privileged group has a higher probability of receiving a positive outcome, while negative values indicate the opposite (Caton and Haas [2020], Mehrabi et al. [2021]).

Disparate Impact Ratio (DIR measures the ratio of the probability of receiving a positive outcome for the unprivileged group to that of the privileged group. A value close to 1 indicates better fairness, while values significantly different from 1 suggest the presence of bias in the AI system's decisions.

Mathematically, the Disparate Impact Ratio can be defined as:

$$\text{DIR} = \frac{P(\hat{Y} = 1|A = \text{unprivileged})}{P(\hat{Y} = 1|A = \text{privileged})}$$

$\hat{Y}$ represents the predicted outcome (1 for a positive outcome and 0 for a negative outcome). D denotes the group membership (privileged or unprivileged). $P(\hat{Y} = 1|A = \text{privileged})$ is the probability of a positive outcome for the privileged group. $P(\hat{Y} = 1|A = \text{unprivileged})$ is the probability of a positive outcome for the unprivileged group. A DIR value of 1 indicates perfect demographic parity, meaning that the classifier treats both groups equally in terms of positive outcomes. Values less than 1 imply that the unprivileged group is less likely to receive a positive outcome, while values greater than 1 indicate the opposite. (Caton and Haas [2020], Mehrabi et al. [2021])

Equal Odds Difference (EOD) assesses whether a classifier maintains equal false positive rates (FPR) and true positive rates (TPR) for different groups, such as privileged and unprivileged groups. This metric aims to ensure that the classifier is equally accurate for both groups and is particularly relevant when assessing the fairness of decisions with significant consequences. Mathematically, the Equal Odds Difference (EOD) can be defined as:

$$\text{EOD} = \max((\text{TPR}_\text{u} - \text{TPR}_\text{p}) + (\text{FPR}_\text{u} - \text{FPR}_\text{p}))$$

TPR_p is the true positive rate for the privileged group. TPR_u is the true positive rate for the unprivileged group. FPR_p is the false positive rate for the privileged group. FPR_u is the false positive rate for the unprivileged group. An EOD value of 0 indicates perfect equal odds, meaning that the classifier has the same TPR and FPR for both groups. Positive values imply that there is a difference in TPR or FPR between the groups, with larger values indicating greater disparities.(Allen et al. [2020], Caton and Haas [2020], Mehrabi et al. [2021]).

Equal Odds Ratio (EOR) is a fairness metric that evaluates the equality of true positive rates (TPR) and false positive rates (FPR) for different groups, such as privileged and unprivileged groups, in a classifier. The EOR considers the ratio of the TPR and FPR between unprivileged and privileged groups.

$$\text{EOR} = \min\left(\frac{\text{FPR}_\text{u}}{\text{FPR}_\text{p}}\frac{\text{TPR}_\text{u}}{\text{TPR}_\text{p}}\right)$$

FPR_u refers to the false positive rate for the unprivileged group. FPR_p is the false positive rate for the privileged group and Here TPR_p is the true positive rate for the privileged group. and TPR_u is the true positive rate for the unprivileged group Allen et al. [2020]. An EOR value of 1 indicates perfect equal odds, meaning that the classifier has the same TPR and FPR for both groups. Values significantly different from 1 suggest the presence of bias in the classifier's decisions (Caton and Haas [2020], Mehrabi et al. [2021]).

2.1.1.3 Individual Fairness

Individual fairness exists if "similar" individuals (ignoring the protected attribute) are likely to have similar predictions (see Allen et al. [2020]).

Between-Group Generalized Entropy Error (BG-GEE) is a fairness metric used to measure the disparities between different groups in classification problems. It evaluates the fairness of a classifier by comparing the probability distributions of predicted scores across various groups (e.g., demographic groups). It is based on the concept of Generalized Entropy (GE) and is specifically designed to capture disparities across multiple groups.

The BG-GEE is defined as:

$$\text{BG-GEE}(\alpha) = \frac{1}{\alpha(\alpha-1)}\left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{p_i}{\bar{p}}\right)^{\alpha} - 1\right]$$

where α is a parameter that determines the sensitivity of the metric to disparities within and between groups. When $\alpha = 0$, the metric is equivalent to the Between-Group Variance. When $\alpha = 1$, it corresponds to the Theil Index, and when $\alpha = 2$, it becomes the Coefficient of Variation. p_i is the predicted score (e.g., probability of belonging to

the positive class) for the i-th individual. N is the total number of individuals and $\bar{p}$ is the average predicted score across all individuals. By calculating BG-GEE for different values of α, we can obtain insights into the disparities between groups at various levels of granularity. This metric is particularly useful when evaluating fairness in classification models, as it allows us to identify and address potential biases in our predictions (Speicher et al. [2018]).

Consistency Score represents an individual fairness metric from Zemel et al. [2013] that measures how similar the labels are for similar instances. It compares a model's classification prediction of a given data item x to its k-nearest neighbors, kNN (x):

$$C = 1 - \frac{1}{Nk} \sum_{n} \left| \hat{y}_n - \sum_{y \in \text{kNN}(x_n)} \hat{y}_j \right|$$

N represents the total number of data items, k is the number of nearest neighbors considered, $\hat{y_n}$ is the model's prediction for the data item x_n, and $\hat{y_j}$ is the model's prediction for the j-th nearest neighbor of x_n ($y \in \text{kNN}(x_n)$).

This consistency score calculates the absolute difference between the model's prediction for a data item x_n and the sum of predictions for its k-nearest neighbors. These differences are then summed over all N data items and divided by $N * k$. Finally, the result is subtracted from 1 to obtain the consistency score, N.

The consistency score ranges from 0 to 1, with higher values indicating more consistent predictions between a data item and its nearest neighbors. In the context of fairness, a model with a high consistency score tends to make similar predictions for similar data items, regardless of the sensitive attributes, which can be an indication of a fair model.

2.1.1.4 Data Metrics

Prevalence of Privileged Class refers to the proportion of individuals in the dataset who belong to a privileged group. In studies of fairness and bias, it is essential to identify and distinguish between privileged and unprivileged classes or groups. A privileged group typically experiences advantages or benefits due to their social or demographic attributes, while an unprivileged group may face disadvantages or discrimination (Allen et al. [2020]).

2.1.2 Metrics for Privacy

> „Machine learning algorithms are under increasing scrutiny from regulatory authorities, due to their usage of a large amount of sensitive data. In particular, vulnerability to membership inference attacks.“ (Ye et al. [2022]).

2.1.2.1 Membership Inference

Membership inference attacks were first described by Shokri et al. [2017]. Since then, a lot of research has been conducted in order to make these attacks more efficient, to measure the membership risk of a given model, and to mitigate the risks. We first give a short definition of what membership inference actually means and how it can be used in order to violate individual privacy.

Definition: Membership Inference

> Given a model f and a data point x_i, determine whether or not x_i was part of the model's training dataset X or not (Shokri et al. [2017]).

We investigate this question in the most difficult setting, where the adversary's access to the model is limited to black-box queries that return the model's output on a given input. In summary, we quantify membership information leakage through the prediction outputs of machine learning models.

One way to create a successful MIA is to observe the loss values for different data points and set a threshold to distinguish between member and non-member data points. This threshold is called the *loss threshold*. A well-chosen loss threshold is essential for the success of the attack because it helps the attacker minimize false positives and false negatives when inferring membership. If the loss threshold is too low, the attacker may incorrectly identify non-member data points as members (false positives). Conversely, if the loss threshold is too high, the attacker may fail to identify actual member data points (false negatives).

Membership Inference Attack To perform a basic membership inference attack, the authors train an attack model whose purpose is to distinguish the target model's behavior on the training inputs from its behavior on the inputs that it did not encounter during training.

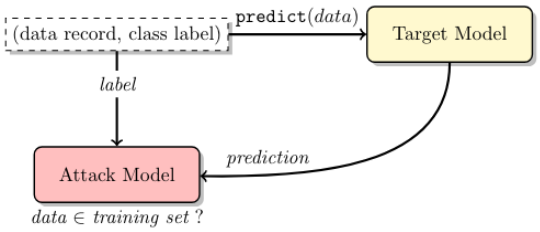


Figure 2.2: Membership inference attack in the black-box setting, (Shokri et al. [2017])

The adversary submits a data record to the target model and receives the model's prediction for that particular record. This prediction consists of a probability vector, with each element corresponding to the likelihood of the record belonging to a specific class. Subsequently, the attack model is provided with both the prediction vector and the target record's label, enabling it to deduce whether the record was included in or excluded from the target model's training dataset.

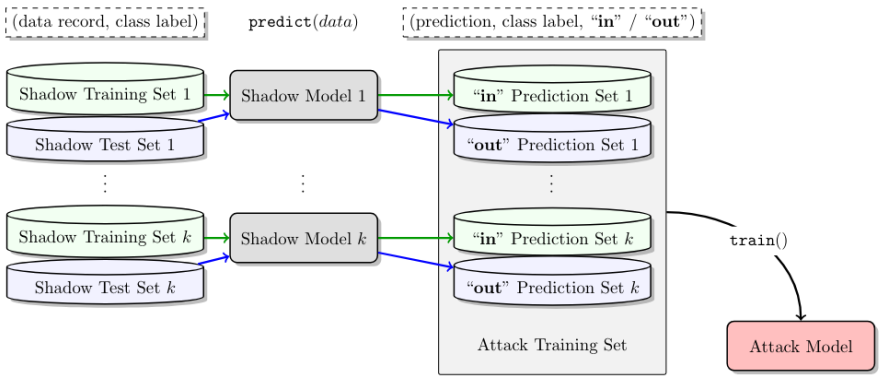


Figure 2.3: Training the attack model on the inputs and outputs of the shadow models, (Shokri et al. [2017])

MIA via shadow metric A method to evaluate this involves constructing a binary meta-classifier. To train this meta-classifier, shadow models are created. These models are designed to replicate the functionality of the original machine learning model. Nonetheless, their training data, as well as the ground truth, for binary classification tasks, are accessible to the attacker.

Leveraging the information about the shadow models' training data, input-output pairs for the meta-classifier can be generated. This enables the meta-classifier to learn the task of discerning between members and non-members based on a machine learning model's performance on these instances (Shokri et al. [2017], Ye et al. [2022]).

Model-dependent MIA via population data is another approach described by Ye et al. [2022]. This Attack does not use a separate classifier or shadow model like the Shadow Attack. Instead, it directly leverages the target model and population data to determine the threshold for membership inference. The key difference is that it exploits the dependency of the loss threshold on the specific target model, whereas Shadow Attack uses shadow models as a proxy.

Here's the overall process: 1. Given a target model, calculate the loss values for a set of population data points. The population data is assumed to be member data (training data) to establish a baseline for what the loss values look like for data points that are known to be members. 2. Determine the α-percentile of the loss values calculated in step 1. This is the threshold, $c\alpha(\theta)$, that represents the desired false positive rate of α for the specific target model. A low false positive rate is desirable in this context, as it means that fewer non-member data points will be incorrectly identified as members by the attack. 3. To perform the attack on a new set of data points (which could include both members and non-members), calculate the loss values for these data points using the target model. 4. Compare the calculated loss value with the threshold. If the loss value is less than or equal to the threshold, it is inferred as a member. Otherwise, it is inferred as a non-member.

The attack simplifies the process by directly working with the target model and population data, avoiding the need to train shadow models or a separate classifier. However, this also means that it is only able to exploit information about the target model and does not take into account any uncertainty regarding the target record. This means that Attack P cannot exploit any differences or patterns in the target records themselves when determining membership.

In comparison, the shadow model-based attack can potentially exploit differences or patterns in the target records themselves when determining membership. Since multiple shadow models are trained on different datasets, a separate classifier is trained to differentiate between member and non-member data points based on the loss values or confidence scores generated by these shadow models.

Since the classifier in shadow model-based attack learns from a more diverse set of models and data points, it may capture patterns in the target records that are indicative of membership or non-membership. This can lead to a more accurate membership inference attack compared to the population Attack, which only works with the target model and population data. However, it's worth noting that a shadow model-based attack is more computationally expensive due to the need to train multiple shadow models and a separate classifier.

Evaluation of Attack Performance The quantification of the attacker's average performance on general targets can be accomplished using two metrics: its true positive rate (TPR), and its false positive rate (FPR), over the random member and non-member data of random target models. We use the ROC curve to capture the tradeoff between the TPR and FPR of an attack, as its threshold c_α is varied across different FPR tolerance α. The AUC score then measures the strength of an attack (Ye et al. [2022])

2.1.3 Metrics for Security

As mentioned in section 2 the field of AI security is rapidly evolving nowadays, with researchers developing new methods and defenses to counter these threats. Szegedy et al. [2013] first noticed the existence of adversarial examples in image classification, showing that especially neural networks are surprisingly vulnerable. Jankovic and Mayer [2022] also emphasizes, that the security of deep learning models has become a major concern, indirectly also affecting safety. In this work, we focus on dealing with measuring the *adversarial robustness* of a model to measure its vulnerability against *adversarial attacks* which will be explained below.

2.1.3.1 Adversarial Attacks

Machine learning models, particularly neural networks, are susceptible to adversarial examples. These examples involve carefully crafted, subtle modifications to the input data with the goal of compromising the model's prediction accuracy. As a result, the model may produce incorrect predictions, posing significant security and safety concerns. Various attack types exist, and not every attack is compatible with all models due to differences in their underlying structures or mechanisms. Given that our framework incorporates multiple model types, we have explored efficient techniques to evaluate their

robustness against adversarial attacks, taking into account the compatibility between the attack methods and the specific models being used.

Definition: Adversarial Attack Let $x_0 \in \mathbb{R}^d$ be a data point belonging to class C_i. Define a target class C_t. An adversarial attack is a mapping $A : \mathbb{R}^d \to \mathbb{R}^d$ such that the perturbed data $x_0 = A(x_0)$ is misclassified as C_t.

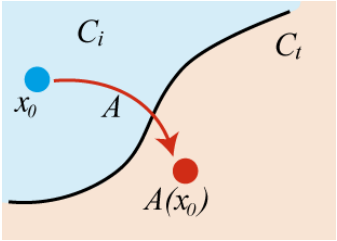


Figure 2.4: Adversarial Attack

Categories of Adversarial Attacks Adversarial attacks can be classified into three categories according to the threat model: white-box, gray-box, and black-box attacks. The distinctions among these categories are determined by the adversaries' level of knowledge about the target model. In white-box attacks, adversaries are assumed to possess complete information about the target model, including its architecture and parameters, enabling them to craft adversarial samples directly using any available techniques. Gray-box attacks, on the other hand, limit adversaries' knowledge of the structure of the target model, without access to its parameters. Lastly, in black-box attacks, adversaries must rely on query access to generate adversarial samples, as they lack any specific information about the target model's architecture or parameters Ren et al. [2020].

Distance Metrics The distance metrics used for measuring the distortion in adversarial attacks indicate the extent of the perturbation added to the input data. These metrics help quantify the similarity between the original input and the adversarial example. By definition, an adversarial sample $x0$ should be close to a benign sample x under a specific distance metric.

The L_2 distance, often called the Euclidean distance, calculates the square root of the sum of the squared differences between corresponding elements of the two vectors. For adversarial attacks, the L2 distance represents the magnitude of the perturbation vector.

A smaller L_2 distance suggests that the perturbation is less perceptible, making the adversarial example more stealthy. The formula for the L2 distance is as follows:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

where x and y are the two vectors, and n is the number of dimensions in Euclidean space.

The most popular distance metric is the L_∞ distance, which measures the maximum absolute difference between corresponding elements of two vectors. In the case of adversarial examples, the maximum element-wise difference between benign and adversarial samples (Ren et al. [2020]). The formula for the L_∞ distance is as follows:

$$d(x, y) = max|x_i - y_i|$$

where x and y are the two vectors, and n is the number of dimensions in Euclidean space.

Zeroth Order Optimization Attack (ZOA) is a type of black-box adversarial attack that relies on zeroth-order optimization methods to approximate gradients, enabling the generation of adversarial examples. These attacks do not require direct access to the model's gradients or parameters, Instead, it typically involves a trial-and-error process of perturbing the input to the model and observing the resulting output, in order to iteratively refine the adversarial example. This makes them applicable to a wide range of machine learning models, including deep learning models like neural networks and conventional machine learning methods like for example Random Forests.

Figure 2.5: Visual illustration of a ZOA Attack, (Chen et al. [2017])

One approach of the ZOA is presented in Chen et al. [2017]. The authors proposed a black-box attack using stochastic coordinate descent to perform numerical estimation of gradients. It aims to minimize an objective function that combines the distortion

introduced by the perturbation (measured using L_2 distance) and the difference between the model's logits for the correct class and the target class.

Fast Gradient Sign Method (FGSM) Goodfellow et al. [2014] invented the Fast Gradient Sign Method (FGSM), an attack strategy for generating adversarial images, aimed at fooling machine learning models, particularly deep neural networks. It was introduced by Ian Goodfellow and his colleagues in 2014. The main idea behind FGSM is to make minimal perturbations to an input image, which can cause the model to misclassify it. The attack is fast and computationally efficient, making it a popular choice for adversarial example generation.

The core of the Fast Gradient Sign Method can be described by the following formula:

$$x' = x + \epsilon \cdot sign(\Delta_x J(\theta, x, y))$$

Where $\Delta_x J$ is the gradient of the model's loss function with respect to the original input x (an image or a tensor of input images) that we want to perturb. Y is the true label vector for x and θ is the model parameter vector. The gradient provides information about the direction in which the input should be modified to increase the loss, and thus mislead the classifier. From the gradient vector (which is as long as the vector of the input pixels) we only need the sign: The sign of the gradient is positive (+1) if an increase in pixel intensity increases the loss and negative (-1) if a decrease in pixel intensity increases the loss. This is then multiplied by the ϵ to control the perturbation's size. Finally, the perturbed image x' is generated by adding the scaled gradient sign to the original input image x.

Adversarial examples generated using FGSM often transfer well across different models, meaning that an adversarial example crafted to fool one model might also fool another model with a similar architecture or trained on similar data.

Projected Gradient Descent(PGD) The Projected Gradient Descent (PGD) attack is an iterative method for generating adversarial examples It was proposed by Madry et al. [2017] as a part of their work on adversarial training. PGD is designed to address the limitations of FGSM by incorporating multiple iterations and random initialization, making it more effective against models with highly non-linear decision boundaries. The goal of the PGD attack is basically to find a point in the region (L_∞ -ball) around

the input x that maximizes the loss. The process can mathematically be described as follows:

$$x_0 \in B_{(x,\varepsilon)} x^{t+1} = \Pi_{B(x,\varepsilon)} \left(x^t + \alpha \cdot \text{sgn}(\nabla_x L(\theta, x, y)) \right)$$

Here, $B_{(x,\epsilon)}$ represents an L_∞ -ball around the input x with a radius of ϵ. This set constrains the allowed perturbations to ensure they are within the range of $\pm\epsilon$. $\Pi_{B(x,\epsilon)}$ is the projection operator that projects the perturbed input back onto the L-infinity ball $B(x, \epsilon)$.

DeepFool Attack Moosavi-Dezfooli et al. [2016] developed an iterative algorithm that aims to find the smallest perturbation required to cross the decision boundary of a deep learning model. It does so by linearizing the model around the input image and iteratively perturbing the image in a direction that minimizes the distance to the decision boundary. (The L_∞ variant of the attack constrains the perturbation to lie within a specified norm bound.)
The algorithm involves the following steps: Starting from the original input image x, while the image is still classified correctly, we perform the following steps: We compute the gradients of the classifier's output with respect to the input image. Then Linearize the classifier around the current image, approximating the decision boundary. Calculate the minimal perturbation required to cross the approximated decision boundary. Finally, we update the image by adding the calculated perturbation, while ensuring that the total perturbation stays within the L-infinity norm bound. We continue iterating until the image is misclassified or a maximum number of iterations is reached.

It demonstrates the vulnerability of deep learning models to small adversarial perturbations by focusing on finding the minimum perturbation required to fool the model.

Additive Uniform Noise Attack This Attack generates adversarial examples by adding uniformly distributed noise to the input image within a specified L-infinity norm bound (Goodfellow et al. [2014]). This attack serves as a baseline method for adding random noise to an image, without exploiting any specific properties of the targeted model.

The algorithm for the Additive Uniform Noise Attack works as follows: Noise is sampled from a uniform distribution within the range of $[-\epsilon, \epsilon]$ for each pixel. Then this sampled noise is added to the input image, creating a perturbed image. Finally, the perturbed

image must be clipped to ensure it stays within the valid pixel range (e.g., [0, 1] or [0, 255]).

This attack serves as a basic benchmark for evaluating the robustness of deep learning models against adversarial perturbations. It is often used as a reference point when comparing the effectiveness of more advanced attacks like FGSM, PGD, or others that exploit gradient information or other properties of the targeted model.

Adversarial Text Attacks

In recent years, adversarial attacks have also gained significant attention in the field of natural language processing. The primary objective of these attacks is to generate adversarial examples that can effectively fool deep learning classifiers while preserving the semantics and readability of the original text.

Text Bugger Li et al. [2018] developed a method designed to create adversarial examples in the text domain. It exploits the unique properties of text data, such as the discrete nature of words and the inherent linguistic structures, to craft adversarial examples that are both effective and imperceptible to humans. It can be applied to various deep learning models, including LSTM, CNN, and a combination of both.

The TextBugger algorithm consists of several steps: First, we identify important words in the input text by calculating the gradients of the classifier's output with respect to the input words (word saliency). The higher the gradient, the more important the word is for the classification decision. Mathematically, this can be represented as: Let x be the input text, y be the true class, and $L(\theta, x, y)$ be the loss function for a classifier with parameters θ. Calculate the gradients with respect to the input words: $\Delta x L(\theta, x, y)$. Secondly generate candidate perturbations by applying word-level or character-level modifications to the important words, such as substitution, insertion, deletion, or swapping. Next, we select the most effective perturbations based on their impact on the classifier's output while considering the readability and semantics of the resulting adversarial text. Finally, we apply the selected perturbations to the input text, creating an adversarial example ($x_{adv} = x + perturbation$)

DeepWordBug The key idea behind the *DeepWordBug* algorithm by Gao et al. [2018] is to identify important words in the input text sequence and apply character-level transformations that maximize the change in the classifier's prediction confidence while maintaining visual similarity. The most important mathematical aspects of the DeepWordBug algorithm include:
Step 1: Token Scoring Function and Ranking:
Replace-1 Score: This score measures the change in the classifier's output probabilities when a token is replaced with other tokens sampled from a predefined distribution. The higher the change in the output probabilities, the more important the token is considered to be.
Temporal Head Score: This score is based on the idea that tokens appearing earlier in the text sequence contribute more to the classifier's decision than those appearing later. The score is calculated by removing tokens sequentially from the beginning of the text sequence and measuring the change in the classifier's output probabilities.
Temporal Tail Score: This score is based on the idea that tokens appearing later in the text sequence contribute more to the classifier's decision than those appearing earlier. The score is calculated by removing tokens sequentially from the end of the text sequence and measuring the change in the classifier's output probabilities.
Combination Score: This score is a combination of the Replace-1 Score, Temporal Head Score, and Temporal Tail Score. The combination is done by averaging the ranks of the tokens based on their scores from each of the other three scoring functions.
After calculating the scores, the tokens are ranked by their importance.
Step 2. Token transformer: The second step is to apply transformations to the most important tokens to generate adversarial examples. The authors propose several character-level transformations, including:
a. Swap: Swap two adjacent characters in the token. *b. Substitute:* Replace a character in the token with a visually similar character. *c. Insert:* Insert a visually similar character adjacent to an existing character in the token. *d. Delete:* Remove a character from the token.

TextFooler Jin et al. [2020] present an approach for generating adversarial examples called TextFooler, which is a black-box attack that aims to deceive various NLP models, including BERT, by generating human-readable adversarial examples with minimal modifications to the input text. The most important aspects of the TextFooler algorithm are:

Step 1. Word importance ranking: For a given input text, a sentence of n words $X = w_1, w_2, ..., w_n$, we observe that only some keywords act as influential signals for the prediction model F. TextFooler first computes the importance of each word in the text. The importance score is calculated based on the decrease in the model's confidence when the word is removed from the input. Words with higher importance scores contribute more to the model's classification decision.
Step 2: Word Transformer: Given a word $w_i \in X$ with a high importance score obtained in Step 1, the aim is to find a suitable replacement word that meets specific criteria: similar semantic meaning, fitting the surrounding context, and forcing the target model to make wrong predictions: *Synonym Extraction:* Gather a candidate set $CANDIDATES$ for all possible replacements of the selected word w_i $CANDIDATES$ is initialized with N closest synonyms according to the cosine similarity between w_i and every other word in the vocabulary, using word embeddings from Mrkšić et al. [2016]. In the paper, they set N to 50 and a cosine similarity threshold δ to 0.7. *POS Checking:* Keep only the words in the $CANDIDATES$ set that have the same part-of-speech (POS) as w_i This ensures that the grammar of the text is mostly maintained. *Semantic Similarity Checking:* For each remaining word $c \in CANDIDATES$, substitute it for w_i in the sentence X, and obtain the adversarial example $X_{adv} = w_1, ..., w_{i-1}, c_{i+1}, ..., w_n$. Use the target model F to compute the corresponding prediction scores $F(X_adv)$. Calculate the sentence semantic similarity between the source X and adversarial counterpart X_{adv} using Universal Sentence Encoder (USE) to encode the two sentences into high-dimensional vectors and compute their cosine similarity score. The words resulting in similarity scores above a preset threshold ϵ are placed into the final candidate pool $FINCANDIDATES$. *Finalization of Adversarial Examples:* In the final candidate pool $FINCANDIDATES$ if there exists any candidate that can already alter the prediction of the target model, choose the word with the highest semantic similarity score among these winning candidates. If not, select the word with the least confidence score of the label y as the best replacement word.

Probability Weighted Word Saliency (PWWS) The algorithm of Ren et al. [2019] generates adversarial examples by perturbing the original text to fool the target model while preserving the semantics of the input text. By iteratively replacing words with their best replacements, the algorithm aims to minimize the number of modifications while maximizing the impact on the model's prediction. As a result, the generated

adversarial examples can effectively evade deep learning classifiers while maintaining the overall meaning and readability of the original text.

The main steps are as follows: We first calculate the word saliency score $S(x_{(0)}, w_i)$ for each word w_i in the input text $x(0)$. The saliency score is based on the PWWS: $PWWS(w_i) = \Delta P(w_i) * P(w_i)$, where $\Delta P(w_i)$ represents the change in probability of the true class when w_i is removed. For each word w_i, we then obtain a set of synonyms $\mathbb{L}_i$ and select the best replacement w_i^* from $\mathbb{L}_i$ based on its semantic similarity and impact on the model's prediction. Then we reorder the words w_i such that the highest impact on the model's prediction comes first (i.e., $H(x, x_1^*, w_1) > \cdots > H(x, x_n^*, w_n)$). Iteratively we replace words in the input text with their best replacements w_i^*, creating a new adversarial example $x(i)$, and stop when the model's prediction changes (i.e., $F(x_{(i)}) \neq F(x_{(0)})$).

2.1.4 Metrics for Explainability

The following section will first deal with the XAI techniques and secondly with evaluating the quality and effectiveness of generated explanations.

2.1.4.1 Explainable AI-Techniques

Among the various XAI techniques, Integrated Gradients and LIME (Local Interpretable Model-agnostic Explanations) have gained popularity due to their effectiveness in providing meaningful explanations for different types of data, including images, text, and tabular data. As we use them in our implementation these methods will be explained in more detail in the following section.

Integrated Gradients Integrated Gradients is a powerful XAI technique by Sundararajan et al. [2017] specifically developed for deep learning models, such as neural networks. It is particularly useful for interpreting image-based models, such as convolutional neural networks (CNNs). Integrated Gradients work by attributing the output of the model to its input features by computing the gradients of the output with respect to each input feature. This process enables the identification of critical features in the input image that contribute to the model's prediction, effectively highlighting regions of interest and offering valuable insights into the model's decision-making process. The authors

propose three key axioms that a good attribution method should satisfy: *a. Sensitivity:* If the model's output changes due to a change in a single feature, the attribution should reflect that change. *b. Implementation Invariance:* The attributions should be consistent across different functionally equivalent implementations of the model. *c. Completeness:* The sum of attributions for all input features should equal the difference between the model's output for the input instance and the baseline input.

Formally, suppose we have a function $F : \mathbb{R}^n \to [0, 1]$ that represents a deep network. Specifically, let $x \in \mathbb{R}^n$ be the input at hand, and $x' \in \mathbb{R}^n$ be the baseline input. For image networks, the baseline could be the black image, while for text models it could be the zero embedding vector. We define a straight-line path between x' and x as: $x(\alpha) = x' + \alpha(x - x')$ We compute the gradient of F with respect to $x(\alpha)$ and integrate it along the path from x' to x:

$$\text{IntegratedGradients}_i(x) = (x_i - x'_i) \int_0^1 \frac{\partial F(x(\alpha))}{\partial x_i} d\alpha$$

where i is the index of the input feature. To approximate the integral, we can use numerical integration techniques such as the Riemann sum or the trapezoidal rule.

The resulting Integrated Gradients provide the attributions for each input feature, reflecting their contributions to the model's output prediction for the given input instance, which is visualized in the following example 2.6. We can see that the model highlights the texture, nose, and fur of the Panda's face.

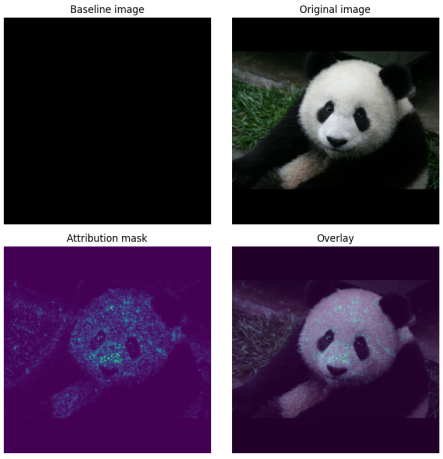


Figure 2.6: Integrated Gradients Example: IG Attribution Mask and Original + IG Mask Overlay (Image from Tensorflow [2023])

Transformers Interpret The Transformers Interpret library by Pierse [2023] is a tool designed to provide insights and interpretability for Transformer models (like BERT, GPTs, etc.). It aims to understand and visualize the contribution/importance of each input token to the output of the Transformer model.

The library uses Layer Integrated Gradients (LIG), an extension of Integrated Gradients, that attributes importance to the neurons in each layer of the model.

For example, in sequence classification tasks like text classification, the *SequenceClassificationExplainer* would tokenize the input text into a sequence of tokens. It then generates a baseline input, typically a sequence of padding tokens of equal length. It finally uses the LIG method to compute the attributions of each token in the input text, i.e., how much each token contributes to the final prediction of the model. Green markers represent a positive and red a negative contribution to the final prediction. The attributions are then returned for visualization or further analysis, as shown in the following figure:

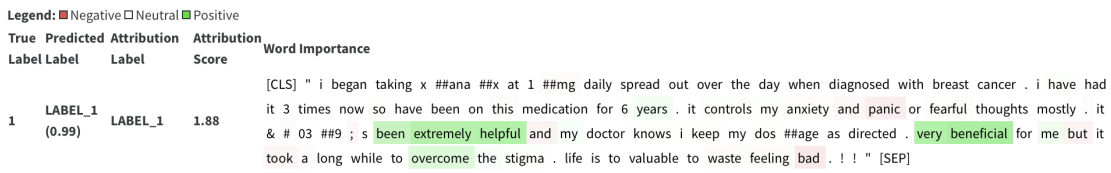


Figure 2.7: Visualization of each token's attributions to the prediction

The use of LIG enables the explainers to provide a fine-grained understanding of the decision-making process within the Transformer models. This can provide invaluable insights for practitioners seeking to understand the underlying model behaviors, correct biases, and improve model performance.

LIME Ribeiro et al. [2016] developed a model-agnostic XAI technique that generates local explanations for individual predictions made by any ML model.

LIME investigates the impact of data variations on the model's predictions by feeding it perturbed versions of the original input. It generates a new dataset comprising perturbed samples and their corresponding predictions from the black-box model. Subsequently, LIME trains an interpretable model on this new dataset, with the training instances weighted by their proximity to the instance under scrutiny. The interpretable model can be any model from the interpretable model's category, such as Lasso or a decision tree. The resulting model should accurately approximate the original machine learning model's

predictions in the local vicinity of the input instance, although it may not necessarily be an accurate global approximation. This type of accuracy is referred to as local fidelity.

Mathematically, local surrogate models with interpretability constraints can be expressed as follows:

$$\text{explanation}(x) = \underset{g \in G}{\text{argmin}}\ L(f, g, \pi_x) + \Omega(g)$$

The explanation model, for instance, x is the model g that minimizes loss L (e.g. mean squared error), which measures how close the explanation is to the prediction of the original model f, while the model complexity $\Omega(g)$ is kept low (e.g. prefer fewer features). G is the family of possible explanations, for example, all possible linear regression models. The proximity measure π_x defines how large the neighborhood around instance x is that we consider for the explanation. In practice, LIME only optimizes the loss part. The user has to determine the complexity, e.g. by selecting the maximum number of features that the linear regression model may use.

Evaluation Metrics for Explanations on Tabular Models

While the importance of XAI is widely acknowledged, evaluating the quality and effectiveness of explanations generated by XAI techniques remains a challenge. To ensure the development of reliable and efficient XAI, it is crucial to establish quantitative metrics that can objectively measure the quality of explanations. In the following sections, we will delve into various metrics and evaluation techniques usable on tabular models and computer vision models. Metrics for evaluation of explanations on NLP Models are not part of this thesis and will be part of future work.

Faithfulness This metric, which is mentioned in the paper of Alvarez-Melis and Jaakkola [2018] aims to assess how well the explanations provided by an interpretability algorithm capture the true decision-making process of a predictive model. As we use the implementation of Arya et al. [2019] because the way it was implemented is more general and can be applied to a wide range of interpretability algorithms, whereas the metric in the paper is specific to the self-explaining neural network architecture proposed by the authors.

The evaluation process involves: Evaluating the correlation between the importance scores assigned to input features by an interpretability algorithm and the actual impact of these features on the model's performance. Given a set of n features, the al-

gorithm assigns importance scores $w = (w_1, w_2, \ldots, w_n)$ to each feature. Next, the features are incrementally removed according to their importance, and the model's performance is measured using a performance metric P, resulting in a performance vector $p = (p_1, p_2, \ldots, p_n)$. The faithfulness metric is then calculated as the Pearson correlation coefficient between the importance scores w and the performance vector p, denoted as $\rho(w, p)$. A high correlation value indicates that the importance scores accurately reflect the contribution of the features to the model's performance, thus suggesting that the explanations provided by the interpretability algorithm are faithful to the model's decision-making process.

A high correlation value indicates that the interpretability algorithm's assigned importance scores align well with the true impact of the features on the model's performance, suggesting that the explanations provided by the algorithm are faithful to the model's decision-making process.

Monotonicity The metric of Luss et al. [2021] measures the effect of individual features on model performance by evaluating the effect on the model performance of incrementally adding each attribute in order of increasing importance. As each feature is added, the performance of the model should correspondingly increase, thereby resulting in monotonically increasing model performance ensuring that the model's behavior aligns with the expected relationships between input features and the target variable.

The summary of the steps to compute the monotonicity metric is as follows: First, we have to rank the input features according to their importance, either as determined by an interpretability (e.g. LIME) algorithm. Next, we incrementally add the ranked features to the model, starting with the least important feature and progressing to the most important feature. Then the model's performance is measured at each step as features are added using a suitable metric, depending on the problem and the model. Assess the monotonicity of the model's performance as features are added. If the performance consistently increases as more important features are included, the model exhibits a strong monotonic relationship between feature importance and performance.

When the monotonicity metric returns false or indicates low monotonicity, it is essential to investigate the underlying cause and make adjustments as needed. This might involve revising the feature importance rankings, selecting a more suitable model, or considering alternative explanation methods that do not rely on monotonic relationships.

Evaluation Metrics for Explanations on Computer Vision Models

Max Sensitivity In the paper Yeh et al. [2019], the authors introduce two new evaluation metrics for assessing the quality of explanations provided by post-hoc explanation methods, such as LIME and SHAP. One of these metrics is called Max-Sensitivity.

Max-Sensitivity is a metric that measures the stability of an explanation method with respect to small perturbations in the input. In other words, it evaluates how sensitive the explanations are to small changes in the input data. The idea is that a good explanation method should provide consistent explanations even when the input data is slightly perturbed, as this would indicate that the explanations are robust and reliable.

To compute the Max-Sensitivity, the following steps are performed:

Generate an explanation for a given instance using the explanation method of interest (e.g., LIME, SHAP, etc.). Perturb the instance by making small changes to its feature values. This can be done by adding small amounts of noise or by making other minor modifications to the input. Generate a new explanation for the perturbed instance using the same explanation method. Compute the max-sensitivity of the explanation method by comparing the original explanation and the explanation for the perturbed instance:

Given a black-box function f, explanation functional Φ, an input instance x, and a neighborhood radius r, the max-sensitivity for explanation is:

$$\mathrm{SENS}_{\mathrm{MAX}}(\Phi, f, x, r) = \max_{\|y-x\| \leq r} \|\Phi(f, y) - \Phi(f, x)\|$$

Where $\|\cdot\|$ denotes an appropriate norm (e.g., Euclidean norm), y represents a perturbed instance within the neighborhood of x, and $\|y - x\| \leq r$ defines the neighborhood.

The max-sensitivity metric evaluates the maximum change in the explanation when the input instance x is perturbed within a defined neighborhood. The lower the max-sensitivity value, the more stable and robust the explanations are to small changes in the input data.

One of the advantages of the max-sensitivity metric is that it can be robustly estimated using Monte Carlo sampling. This makes it a practical choice for evaluating the stability of different explanation methods when applied to machine learning models.

Sparseness The sparseness metric by Chalasani et al. [2020] is a quantitative measure used to assess the conciseness of explanations provided by attribution-based explanation methods for machine learning models, particularly deep neural networks. It helps determineing the significance of input features in relation to the output of a model. The metric is based on the Gini Index, denoted as $G(v)$, applied to the vector of absolute values of attributions. By definition, the Gini Index lies in the range of [0, 1], with a higher value indicating greater sparseness.

Attribution methods, such as Integrated Gradients (see subsection 2.1.4.1) and DeepSHAP, assign importance scores to input features, to explain the output of a model. The sparseness metric evaluates how well these attribution methods can provide concise explanations by emphasizing truly predictive features and minimizing the contributions of irrelevant or weakly-relevant features.

According to Chalasani et al. [2020] To calculate the sparseness metric for a given model and attribution method A, the Gini Index is applied to the absolute values of the attribution vector $A(x)$ for each input instance x. The sparseness metric for an input instance x is denoted as $G\left(|A(x)|\right)$.

The sparseness metric can be applied to various types of models, including naturally-trained models without adversarial perturbations or regularization (n-model), models trained with adversarial perturbations using $\infty(\epsilon)$-bounded adversaries (a-model), and models trained with L1-regularization (l-model).

Faithfulness Correlation The Faithfulness Correlation metric by Bhatt et al. [2020] intends to capture an explanation's relative faithfulness (or 'fidelity') with respect to the model behavior.

Faithfulness correlation scores show to what extent the predicted logits of each modified test point and the average explanation attribution for only the subset of features are (linearly) correlated, taking the average over multiple runs and test samples. The metric returns one float per input-attribution pair that ranges between -1 and 1, where higher scores are better.

For each test sample, |S| features are randomly selected and replaced with baseline values (zero baselines or average of set). Pearson's correlation coefficient between the predicted logits of each modified test point and the average explanation attribution for only the

subset of features is calculated. Results are averaged over multiple runs and several test samples.

Random Logit Sixt et al. [2020] introduce the Random Logit Metric, which aims to assess the quality of explanations by comparing them to a reference explanation for a randomly chosen non-target class.

The Random Logit Metric calculates the similarity between the original explanation (e.g., an attribution method like Integrated Gradients or DeepLIFT) and the reference explanation of a randomly chosen non-target class. The idea is that a good explanation should be significantly different from the explanation of an unrelated, non-target class.

To compute the Random Logit Metric, the authors perform the following steps: Choose a random non-target class. Compute the explanation for the input and the selected non-target class using the same attribution method. Calculate the similarity (e.g., using cosine similarity) between the original explanation and the reference explanation of the non-target class.

The resulting value represents the distance between the original explanation and the reference explanation. A lower similarity score indicates that the original explanation is significantly different from the non-target class, suggesting a better-quality explanation.

2.2 Toolkits

Since the goal of the project was to verify the fairness, robustness, privacy leakage, and explainability of the machine learning models, we explored state-of-the-art toolkits for testing these aspects. Other toolkits exist to evaluate these aspects, so this is not an exhaustive list. The toolkits listed here have been chosen for implementation based on their performance and compatibility with the models used.

Toolkits for Fairness Evaluation

FairMLHealth Allen et al. [2020] proposed a Python package designed to help researchers and practitioners analyze and report fairness in machine learning models. It

provides a variety of fairness metrics and visualization tools to understand and communicate fairness outcomes. The Toolkit was chosen for assessing the fairness of the tabular model due to its ability to provide a comprehensive suite of 17 different metrics. These metrics encompass various aspects of fairness, such as group and individual fairness, providing a well-rounded evaluation of the models. The only limitation is that it only applies to tabular data and binary classification.

Toolkits for Explainability Evaluation

Quantus Hedström et al. [2022] developed a comprehensive evaluation framework that provides standardized benchmarks and evaluation tools for assessing the quality of explanations produced by feature-based explanation methods. It offers a variety of metrics for evaluating explainability, such as faithfulness, stability, and sparseness.

IBM AI Explainability 360 Toolkit (AIX) Arya et al. [2019] developed an open-source toolkit that offers a collection of explainability methods to help users understand and interpret machine learning models. It covers various explanation techniques, from local to global explanations, and from feature-based to example-based explanations.

Toolkits for Robustness Evaluation

IBM Adversarial Robustness Toolkit (ART) Nicolae et al. [2018] developed an open-source library that provides tools for evaluating and improving the robustness of machine learning models against adversarial attacks. It includes a wide range of attack and defense methods, as well as evaluation techniques to assess model robustness.

Foolbox Rauber et al. [2020] developed a Python library that provides a collection of adversarial attack methods for evaluating the robustness of machine learning models. It supports various model types and It is built on top of EagerPy and works natively with models in PyTorch, TensorFlow, and JAX.

Toolkits for Privacy Evaluation

Privacy Meter Murakonda and Shokri [2020] created an open-source library that helps users measure the privacy of their machine learning models by estimating the information leakage from the model's predictions. It provides various privacy metrics and enables users to evaluate the privacy-preserving properties of their models.

TensorFlow Privacy Andrew et al. [2022] proposed a library that provides privacy-preserving mechanisms for machine learning models developed using TensorFlow. It offers a variety of privacy techniques, such as differentially private stochastic gradient descent, which helps users build privacy-preserving models without sacrificing utility.

3 Methodology

In this chapter, we build upon our previous work (see Göllner and Tropmann-Frick [2023]), where we developed a small prototype for VERIFAI. This prototype provided a foundational framework but it was limited to the computer vision model and a limited set of metrics. In this thesis, we expand upon this foundation by incorporating additional metrics, datasets, and models to further enhance the comprehensiveness and applicability of VERIFAI.

The following section will therefore outline the methodology employed in our research, detailing the initial use cases that serve as the foundation for testing the framework.

First, we will describe the models and datasets utilized in our study, emphasizing their relevance to the core aspects of RAI, such as privacy, fairness, explainability, and robustness. Next, we will discuss the rationale behind selecting appropriate metrics from the libraries mentioned in the technical background. This will provide a clear understanding of the evaluation criteria employed to assess the performance of AI models in terms of ethical dimensions.

Following this, we will delve into the application architecture and design, explaining how these components facilitate the implementation of the framework in real-world scenarios. This will include a discussion of how the various elements of the framework interact with one another, as well as how the framework can be scaled and adapted to different use cases.

Finally, we will provide an overview of the tools, libraries, and hardware resources used to conduct the experiments. This section will offer insights into the computational requirements and practical considerations for implementing and testing the RAI framework.

This comprehensive methodology section will enable readers to understand the framework's development and application, as well as its potential for advancing the field of RAI.

3.1 VERIFAI -Framework

This section provides an overview of the VERIFAI framework, explaining the selection of data, model architecture, and evaluation toolkits. The framework is designed to assess RAI principles, specifically focusing on security, fairness, privacy, and explainability concerns.

3.1.1 Healthcare Scenario

In this study, we have chosen the Healthcare Domain as the primary scenario for evaluating our RAI framework. Since this field represents an area where automatic decisions significantly affect human lives, building RAI in this domain is indispensable. We aim to put the defined aspects of RAI in a clinical context, shedding light on the importance of addressing security, fairness, privacy, and explainability concerns.

Secure AI in Healthcare

Security is a critical aspect of RAI in healthcare, as vulnerabilities in AI systems can lead to severe consequences for patients. Adversarial attacks, where an attacker manipulates the input data to mislead the model, may result in incorrect predictions and diagnoses. Therefore, it is essential to ensure that the models are robust against such attacks and maintain their performance in the presence of adversarial perturbations. By addressing security concerns within our framework, we seek to mitigate the risks associated with deploying AI models in healthcare, where undetected diseases or incorrect treatment recommendations can have life-threatening implications.

Fair AI in Healthcare

Fairness is a crucial aspect of RAI in healthcare, as it ensures that AI models do not discriminate against or favor specific patient groups. Ensuring fair data representation and utilizing stratified sampling techniques contribute to unbiased and balanced training data. Addressing fairness concerns within our framework aims to identify potential sources of bias and reduce discriminatory decisions in medical applications, which is critical for fostering trust in AI systems among healthcare professionals and patients.

Privacy-preserving AI in Healthcare

Privacy is an essential consideration in the healthcare domain, as AI models must be trained using sensitive patient data. Ensuring that the model's training data cannot be traced back to individuals is crucial for maintaining patient privacy and adhering to data protection regulations. Addressing privacy concerns within our framework involves assessing potential privacy risks and implementing techniques to protect sensitive information. This helps ensure that patient privacy is preserved and that the deployment of AI models in healthcare complies with applicable data protection laws.

Explainable AI in Healthcare

Explainability is a vital component of RAI in healthcare, as it allows medical professionals to understand and validate the decision-making process of AI models. Ensuring that AI models are explainable and interpretable is essential for gaining the trust of healthcare professionals and patients. Within our framework, we aim to evaluate the explainability of AI models and identify the characteristics that contribute to effective explanations. Additionally, we assess the compatibility of interpretability algorithms with the models to ensure reliable and comprehensible explanations for their predictions.

3.1.2 Data Sources

Within the healthcare domain, we have selected three initial use cases to demonstrate the applicability of the framework across different tasks and data types: detecting Skin Cancer, recommending medicines based on Sentiment Analysis, and detecting Heart Diseases. To train the models for these use cases, we have utilized the following datasets:

Tabular data

To train our tabular model for detecting heart diseases, we utilized the Heart Disease Dataset by Janosi et al. [1988] hosted in the UCI Machine Learning Repository. This tabular dataset consists of various medical attributes related to heart disease, providing a suitable foundation for evaluating the performance of our tabular model in the healthcare scenario.

Image Data

For detecting skin cancer we employed the HAM10000 dataset by Tschandl et al. [2018], a comprehensive collection of dermatoscopic images of various skin lesions, to train the Xception model for classifying skin cancer. This dataset provides a challenging and relevant testbed for evaluating the performance of our computer vision model within the healthcare domain and satisfies a second criterion: it consists not only of image data but also of metadata for the analysis.

Text Data

For the Use Case of recommending medicines based on Sentiment Analysis, we used the *Drug Review Dataset* published by Gräßer et al. [2018] to perform sentiment analysis on medication reports. By training the model on this dataset, we aimed to recommend medicines based on the sentiment expressed in user reviews, thus demonstrating the applicability of our framework to natural language processing tasks in healthcare.

By leveraging these diverse datasets and use cases, we can effectively assess the performance and capabilities of our RAI framework in the context of healthcare, while also ensuring the framework's relevance and applicability to real-world problems. Next up we will present the corresponding models which were trained using those datasets.

3.1.3 Models

In the initial phase of our framework development, we selected one representative model for each of the three primary ML tasks: natural language processing, computer vision, and tabular data analysis. These models have been chosen based on their suitability for the specific datasets used in our study, as well as their proven performance in their respective domains. The models are as follows:

Tabular Model

We have opted for the *Random Forest* (RF) model (from the Scikit-Learn Library proposed by Pedregosa et al. [2011]) for handling tabular data, due to its proven efficacy in processing structured data and its ability to address a wide range of problems. The

Random Forest model's ensemble learning approach, which combines multiple decision trees, allows it to capture complex patterns and relationships in tabular data, making it an ideal choice for our initial tests. Another reason was the need to have a good example for measuring model explainability since RF belongs to the category of black-box models.

Computer Vision Model

For image classification, we have selected the *Xception* model by Chollet [2017], an advanced convolutional neural network (CNN) architecture that employs depthwise separable convolutions and has demonstrated impressive performance on various computer vision benchmarks. As these models are effective in processing and analyzing visual data, such as images or videos. They can be used for various tasks, such as object recognition, image segmentation, and scene understanding. The Xception model's architecture enables efficient learning of features from images, making it a good choice for our framework. It achieved the best results on the dataset compared to other architectures (like Inception by Szegedy et al. [2014] or ResNet by He et al. [2015])

Language Model

For the task of *Natural Language Processing* (NLP), we have chosen the *DistilBERT* Architecture by Sanh et al. [2019], as it is a smaller general-purpose language representation model, which can then be fine-tuned with good performances on a wide range of tasks.

An important consideration in selecting these models is their compatibility with the evaluation metrics used in our framework. The chosen models — DistilBERT for natural language processing, Xception for computer vision, and Random Forest for tabular data analysis—have been carefully evaluated to ensure they align with the metrics for privacy, fairness, explainability, and robustness. This compatibility is crucial to accurately assess the performance of the models in terms of RAI and enables a comprehensive evaluation of the framework's effectiveness across different tasks and domains.

3.1.4 Selection of Metrics

In the following section, we will discuss the evaluation metrics selected for each model, addressing particularly the research questions RQ2 and RQ3. Our decisions were informed by analyzing the features, and usage limitations of the metrics, as well as the specific characteristics of the models and their intended applications.

Overview and Scope of Metrics

In our evaluation framework, we utilized a variety of metrics to evaluate the different aspects of our models - robustness, fairness, privacy, and explainability. The specific metrics used vary depending on the type of model - image, text, or tabular.
The scope of these metrics relates to fairness within the ethics aspect, robustness within the security aspect, and privacy leakage within the privacy aspect. For the aspect of explainability, we focus on the quantitative evaluation of the XAI methods in connection with the respective model (see sub-aspects in 3.1). The selection is based on identifying the most important findings of our systematic literature review.

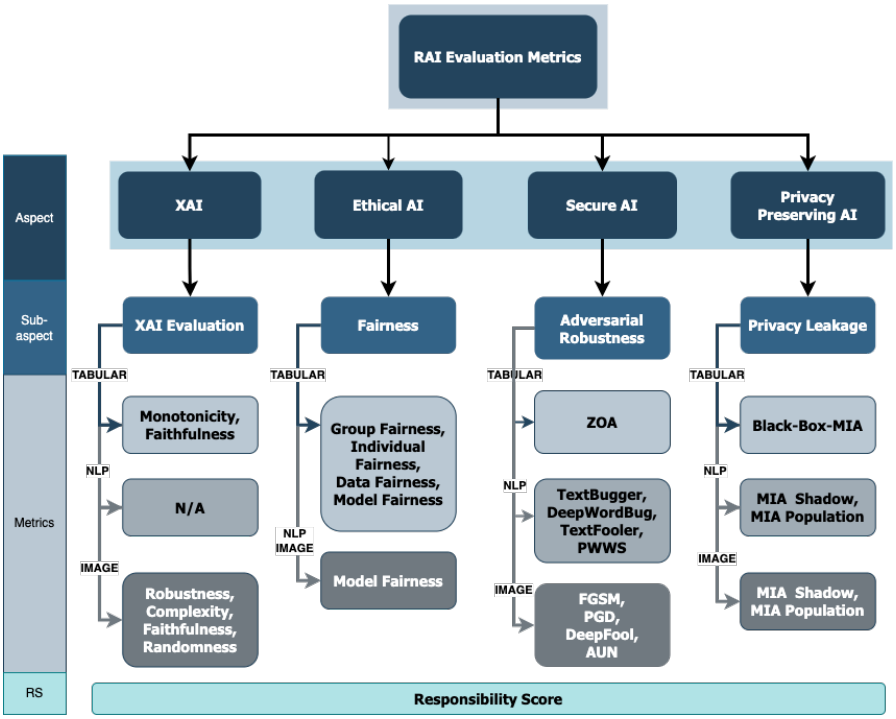


Figure 3.1: RAI Evaluation Metrics Overview (high-level)

3.1.4.1 Evaluation of Tabular Models

Aspect		Metrics
Fairness	Group Fairness	AUC Difference, Balanced Accuracy Difference, Balanced Accuracy Ratio, Disparate Impact Ratio, Equal Odds Difference, Positive Predicted Parity Difference, Positive Predicted Parity Ratio, Statistical Parity Difference
	Individual Fairness	Between Group Entropy Error, Consistency Score
	Model Performance	Accuracy, F-1 Score, FPR, Mean Target, Precision, TPR
	Data Metrics	Prevalence of Privileged Class
Privacy Leakage		Black-Box-MIA
Robustness		ZOA Attack
Explainability		Monotonicity, Faithfulness, (XAI Method: LIME)

Table 3.1: Metrics for the Tabular Model

Fairness For Group Fairness, a range of metrics was selected to capture different dimensions of fairness, including *AUC Difference*, *Balanced Accuracy Difference*, *Balanced Accuracy Ratio*, *Disparate Impact Ratio*, *Equal Odds Difference*, *Positive Predicted Parity Difference*, *Positive Predicted Parity Ratio*, and *Statistical Parity Difference*. These metrics help to ensure that the models perform fair across various groups and do not disproportionately favor any specific group.

Individual Fairness metrics, such as *Between Group Entropy Error* and *Consistency Score*, were chosen to evaluate how fairly the models treat individual data points. This is essential for understanding the fairness of the models at a granular level.

Standard metrics like *Accuracy*, *F-1 Score*, *FPR*, *Mean Target*, *Precision*, and *TPR* were chosen to assess model performance. These metrics provide insights into the overall performance and effectiveness of the models, ensuring that they can make accurate and reliable predictions.

The data metric, *Prevalence of Privileged Class*, was selected to gain insights into the distribution of privileged and non-privileged groups in the data, which is crucial for understanding potential biases and imbalances.

In order to evaluate the fairness of the tabular model, we calculated a composite fairness score. To compute this score, we first identify the number of biased metrics (biased)

out of the total number of metrics (total). Then, we calculate the proportion of non-biased metrics (or fairness proportion) by dividing the number of biased metrics by the total number of metrics and taking the inverse since we want to know the percentage of fairness: fairness $= 1 - \left(\frac{\text{biased}}{\text{total}}\right)$

Privacy For evaluating privacy leakage, the *Black-Box-MIA* attack metric from the *IBM ART library* was selected. This metric enables a comprehensive assessment of potential privacy risks in the models by quantifying privacy leakage.

The resulting score is determined by calculating the Area Under the Curve (AUC) of the best attack performance based on the TPR and FPR. A higher value signifies a greater risk of privacy leakage. Consequently, an AUC value near 0.5 indicates a low privacy risk, while a value approaching 1 signifies a substantial privacy risk.

Robustness The *ZOA Attack* was selected as a robustness metric since it can be applied to non-differentiable models, such as random forests, due to its independence from gradient information. This makes it a versatile and effective method for evaluating the models' robustness against adversarial attacks.

Another reason why ZOAs can be effective with random forest models is that these models are sensitive to small perturbations in the input data. Random forests make decisions based on the majority vote of the individual decision trees, so if a small perturbation causes a few of the decision trees to change their predictions, the overall prediction of the model can be flipped.

Overall, zeroth order optimization attacks are considered more challenging to defend against than attacks that rely on gradient information, because they are more difficult to detect and mitigate.

Explainability Finally, the *LIME* explainer was chosen for explainability assessment, using the metrics of *monotonicity* and *faithfulness*. These metrics help to evaluate the influence of individual attributes on the performance of the predictive models and understand how each attribute contributes to model performance. By doing so, they provide valuable insights into the inner workings of the models, making them more transparent and interpretable. Since we choose LIME as our XAI method for the tabular model,

we need to iterate over all local explanations so we can average them finally and get an overall result metric

3.1.4.2 Evaluation of Computer Vision Models

The metrics for evaluating the computer vision model, Xception are described in table 3.2 as follows:

Aspect	Selected Metrics
Fairness	F-1 Score
Privacy	MIA via Shadow Models Model Dependent MIA via population data
Security	FGSM Attack PGD Attack DeepFool Attack Additive Uniform Noise Attack
Explainability	(Robustness) Max Sensitivity (Complexity) Sparseness (Faithfulness) Faithfulness Correlation (Randomness) Random Logit (XAI Method: Integrated Gradients)

Table 3.2: Metrics for the Computer Vision Model

Fairness For evaluating the fairness of the computer vision model, the F-1 metric is utilized. In these cases, we have imbalanced datasets and lack protected attributes in the dataset, which inhibits the proper use of group-fairness metrics. Consequently, the mean F-1 score is calculated across all tested classes within the dataset to assess the model's performance. This approach ensures a comprehensive assessment of the model's performance by accounting for the average classification accuracy across all the classes while considering the trade-off between precision and recall, which is particularly important in imbalanced datasets.

However, it is important to note that the F-1 score does not directly measure bias or discrimination in the model's predictions. Ideally, a more complete understanding of the model's fairness would be obtained through the use of additional fairness-specific metrics, provided that the required protected attributes are available. In this study, due to the absence of these attributes, we are limited to the F-1 score. As a result, while we

can comment on the model's performance in handling imbalanced data, we cannot fully assess its performance in terms of fairness.

Privacy For evaluating privacy leakage through Membership Inference Attacks, we selected the metrics *MIA via Shadow Models* and *Model Dependent MIA via population data* from the *Privacy-Meter* Library. These two metrics try to measure privacy leakage through two different approaches to make it more comprehensive.

The resulting score is determined by calculating the Area Under the Curve (AUC) of the attack performance. A higher value signifies a greater risk of privacy leakage. Consequently, an AUC value near 0.5 indicates a low privacy risk, while a value approaching 1 signifies a substantial privacy risk. Finally, we take the worst case for calculating the privacy score (higher privacy leakage).

To obtain a measure of privacy goodness, the inverse of the AUC value is considered. This allows for a more intuitive interpretation of the results, where a higher value represents better privacy preservation, while a lower value indicates a higher privacy risk.

Robustness For security assessment, the metrics *FGSM Attack*, *PGD Attack*, *DeepFool Attack*, and *Additive Uniform Noise Attack* were applied to test the model's robustness.

They cover a range of adversarial attack strategies, from simple to sophisticated gradient-based attacks. The chosen metrics include both targeted and non-targeted perturbations, providing a comprehensive evaluation of the model's robustness. The combination of these metrics enables the identification of potential vulnerabilities, guiding the development of more resilient models. By using both iterative and non-iterative attacks, as well as noise-based perturbations, the evaluation assesses the model's robustness against various types of input data manipulations.

This selection of metrics allows for a thorough and multifaceted assessment of the model's security, ensuring that it can withstand different adversarial attempts.

These attacks were applied using the *Foolbox* library because the calculation is fast, and the library has many metrics to compare.

The robustness score is calculated over 15 iterations. In each iteration, the perturbation rate (epsilon) is incrementally increased. The model's robustness is assessed based on its classification accuracy on the corrupted images. This value is a standard in adversarial

robustness for image data, representing the maximum allowable change to any pixel's intensity value as a fraction of its possible range, which is typically 0 to 255 for standard 8-bit images. Therefore, perturbations with $\epsilon_\infty = \frac{8}{255}$ ensures that the perturbations made to the image are small enough to be virtually indistinguishable to the human eye, while still potentially significant enough to cause a model to misclassify the image. This is considered the benchmark metric for Adversarial Robustness (see Croce et al. [2021]). The final so-called Robust Accuracy is then determined by taking the worst-case result from all four attack metrics. This approach ensures that the evaluation captures the model's performance under the most challenging adversarial conditions.

Explainability The *Integrated Gradients* method was utilized to assess the explainability of the model.

As visual explanations alone are often insufficient and to provide a comprehensive evaluation, four metrics were chosen, which measure explanations from different perspectives: *Robustness* (Max Sensitivity), *Complexity* (Sparseness), *Faithfulness* (Faithfulness Correlation), and *Randomness* (Random Logit). These metrics ensure the explanations are stable, concise, faithful to the model's behavior, and robust to input perturbations.
The overall explainability score is calculated as the average of the four metric results, and the *Quantus* library was used for its wide range of metrics to evaluate explainability.

3.1.4.3 Evaluation of NLP Models

The metrics for evaluating the language model, DistilBERT, were as follows:

Aspect	**Metrics**
Fairness	F-1 Score
Privacy	MIA via Shadow Models Model Dependent MIA via population data
Security	TextBugger DeepWordBug TextFooler PWWS
Explainability	Metric: None, XAI Method: Transformers Interpret

Table 3.3: Metrics for the NLP Model

Fairness For the evaluation, we used the F-1 Score for fairness evaluation, similar to the computer vision model. This was done to ensure consistency and coherence across neural network models when evaluating fairness. The choice of the F-1 score for the language model is particularly appropriate given the potential for imbalanced classes in our medical reviews dataset.
However, it's important to consider the nature of bias in this context, which may not relate to protected attributes but could instead stem from the specific domain. For instance, biases in the recommendation of certain medicines for specific symptoms or conditions could be present in the reviews. This is a potential limitation of using the F-1 score as our primary measure of fairness.

Privacy For privacy evaluation, we also used the same MIA metrics as for the computer vision model (*MIA via Shadow Models* and *Model Dependent MIA via population data* from the *Privacy-Meter Library*) to maintain consistency in assessing privacy leakage across the neural networks. Importantly, our text data, consisting of medical reviews, do not contain personally identifiable information (PII), which alleviates some common privacy concerns associated with text data. Therefore, the MIA metrics used in our evaluations are both relevant and sufficient for assessing the privacy concerns in our specific use case. They provide a comprehensive measure of how well our models protect against potential privacy attacks, making them applicable to both the computer vision and language models in our framework. In future work, if datasets containing sensitive personal information are used, additional privacy protection measures, such as differential privacy or secure multi-party computation, might be needed.

Security For security assessment, since adversarial attacks must be tailored to the specific domain, we selected the following metrics that are specifically designed for text models: *TextBugger*, *DeepWordBug*, *TextFooler*, and *PWWS*.
We have chosen these four metrics because they address different aspects of adversarial attacks on text models, providing a comprehensive assessment of the model's robustness. *TextBugger* is a universal attack method that can be applied to various NLP models and tasks. *DeepWordBug* focuses on generating minimal perturbations to fool deep learning-based text classifiers. *TextFooler* is a simple yet effective method that creates adversarial samples by substituting a few important words while maintaining the semantic meaning of the text. *PWWS* is a white-box method that generates adversarial examples by replacing words in the input with semantically similar words, taking into account the model's

behavior. Together, these metrics thoroughly evaluate the model's security against various attack strategies.
The final so-called *Robust Accuracy* is then determined by taking the worst-case result from all four attack metrics. This approach ensures that the evaluation captures the model's performance under the most challenging adversarial conditions.

Explainability In terms of explainability, we utilized the *Transformers Interpret* XAI method. This technique provides visualization of the contributions of individual features, in our case tokens, in the prediction made by the model. It is particularly useful in interpreting models like DistilBERT, shedding light on how the model processes and weighs different elements of the input data.
For this iteration of the framework, we did not include any specific metrics for evaluating the explainability of the text models. The reason for this omission was due to challenges encountered in defining and quantifying explainability for text models, which often involve subjective and context-dependent aspects that are difficult to capture with standard metrics.

3.1.4.4 Responsibility Score

Now we discuss how the final *Responsibility Score* was calculated using the obtained metrics from each category. First, the metrics from each category, such as fairness, were normalized and scaled to a value between 1 and 10. This process ensures that all the metrics are on a comparable scale and contribute equally to the final score. At the current stage of development, we calculate the *Responsibility Score* by taking the mean of the four normalized scores from fairness, privacy, security, and explainability. This approach considers that each category contributes equally to the overall responsibility of the model. The resulting *Responsibility Score* is then converted into a percentage value, representing the percentage of responsibility achieved by the model with respect to the evaluated metrics. Scores from 0 to 4 (0 - 40%) indicate, that the model's performance in terms of responsibility is unsatisfactory or potentially harmful. Scores from 5 to 7 (50 - 70%) represent a medium level of responsibility, suggesting that while the model shows some promise, improvements are still needed. Scores from 8 to 10 (80 - 100%) indicate that the model has achieved a high level of responsibility according to the respective metrics.

3.1.5 Implementation Details

To answer the research question RQ4, in this section, we will discuss the implementation details of the VERIFAI application. This includes an overview of the chosen framework, the application architecture, and the rationale behind these choices. We will delve into the various components of the architecture and highlight the key aspects that contribute to the overall functionality and extendability of the application. This discussion aims to provide a comprehensive understanding of the design decisions and their implications on the performance and scalability of the system.

Dash

We have chosen Dash as the fundamental framework for building VERIFAI due to its numerous advantages. Dash is written on top of popular frameworks and libraries like Flask, React, and Plotly, making it ideal for data visualization apps. As a low-code framework, Dash enables the rapid development of data apps in Python. Because Dash Apps are using Flask as the backend, we can run them using Gunicorn, so it is easy to scale these apps to serve more users by scaling up the number of worker processes. Moreover, Dash apps can be deployed in the same way as Flask apps, simplifying the process. As an open-source library released under the permissive MIT license, Dash is well-documented and supported by an active and responsive community.

Application Architecture

The architecture of the VERIFAI application is based on the well-established Model-View-Controller (MVC) design pattern. This pattern promotes modular design by dividing an application into separate units, each responsible for a specific task. This approach ensures easy maintenance, testing, and reusability of components while providing a clean separation of concerns (SoC).

The application's structure is organized as follows:

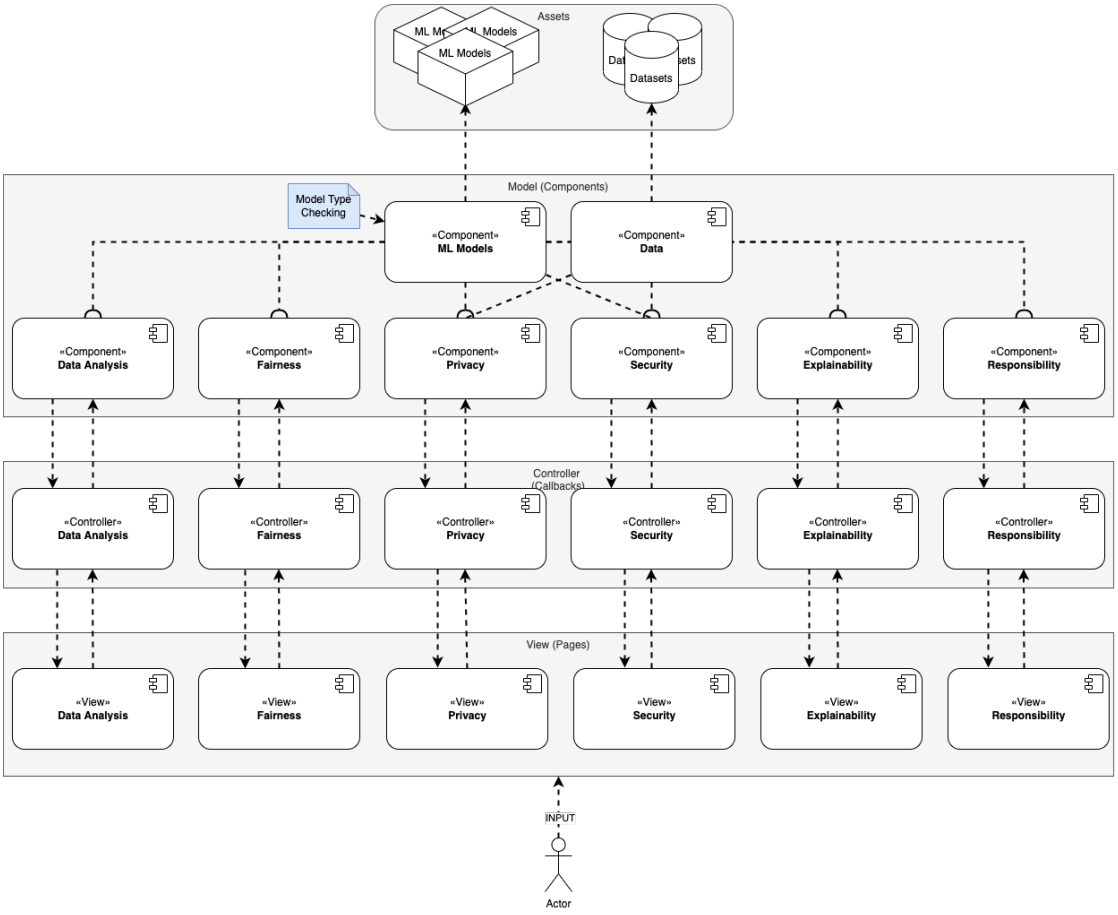


Figure 3.2: VERIFAI Component Diagram (high level)

- **Assets**
  - Datasets, trained ML-Models, Images, Infographics
- **Components**
  - Backend logic of the app
  - Logical components for each RAI aspect
  - Metric calculations and data preparation for user presentation
  - Recurring components of pages, such as headers and navigation, model cards, and plot templates

- **Controller**
  - Controller logic (MVC), implemented in Dash using callbacks to process requests from the view and retrieve data processed in the backend
  - The data and models are retrieved from the system depending on the use case. Depending on the model architecture and data set, this is provided accordingly
- **View**
  - Views that display results and plots to the user
  - Designed as a step-by-step walkthrough, allowing users to progress through each aspect sequentially
- **Utils**
  - Utility functions, such as app configuration files (not in the diagram)

This architecture is designed with extensibility and future development in mind. It ensures a modular and maintainable design, making extending and adapting the application for future requirements easier.

Flowchart

Figure 3.3 shows the flowchart of the application:

- We start with an overview of the process, called VERIFAI-Lifecycle on the index page.
- Then the user can switch to the implemented overview of use cases (Image, Text, and Tabular Data). The selected dataset gets saved as a variable in the session storage used by Dash (browser) so that it remains available throughout the session.
- In the next step, we can view the dataset via charts to see the characteristics.
- After clicking 'next' we can choose a pre-trained model, which was trained using the dataset we selected before. This model can be tested with a batch of test data. The test size of the dataset can be selected separately via a slider.

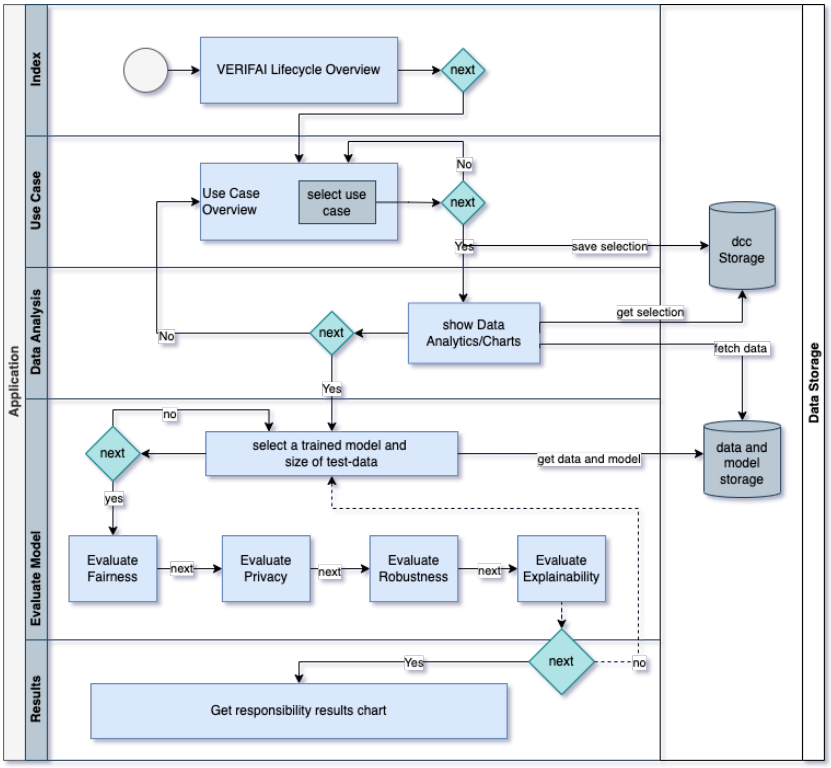


Figure 3.3: VERIFAI Flowchart (high level)

- After clicking the button 'submit' the selection gets stored as a variable in the current session.
- Now we can switch though each of the evaluation aspects (Fairness, Privacy Leakage, Robustness, and Explainability) separately in single views.
- Only if all the metrics have been calculated we can switch to the 'responsibility view' to see the final result.

User Interface and Usability

One essential concept in human-centered AI is the Human-in-the-loop (HITL) which involves incorporating human input and feedback throughout the AI system's development and decision-making processes. Since *Human-centeredness* is a fundamental aspect of RAI, it emphasizes the need to consider user interaction and understanding when designing AI systems. Therefore VERIFAI needs also to have an intuitive User Interface.

The UI is structured to facilitate a seamless and intuitive user experience, enabling users to easily navigate through the various sections and comprehend the evaluation results.

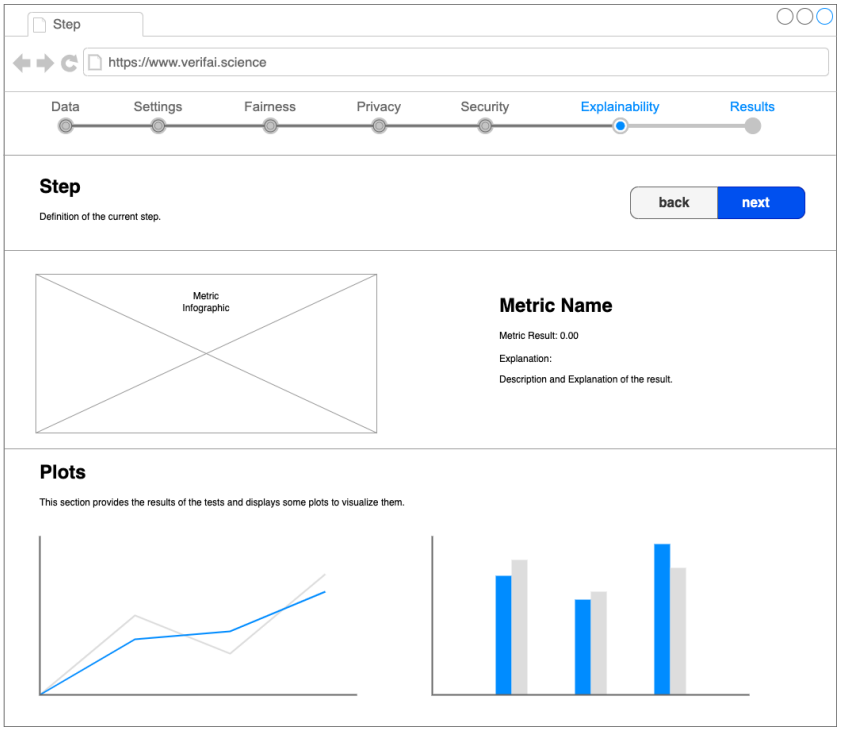


Figure 3.4: Wireframe of the User Interface

The interface is divided into four primary sections. At the top, a step bar guides users through the different stages of the evaluation process, allowing them to visualize their progress. Next, a description of the current step is displayed, such as "Model Fairness", along with two navigation buttons for moving back and forth between steps.

Below the description, a small infographic is presented, which provides a visual representation of the metric being evaluated in the current step. Adjacent to the infographic is an explanation of the metric's performance, giving users a clear understanding of the model's performance in that specific aspect.

Finally, at the bottom of the page, plots are displayed to showcase the test results visually. This enables users to easily interpret the evaluation outcomes and derives insights from the data.

The UI design leverages the capabilities of the Dash framework, ensuring a responsive and visually appealing interface that caters to users with varying levels of expertise.

The layout and organization of the elements contribute to a cohesive and user-friendly experience, promoting efficient navigation and comprehension of the evaluation results.

4 Results

In this chapter, we present the results of the VERIFAI implementation. The chapter is divided into two parts: the first part deals with the results of the implementation, while the second part focuses on the technical challenges faced. The overall workflow of the application is structured as follows: We begin by describing the application's workflow and providing an overview of the user experience. Next, we delve into the data preparation process and discuss the step where the user can select the use case. We then proceed with the run-through of VERIFAI, analyzing various use cases, including tabular, image, and text datasets. In each case, we evaluate the model's fairness, privacy-leakage, robustness, and explainability, discussing the plots and their implications, and calculating the Responsibility Score. The second part of this chapter addresses the challenges encountered during the implementation process, discussing potential mitigations and outlining areas left for future work.

4.1 Implementation Results and Use Case Insights

VERIFAI- Lifecycle

On the index page, as depicted in figure 4.1, our aim is to familiarize the user with the workflow of our system. We provide a visual representation of the *VERIFAI-Lifecycle*, which extends the data science lifecycle by incorporating responsibility checks. The webpage displays the current step, which is explained at the top of the page, and this feature is consistent throughout the application. Additionally, we have included a section at the top of the page that outlines the pipeline we follow during the entire assessment process.

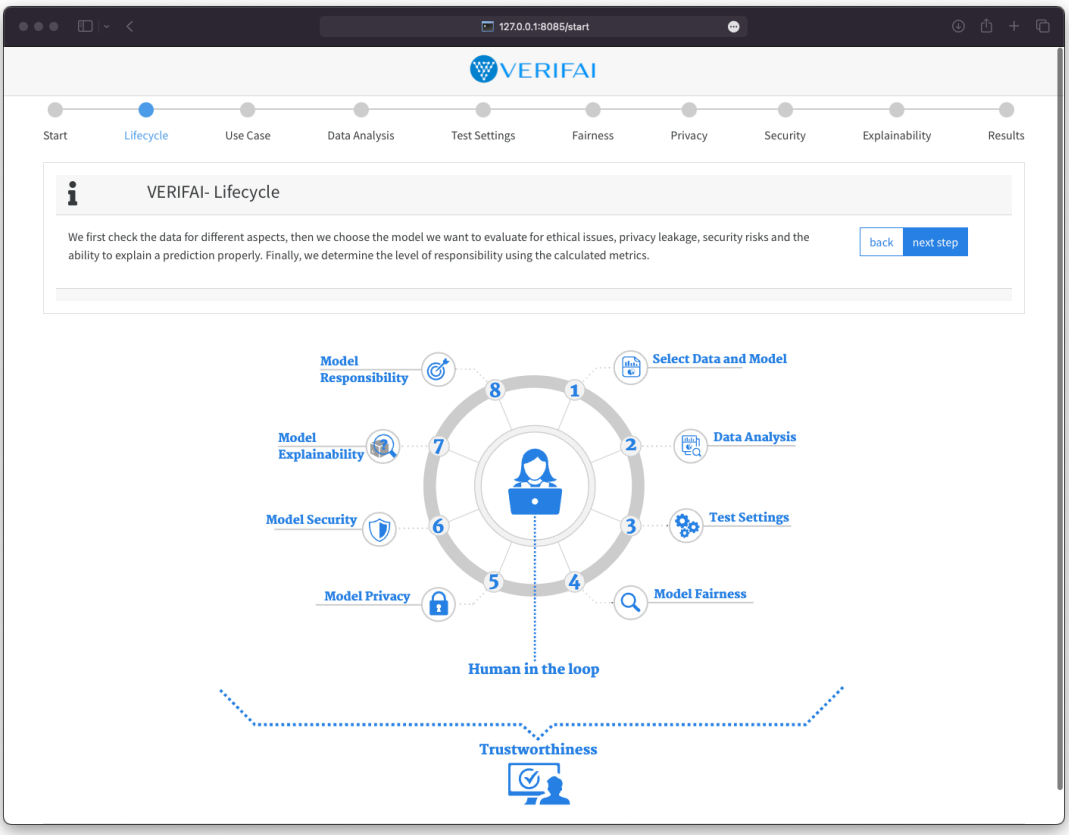


Figure 4.1: VERIFAI-Lifecycle

Data Preparation, Feature Engineering, and Model Training

In the test scenarios, the three prepared datasets underwent cleaning and, when feasible, stratified sampling was applied. Data augmentation was not utilized, as it is unsuitable in the medical domain and may result in inaccurate data. Feature engineering and model training was conducted during the preparation phase. The data was subsequently exported in numpy-format for the tests for faster processing and that was most compatible with the processing libraries Tensorflow, PyTorch, and Scikit-learn. The models were saved in the format of the corresponding library (such as *h5* for Tensorflow-models). Each model needs therefore special attention on the way it is loaded as well. Thus, the subsequent step involved selecting the use case, after which the corresponding data and model were then utilized for further processing.

Select use case

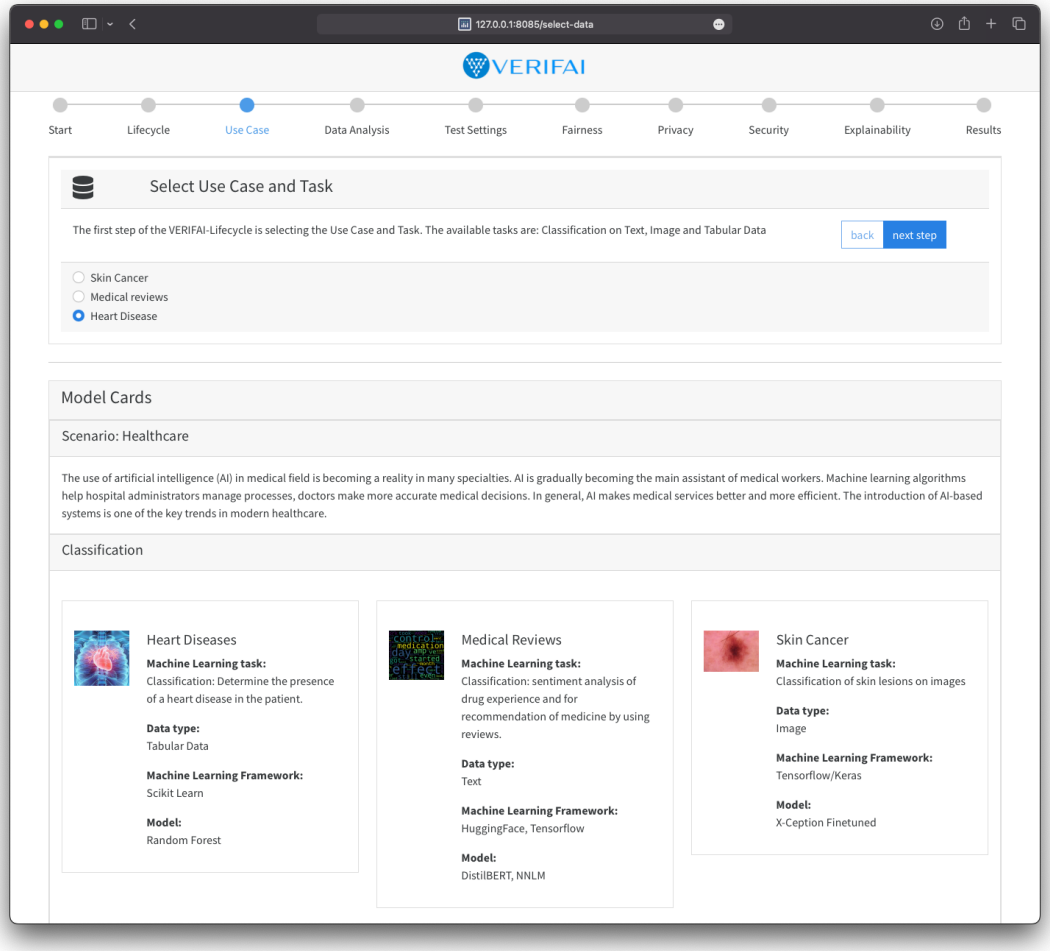


Figure 4.2: Select Use Case and Task

Figure 4.2 showcases a screenshot of the step in which users can select from various use cases and corresponding datasets for evaluation purposes.

As illustrated, there are three different application examples, which can be regarded as model cards. These cards provide information on the model architecture used for training and the datasets on which they were trained.

We will conduct three separate runs, one for each model. By presenting various use cases, we demonstrate the versatility and adaptability of our application across different domains and data types.

4.1.1 Results of the Tabular Model Evaluation

Our first run is evaluating the tabular model trained on the *Heart Disease* data. It is used for classifying whether an individual suffers from heart disease or not.

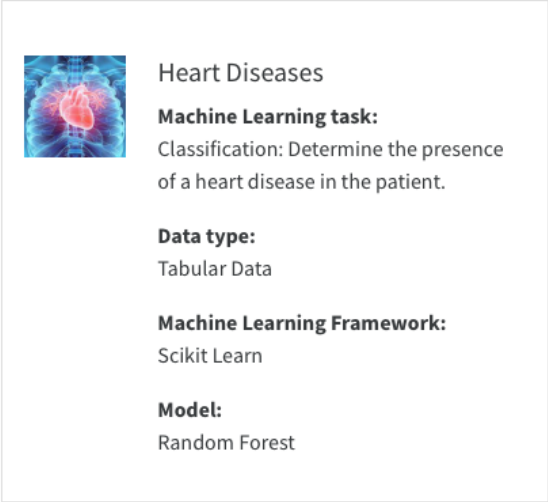


Figure 4.3: Select use case: Heart Disease

Data Analysis

As previously mentioned, the data was cleaned and prepared for visualization before being integrated into the software. In the initial step, we loaded the dataset and carried out a visual analysis employing various plots to uncover diverse aspects of the data. This examination allowed us to pinpoint potential biases. The dataset (see 4.4) consists solely of tabular data, and the visualizations display the most important features.
Age and gender distribution: We can also see that the risk of heart disease is higher among people of ages up to 55 and is drastically low among adults above 55 years of age. The risk of heart disease is seen to be more prevalent among women than men.
Chest Pain and Max Heart Rate: People having type 1 chest pain have a high risk of high disease as compared to other chest pain types. A higher max heart rate among younger candidates is seen to be a major symptom of heart disease Nonetheless, our emphasis for the prototype is on model validation. Consequently, we proceed directly to the subsequent step.

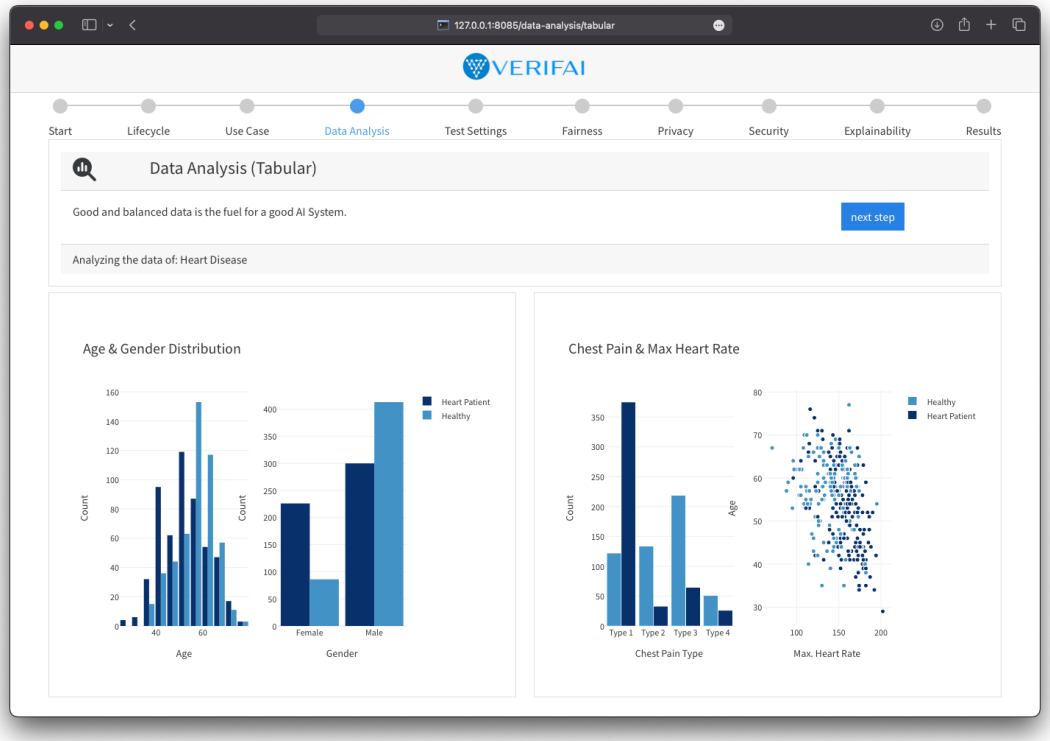


Figure 4.4: Data visualization of the Heart Disease dataset

Test Settings

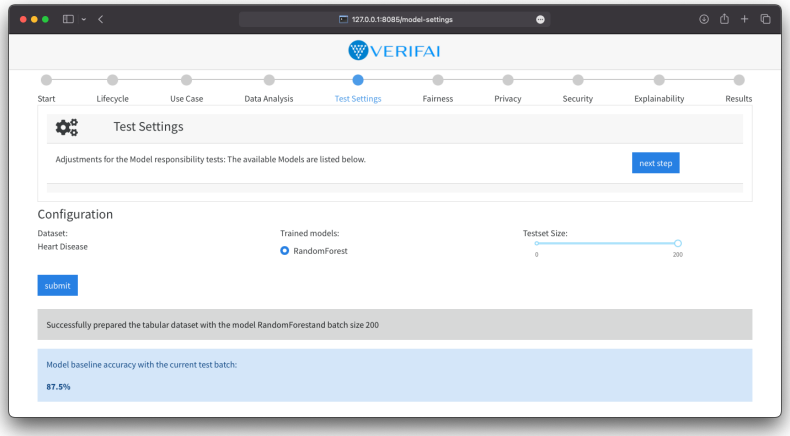


Figure 4.5: Test settings for the assessment of the tabular model

In this stage, we can configure the subsequent tests. We load our model and set the size of the test dataset, which consists of 200 data points that serve as a test batch. Here, we load our trained *Random Forest* model and evaluate it on the loaded data. This serves as the foundation on which the same tests are conducted. The model has a validation accuracy of 87.5%, but a good accuracy is not enough, furthermore, we will assess the model on our aspects of responsibility.

Fairness

In this section, we will first delve into the fairness analysis. An overview of the fairness assessment is shown in figure 4.6 provides a complete overview of the results.

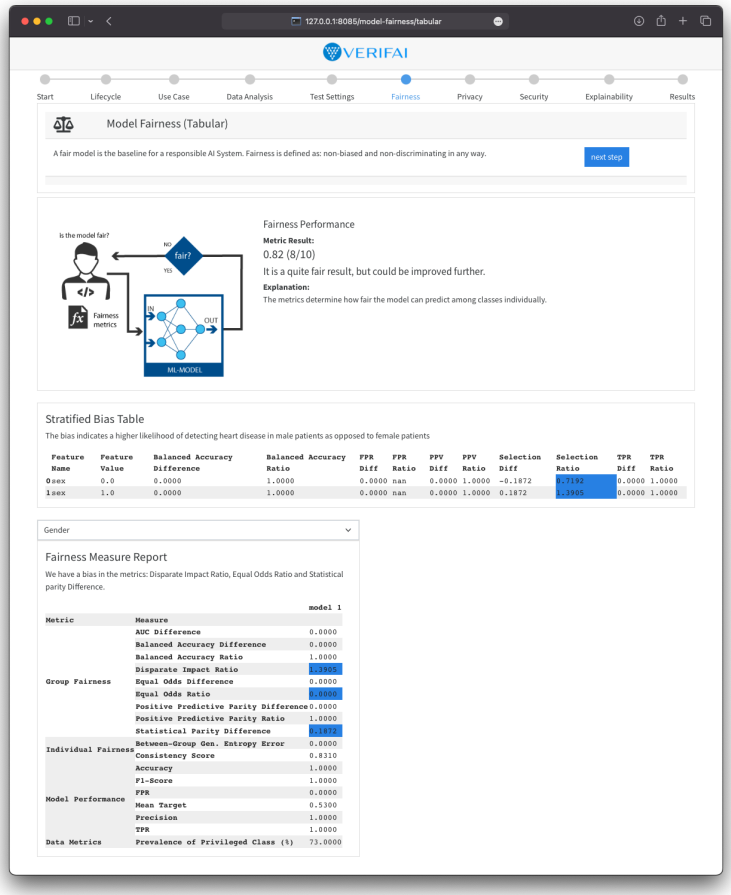


Figure 4.6: Tabular Model Fairness Evaluation (full-page screenshot)

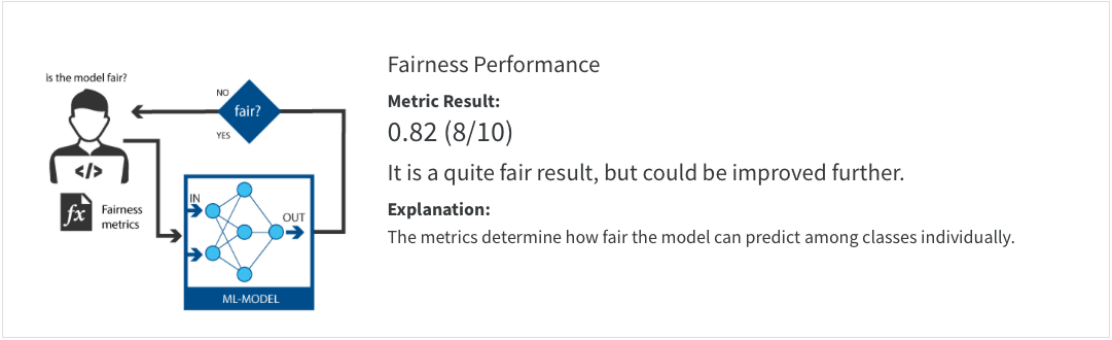


Figure 4.7: Tabular Model Fairness Evaluation (results details)

The top of the page features a helpful illustration of the current process, and the accompanying metrics results are conveniently displayed nearby. Each figure used in these sections is designed to give a comprehensive visual representation of our evaluation metrics. They serve to complement the accompanying discussion and provide an intuitive understanding of the outcomes of our analysis. Similar visualizations will be used in the following sections for each of the additional evaluation metrics, ensuring a consistent and accessible presentation.

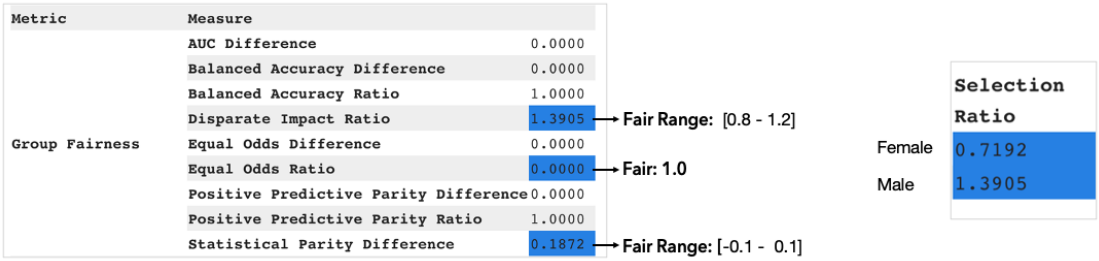


Figure 4.8: Tabular Model Fairness Evaluation (details)

In this part of the assessment, we aim to answer the questions if the model's prediction is fair and whether there is there a higher likelihood for a certain gender to detect heart diseases, therefore we focus on the sensitive feature: gender.

The detailed view of the results is displayed in figure 4.8 reveals that the model exhibits bias in several metrics. The *Disparate Impact Ratio*, which measures the ratio of favorable outcomes for one group compared to another, shows that the model favors one group significantly more than the other, with a value of 1.5 compared to the fair value range of 0.8 - 1.2. The *Equal Odds Ratio*, which assesses whether the model predicts equally well for each group, shows a value of 0 for our model, indicating that it does not predict

equally well for both groups, while the fair value is 1. The *Statistical Parity Difference*, which measures the difference in favorable outcomes between groups, shows a value of 0.2 for our model, while the fair value range is -0.1 - 0.1.

The excerpt of the Stratified Bias Table (figure 4.8 on the right) shows that the model stratifies patients based on gender, with value 0 representing female and value 1 representing male. Here we can see, that the Selection Ratio indicates a higher probability that the model will detect heart disease in a male patient. Overall, while the model has a bias toward male patients, it still maintains a good overall rating of 8 out of 10 (82%).

Privacy

Now we turn our attention to the topic of data privacy. In this part of the assessment, we aim to answer whether it is robust to membership inference attacks and whether we infer training data, that was used to train the model, which indicates privacy leaks.

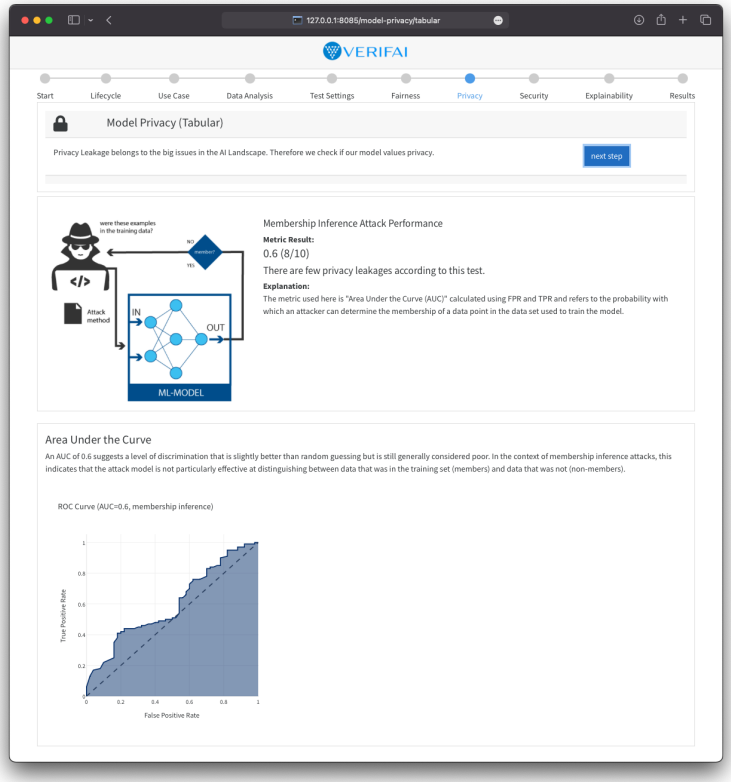


Figure 4.9: Tabular Model Privacy Leakage Evaluation (full-page screenshot)

We measured privacy leakage through a membership inference attack using the Black-Box MIA Attack. This attack method trains a classifier based on the loss values of the model's output to distinguish between members and non-members.

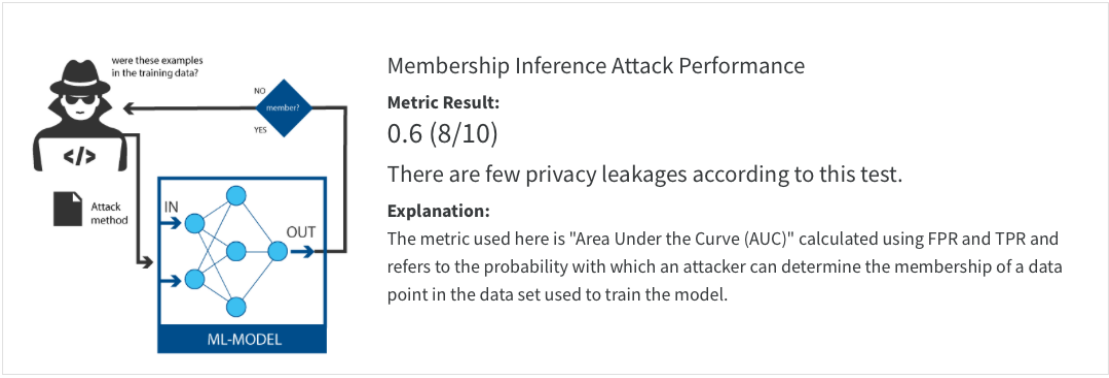


Figure 4.10: Tabular Model Privacy Leakage Evaluation (results details)

The AUC-curve in figure 4.11 shows that the attacker was able to infer some of the training points. The higher the value towards 1, the worse the situation. The lowest point is represented by the dashed line, which would be equivalent to random guessing.

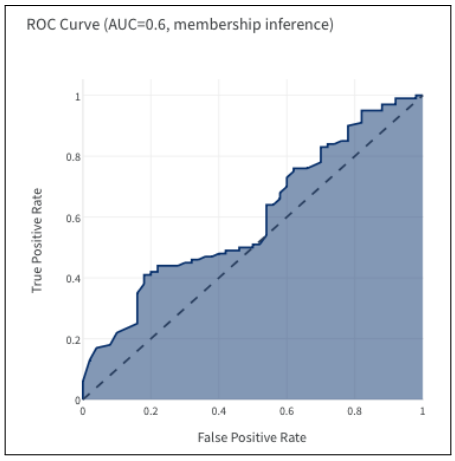


Figure 4.11: Tabular Model Privacy Leakage Evaluation (details)

The lower success rate of the Membership Inference Attack (MIA) on the Random Forest model trained on the heart disease dataset can be attributed to several factors. These factors include the inherent robustness of Random Forest models against overfitting, the

model's ensemble nature, dataset properties, and the limitations of the attack model and data. Additionally, preprocessing and feature engineering steps applied to the dataset might have improved the model's generalization capabilities, making the MIA less effective.

As we can see in figure 4.10 overall, while the model has a weak privacy leak, it still maintains a good overall rating of 8 out of 10 (AUC 0.6).

Robustness

Next, we attempt to deceive the model into assessing its robustness against adversarial attacks. This metric is referred to as *adversarial robustness*.

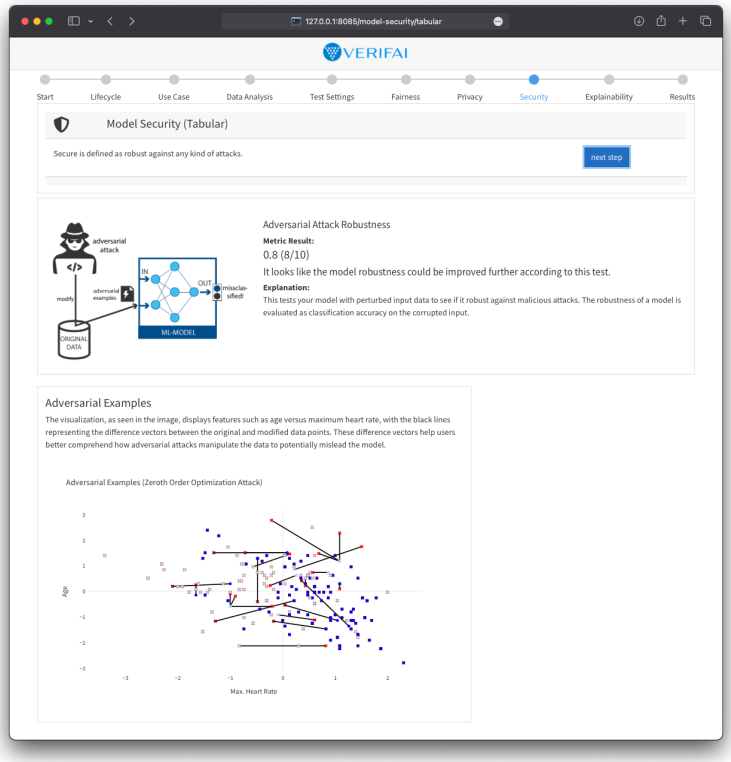


Figure 4.12: Tabular Model Robustness Evaluation (full-page screenshot)

The outcome (see 4.13) reveals that the model maintains its reasonably good accuracy (88%) in the face of adversarial attacks.

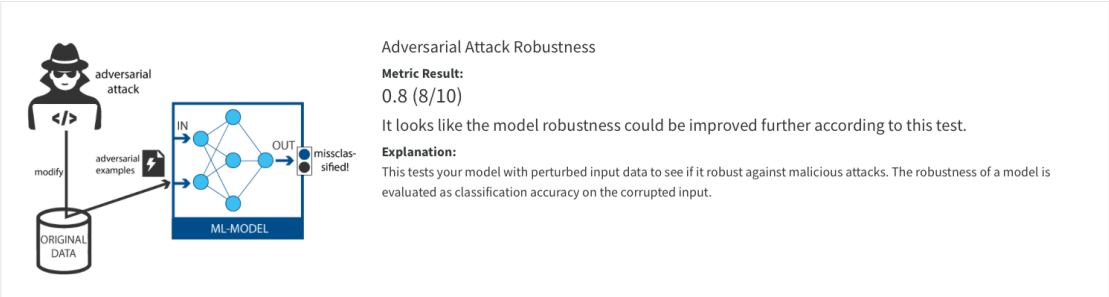


Figure 4.13: Tabular Model Robustness Evaluation (results)

Figure 4.14 visualizes the *Adversarial Examples* to provide users with a better understanding of the attack's impact on the model's decision-making process. The visualization, as seen in the image, displays features such as age versus maximum heart rate, with the black lines representing the difference vectors between the original and modified data points. These difference vectors help users better comprehend how adversarial attacks manipulate the data to potentially mislead the model.

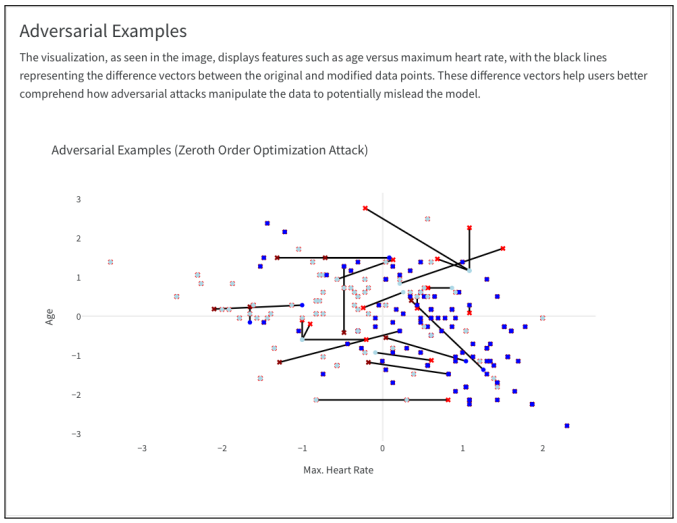


Figure 4.14: Tabular Model Robustness Evaluation (details)

The model's robustness against adversarial attacks could be attributed to several factors, including data preprocessing using scaling and normalizing in a way that reduces the model's sensitivity to small perturbations, making it more robust. The model's resistance to attacks could be due to some features being more significant in determining the outcome than others, causing it to be less affected by attacks that primarily target less critical features. The model's simplicity, having fewer decision nodes, may make

it less susceptible to adversarial attacks, as there are fewer decision boundaries for an attacker to exploit. The robustness of the model could be also a result of the adversarial attack being less effective, possibly due to the attack algorithm not being optimized for the specific model or not finding the most effective perturbations.

Figure 4.13 shows, that overall the model has strong robustness against this attack, with a good overall rating of 8 out of 10 (80 %).

Explainability

Figure 4.15: Tabular Model Explainability Evaluation using LIME explainer (full-page screenshot)

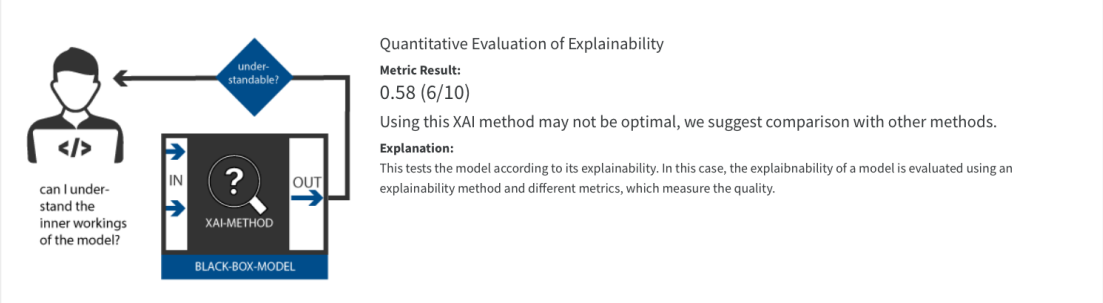


Figure 4.16: Tabular Model Explainability Evaluation using LIME explainer (results)

We will now investigate whether the model's decision-making process for predicting heart patients can be understood and if the explanations are of high quality. Additionally, we will assess if the interpretability algorithm is appropriate for the model.

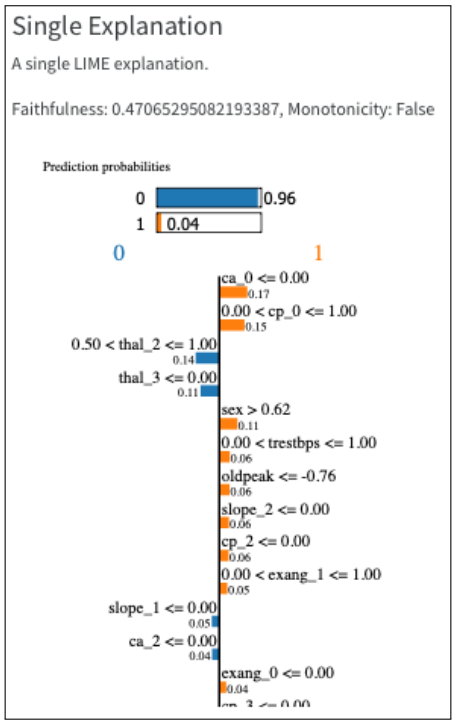


Figure 4.17: Tabular Model Explainability Evaluation using LIME explainer (truncated)

In the figure 4.17 above, we can visualize a single LIME explanation for an individual data point with one-hot encoded features.

Our metrics for evaluating the quality of the explainer and explainability were Faithfulness and Monotonicity. The results show, that the faithfulness score is high (see fig. 4.18) while the explanations are not monotonic (monotonicity is false) (see fig. 4.19).

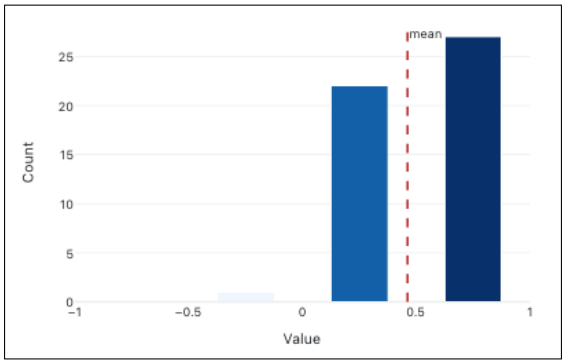


Figure 4.18: Tabular Model Explainability Evaluation using LIME explainer (faithfulness metric)

A high Faithfulness score indicates that LIME is effectively capturing the contribution of each attribute to the model's predictions. However, the falsy monotonicity shows, it doesn't guarantee that the importance assigned to each attribute will always be the same across different instances or scenarios. The importance of attributes can vary depending on the specific data points being explained or other factors related to the model's internal decision-making process.

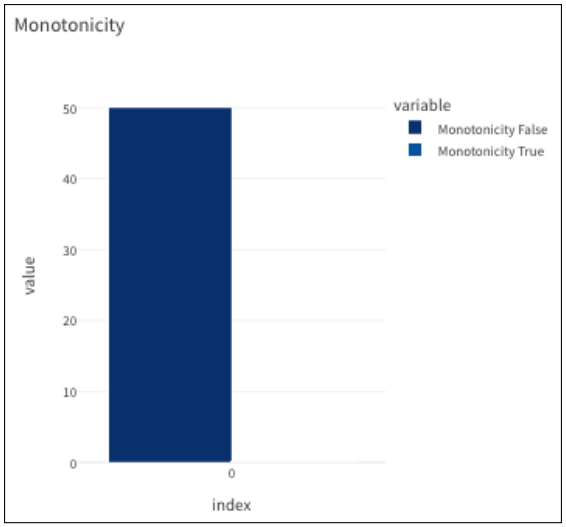


Figure 4.19: Tabular Model Explainability Evaluation using LIME explainer (monotonicity metric)

Monotonic explanations are generally considered more intuitive and easier to understand, as they show a consistent relationship between the importance of attributes and their impact on the model's predictions. This makes it easier for users to trust and rely on the explanations provided by the algorithm.

This is why it's important to consider both Faithfulness and Monotonicity when evaluating the quality of explanations provided by an interpretability algorithm like LIME. Therefore results for the *explainability score* indicate, that the quality of the model's explainability, as measured by LIME, is 58% on average, based on both Faithfulness and Monotonicity metrics.

Overall, the explainability overall rating is 6 out of 10 (58%) as we can see in the results section in 4.16, which suggests a comparison with other XAI-Methods.

Responsibility Score

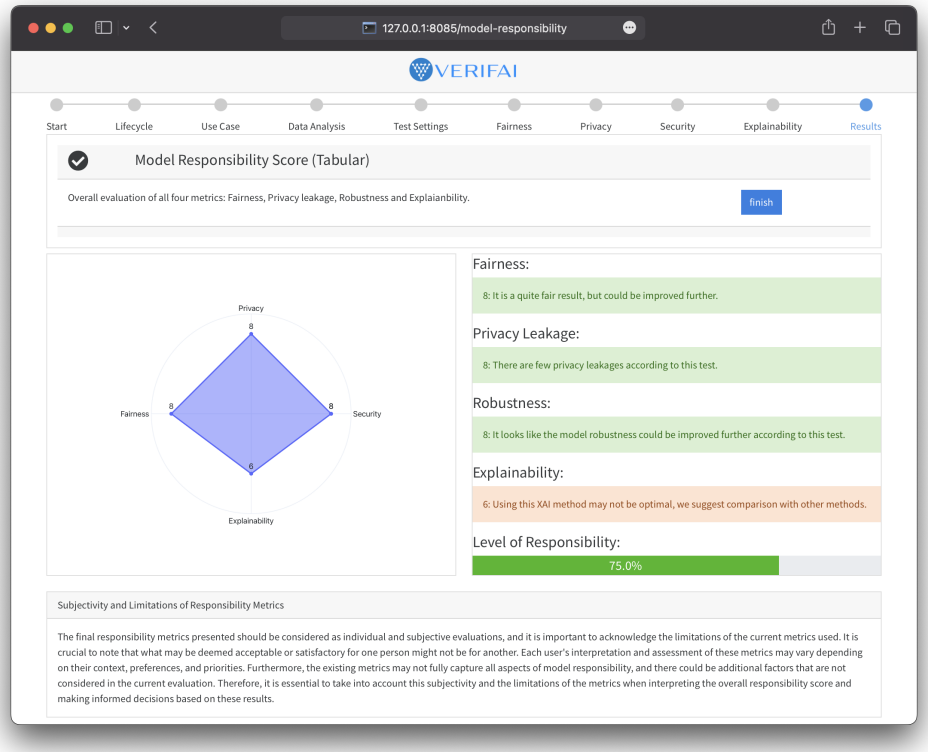


Figure 4.20: Tabular Model Responsibility Evaluation

In the final step, we calculate the overall responsibility score by converting the results onto a scale ranging from 1 to 10 and displaying them in a plot. Additionally, a brief explanatory text is provided to ensure users understand the meaning and interpretation of the score. This addresses inquiries regarding the model's fairness, resilience against privacy and adversarial attacks, and the extent to which it can be explained using suitable XAI methods. A summary overview is presented, highlighting the areas where the model performance is good (in the green zone) and where improvements are suggested (orange and red zone). For the tabular model in this example, we obtain a 75% responsibility score.

4.1.2 Results of the Computer Vision Model Evaluation

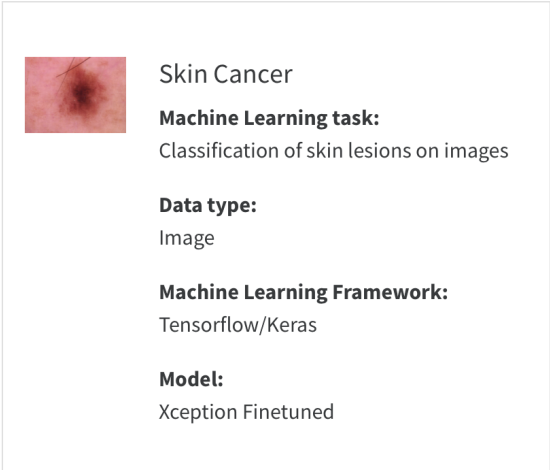


Figure 4.21: Select use case: image data

As mentioned earlier, the dataset also addresses a crucial health issue: skin cancer. We now have images representing seven different types of skin cancer. For training purposes, we utilized a convolutional neural network (CNN) using the Xception architecture, which is well-suited for handling image data. We proceed by selecting the relevant use case and continuing with the data analysis.

Data Analysis

Upon briefly examining the data in figure 4.22, we observe that the features exhibit a distinct inclination towards the Melanocytic-nevi type of skin cancer. Most images

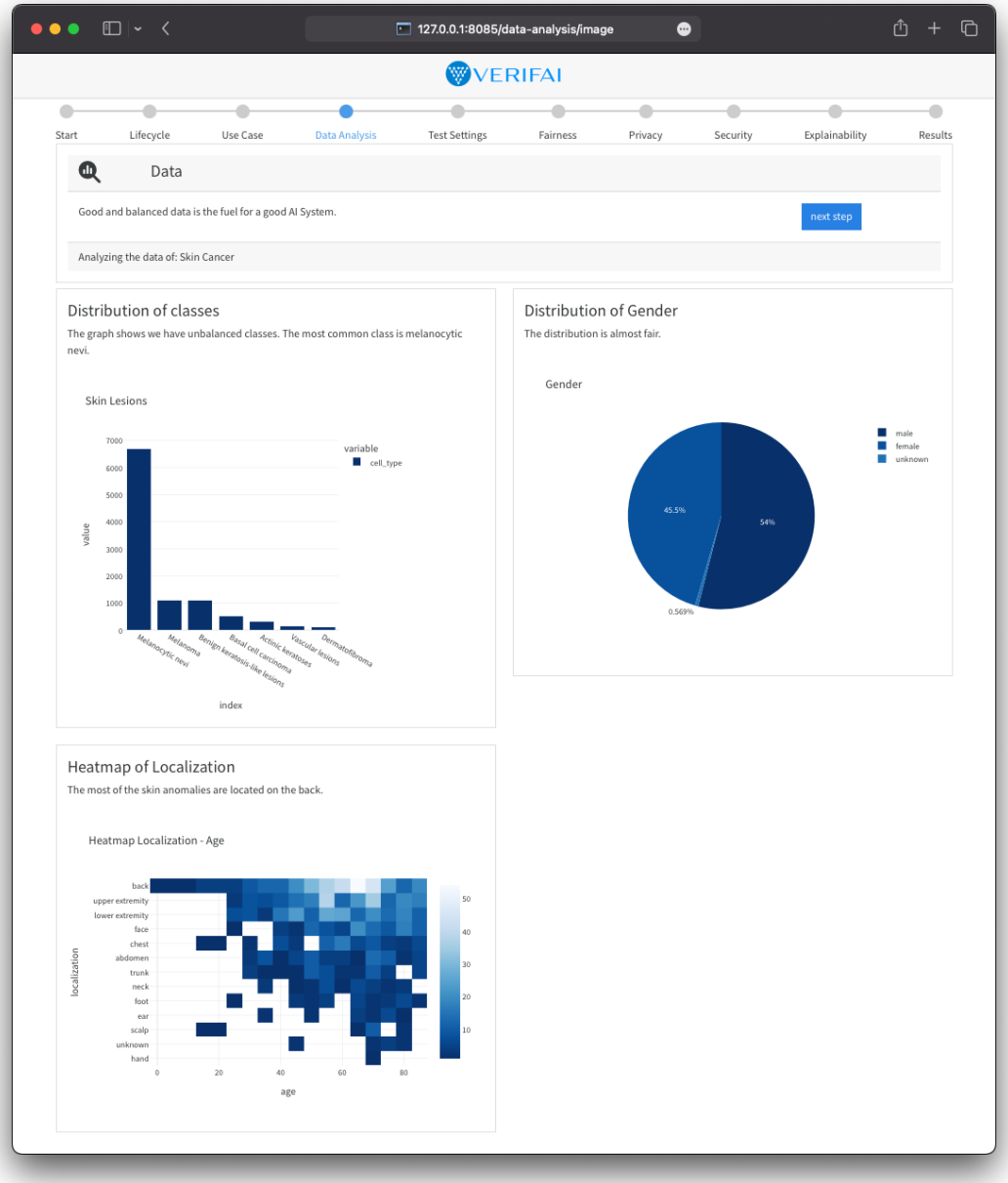


Figure 4.22: Data Visualization of the Skin Cancer dataset

capture the back or upper extremities of the subjects. Additionally, all images display individuals with lighter skin tones which is not immediately evident but should be noted, as it potentially introduces bias.

Test settings

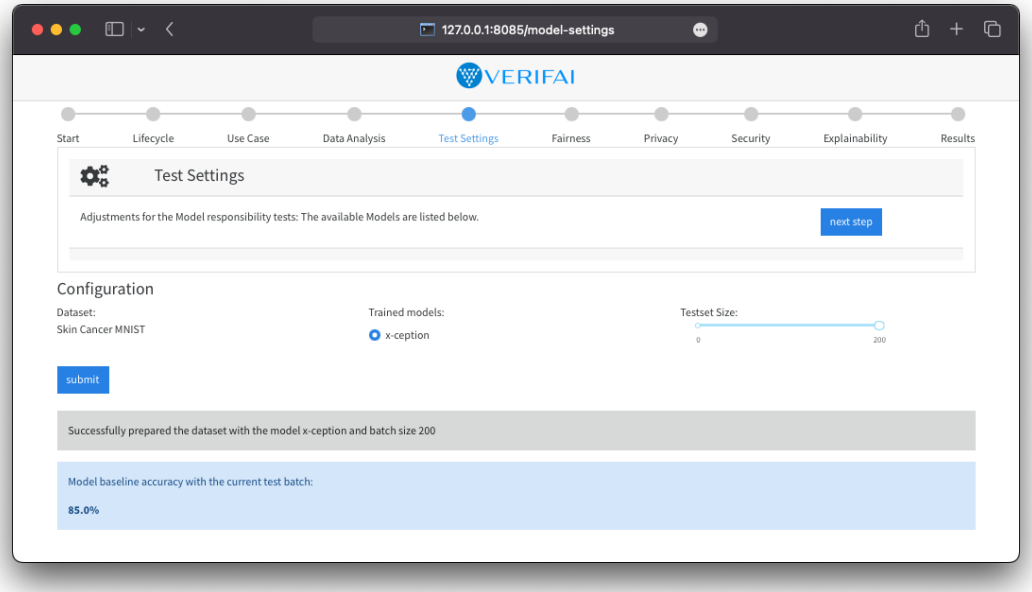


Figure 4.23: Test settings: image data (full-page-screenshot)

The test settings shown in 4.23 are also adjusted to 200 data points for equal test conditions. We select the Xception Model and evaluate the test batch. The model has a validation accuracy of 86%, but a good accuracy is not enough, furthermore, we will assess the model on our aspects of responsibility.

Fairness

In this part of the assessment, we aim to answer the questions if the model's prediction is fair and whether there is there a higher likelihood for a certain class to detect skin cancer. In figure 4.24 we can see a full-page screenshot of the fairness assessment and in figure 4.25 we show the details of the results section. It indicated that the model represents a medium level of fairness.

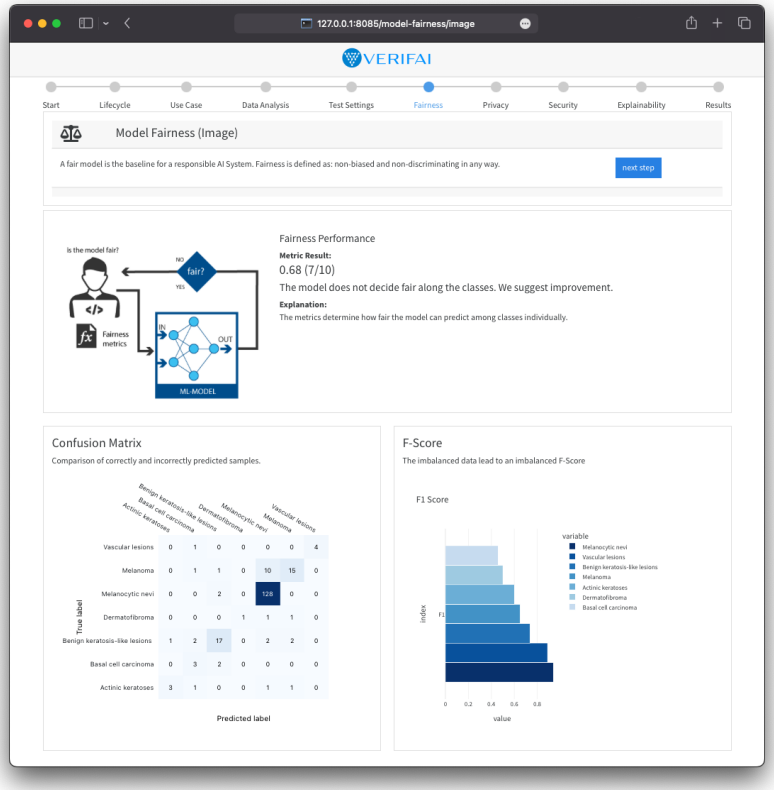


Figure 4.24: Computer Vision Model Fairness Evaluation (full-page-screenshot)

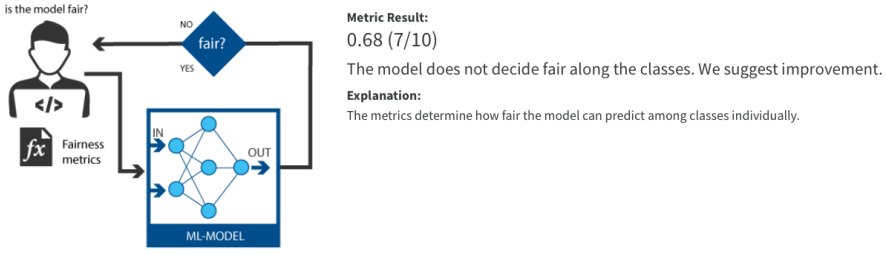


Figure 4.25: Computer Vision Model Fairness Evaluation (results)

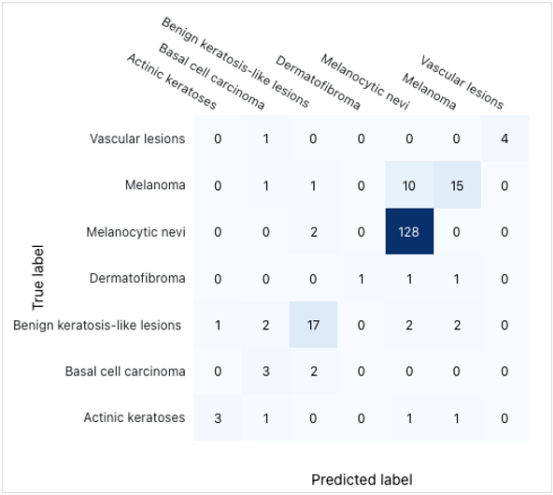


Figure 4.26: Computer Vision Model Fairness Evaluation (confusion matrix)

Figure 4.26 shows, that our model exhibits bias due to the imbalanced data distribution. Examining the confusion matrix, we find that the majority of correctly classified images belong to the Melanocytic-nevi type, which is the most prevalent category in the dataset. Additionally, some images were misclassified, where Melanomas were erroneously identified as Melanocytic-nevi.

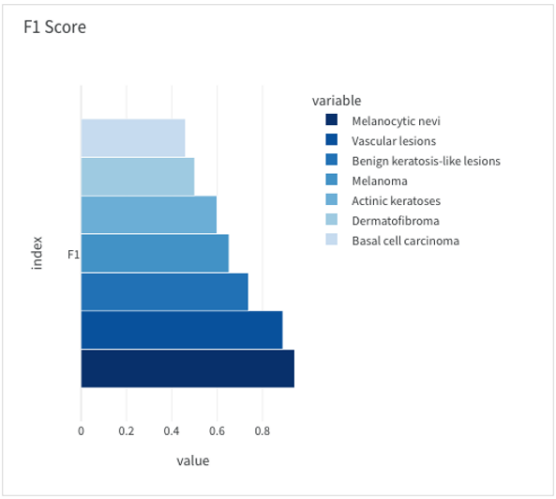


Figure 4.27: Computer Vision Model Fairness Evaluation (F1-Score)

For the evaluation, we calculate the F1-score, as shown in figure 4.27 for each feature and compute the average to obtain our final score. As demonstrated, we rely solely on

the F1-score for assessment in this iteration. As a result, we get a Score of 0.68, which is a 7 out of 10 in fairness performance (see 4.25 in the top section).

Privacy

In this section, we also examine the privacy metric, specifically through the lens of Membership Inference Attacks. The following image shows a full-page screenshot of the privacy leakage assessment.

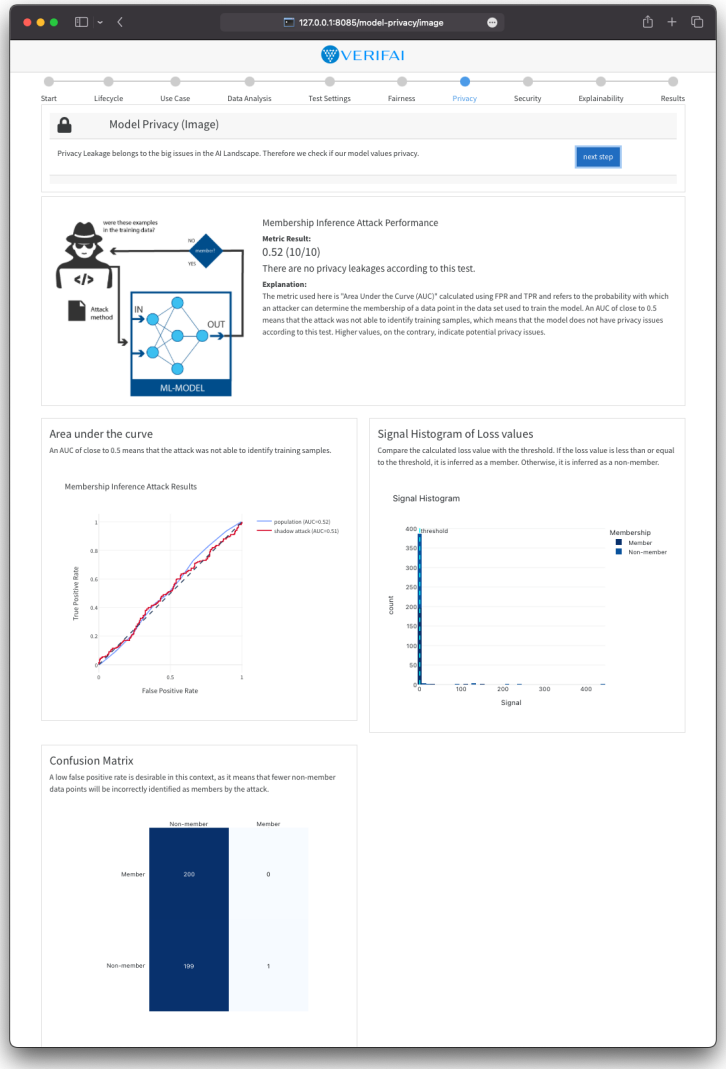


Figure 4.28: Computer Vision Model Privacy Leakage Evaluation (full-page-screenshot)

In order to be able to go into the details of the results, we show the different sections separately. The next figure shows the top section with the performance results.

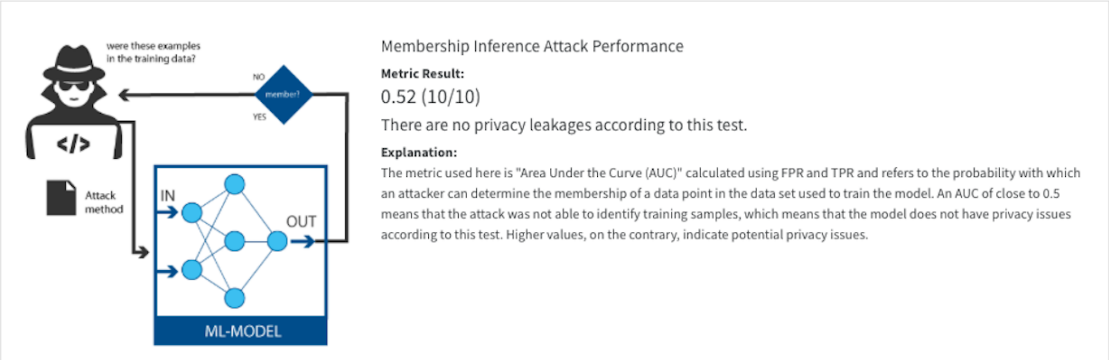


Figure 4.29: Computer Vision Model Privacy Leakage Evaluation (results)

We performed the assessment using two different attacks: the *MIA via Shadow Model* metric and the *MIA via population data*. In the first case, a shadow model was trained on the test data and subsequently applied as the attacker model on our target model. We also employed the *MIA via population data* metric, but the former proved to be more effective.

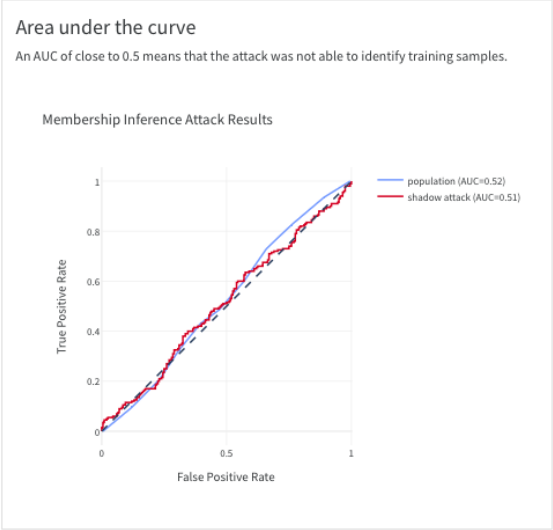


Figure 4.30: Computer Vision Model Privacy Leakage (AUC Score)

Given that we must consider the worst-case scenario, where the highest privacy leakage occurs. Consequently, we arrived at a result of 0.52 of the population-attack measured

using the AUC, indicating that we were nearly unable to recover any data using this attack.

We now will delve into the results of this attack by looking at the signal histogram of the loss values (see 4.31). We aim to find a threshold to separate between member and non-member data points. The result shows similar loss distributions, which indicates it is not possible to separate them.

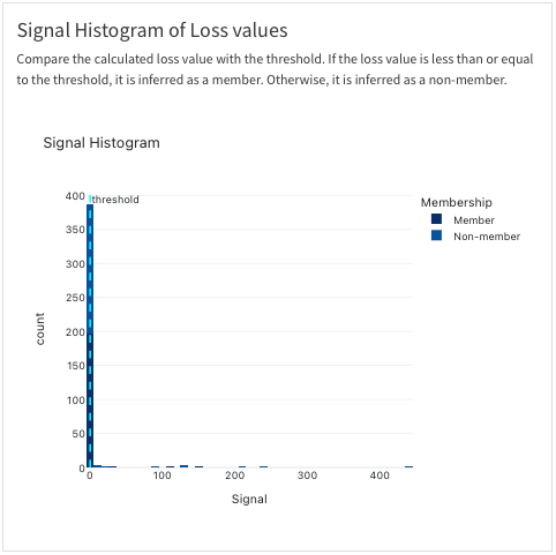


Figure 4.31: Computer Vision Model Privacy Leakage (Signal Histogram of loss values)

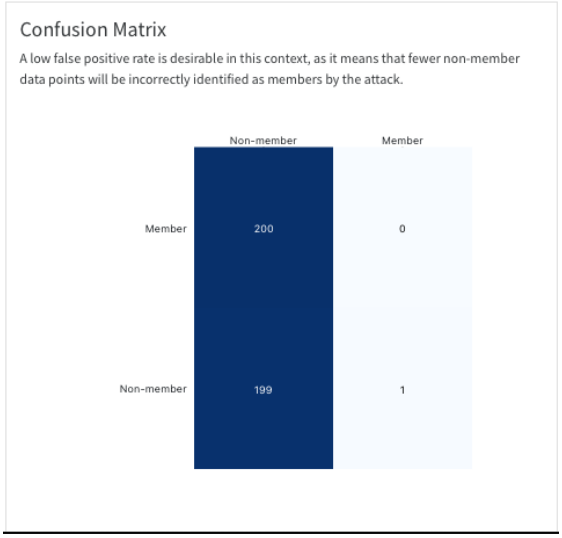


Figure 4.32: Computer Vision Model Privacy Leakage (Confusion Matrix)

The Confusion Matrix in 4.32 is also an indication that, while we were able to achieve a low false positive rate but the true positives are also low, which makes the attack unsuccessful. So our model is robust against this kind of attack because it can generalize well and the loss values do not differ between members and non-members.

Robustness

In this section, we apply various adversarial attacks to deceive our Computer Vision model and induce inaccurate predictions. The following figure gives an overview of the results. Afterward, we want to delve into the details.

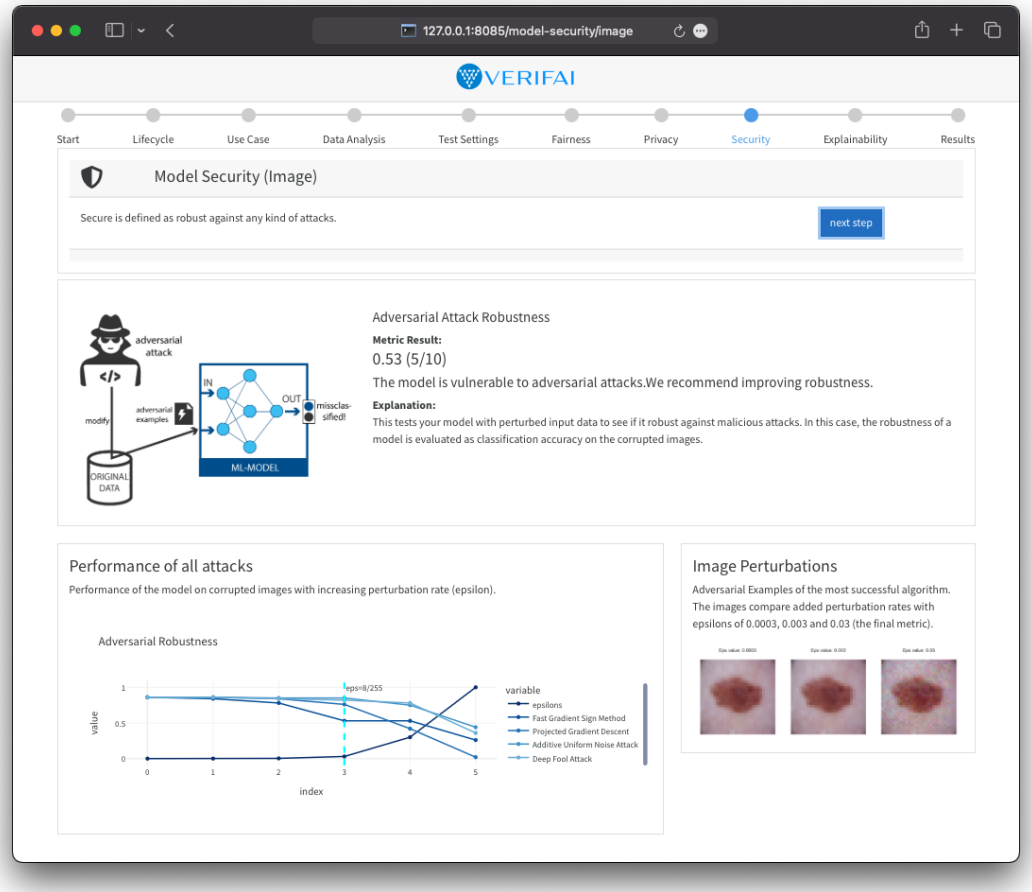


Figure 4.33: Computer Vision Model Adversarial Robustness Evaluation (full-page-screenshot)

Based on the figure 4.34 we can determine the robustness results, which we now discuss further.

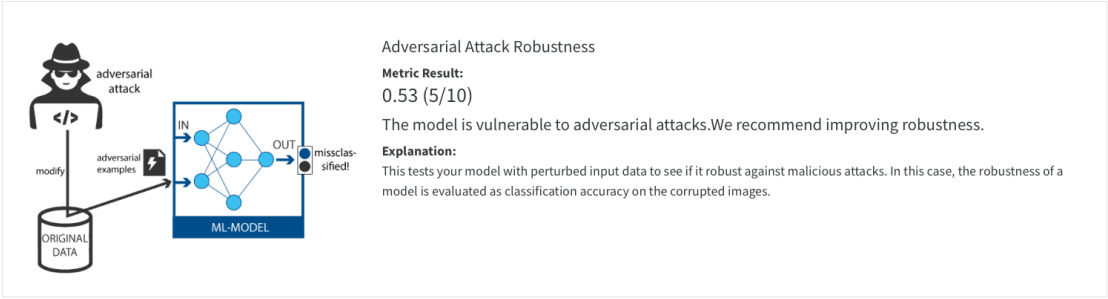


Figure 4.34: Computer Vision Model Adversarial Robustness Evaluation (results)

The adversarial attacks (see fig. 4.35) were performed using four different algorithms for adversarial attacks: FGSM Attack, PGD Attack, DeepFool Attack, and Additive Uniform Noise Attack. We introduce perturbations to the input images using these methods, increasing the disturbance rate represented by the ascending dark blue curve in the illustration. We measure the accuracy in each round on these perturbed images, and the final metric is set at an epsilon of 0.03 (corresponding to round 3). This epsilon value is also utilized for benchmarking in Croce et al. [2021].

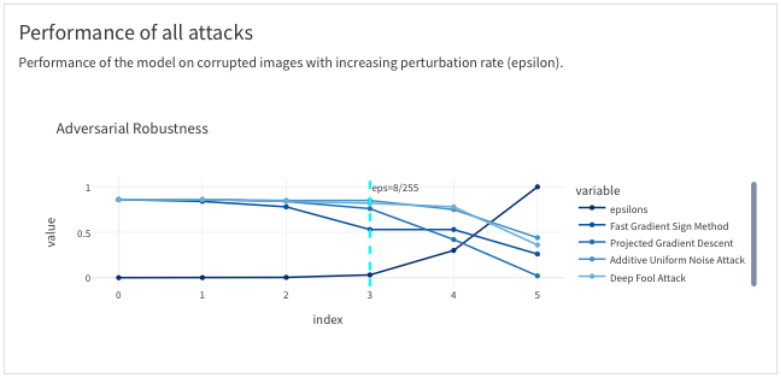


Figure 4.35: Adversarial Attacks for measuring Adversarial Robustness of the Computer Vision Model (tested epsilons = [0.0, 0.0003,0.003,0.03,0.3,1.0])

The images of figure 4.36 show an example of this process. Now we can determine, that the robustness of the model was 0.53, which means it is vulnerable to adversarial attacks. The results indicate that the model can be deceived with only a few pixel modifications. In the worst-case scenario, we observe an *Accuracy under Attack* of 52% using the *FGSM Algorithm* which is then calculated as a privacy score of 5 out of 10 (see 4.34).

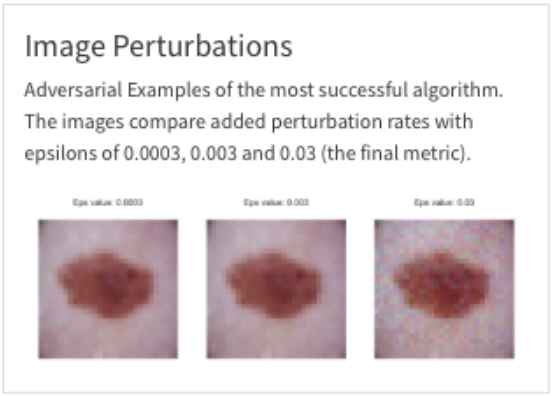


Figure 4.36: Adversarial Attacks through image perturbations with increasing perturbation rates (epsilon)

Explainability

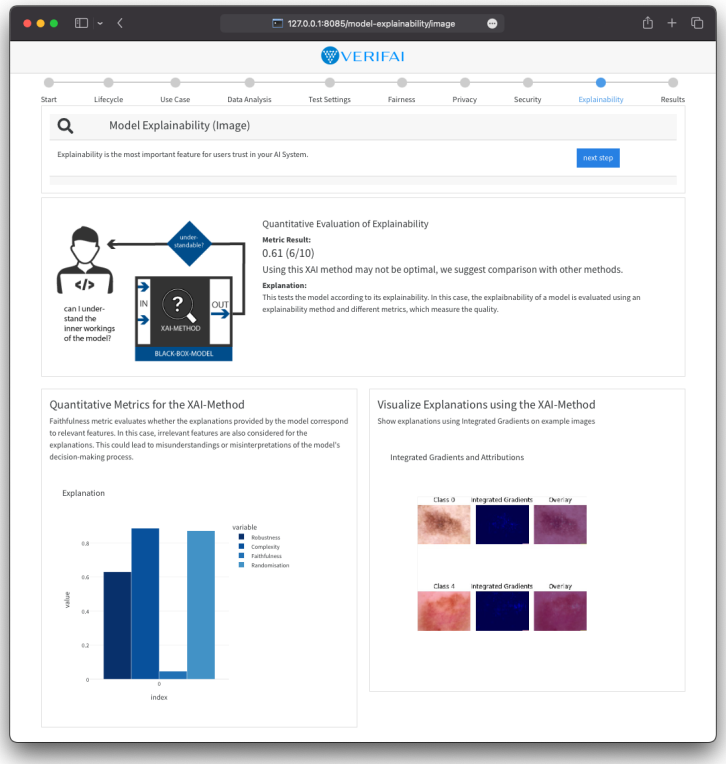


Figure 4.37: Computer Vision Model Explainability Evaluation (full-page-screenshot)

In this section, we deal with measuring explainability using XAI methods and evaluation metrics, as it is crucial as it is essential to understand the decision-making process of the model.

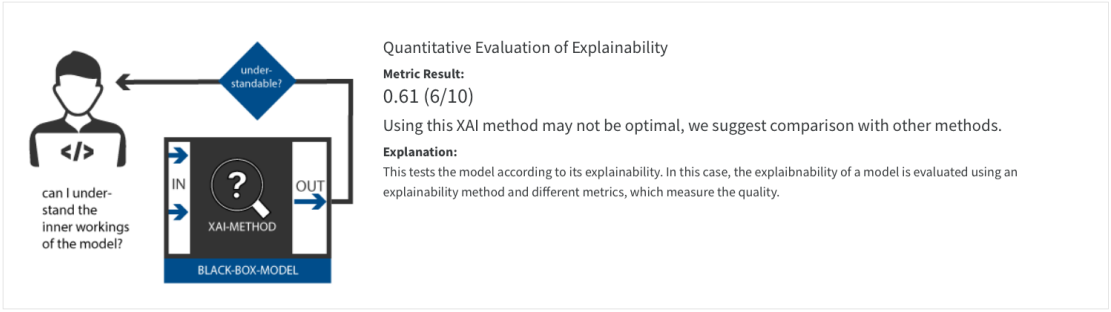


Figure 4.38: Computer Vision Model Explainability Evaluation (results)

We utilize four metrics to assess explanations from various perspectives:

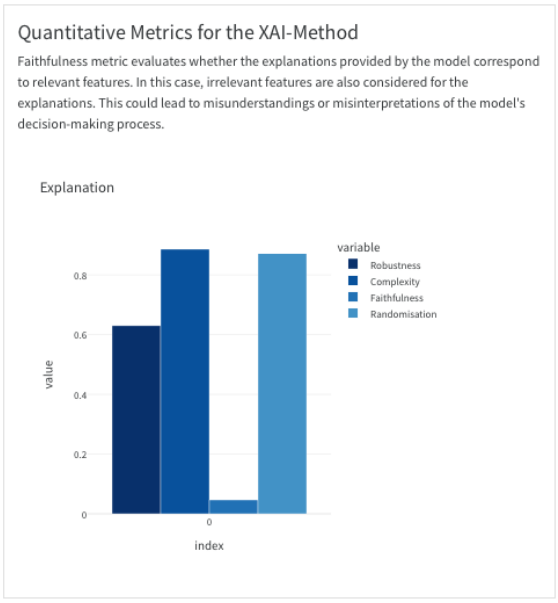


Figure 4.39: Computer Vision Model Explainability Evaluation Metrics

In the results, one metric stands out having a very low score: *Faithfulness*. The metric evaluates whether the explanations provided by the explainer correspond to relevant features. In this case, irrelevant features are also considered for the explanations. This could lead to misunderstandings or misinterpretations of the model's decision-making process. The other metrics achieve higher scores. In this case *Robustness* refers to the

stability of explanations against minor input perturbations in the images. *Complexity* measures the conciseness of a model's predictions and whether the model can make its predictions with only a small number of features. *Randomization* investigates the effect of increasingly randomized parameters on the quality of explanations provided by the explainer, such as the distance between the original explanation and the explanation for a randomly chosen class.

In 4.40 a visualization of an explanation using an image is displayed. The visualization of the Integrated Gradients explanations is overlaid onto the image to make it visible to the user.

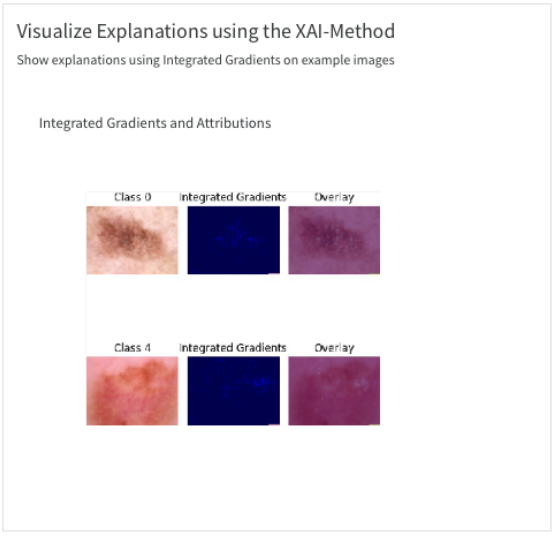


Figure 4.40: Visualized explanations using the Integrated Gradients Explainer

This results in an average score of 0.61 (see figure 4.38) for our final evaluation based on all metrics, leading to a score of 6 out of 10. The tool also recommends comparing the explainer with others, as it may not be optimal.

Responsibility Score

Through looking at the responsibility score in figure 4.41 we can now conclude that the computer vision model does not appear sufficiently prepared for using it in a production environment. It exhibits significant weaknesses in fairness, security, and explainability, although privacy does not seem to be at risk. The overall result is 70% with a recommendation for improvement.

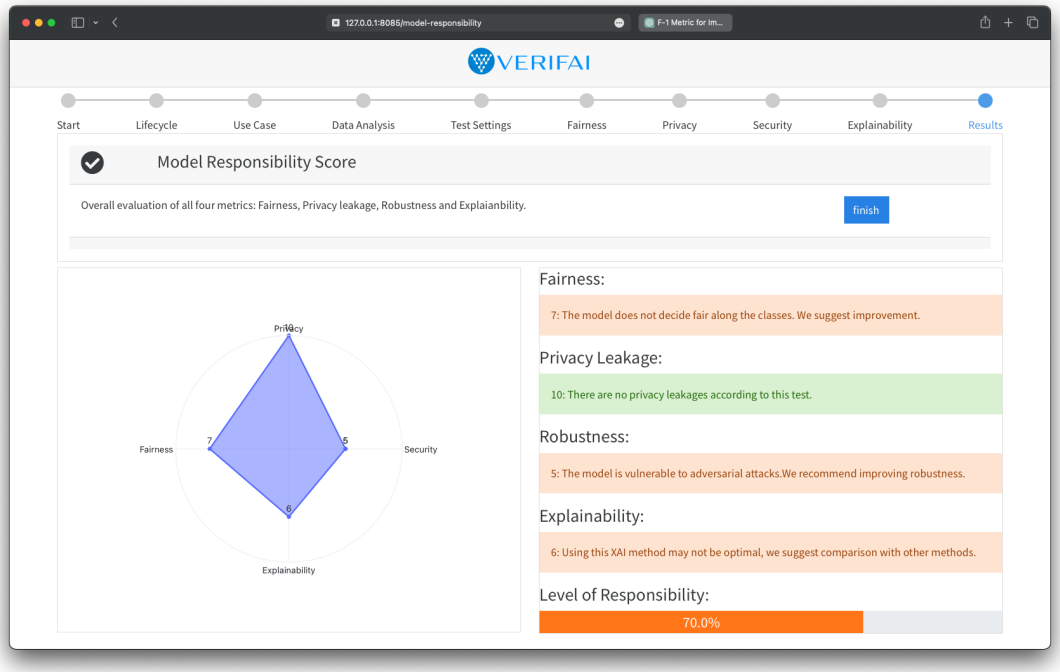


Figure 4.41: Computer Vision Model Responsibility Evaluation

4.1.3 Results of the NLP Model Evaluation

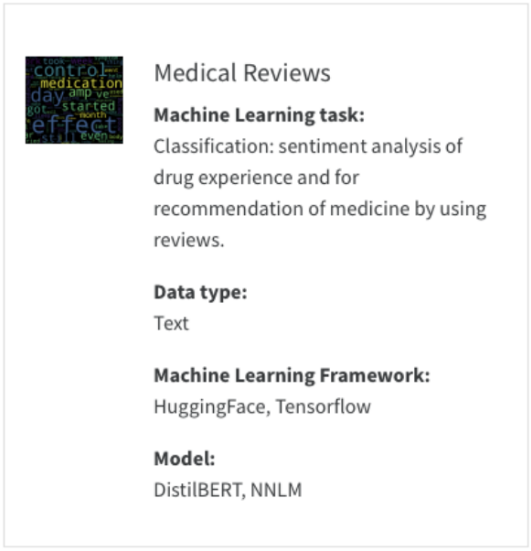


Figure 4.42: Use Case: Medical Reviews

Now, we delve into the evaluation results of the Natural Language Processing (NLP) model. This model has been trained on textual data from medical reviews, with the primary objective of offering recommendations for medications, and discerning whether they are efficacious or not. In the figure 4.42 we can also notice that we could select from two different models (DistilBERT and NNLM) but we focus on the DistilBERT model here, which was trained using the HuggingFace Framework.

Data Analysis

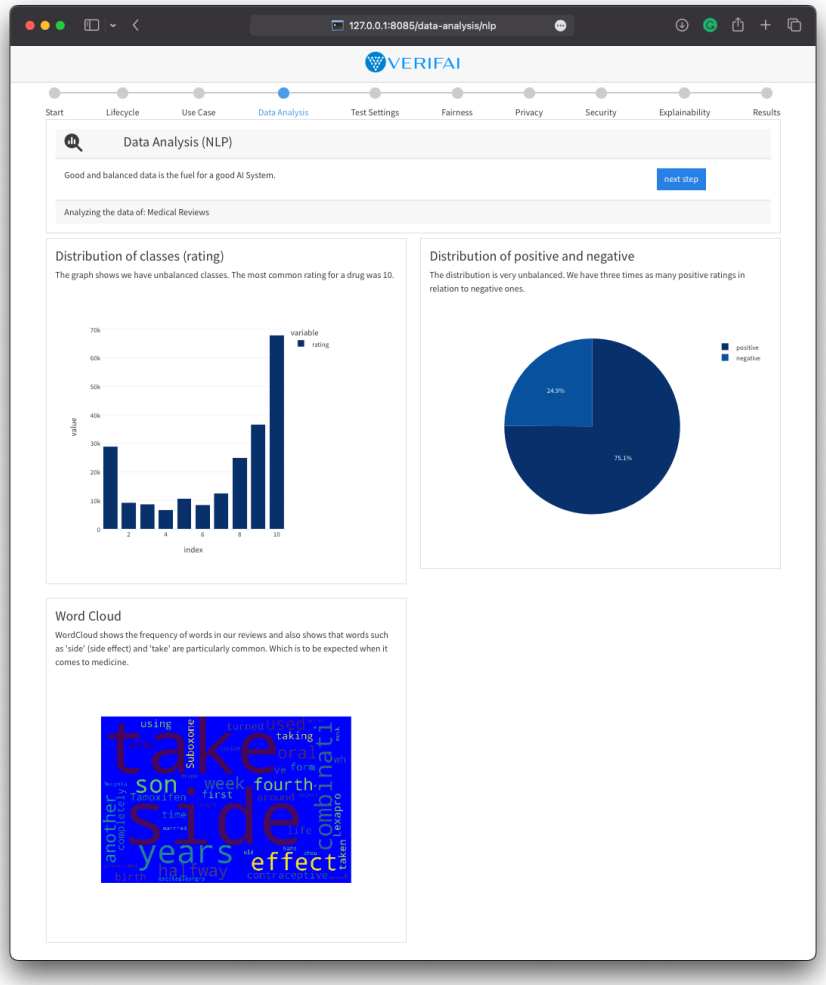


Figure 4.43: Data analysis on text data (full-page-screenshot)

Upon loading the dataset, we can already observe a significant bias in the data (see Figure 4.43). There are considerably more positive than negative review texts. This can be attributed to the preprocessing of the data, wherein data points with a rating of 5 or higher are classified as positive, and those with a rating of 0 to 4 are considered negative. Consequently, the model was trained as a binary classifier.

Unfortunately, the initial distribution was already biased, resulting in a significantly higher number of positive texts. In fact, the majority of the texts have a high rating of 10. While other features, like the distribution of words, have been analyzed during the data analysis phase, they will not play a role in the subsequent analysis. Instead, we will focus on classifying the text and the targets as positive and negative.

Fairness

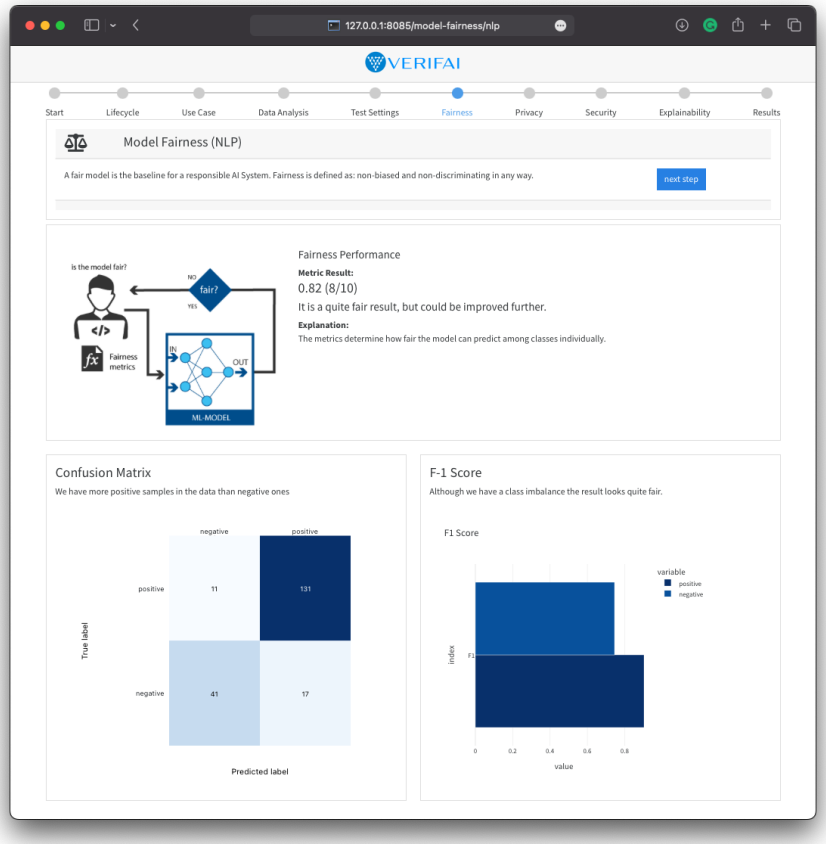


Figure 4.44: NLP Model Fairness Evaluation

In this respect, we want to answer the question, of whether the model's prediction is fair and if there is a higher likelihood for a certain class to detect the sentiment.

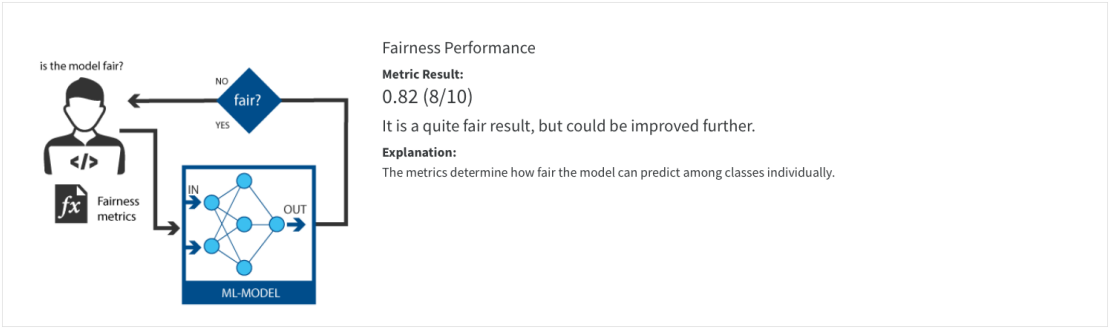
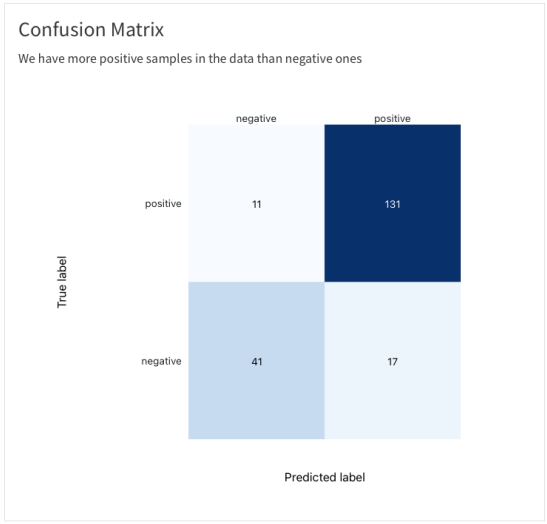


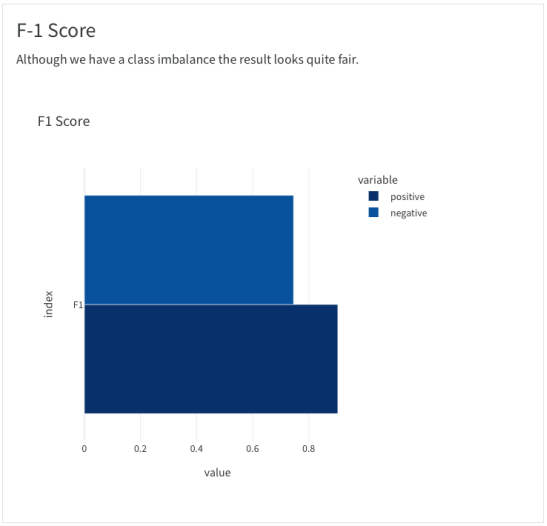
Figure 4.45: NLP Model Fairness Evaluation Results

The plot of the Confusion matrix (see fig. 4.46a) as well as the plot of the F-1 Score (see fig. 4.46b) indicates a bias toward positive sentiments, which was caused by unequal data distribution. Therefore there is a higher likelihood of detecting positive sentiment.

This results in an average score of 0.82 (see figure 4.45) for our final evaluation based on all metrics, leading to a score of 8 out of 10.



(a) Confusion Matrix of positive and negative predictions



(b) NLP Model F-1 Score

Figure 4.46: Membership Inference Attack Results (details)

Privacy

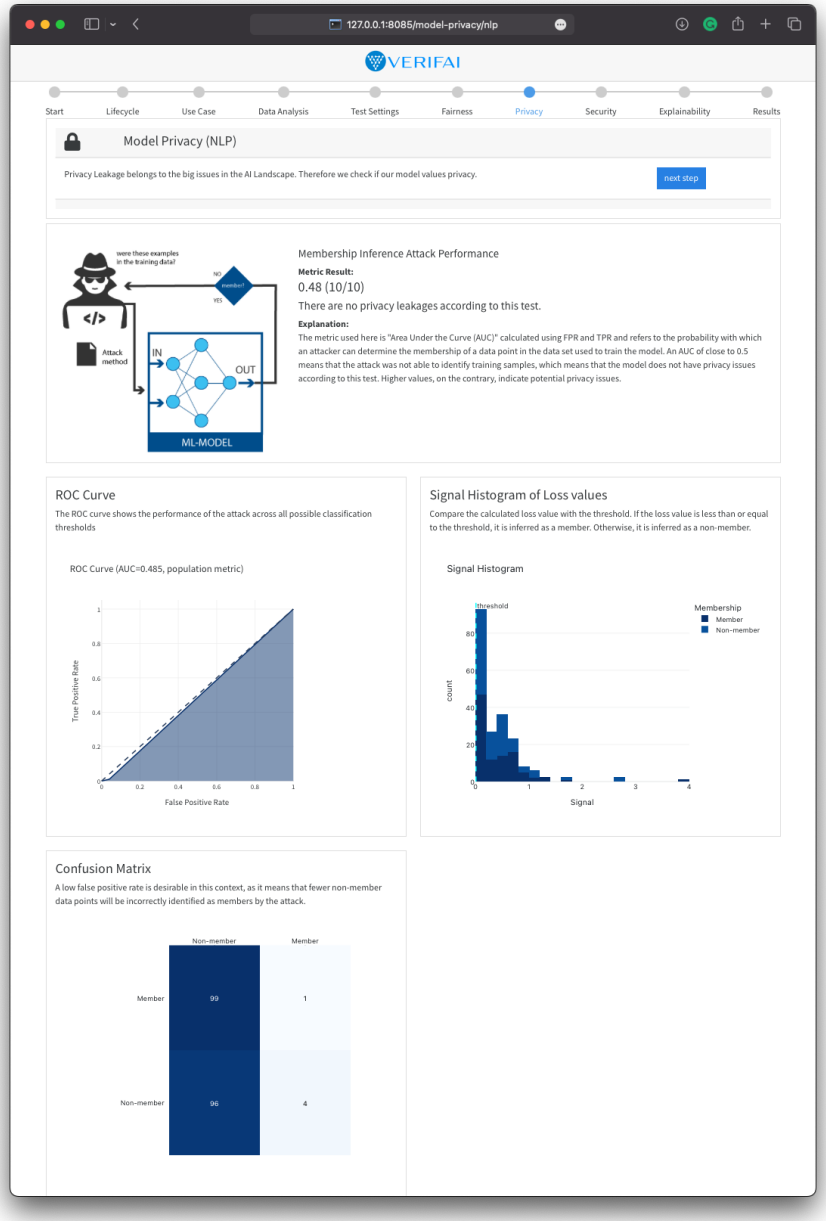


Figure 4.47: Model privacy on image data (full-page-screenshot)

In means of privacy leakage, our NLP model seems to be robust against it. For testing, we conduct also Membership Inference Attack on the NLP model. The model gets the

highest score for privacy 10 out of 10 for the privacy score. Using the following plots will explain the reasons for this in detail.

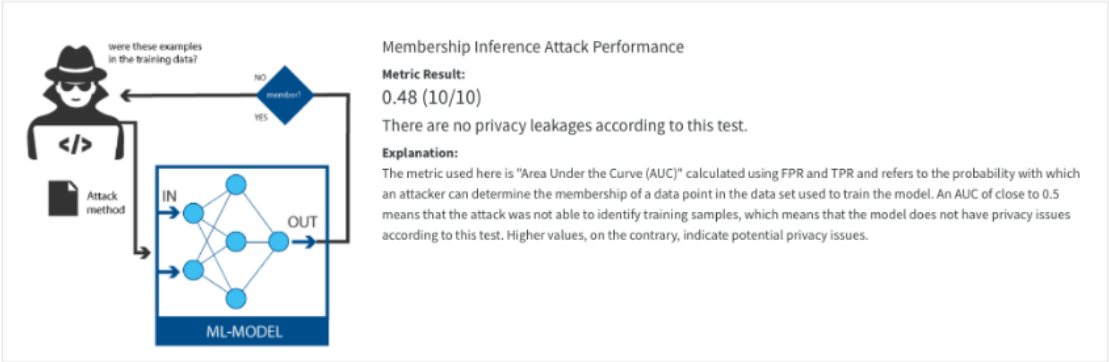
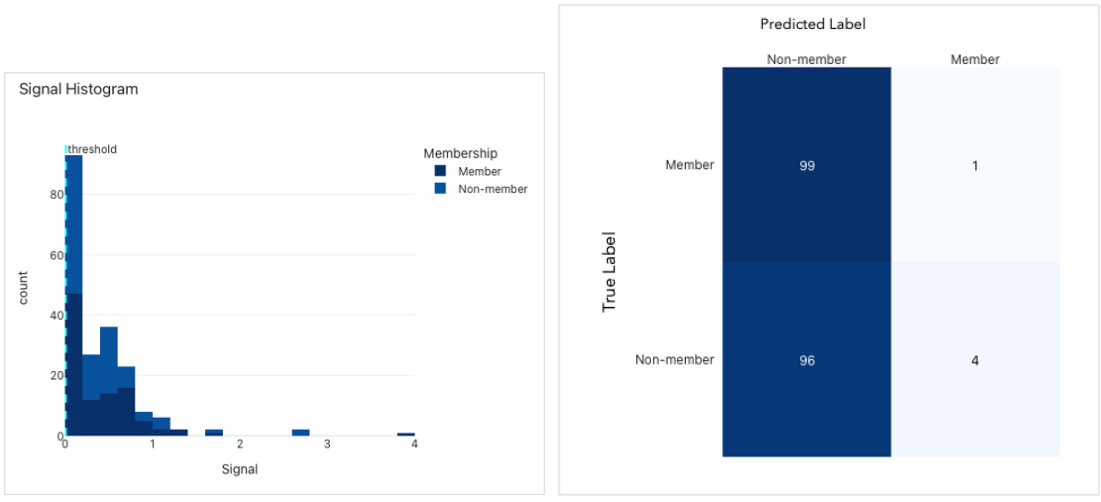


Figure 4.48: NLP Model Privacy Leakage Evaluation (results)

We have carried out one test using the population metric (directly testing the target model). The histogram in 4.49a of the results indicates that the attacker could not infer training data. We aimed to find a threshold to separate between members and non-members, and the result was showing a similar loss distribution, not possible to differentiate between them (signal histogram). The confusion matrix on the right plot confirms the result in low false positives (4) which is desirable, but also low true positives (1) which is not desirable.



(a) Signal Histogram **(b)** Confusion Matrix

Figure 4.49: NLP Model Membership Inference Attack Results (details)

Therefore in means of privacy leakage, our NLP model seems to be robust against it, which gets also confirmed by the AUC of ≈ 0.5.

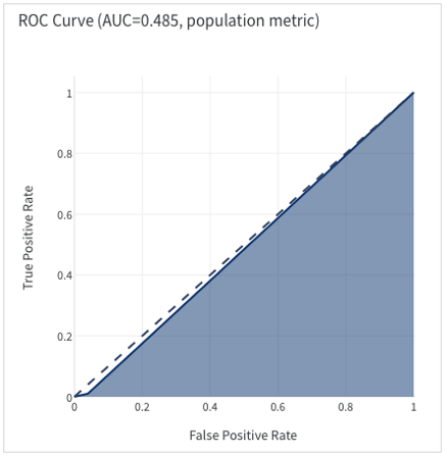


Figure 4.50: NLP Model Membership Inference Attack AUC Score

Robustness

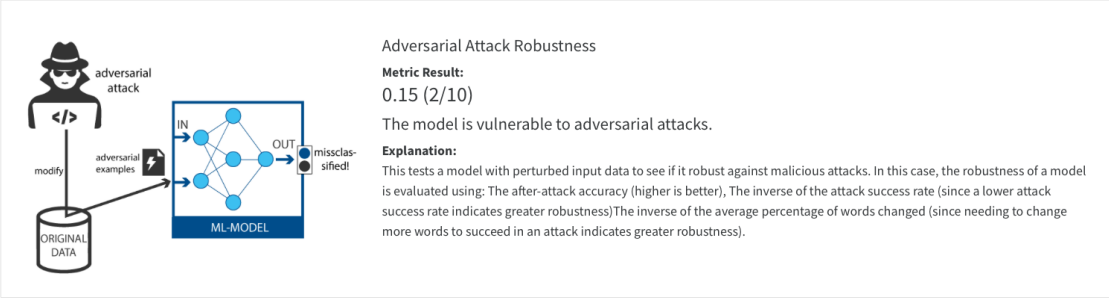


Figure 4.51: NLP Model Adversarial Attack Robustness (details)

Attackers can also deceive our NLP model and cause it to make incorrect predictions. As a consequence, an incorrect medication recommendation could have severe repercussions. We can determine from the results, that the model has only an adversarial robustness of 2. In the following, we will go through the details leading to those results.

In this case, we have employed four distinct algorithms that modify the input text in various ways, such as inducing word swaps. The algorithms used are TextBugger, DeepWordBug, TextFooler, PWWS.

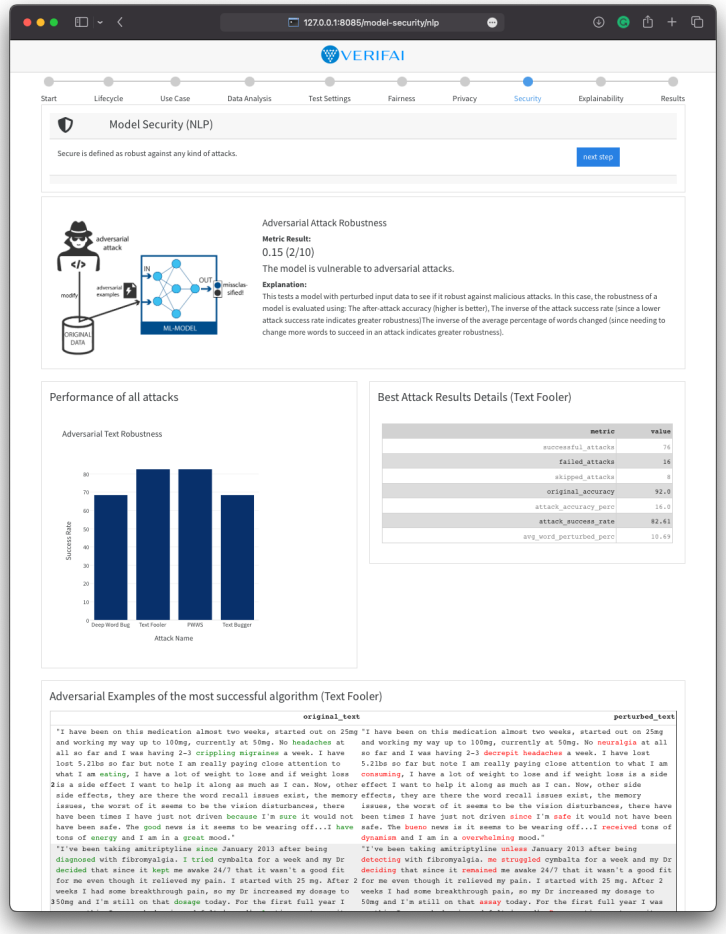


Figure 4.52: NLP Model Adversarial Attack Robustness (full-page-screenshot)

metric	value
successful_attacks	76
failed_attacks	16
skipped_attacks	8
original_accuracy	92.0
attack_accuracy_perc	16.0
attack_success_rate	82.61
avg_word_perturbed_perc	10.69

Figure 4.53: Best Adversarial Attack Results (TextFooler)

This attack was run on 200 examples. Out of those 200, the model initially predicted 8 of them incorrectly (skipped_attacks); this leads to an original accuracy of 92.0% (original_accuracy). TextAttack ran the adversarial attack process on the remaining examples to find a valid adversarial perturbation for each one. Out of those 16 attacks failed (failed_attacks), leading to a success rate of 82.61% (attack_success_rate). Another way to articulate this is that the model correctly predicted and resisted attacks for 16 out of 200 total samples, leading to an accuracy under attack (attack_accuracy_perc) of 16.0%. Among the 76 successful attacks (successful_attacks), on average, the attack changed 10.69% of words (avg_word_perturbed_perc) to alter the prediction.

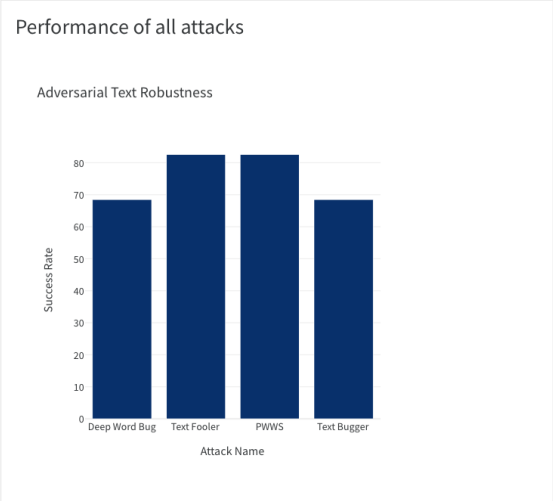


Figure 4.54: Adversarial Attack Results Comparison (details)

For the robustness score, we are taking three of the metrics into account: the model's after-attack accuracy, the inverse of the success rate of adversarial attacks, and the extent of input modifications needed by the adversary (avg_word_perturbed_perc). A higher robustness score indicates a model that effectively maintains accuracy, resists adversarial attacks and necessitates more significant alterations by the adversary to succeed. This provides a comprehensive view of the model's ability to withstand adversarial manipulation.

Figure 4.55 displays a comparison between the original text and the modified text using one of the best-performing algorithms, in this case, the examples are generated using *TextFooler*. In this example, the model has a robustness score of 2 out of 10 (15%) using the most successful attacks which means, that it needs to be improved using mitigation techniques.

	original_text	perturbed_text
2	"I have been on this medication almost two weeks, started out on 25mg and working my way up to 100mg, currently at 50mg. No headaches at all so far and I was having 2-3 crippling migraines a week. I have lost 5.21bs so far but note I am really paying close attention to what I am eating, I have a lot of weight to lose and if weight loss is a side effect I want to help it along as much as I can. Now, other side effects, they are there the word recall issues exist, the memory issues, the worst of it seems to be the vision disturbances, there have been times I have just not driven because I'm sure it would not have been safe. The good news is it seems to be wearing off...I have tons of energy and I am in a great mood."	"I have been on this medication almost two weeks, started out on 25mg and working my way up to 100mg, currently at 50mg. No neuralgia at all so far and I was having 2-3 decrepit headaches a week. I have lost 5.21bs so far but note I am really paying close attention to what I am consuming, I have a lot of weight to lose and if weight loss is a side effect I want to help it along as much as I can. Now, other side effects, they are there the word recall issues exist, the memory issues, the worst of it seems to be the vision disturbances, there have been times I have just not driven since I'm safe it would not have been safe. The bueno news is it seems to be wearing off...I received tons of dynamism and I am in a overwhelming mood."

Figure 4.55: Adversarial examples using TextFooler (single example)

Explainability

In this section, we aim to understand the decision-making process of the NLP model and assess whether its explanations are sufficiently clear and informative.

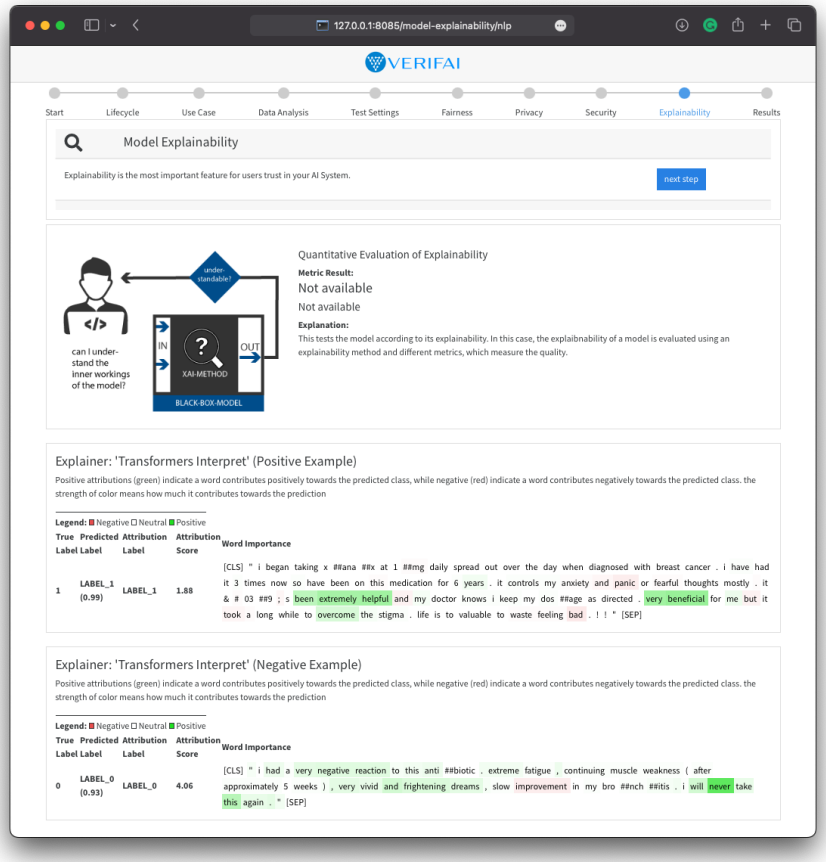


Figure 4.56: NLP Model Explainability Evaluation

We employ the Transformers Interpret Explainer to provide an explanation for individual data points, which can be visualized for the user's understanding. Currently, there are no implemented metrics for evaluating explainability in NLP models due to several technical issues. However, we can still involve a human-in-the-loop to assess the quality and accuracy of the provided explanations and determine if they align with human reasoning.

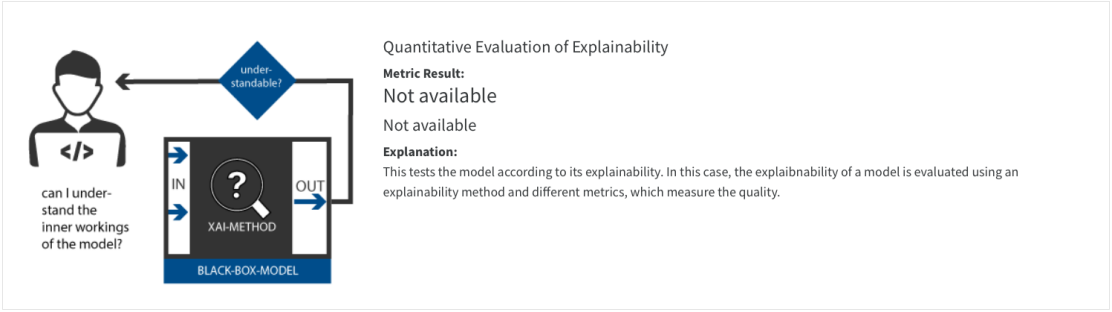


Figure 4.57: NLP Model Explainability Evaluation

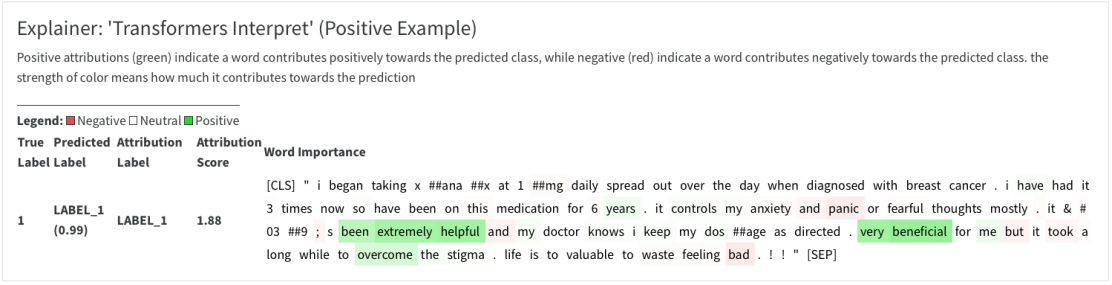


Figure 4.58: NLP Model Explainability Evaluation using Transformers Interpret

In this example, green markers indicate a positive contribution toward the prediction e.g. *extremely helpful, very beneficial, overcome* are recognized as positive words

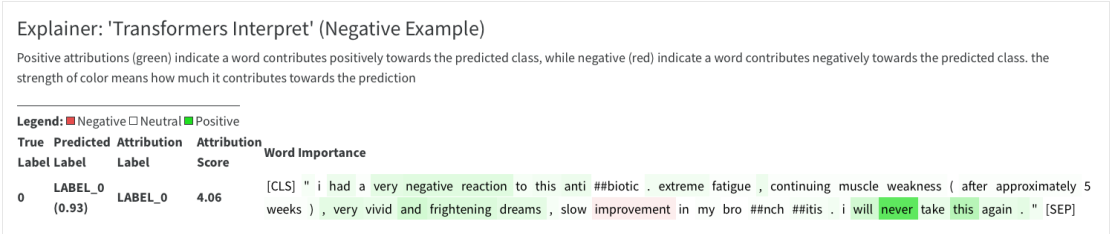


Figure 4.59: NLP Model Explainability Evaluation using Transformers Interpret

Red markers indicate a negative contribution toward the prediction, e.g. *panic, bad* are recognized as negative words. In this example, the explainer seems to caption the right attributions for the predictions, which is human-understandable.

Responsibility Score

The evaluation of our NLP model reveals considerable shortcomings in terms of fairness and security, necessitating substantial improvements before deployment.

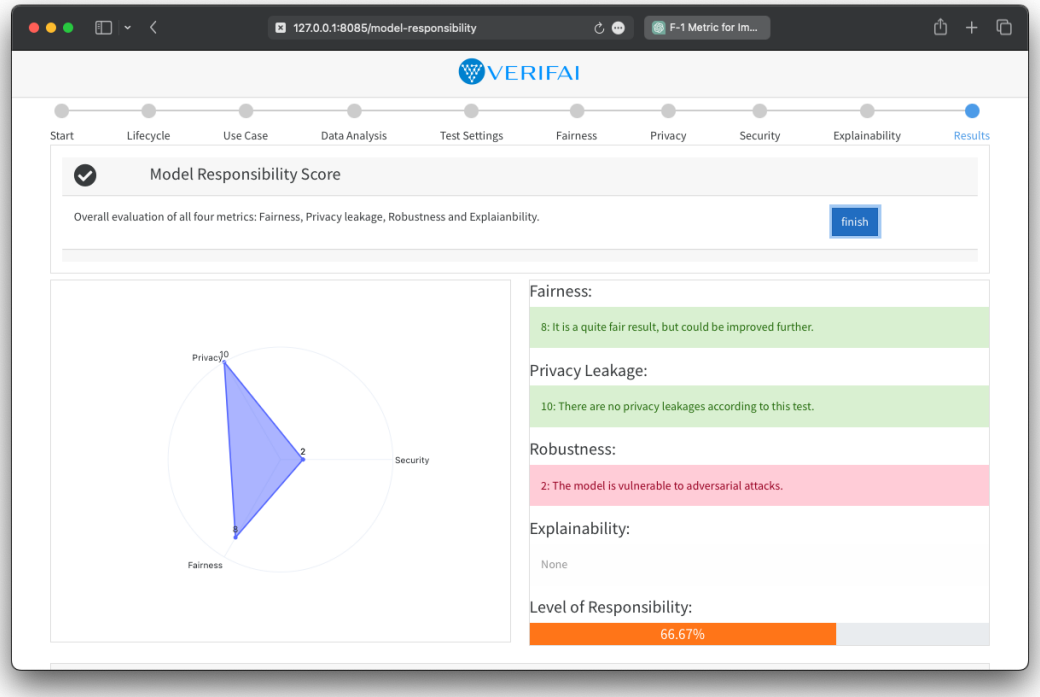


Figure 4.60: NLP Model Responsibility Score

With an overall score of 66.67%, the model exhibits some weaknesses that could impact its reliability and trustworthiness. Employing this model in its current state would not be advisable, given the potential risks associated with its robustness. Further investigation and refinement are essential to address these issues and enhance the model's overall responsibility.

4.2 Technical Challenges

This section discusses the technical challenges faced during the implementation of our framework.

Limitations of the available Metrics

The F-1 Score was the only fairness measure available for neural networks, which is insufficient for a comprehensive fairness assessment. This limitation arose because it was difficult to find a tool for evaluating neural networks on healthcare datasets using all fairness metrics. Most toolkits are either domain-specific or only support simple binary classifiers trained on tabular data. Additionally, our current dataset lacks protected attributes like gender or age to measure bias.

Challenges with Dataset Quality Assessment

In the initial stage, we visually analyzed the data to identify potential biases but did not perform additional assessments (e.g., detecting privacy issues or adversarial images) as the data was unsuitable for such evaluations. Some libraries attempt to detect biases in images or text but often focus on more general data like people or objects. Detecting biases in skin data or medical review texts would require specific measures, which were not available.

We encountered compatibility issues when attempting to evaluate datasets with some libraries, such as those for detecting adversarial examples in images. As a result, we deferred the development of these measures for future work, focusing on model validation for this stage.

Limitations of the available Toolkits

Several limitations were encountered with XAI and NLP toolkits. For example, no library currently supports verifying explanations for language models. Although an update addressing this issue is forthcoming, it was not available during our project timeline.

None of the libraries support multi-label classification tasks, and do not support one-hot encoded features (e.g., requiring one-dimensional labels).

Limitations of the Dash Framework

The Dash framework posed several challenges during implementation. These included the need for separate functions for each Output, which necessitated code restructuring. Additionally, two Python callbacks could not update the same element, and callbacks with no Inputs or Outputs were not allowed. Furthermore, we can't perform REST API calls which will be needed in future works. Therefore we need to pay attention to this issue, find a better solution, or combine it with another framework.

Despite these challenges, we successfully implemented three working use cases, addressing most of the aspects effectively. Future work will focus on improving and expanding the framework's capabilities.

5 Conclusion & Future Work

In this chapter, we discuss the contributions to the field, offer recommendations for future research, and share our final remarks.

We addressed the research questions by developing a base for a unified framework, VERIFAI, that incorporates various metrics for assessing AI responsibility. Our work contributes to the field by providing a comprehensive method for evaluating the ethical, secure, private, and explainable aspects of AI systems.

We will answer the research questions below:

RQ1: What constitues *responsible* AI?

To address this, we first provided a definition for *Responsible AI* in chapter 2 based on a structured literature review, identifying the key facets that compose it.

RQ2: What are the most appropriate metrics for assessing the aspects of Responsible AI?

To answer this research question, we leveraged insights derived from the literature review, focusing on the most critical aspects identified, namely *Fairness, Privacy, Robustness, and Explainability*. We then aimed to identify and employ metrics to effectively evaluate these aspects on a Tabular Model, a Computer Vision Model, and NLP Models in chapter 3. Therefore the metrics were Fairness, Privacy Leakage through MIA-Attacks, Robustness through Adversarial Attacks, and the evaluation of the XAI methods in terms of whether they capture the decision-making process of the model.

RQ3: To what extent are the identified metric settings applicable to various types of AI models trained on diverse datasets, such as images, text, and tabular data?

We tested the applicability of the identified metrics across different model architectures and datasets of three different types: We used Random Forest as Tabular Model, the

X-Ception architecture for the Computer Vision Model, and DistilBERT as the NLP model architecture. Although we anticipated that different metrics would be required depending on the model type and training data, our goal was to create a generally comparable evaluation procedure. Therefore, we used the same metrics for all models, but adapted them to the specific model.

RQ4: How can we assess the aspects using the metrics on different model types within an application framework?

We then intended to design an appropriate application architecture that incorporates use cases into an overarching scenario to demonstrate the practicality of the proposed framework for assessing AI responsibility along a defined pipeline. We realized the VERIFAI application using the MVC architecture and used the Python framework Dash for the implementation with the help of various toolkits to verify the metrics already presented in chapter 2.

Future Work

Sub-Aspects and Metrics We focused on discriminative models and specific sub-aspects within each area: fairness in the case of ethics, robustness for security, membership inference for privacy, and a limited set of metrics for explainability. These limitations restrict the applicability of our framework and highlight areas for improvement.

Future research should aim to address these limitations and expand the range of sub-aspects studied. For instance, researchers can incorporate additional fairness metrics, study other security aspects, such as model inversion attacks for privacy, and explore more sophisticated explainability techniques.

Regarding the metrics, more in-depth analysis could be performed, especially for each sup-aspect. We can examine whether metrics change per class or if specific classes are more affected, and closely inspect targeted data points. This approach can help gain insights and establish cross-references, similar to the Google What-if Tool, which allows sorting and examining data points individually.

Generative Models As AI systems continue to evolve, it is essential to develop metrics for measuring both discriminative models and generative models, while understanding the nature and mechanisms of generative AI systems like GPT-4 remains a formidable challenge that has become crucial and urgent (Bubeck et al. [2023]).

Data and Application scenarios Moreover, the aims for future work encompass a variety of improvements and extensions to our current framework. These include expanding the range of datasets and models evaluated by the framework. We also plan on exploring additional application scenarios beyond healthcare to demonstrate the versatility of our approach in diverse contexts.

Suggestions for Mitigation Our future work also includes providing actionable suggestions for addressing identified shortcomings, such as improving privacy protection measures within AI systems.

Comparison of Models Supporting model comparison will allow users to assess the relative performance and responsibility of various AI models.

Adaptability to the needs of users To tailor the evaluation process to specific user needs, we will allow for selectable target users and customizable aspects most important to them. Defining tolerance thresholds for each aspect will enable stakeholders to set acceptable levels of responsibility. We aim to enable users to upload or connect their models and data for evaluation, fostering a more inclusive and practical framework. Lastly, we plan to integrate AI-generated explanation texts, leveraging advancements in natural language processing to enhance the framework's explainability capabilities.

Final Remarks

In conclusion, we have demonstrated the capabilities of the VERIFAI framework in assessing the responsibility of AI systems, acknowledging the current limitations, and outlining the directions for future research.

Our comprehensive results are available as a live demo version on our project website[1] and will be updated as new enhancements become available.

The ongoing advancements in AI have the potential to transform various industries, including healthcare, and it is imperative to ensure that these systems are responsible. As AI technologies continue to evolve, the need for robust evaluation frameworks will only grow. By expanding the scope and depth of VERIFAI, we aim to make it a valuable tool for researchers, practitioners, and organizations in assessing and improving the responsibility of their AI models across different domains.

To achieve this, we encourage the research community to actively contribute to the development of new metrics, methodologies, and best practices for assessing AI systems, ensuring that they are inclusive, fair, and respectful of privacy. Furthermore, collaboration between AI experts, domain specialists, and ethicists will be essential in navigating the complex landscape of responsible AI, addressing these technologies' social and ethical implications.

Together, we can work towards creating AI systems that not only exhibit remarkable capabilities but also uphold the responsible values and principles that are important to us as a society.

[1] https://www.verifai.science

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original

Bibliography

C. Allen, M.A. Ahmad, C. Eckert, J. Hu, V. Kumar, and A. Teredesai. fairmlhealth: Tools and tutorials for fairness evaluation in healthcare machine learning. https://github.com/KenSciResearch/fairMLHealth, 2020.

David Alvarez-Melis and Tommi S. Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks, December 2018. URL http://arxiv.org/abs/1806.07538. arXiv:1806.07538 [cs, stat].

Galen Andrew, Steve Chien, and Nicolas Papernot. Tensor flow privacy. https://github.com/tensorflow/privacy, 2022.

Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. arXiv preprint arXiv:1909.03012, 2019.

Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and Aggregating Feature-based Model Explanations, May 2020. URL http://arxiv.org/abs/2005.00631. arXiv:2005.00631 [cs, stat].

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In 2010 20th International Conference on Pattern Recognition, pages 3121–3124, 2010. doi: 10.1109/ICPR.2010.764.

Boštjan Brumen, Sabrina Göllner, and Marina Tropmann-Frick. Aspects and Views on Responsible Artificial Intelligence. In Machine Learning, Optimization, and Data Science: 8th International Workshop, LOD 2022, Certosa di Pontignano, Italy, September 19–22, 2022, Revised Selected Papers, Part I, pages 384–398. Springer, 2023.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4, March 2023. URL http://arxiv.org/abs/2303.12712. arXiv:2303.12712 [cs].

Simon Caton and Christian Haas. Fairness in Machine Learning: A Survey, October 2020. URL http://arxiv.org/abs/2010.04053. arXiv:2010.04053 [cs, stat].

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Somesh Jha, and Xi Wu. Concise Explanations of Neural Networks using Adversarial Training, July 2020. URL http://arxiv.org/abs/1810.06583. arXiv:1810.06583 [cs, stat].

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec '17, page 15–26, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450352024. doi: 10.1145/3128572.3140448. URL https://doi.org/10.1145/3128572.3140448.

Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2021.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In 2018 IEEE Security and Privacy Workshops (SPW), pages 50–56. IEEE, 2018.

Sabrina Göllner, Marina Tropmann-Frick, and Boštjan Brumen. Towards a Definition of a Responsible Artificial Intelligence. In Proceedings of the 33rd International Conference on Information Modelling and Knowledge Bases EJC 2023. University of Maribor, University Press, 2023. doi: 10.18690/um.feri.5.2023.2.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In Proceedings of the 2018 International Conference on Digital Health, DH '18, page 121–125, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450364935. doi: 10.1145/3194658.3194677. URL https://doi.org/10.1145/3194658.3194677.

Sabrina Göllner and Marina Tropmann-Frick. VERIFAI - A Step Towards Evaluating the Responsibility of AI-Systems. In Birgitta König-Ries, Stefanie Scherzinger, Wolfgang Lehner, and Gottfried Vossen, editors, BTW 2023. Gesellschaft für Informatik e.V., 2023. doi: 10.18420/BTW2023-63.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Anna Hedström, Leander Weber, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. 2022.

Aleksandar Jankovic and Rudolf Mayer. An Empirical Evaluation of Adversarial Examples Defences, Combinations and Robustness Scores. In Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics, pages 86–92, Baltimore MD USA, April 2022. ACM. ISBN 978-1-4503-9230-3. doi: 10.1145/3510548.3519370. URL https://dl.acm.org/doi/10.1145/3510548.3519370.

A Janosi, W Steinbrunn, M Pfisterer, and R Detrano. Uci machine learning repository-heart disease data set. School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 1988.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 8018–8025, 2020.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. arXiv preprint arXiv:1812.05271, 2018.

Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Yunfeng Zhang, Karthikeyan Shanmugam, and Chun-Chen Tu. Leveraging latent features for local explanations, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning, jul 2021. ISSN 0360-0300. URL https://doi.org/10.1145/3457607.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 142–148, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1018. URL https://aclanthology.org/N16-1018.

Sasi Kumar Murakonda and Reza Shokri. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. arXiv preprint arXiv:2007.09339, 2020.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. Adversarial robustness toolbox v1. 0.0. arXiv preprint arXiv:1807.01069, 2018.

OpenAI. GPT-4 technical report, 2023. URL https://arxiv.org/abs/2303.08774.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. Journal of machine learning research, 12(Oct):2825–2830, 2011.

Charles Pierse. Transformers interpret. https://github.com/cdpierse/transformers-interpret, 2023.

Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. Journal of Open Source Software, 5(53):2607, 2020. doi: 10.21105/joss.02607. URL https://doi.org/10.21105/joss.02607.

Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial Attacks and Defenses in Deep Learning. Engineering, 6(3):346–360, March 2020. ISSN 20958099. doi: 10.1016/j.eng.2019.12.012. URL https://linkinghub.elsevier.com/retrieve/pii/S209580991930503X.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1103. URL https://aclanthology.org/P19-1103.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, August 2016. URL http://arxiv.org/abs/1602.04938. arXiv:1602.04938 [cs, stat].

Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. Pearson Series in Artificial Intelligence, 4 edition, 2020. ISBN 0134610997,9780134610993.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv, abs/1910.01108, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In 2017 IEEE Symposium on Security and Privacy (SP), pages 3–18, San Jose, CA, USA, May 2017. IEEE. ISBN 978-1-5090-5533-3. doi: 10.1109/SP.2017.41. URL http://ieeexplore.ieee.org/document/7958568/.

Leon Sixt, Maximilian Granz, and Tim Landgraf. When Explanations Lie: Why Many Modified BP Attributions Fail, August 2020. URL http://arxiv.org/abs/1912.09818. arXiv:1912.09818 [cs, stat].

Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P. Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual &Group Unfairness via Inequality Indices. In Proceedings of the 24th ACM SIGKDD International Conference on

Knowledge Discovery & Data Mining, pages 2239–2248, London United Kingdom, July 2018. ACM. ISBN 978-1-4503-5552-0. doi: 10.1145/3219819.3220046. URL https://dl.acm.org/doi/10.1145/3219819.3220046.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks, June 2017. URL http://arxiv.org/abs/1703.01365. arXiv:1703.01365 [cs].

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. CoRR, abs/1312.6199, 2013.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

Tensorflow. Tensorflow Integrated Gradients. https://www.tensorflow.org/tutorials/interpretability/integrated_gradients, 2023. Accessed: 2023-03-28.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data, 5(1):1–9, 2018.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced Membership Inference Attacks against Machine Learning Models, September 2022. URL http://arxiv.org/abs/2111.09679. arXiv:2111.09679 [cs, stat].

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (In)fidelity and Sensitivity for Explanations, November 2019. URL http://arxiv.org/abs/1901.09392. arXiv:1901.09392 [cs, stat].

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, Proceedings of the 30th International Conference on Machine Learning, volume 28 of Proceedings of Machine Learning Research, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/zemel13.html.