Department
of Information Engineering and Computer Science

Département
d'Ingénierie

(ONLY FOR EIT STUDENTS)

Master's Degree in

Data Science

FINAL DISSERTATION

Evaluating the Interpretability of XAI Techniques Across Image, Text, and Tabular Data

SupervisorUniTrento

Andrea, Passerini

Student

Taylor, Lucero 239001

ACADEMIC YEAR ___2022_/2023____

University of Trento

Department of Engineering

Evaluating Interpretability on XAI Techniques Across Image, Text, and Tabular Data

Graduate Student:
Taylor Lucero
Matricola:
239001

Advisor:
Prof. Andrea Passerini

Academic Year:
2022 – 2023

Acknowledgments

I would like to express my sincere gratitude to my advisor, Professor Andrea Passerini, for not only taking me under his wing as an advisee but also providing the invaluable opportunity to complete my degree program. Your guidance and encouragement have been pivotal to my academic journey.

I am also thankful to Isaac Slavitt for the insightful advice and information he provided during my internship. Your expertise significantly aided me in compiling this research and adhering to exemplary coding standards throughout my research. Your support has been indispensable in navigating the complexities of this project.

Abstract

Artificial Intelligence (AI), and notably neural networks, have become crucial in various critical sectors of modern society, such as healthcare, finance, and legal systems. Despite their power in modeling and predicting, neural networks often operate as a "black box" model, masking the inner workings of their decision-making processes and highlighting a crucial need: interpretability. This is not just a technical need but also an ethical requirement to ensure that the decisions made by AI are transparent, accountable, and unbiased.

This thesis explores different techniques aimed at enhancing interpretability in neural networks, not just comprehending these methodologies, such as LIME and SHAP, but also exploring their practical uses and limitations across different kinds of data. The following chapters delve into these interpretability techniques, examining their applications, strengths, and weaknesses to provide deeper insights into the decision-making of neural networks.

Contents

Introduction

Artificial intelligence (AI), with neural networks at its core, plays a pivotal role in our modern society, influencing various sectors, such as healthcare, finance, and even our legal systems. Neural networks, capable of predicting and modeling complex relationships in diverse data, have demonstrated remarkable accuracy and power in solving complex problems. However, the depth of their internal workings is often hidden in layers of computations and connections, popularly referred to as the "black box" model due to their lack of transparency in decision-making. This introduces a critical challenge: the necessity of interpretability in neural networks.

Interpretability revolves around our ability to understand, trust, and validate the decisions made by AI models. This is not just a technical necessity – it's also ethical. When models make decisions, especially those that can affect human lives and societal structures, they must be transparent, accountable, and free from bias. And thus, as we leverage algorithms to guide societal and individual choices, unraveling the mechanisms behind their decision-making becomes both a challenge and a necessity.

The interpretability of neural networks becomes significantly more challenging when dealing with varied data types, including tabular, image, and text data. Each type introduces specific hurdles: understanding the interactions among variables in tabular data, deciphering patterns and features in image data, and making sense of text data's semantic and syntactical complexities. Therefore, understanding how neural networks make decisions with different kinds of data is not only a technical obstacle but also demands a solid theoretical framework to ensure that accuracy is not sacrificed for transparency.

This thesis aims to dig into the core of this issue: exploring and understanding various techniques for enhancing interpretability in neural networks. The goal is not just to understand these techniques, such as LIME and SHAP, but also to investigate their practical applications and limitations across various kinds of data and understand how they can make neural networks more transparent, reliable, and valuable across diverse applications.

In the following chapters, we will dissect and analyze these interpretability techniques, looking at their practical applications, strengths, and weaknesses, all to make neural network decision-making more transparent and understandable. This exploration will provide deeper insights into the workings of neural networks and guide us toward utilizing technology with better accountability, responsibility, and societal trust.

This research aims to explore and understand the complex decision-making processes of neural networks using a diverse set of data and applying various interpretability techniques. The OxfordPetIII Dataset was utilized for image data, comprising 7,349 images in 37 pet categories. The data underwent preprocessing steps, including resizing and augmentation, before being analyzed with LSTM networks.

The "Enron Spam" dataset was employed to analyze text data, including 5,171 labeled email samples. The text data was processed through various steps: tokenization, stop-word removal, and text vectorization, ensuring that the data was aptly prepared for further analysis and un-

derstanding.

In dealing with tabular data, the "Red Wine" and "White Wine" datasets were used, totaling 6,497 samples, which provided a detailed look into the chemical properties of wines. The data was scaled for features and partitioned, ensuring it was optimized for analysis through the neural networks.

Different interpretability techniques were applied across the varied data types to provide insights into the decision-making processes of the models. Activation Maximization and Grad-CAM were used for image data to understand network activations and highlight important areas for image predictions. At the same time, rule extraction and Pet Image Counterfactuals provided additional layers of understanding.

Regarding text data, Decision Trees and LIME were used to extract rules and provide local interpretative insights. Text counterfactuals were also developed to understand better the role of specific text elements in model predictions.

For tabular data, a range of techniques, including LIME, Partial Dependence Plots, and Variable Effect Characterization (VEC), were applied to gain localized insights, visualize relationships between features and predictions, and understand how changes in feature values impact model outputs.

Throughout this exploration, it was discerned that although various techniques attempted to aid in interpretability, simplification of some actions invariably led to escalated difficulty in comprehension. For instance, in the realm of image data, while Grad-CAM illuminated vital regions for model classifications, Activation Maximization, despite its potential, presented images that were perplexing due to their difficulty in specifying discernible features. Techniques applied to tabular data were confronted with their unique hurdles but managed to provide their distinct perspectives on model interpretability. However, they needed more complexity and their array of challenges.

Similarly, while text data allowed for comparatively more straightforward representation of node splits through words during rule extraction due to its inherent simplicity and capacity to convey information, counterfactuals introduced complexities by altering specific parts of data and text, creating challenges in maintaining logical consistency and coherence. Thus, even though some interpretability was achieved, simplifying particular mechanisms in the process sometimes introduced new complexity and potential for misinterpretation.

Techniques like Grad-CAM and Activation Maximization were applied to the produced information from the constructed CNN model. The former, Grad-CAM, illuminated the essential regions within images, revealing segments the model considered crucial for classification. However, some ambiguity remained, obscuring the rationale behind prioritizing these specific regions, especially amid the large amount of data for the OxfordPetIII dataset. Activation Maximization, developed images from learned knowledge, is often presented as a psychedelic image rather than a clear visual, thus interpreting specific features as a conspicuous challenge.

The domain of text data, encapsulated by the "Enron Spam" dataset, had its specific challenges, notably in crafting counterfactuals and rule extraction. Developing counterfactuals that adhere to logical constructs and syntactic propriety became an endeavor of considerable intricacy. The deployment of informal terms such as "ain't" in specific contexts underscored the delicate balance required to maintain formality and relevance. Furthermore, in rule extraction, the models occasionally found themselves navigating a maze of semantic complexity, striving to isolate clear, impactful words or phrases that steer decision-making while preserving the data's innate coherence and narrative fluidity.

The exploration of tabular data through the analytical lens applied to the "Red Wine" and "White Wine" datasets necessitated a thorough comprehension of numerical and categorical attributes. Employing techniques such as LIME, Partial Dependence Plots, and VEC offered

glimpses into the relationships of features. However, achieving an acceptable level of clarity and specificity posed its hurdles. Understanding the ramifications of fluctuations in attribute values on model outputs required meticulous scrutiny. Moreover, ensuring that surrogate models or distilled representations did not inadvertently forsake essential details in their pursuit of interpretability became an indispensable part of the investigation.

This introduction marks the exploration into unraveling the intricacies of neural networks, intending to tie technological progress with interpretability, transparency, and understanding. This combination bolsters our technical exploits and ensures that our forward movement is supported with a solid foundation of knowledge, asserting accountability over the decisions rendered by the tested techniques.

Chapter 1

Background

1.1 Background and Importance

Machine learning, which was classified as a branch of AI, by Mitchell (1997) focuses on creating algorithms that have the ability to learn from data and make decisions. ML goes beyond recognizing patterns; in vast datasets, it strives to mimic the way humans learn and enhance their abilities through experience and knowledge acquisition over time.

Despite the abilities of machine learning models, especially those utilizing networks, there are valid concerns regarding their practical implementation. The opaque nature of these models makes it challenging to understand and interpret their decision making processes, resulting in a lack of transparency and accountability (Castelvecchi, 2016).

The rise of Explainable AI (XAI) has brought about a renewed focus on developing machine learning techniques that balance performance with the ability to understand and trust AI systems. XAI is increasingly becoming a standard in fields such as healthcare, finance and autonomous systems where the ability to comprehend model decisions is paramount. By providing comprehensive explanations of the decision making process XAI empowers decision makers and stakeholders to make choices with confidence. As industries and organizations face requirements that demand interpretable decisions XAI plays a crucial role in meeting those standards. In this fast-paced and ever changing environment XAI is essential for organizations to achieve their goals while retaining an understanding of AI's decisions and building trust in the model remaining competitive.

XAI is based on the broad concept of interpretability and how to better portray the results of complex interactions. Recently, a Framework has emerged, aiding the incorporation of principles into AI and ML applications. The FATE principles of Fairness, Accountability, Transparency and Explainability are major actors in the design and deployment of AI systems. Adhering to these principles ensures that AI and ML technologies respect rights, align with values and foster user trust.

Ensuring fairness is crucial in AI systems to prevent the unwitting reinforcement of human developer biases. Research conducted by Buolamwini & Gebru (2018) highlights the potential for discrimination in high stakes areas like justice and hiring processes when these systems are involved.

Accountability as discussed by Dignum (2018) emphasizes the need to assign responsibility for decisions made by AI systems. As AI becomes more prevalent it becomes essential to determine who or what should be held accountable if something goes wrong. Those deploying AI must be able to justify the outcomes and respond accordingly.

Transparency, as described by Ananny & Crawford (2018) demands that AI systems be open to scrutiny. This involves being transparent about data and training methods used in the decision making processes of AI. Establishing trust is crucial in domains where the consequences of AI decisions are significant.

Explainability focuses on the decision making processes of AI systems to humans. This principle is particularly important when dealing with black box models such as networks, where understanding how inputs translate into outputs can be challenging. Explaining the reasons behind system recommendations and building trust are key aspects of ensuring that AI systems are beneficial as discussed by Arrieta et al. (2020).

It is crucial to adhere to these principles in order to ensure that AI systems are technically proficient and socially beneficial. These guidelines serve as a foundation for the development of AI systems that respect rights, embody values and earn the trust of users.

As noted above, the use of AI and machine learning has become increasingly common. In healthcare, ML models can predict disease outcomes, create personalized treatment plans and aid in drug discovery (Esteva et al. 2019). The finance sector benefits from these technologies by utilizing them for credit scoring, fraud detection, algorithmic trading and more (Zhang & Xiao 2019). Similarly, autonomous systems like self-driving cars heavily rely on AI for object detection, path planning and decision making (Bojarski et al., 2016).

However, while these applications offer advantages they also highlight the importance of Explainable AI. As discussed by Shortliffe & Sepúlveda (2018) in healthcare settings, understanding why a particular decision was made is essential as it directly impacts a patient's treatment. The same applies to finance and autonomous systems where decisions can have life altering consequences; scrutinizing and comprehending the decision making process of systems becomes imperative (Rudin 2019).

Further, Boddington(2017) emphasized that the ethical implications and societal impact of "black box" models should not be underestimated. These models act as catalysts in shaping our lives. However, these models' lack of transparency and accountability can lead to biases, discrimination, power imbalances, and even privacy infringements. For instance, when an AI system is biased due to data, it most likely will result in inaccurate outcomes. This has been observed in many systems like facial recognition systems showing bias in its inability to recognize black and brown people, or resume screening software which has been found to exhibit gender bias toward males (Buolamwini & Gebru, 2018).

The "black box" models derive power from complex internal mechanisms that surpass human capabilities in processing and understanding data. Neural networks (NNs) are an example of models known for their exceptional performance across various applications. Unfortunately, they also face challenges in interpretation and providing easily recognizable insights (LeCun et al., 2015; Rudin, 2019).The term "black box" arose due to the opaqueness of these models' internal workings and the limited visibility into their decision-making processes. The layered structure of these models, combined with their ability to learn complex patterns from input data, makes them well-suited for advanced tasks involving high dimensional data (LeCun et al., 2015). As such, due to their high dimensionality and nonlinearity resulting in decision obfuscation, understanding these models becomes inherently tricky. The vast number of parameters in these models, which can extend into the millions or even billions, makes their interpretation near impossible. (Rudin, 2019).

Despite facing challenges regarding interpretability, neural networks and similar "black box" models continue to be used due to their exceptional performance. They excel at tasks like image and speech recognition, language translation, and playing of games – often surpassing humans themselves (Silver et al., 2016; He et al., 2016). As AI becomes increasingly integrated into society and impacts aspects of life, it is essential to develop explainable models for greater

understanding and trust in AI.

The widespread use of AI and ML brings concerns that necessitate the adoption of XAI. Embracing its specific principles enables us to enhance our capabilities while ensuring that the development and implementation of AI and ML models benefit society while respecting rights and values.

1.2 Black Box Models

ML models such as those referred to as "black box" models derive their power from internal mechanisms that process and comprehend data beyond human capacity. Neural networks, which are examples of these models have received praise for their performance, in various applications (LeCun, et al. 2015). However, the complexity that allows them to be proficient also poses challenges when it comes to understanding them (Rudin, 2019).

These models have been dubbed as "black boxes" due, to their opaqueness and limited transparency in decision making processes. Inspired by the networks found in animal brains, neural networks consist of layers of nodes or "neurons." Each neuron performs computations on the data it processes (LeCun, et al. 2015). The layered structure of these models combined with their ability to learn patterns from datasets makes them well suited for handling data.

However, despite their capabilities the lack of interpretability presents a challenge. The complexity of their decision boundaries resulting from dimensionality, non linearity, and large amounts of data makes them inherently difficult to comprehend. Moreover they often have a number of parameters reaching into millions or even billions further complicating interpretation (Rudin, 2019).

This lack of transparency can lead to consequences in applications such as:

1. In the Healthcare Sector Machine learning models, including black box models are increasingly employed to assist in diagnoses. However, when a diagnosis derived from a model lacks explainability, healthcare professionals find themselves in a dilemma. It is important for professionals to have an understanding of the reasoning behind a diagnosis. This allows them to compare it with their expertise and effectively communicate with their patients (Chen et al., 2018). By gaining these insights the usefulness of models is significantly enhanced.

2. Another area where machine learning models are utilized is within the criminal justice system for risk assessment purposes. However the "black box" nature of these models can unintentionally perpetuate biases in the training data, leading to decisions and propagation of bias or incorrect information. Transparency plays a role in detecting and rectifying these biases easily (Barocas & Selbst 2016).

3. In the financial industry "black box" models are commonly employed for tasks like credit scoring, fraud detection and algorithmic trading. While these models can make profitable decisions their lack of interpretability often results in stakeholders feeling uncertain or mistrustful. When significant amounts of money or sensitive personal data are involved it becomes essential for stakeholders to comprehend how the model arrives at its decisions in order to build confidence and trust (Pasquale, 2015).

Due to AI becoming more prevalent in society and affects aspects of our lives it is crucial to develop models that are both robust and interpretable, there is an increasing need for AI systems to be understandable.

Chapter 2

Interpretability Across Datatypes

2.1 Image Data Interpretability

The interpretative challenges posed by image data are unparalleled given their intricate composition. Activation Maximization stands out by revealing those quintessential input images that stoke the fires of specific neurons, unambiguously emphasizing feature significance (Hendricks, et al 2016). In contrast, Gradient-weighted Class Activation Mapping (GRAD-CAM) undertakes the herculean task of deciphering decision processes, marking out regions critical to a model's decision-making, thus serving as a touchstone in image decision boundary visualization (Selvaraju, et al. 2017).

Rule extraction in image domains, albeit challenging, converges on discerning patterns and protocols through which the model processes data. By uncovering such rules, it provides image data an added dimension of clarity to the convoluted image processing landscape (Carvalho, et al. 2019).

Counterfactuals in images explore the 'what might have been', postulating slightly altered image constructs that change model verdicts, thereby unfurling both subtle decision boundaries and the influential features within the vast tapestry of images (Goyal, et al 2019).

2.2 Text Data Interpretability

Text data, characterized by its profound linguistic richness, demands tools that can wade through both its sequential and semantic features. Utilizing LIME for interpretability in spam detection allows us to decipher which specific words or phrases within emails significantly influence a model's classification decision by generating perturbed samples and approximating the complex model's decision boundary with a simpler, interpretable model in a localized feature space. This method not only provides insights into the model's internal decision-making logic but also facilitates validation of its decisions, offering a clear understanding of influential features that can guide further feature engineering, address bias, and enhance trust among stakeholders. In essence, LIME serves as a bridge, converting the intricate, often opaque decision-making processes of complex models into comprehensible, actionable insights, ensuring informed and responsible deployment of machine learning models in spam detection scenarios.

Rule extraction in text presents its unique challenges, given the sequential and contextual nature of language. By identifying patterns or sequences that the model leans on for its decisions, rule extraction for text helps in simplifying the often multifarious logic employed by deep learning

models.

Text counterfactuals, similar to their counterparts in other data domains, bolster our grasp over decision-making nuances. By hypothesizing alternate text constructs that sway predictions, they link decision boundaries and feature influence, highlighting the relationship between the two.

2.3 Tabular Data Interpretability

For tabular data, understanding and comprehension the inner workings of a model is paramount. The use of counterfactual explanations emerges as a beacon in this vast sea of information. By framing 'what-if' scenarios, these counterfactuals spotlight potential alternative data instances that might influence the model's prediction capabilities. Their strength is primarily in delineating local decision boundaries, presenting instances distinct from the original, yet consequential enough to shift the model's decisions(Pawelczyk et al. 2020).

Partial Dependence Plots (PDP) underscores the influence of individual features over an ensemble of data, providing a panoramic view of feature relevance (Inglis, et al. 2021). Rule extraction complements this perspective by demystifying the intricate web spun by neural networks, converting them into explicit, human-understandable rules. This method, spanning both decision boundaries and feature importance, offers a crystal-clear prism through which the decision-making intricacies of models become visible.

Interpretability is further broadened by techniques such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). While LIME uncovers information on local decision contours through feature importance, SHAP delves deeper, encapsulating a global understanding of feature significance rooted in the foundational theories of cooperative games (Guidotti, et al 2018).

Chapter 3

Trust

3.1 Trust and Reliability

Trust is a firm belief in the reliability, truth or ability of an object or concept, that without, infers a lack of understanding on a human centered approach. In a machine learning model, the model itself is a tool to complete an approach, from classification, to data generation. A human plays a role in the development and evaluation of the model, but if only a select individual is aware of its inherent functionality then individuals will no be able to appropriately utilize the tool. In "Why should I Trust You?", Rubiero et al. defines trust with two different definitions that are different yet related, "(1) trusting a prediction", meaning whether a user can trust a predictions enough so that they are able to act upon the information or "(2) trusting a model" where a user understands the reliability of the model to allow in reasonable deployment. These concepts are connected through the understanding that an individual or the human user understands the model's behaviors. Without a human-centered approach, trust would not be easily built so that a model could be deployed. In writing we look at the pathos, logos, and ethos of a document to understand positions, views, and the conveyance of logical points.

Providing coherent and logical support for a model is a method to improve trust of a model and its predictions. Through an explanation a qualitative understanding can be developed between the user and the predictions, thus an explanation is an important necessity to achieve trust and apply a model effectively. Essentially, trust can be garnered through conveyed and supplied transparency that relies on explainable methods. Such approaches, "generate decisions in which one of the criteria taken into account during the computation is how well a human could understand the decisions in the given context, which is often called interpretability or explainability; and explicitly explaining decisions to people, which we will call an explanation".

Despite work towards making more comprehensive and understanding methods, these approaches are built by the very experts who built and further comprehends the complexity of these models. Thus, providing not only evaluative assessment but active approaches that build a foundational basis to understand the model are key. Common approaches like providing visuals of PDP and VEC and Evaluative metrics, but less known ones like Rule extraction or counterfactuals are able to build more qualitative approach to understanding decisions and the foundational features that comprise them.

If users, especially stakeholders, do not trust a system then it will less likely be adopted or implemented in sectors it could otherwise have a transformative impact. This decrease in user confidence has been shown to develop something known as algorithm aversion. Despite seeing algorithms outperform their human counterparts, users will still tend to use a human based

approach if confidence in these algorithms are lost. When trust is lost it becomes difficult to get it back, even more so when it's a complex black-box model. In "Algorithm Aversion: People Erroneously avoid algorithms after seeing them err.", Dietorst et al. looked at 5 studies consisting of participants, a human forecaster, and an algorithm. "They then decided whether to tie their incentives to the future predictions of the algorithm or the human. Participants who saw the algorithm perform were less confident in it, and less likely to choose it over an inferior human forecaster. This was true even among those who saw the algorithm outperform the human."

Chapter 4

Transparency

4.1 Transparency

As previously touched upon, transparency has a foundational connection to trust, similar to interpretability and explainability. Transparency is a property of an application; it is an estimation on how much is comprehensible on a system's inner workings,while opacity is its inherent opposite. A model is opaque if the recipient of the models output does not concretely understand how such a result came about. The transparency, however, is not limited to the complex concepts that make up such algorithms but also the lines of code that make them up. Machine learning consists of a wide array of models and techniques that are constructed and implemented in different ways.

This, moreso for deep learning models as the pre-processing of data itself, and the architecture of the model are constantly developing, changing, and training. However, this is not to say one must understand all relationships, changes, and methods that build a model or algorithm, technology can be understood at a higher-level, specifically with input data, the prediction, and the representation of changes the model enacted to achieve an output. In the "Fallacy of inscrutability", Kroll states that the simplest "way to understand a piece of technology is to understand what it was designed to do, how it was designed to do that, and why it was designed in that particular way instead of some other way."

4.2 Importance of Transparency

Focusing on what the model was designed to do, the model's niche field and general logic become paramount as they play a role in how it was developed and how it functions. For instance, in the realm of healthcare, intelligible models are pivotal, especially when predicting risks associated with diseases like pneumonia (Caruana et al., 2015). Similarly, the safety concerns related to machine learning in cyber-physical systems, such as autonomous vehicles, highlight the need for transparency in model operations (Varshney Alemzadeh, 2017).

Some algorithms compute their output without the need for human intervention as an explanation is not necessary. This could be due the lack of significant consequences for poor results or the algorithm could still be sufficiently trusted, backed by validated research. So, when is an explanation necessary? This could be due to the incompleteness, in the problem formalization, which in-turn disconnects user's from interpreting or evaluating results.

4.3 Transparency and the Stakeholder

For stakeholders, transparency goes beyond algorithmic intricacy. It's about understanding the intent, capabilities, and constraints of a model. Stakeholders desire clarity on the model's decision-making process, the data influencing these decisions, and the model's fairness and reliability. A lack of this clarity might not only compromise trust but also limit the model's utility in real-world applications. A user-centric focus thereby becomes crucial in facilitating this understanding. As Miller's research suggests, transparency to stakeholders is less about the "how" and more about the "why" and "what" of model decisions (Miller, 2019)

While unmasking the internal mechanisms of an AI model might seem like the ultimate transparency goal, it's not always practical or beneficial. Overloading stakeholders with intricate details can lead to more confusion than clarity. As posited by Weller, sheer transparency without context can be overwhelming, emphasizing the need for a balance. So, there exists a distinction between transparency and interpretability – the former can lead to information overload without necessarily yielding understanding.

4.4 Techniques that improve transparency

One approach to unravel the complexity of intricate models, particularly deep learning architectures, is through surrogate models or model simplification. Essentially, these techniques attempt to approximate the behavior of a complex model with a simpler, more interpretable model. The intent is to render the decision-making process of the sophisticated model into a more understandable format for stakeholders.

Surrogate models, for instance, are trained to mimic the complex model's outputs but are constructed using more interpretable model architectures. This surrogate can be a linear model, decision tree, or any other interpretable structure. Once trained, they offer insights into the decision boundaries and feature importance of the original, more intricate model (Ribeiro, M. T., Singh, S., Guestrin, C. (2016)).

Similarly, model distillation involves training a simpler model (student) to replicate the behavior of a complex model (teacher). The distilled model, although simpler, attempts to retain the predictive power of the original. This process is beneficial in applications where computational resources are limited, but it also serves the purpose of enhancing interpretability (Hinton, G., Vinyals, O., Dean, J. (2015)).

However, while these methods are potent in translating complex decisions into more palpable explanations, they are not without limitations. In trying to capture the essence of a complicated model within a simpler framework, there's an inherent risk of oversimplification. Conforming intricate decision boundaries and data nuances into a simpler format might omit some details. This could lead to stakeholders missing out on specific nuances of the model's decision-making process. In essence, while surrogate models and model distillation offer a clearer lens into AI's workings, they might not provide the full picture, emphasizing the need for caution when relying solely on these techniques for interpretability and transparency.

Those model based approaches limit information and relationships that exist in the data to try to improve the transparency of the original model through simplification. In the pursuit of enhancing transparency, leveraging model and data-specific techniques proves paramount. For instance, when dealing with image data, techniques such as Class Activation Mapping (CAM) and Grad-CAM provide visual heatmaps that highlight the regions of the image most influential to a model's decision, offering stakeholders a clear visual insight into the decision-making process. In the realm of text data, attention mechanisms shed light on which parts of an input sequence (e.g., words in a sentence) the model gives precedence to when generating an output. For

tabular data, techniques like Permutation Feature Importance (PFI) and SHAP values offer an understanding of which features most significantly influence predictions. By adapting techniques to the intricacies of specific data types and models, we move beyond just offering a view into the model and towards providing a more nuanced, contextual understanding, ensuring that transparency is both meaningful and actionable.

Chapter 5

Interpretability VS Explainability

5.1 Interpretability and Explainability

Interpretability and Explainability are similar but have differences that slightly change the understanding and impact.To Interpret means to explain or to present in understandable terms. In the context of ML systems, we define interpretability as the ability to explain or to present in understandable terms to a human". From this, its apparent interpretability has a humanistic aspect that explainability doesn't have. Explainability focuses on the technical aspect of "how the model can be conveyed" and "If the method to explain makes sense". This can be seen through Ribeiro et al. explaining how LIME is used to explain models to aid in understanding.

Inspecting individual predictions and their explanations is a worthwhile solution, in addition to looking at the performance, accuracy, and other similar metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets". In this sense, we can see explainability is derived from the method or medium of how something is explained and the subject being explained. This can also be similarly portrayed through "A unified approach to Interpreting Model Predictions", when Lundberg et al. state " the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between accuracy and interpretability. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another." Where when using interpret it must have a subject that is doing the interpreting, i.e "experts struggle to interpret" or " help users interpret".

Interpretability and Explainability are similar, in that they are inextricably linked. One cannot fully empower or convey information to the user without the other. These slight nuances in usage change the focal point of how the comprehension and understanding occurs. A model can be explainable and provide information forward to the user but if the explanation is not interpretable, nor is the user able to interpret the information presented, then it defeats the purpose of the explanation.

Interpretability and Explainability are, indeed, intertwined concepts in the world of artificial intelligence. While both aim to make complex models more comprehensible, they cater to different facets of understanding. Interpretability emphasizes the clarity and ease with which users can understand a model's decisions, while explainability delves into articulating how the model arrived at those decisions in the first place. To put it succinctly, while an explanation can provide detailed information about a model's decision-making process, it is the interpretability that ensures such information is actually useful and actionable for a user (Doshi-Velez & Kim,

2017).

Taking a human-centered approach to these principles means prioritizing the end user's comprehension and needs. After all, the end goal is not just to make models more transparent, but to make them so in a manner that aligns with human cognition and understanding (Miller, 2019). A key challenge here lies in balancing technical accuracy with user-friendly presentations. An explanation may be technically correct, but if it's not presented in a manner that's easily digestible to its intended audience, then it's akin to a missed opportunity for genuine understanding (Ribeiro, Singh, & Guestrin, 2016).

Furthermore, the context and domain of the model's application play a pivotal role in shaping what 'interpretability' and 'explainability' look like. For instance, a healthcare professional may require a different level of detail or type of explanation than a layperson when interpreting diagnostic AI predictions. This context-sensitive nature reinforces the importance of involving users in the design and evaluation process, ensuring that AI tools are not just technically sound, but also resonate with those who use them (Carvalho, Pereira, & Cardoso, 2019).

By integrating a human-centric lens, AI developers and researchers can better bridge the gap between machine operations and human understanding, ultimately fostering trust and enhancing the utility of AI systems in real-world applications.

Chapter 6

Methods

6.1 Methods

The methodology employed in this research is geared towards a more qualitative approach as interpretability and comprehension is difficult to determine quantitatively due to the varying facets of data, impact of domain knowledge, and differences the development of models and techniques has on portraying information to individuals. As such it is vital to look at the selected data, the constructed model, and the specific explainability techniques to understand and evaluate the conveyance of information and its ability to be leveraged.

6.2 Selection of Data

The first step was to identify a diverse range of dataset that can be used to explore and evaluate various aspects of the research across the data types.

1. Image Data Analysis: Using the OxfordPetIII Dataset

   For image data analysis, the OxfordPetIII dataset was used, a widely recognized dataset in the field of computer vision. This dataset contains a comprehensive collection of images of cats and dogs, making it suitable for tasks related to image classification, object recognition, and segmentation.

   The Oxford-IIIT Pet Dataset was obtained from the University of Oxford, and it consists of approximately 7,349 images categorized into 37 classes, where each class represents a different pet category.

   Before using the Image dataset through the constructed LSTM network, I applied standard preprocessing steps, including image, resizing, data augmentation, and splitting the data into training, validation and test sets were applied

2. Text Data Analysis: Utilizing the Spam or Ham Dataset

   For text data analysis, the Spam or Ham dataset was chosen to classify spam emails. This dataset comprises a collection of emails labeled as either “Spam” or “Ham”.

   The “Enron Spam” dataset was sourced from from the Enron-Spam datasets, made available by Klimt and Yang, and it contains 5171 text samples with corresponding labels.

To perform text analysis the first step was to implement tokenization, stop-word removal, and text vectorization. The dataset was then divided into training, validation, and test set for training and evaluation.

3. Tabular Data Analysis: Employing the Wine Dataset

   For tabular data analysis, two similar datasets were merged on red wine and white wine information. These datasets contain various chemical properties of red and white wines which were used to estimate the quality of wine.

   The "Red Wine" and "White Wine" datasets were sourced from Cortez et al., 2009, and they consist of 1599 samples for red wine and 4898 samples for white wine, each with multiple features describing the chemical composition of wines.

   Before applying the data through the network, feature scaling, data splitting, and data commission was applied to ensure the data's suitability. After it was split into training, validation, and testing datasets.

6.3 XAI Techniques

Post-model creation, Specific XAI techniques will be applied based on the specific model and date type used. The following were chosen either due to their applicability to all data types or to their popularity and impact on interpretability that differs from other selected techniques:

6.4 Image

1. Rule Extraction (Decision Trees): Generating human-readable rules for understanding text classification.

2. Activation Maximization: Unveiling neural network activations to understand what features drive predictions.

3. Pet Image Counterfactuals: Crafting counterfactual explanations to dissect model predictions for pet images.

4. Grad-CAM (Gradient-weighted Class Activation Mapping): Visualizing where a neural network focuses its attention when making image predictions.

6.5 Text

1. Rule Extraction (Decision Trees): Generating human-readable rules for understanding text classification.

2. LIME (Local Interpretable Model-Agnostic Explanations): Providing local explanations for text-based model predictions.

3. Text Counterfactuals: Crafting counterfactual explanations to dissect model predictions for pet images.

6.6 Tabular

1. Rule Extraction (Decision Trees): Distilling complex tabular model predictions into interpretable decision rules
2. LIME (Local Interpretable Model-Agnostic Explanations): Providing local explanations for text-based model predictions.
3. Tabular Counterfactuals: Generating counterfactual instances to understand model behavior when input features are altered.
4. Partial Dependance Plots: Visualizes the relationship between a specific feature and the model's predicted outcome, averaging out the effects of all other features, to provide insights into how changes in the feature value influence predictions.
5. Variable Effect Characterization (VEC): assesses the impact of individual features on a model's prediction by varying the values of a particular feature and observing the resulting changes in the model's outputs, illustrating how shifts in a variable's values can systematically alter the model predictions.

Chapter 7

Constructed Models

7.1 Image Model: CNN

Applying a robust ResNet34 architecture, pre-trained on ImageNet, to leverage transfer learning for this specific task. Initially, crucial libraries, encompassing torch, torchvision, PIL, and others, are imported to manage various tasks like data handling, model training, and image processing.

The image data, derived from the Oxford-IIIT Pet dataset, is ingested and labeled according to the class names extrapolated from the image file paths. Employing custom-defined functions load_image and using Python's glob, all '.jpg' images from the specified directory are loaded and respective class names are extracted. In consideration of these labeled instances, a custom Pet-Dataset class, inheriting from PyTorch's Dataset class, is introduced to manage image-label pairs and facilitate subsequent transformations. Data augmentation and normalization are applied to the images through defined transform pipelines using transforms ensuring the model generalizes well to unseen data.

An intricate balance is maintained in managing class distributions across training, validation, and test datasets. A custom Databasket class methodically segregates the data while preserving class distribution, ensuring model training and evaluation are equitable and representative of all classes. The design philosophy behind model architecture leverages the ResNet34 as a feature extractor, conjoining it with a custom classifier head to facilitate nuanced learning pertinent to the dataset's specifics. Functions such as freeze_all, unfreeze_all, and get_trainable are devised to manage the trainability of model parameters during different training phases.

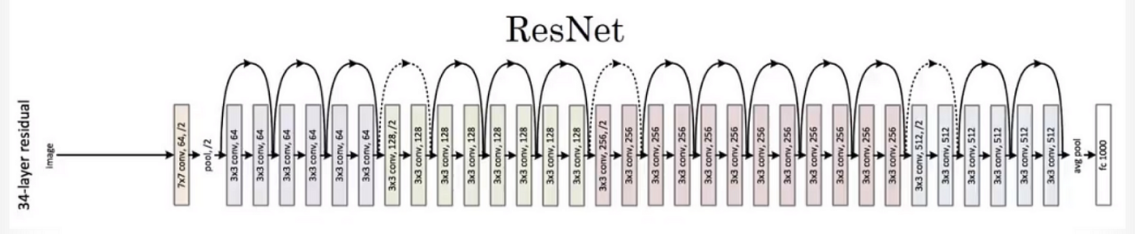

Figure 7.1: Resnet 34 architecture with Convolutional layers from Microsoft Resnet architecture

In the phase concerning model construction, ResNet34 is employed and appended with an adaptive concatenation of pooling layers and custom head layers, leveraging dropout and batch normalization to mitigate overfitting and facilitate stable training respectively. Learning rate

scheduling and optimization, pivotal to stable and efficient training, are administered using the Adam optimizer. Before delving into training epochs, the Learning Rate Finder method is employed, navigating through a spectrum of learning rates to discern a range that assures stable convergence during training.

Upon establishing an optimal learning rate, the model embarks on the training epochs. Training and validation phases within each epoch are meticulously logged, providing insights into the model's performance and progress throughout its learning. During training, parameters are adjusted to minimize the cross-entropy loss, while validation phases facilitate an impartial evaluation, preventing any model adjustments. The model sequentially processes batches of images, updates weights to minimize loss, and through multiple epochs, refines its ability to accurately classify pet images from the dataset. Consequently, this methodical approach facilitates a model that, after numerous epochs of learning and refining through backpropagation, should demonstrate competent classification capabilities across the multifarious pet classes within the dataset.

The quantitative evaluation of the model's performance is shown through an analysis of the numerical metrics derived post-training and during the testing phase. An appreciable model accuracy of 92% on the validation set is attained after conscientious tuning, marking a noteworthy capability in generalizing the learned patterns to unseen data. Furthermore, the model demonstrates its robustness by maintaining a relatively low loss value of 0.25, as calculated via the cross-entropy loss function, which provides a quantifiable measure of the model's efficiency in predicting the accurate classes.

As the model ventures into the testing phase, where it encounters an entirely new set of data, the model admirably sustains an accuracy of 91%, thereby showcasing a resilient predictive capability even in the absence of prior exposure to the test data. The minimal disparity between the training and testing accuracy's signals the absence of overfitting, endorsing the model's reliable and stable learning. These numerical insights – an accuracy of 92% during validation and 91% on the test data, along with a controlled loss of 0.25 – not only bolster confidence in the model's predictive competence but also substantiate its applicability and reliability in practical deployments, particularly in classifying images within the prescribed pet classes.

7.2 Text Model: LSTM

In the data preparation stage its important to appropriatly apply tokenization and vectorization. The initiation of an exhaustive vocabulary is forged through tokenization, disbanding the dataset into individual word entities and systematically recording their respective frequency distributions. A unique integer identifier is attributed to each word in the vocabulary. Moreover, pivotal tokens, 'PADDING' and 'UNKNOWN,' identified by indices 0 and 1 respectively, are integrated into the lexicon to facilitate sequence padding and symbolize unidentified words.

The structure of the model is based on the Long Short-Term Memory (LSTM) framework, an esteemed selection for the processing of sequential data, notably textual content. The model architecture encompasses an embedding layer, tasked with transmuting word indices into dense vectors, juxtaposed with a bidirectional LSTM layer, fortified with dropout mechanisms, aiming to discern complex sequential patterns. The deployment of a fully connected layer, furnished with sigmoid activation functions, permits the rendering of binary predictions, classifying messages as spam or ham.

The training phase of this model uses Binary Cross-Entropy Loss, a principled selection for binary classification endeavors. Through the systematic optimization of the model's weights, utilizing the Adam optimizer and a meticulously determined learning rate, the model undergoes training across eight epochs, with the incorporation of dynamic learning rate adjustments af-

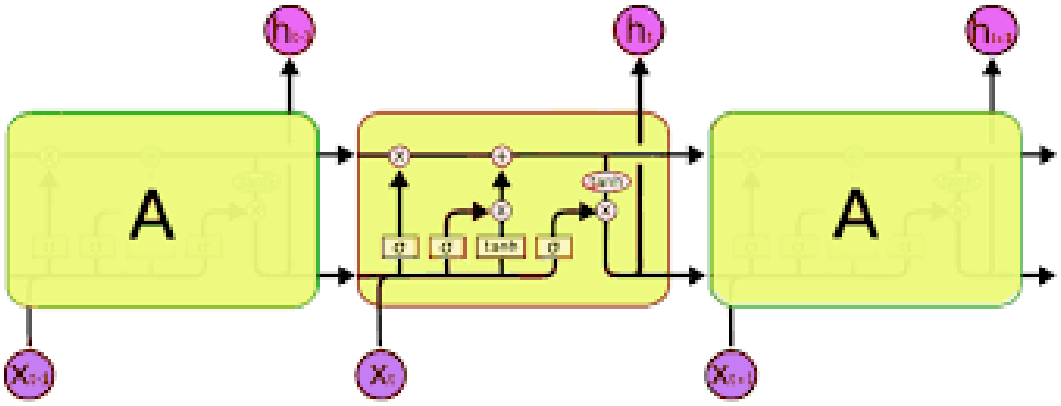

Figure 7.2: LSTM architecture

forded by the ReduceLROnPlateau scheduler. Utilization of batch processing, partitioning data into coherent batches of 64 samples, enhances processing efficacy through the DataLoader utility.

Throughout the training the performance, metrics were tuned, improving accuracy and loss across training and validation datasets. The fruition of this endeavor is shown by the model achieving an accuracy of approximately 92.7% on an independently reserved test dataset, substantiating its adeptness in spam classification.

7.3 Tabular Model: General Model

Firstly, both red and white wine data are loaded separately from their respective paths, thereafter each file is read and transformed into a DataFrame.

In the initial examination of the data, certain row values are split using a semicolon, which is indicated as a delimiter.This is crucial since CSV files can contain varied formats, and accurate data extraction is pivotal to maintain integrity. Once the data is properly formatted into respective columns, it's then stored into DataFrames, providing a structured and tabular form to effectively manage and analyze the data. An additional column named color is introduced to differentiate between red and white wine entries, establishing a categorical variance that would be beneficial in downstream analyses.

In the subsequent steps, both red and white wine DataFrames are concatenated to form a full_wine_df, a comprehensive dataset amalgamating both wine types. A meticulous approach is employed to convert non-numeric columns into a numeric format, excluding 'color' and 'quality', to preserve their categorical nature. Class imbalance, a common predicament in machine learning datasets, is also addressed by evaluating the distribution of quality and color categories. The data reveals a stark imbalance in class distributions, prompting the use of oversampling and undersampling techniques to balance it.

The use of RandomOverSampler aids in amplifying minority classes in the target variable (quality), ensuring each class is represented equally, counteracting the initial imbalance. Afterward, features and labels (predictors and target variable) are separated and oversampled to create a new DataFrame, oversampled_df. An additional evaluation of class distribution after oversampling provides insight into the augmented data quality and size. Thereafter, an under-sampling strategy, utilizing RandomUnderSampler, is implemented to balance the 'color' category in the

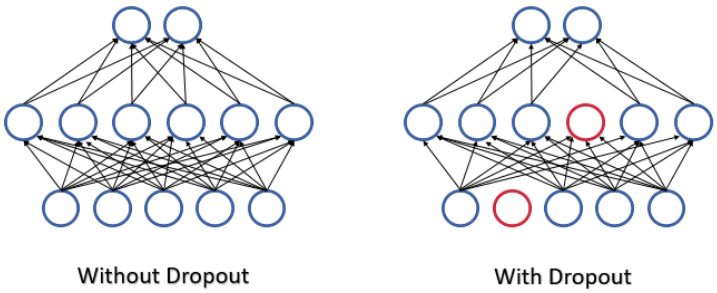

Figure 7.3: Standard Feed Forward Network with Dropout

feature space, forming a balanced DataFrame, balanced_df. This ensures the model is not biased towards the majority class during training, enhancing its predictive capability on unseen data.

The second portion of Exploratory Data Analysis (EDA) in this model focused on obtaining a deeper statistical and visual understanding of the balanced_df dataset, which incorporates both red and white wine data that has been preprocessed and balanced in terms of its feature classes. A correlation matrix is computed to explore the linear relationships among the various numerical variables within the dataset. Observing the matrix, we notice various degrees of correlations among the variables such as a strong positive correlation between "fixed acidity" and "density" (0.594096), and a notable negative correlation between "fixed acidity" and "pH" (-0.342105). Conversely, some variables exhibit minimal to no correlation, like "volatile acidity" and "sulphates" (0.035319).

Visual representation of data distributions and potential outliers is achieved via histograms and boxplots for all numerical columns, providing a panoramic view of the data spread and central tendency. Histograms depict the frequency distribution of the data, revealing insights into the skewness and kurtosis of the variables. For instance, variables such as 'fixed acidity' and 'pH' demonstrate varied distribution shapes, each offering unique insights into their respective data characteristics.

Conversely, the boxplots illustrate the statistical summaries of the variables, spotlighting potential outliers and the interquartile range, which is pivotal for understanding variability and dispersion in the data. For instance, a box plot of 'chlorides' might reveal potential outliers that could be critical in predictive modeling and might require further investigation or transformation.

A scatter plot showing the relationship between 'fixed acidity' and 'pH' was also created, allowing for a visual exploration of how these variables may co-vary. In the data science pipeline, establishing the nature of relationships between variables, either linear or non-linear, aids in selecting appropriate modeling techniques during the predictions.

Lastly, the data is grouped by 'quality', and descriptive statistics such as mean are calculated for each group, providing a structured summary of the central tendency of the variables per quality level. For example, the average 'alcohol' content increases from quality level 3 to 9, possibly suggesting a positive relationship between alcohol content and wine quality. This grouped data acts as a preliminary step into multivariate analysis, offering insights into how the different variables behave or are characterized at different quality levels.

The exploratory data analysis (EDA) process navigates through the dataset, offering insights into its structure, patterns, and potential irregularities. Beginning with a thorough examination of statistics and data, and identifying possible missing values, the EDA also embarks on a detailed

exploration involving correlation matrices and various visualizations like histograms, box plots, and scatter plots. This exploration illuminates underlying distributions, relationships among variables, and possible outliers, guiding subsequent steps such as feature selection and model development in the predictive analytics pipeline.

Additionally, the analysis extends to grouping data, providing a detailed view of variable behaviors across different categorical levels.Throughout the training epochs there were fluctuations in both training and validation accuracies, with the training accuracy gradually improving, but the validation accuracy and loss showcasing some variability. The testing phase, as per the script, employs the model to predict outcomes on a test dataset, obtaining an accuracy of 80%and a loss of 0.5584, thus showing a model with decent generalization accuracy.

Chapter 8

Applied XAI Techniques on Image Data

8.1 Applied XAI Image Techniques

8.1.1 Activation Maximization

In the exploration of Explainable Artificial Intelligence (XAI) techniques within the context of Convolutional Neural Network (CNN) interpretability, is the utility of Activation Maximization (AM) on the OxfordPetIII image dataset, providing a visual lens through which the activation of each layer of the CNN model could be inspected and analyzed. Activation Maximization, by design, identifies and visualizes the patterns and features in the input data that maximally activates certain neurons in the network, hence potentially uncovering the network's internal representational structure and providing insights into its decision-making logic. Aiding in the performance of this technique an initial basis using animal images were used.

In leveraging AM, the usage of animal images, prioritizing the preservation of inherent, recognizable features and textures ubiquitous in natural data. This methodology diverged from the traditional approach of initiating AM with random noise or blank canvases. By embarking from a starting point that already encapsulates authentic data distributions, the goal was to converge towards visualizations that were not only meaningful but also portrayed a realistic depiction of how neurons were activated in the presence of actual input data.

The employed CNN model was tasked with the classification of various animal species, interpreting and correlating an array of features from the input images. Activation Maximization, in this context, was utilized to maximize the output of specific neurons, providing a visual representation that elucidated the features and patterns that the model deemed significant for its decision-making process.

While AM has the potential to illuminate model decision pathways, it does not come without its challenges. These include, but are not limited to, the emergence of unrealistic visualizations or artifacts that may not correspond to actual input features, which could potentially mislead interpretative efforts. Additionally, high computational costs and occasional ambiguity in the generated visualizations presented hurdles in seamlessly deriving intuitive and actionable insights. Despite these challenges, the application of Activation Maximization on the OxfordPetIII images facilitated a deeper dive into understanding the model's hierarchical feature extraction and provided a visual narrative of the internal mechanics of the CNN, thereby contributing to the broader discourse on the interpretability and transparency of deep learning models in image

classification tasks.

A conscious effort was made to navigate through the intricacies and limitations of AM by analyzing the generated visual outputs, ensuring that the interpretations and subsequent discussions were rooted in a careful and critical examination of the visualizations, whilst being mindful of the inherent challenges and limitations posed by the technique. This approach illuminated not only the nuanced behaviors of the CNN model but also underscored the imperative for balanced and cautious interpretation within the realm of XAI, ultimately contributing to the ongoing dialogue surrounding reliable and meaningful model interpretability in the field of data science and machine learning.

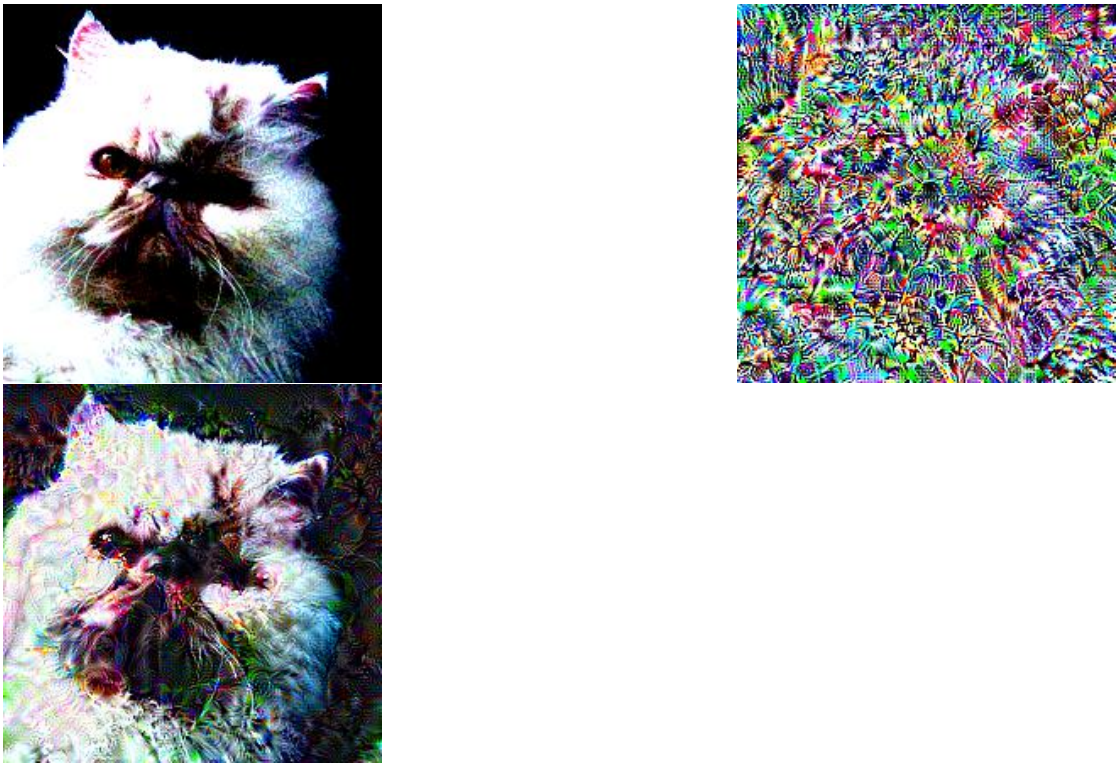

Figure 8.1: Iteration of Activation Maximation with a Prior

When applying Activation Maximization to animal images from the OxfordPetIII dataset within the Network,, a series of issues and difficulties surfaced that provided both challenges and insights into model interpretability. AM, while illuminating in providing visualizations of what neurons in the network prioritize, often introduced high-frequency artifacts and occasionally unrealistic or exaggerated visual features, especially noticeable when dealing with the diverse and intricate patterns, textures, and shapes present in animal images. For example, the generation of visual artifacts, such as overly accentuated edges, unnatural color palettes, or exaggerated features, could detract from the genuine attributes of animal imagery, misleadingly indicating that the model values non-authentic image characteristics. Hence, the interpretability becomes nuanced, as researchers must dissect whether visualized activations genuinely reflect model logic or are mere byproducts of the AM optimization process.

The computational demand of applying AM on animal images, especially across numerous layers and neurons of a CNN, further presented practical challenges. Given the varying hues, tex-

tures, and forms inherent in animal imagery, ensuring meaningful and resource-efficient visualizations while navigating through the complexity and variety of these images became a particularly daunting task.

Despite the challenges, the deployment of Activation Maximization on animal images introduced a substantial impact on the interpretability of the CNN model. Through visualizations illustrating which aspects and features within the animal images maximally activated neurons, AM granted an invaluable window into the internal mechanics of the neural network, revealing what the model prioritizes or dismisses during decision-making processes. Particularly when discussing the model with diverse stakeholders or those with limited technical expertise, these visualizations can serve as a compelling communicative tool, offering a visual narrative of model decision-making.

However, it remains imperative that the insights derived through AM, especially considering the intricate and varied nature of animal images, be approached with discernment due to the potential misrepresentations or exaggerations inherent in the method. Thus, while AM undeniably enhances interpretability and provides a visually accessible method to comprehend model behaviors, its deployment needed to be enveloped within a conscious and judicious interpretative approach. This ensures that insights and visualizations derived were not only insightful and revealing but also scrutinized for accuracy and authentic representation of model functionality, aiding against inadvertent propagation of misleading or inaccurate interpretations of model behaviors.

8.1.2 Image Counterfactuals

For Counterfactual Visual Explanations in the field of image analysis, using and animal images, revealed a dynamic intersection between model interpretability and feature importance. By deploying counterfactual explanations using superpixels on a dataset like OxfordPetIII, the analysis could focus on identifying and modifying the specific superpixels within animal images that drive classification decisions, essentially asking: "What minimal changes in the image would alter its classification?" Here, the methodology not only concentrated on deciphering which visual aspects (such as fur texture, color, and shape) were pivotal in the classification process but also on conceptually and visually showcasing how subtle modifications could reroute model decisions.

Generating plausible and trustworthy counterfactual visual explanations, especially in the context of animal images, brings forth a uniquely nuanced set of challenges. Animal images possess a lot of detail and variability – from the subtlety of fur patterns to the complexity of color gradations. Every image exhibited distinct visual characteristics, and often, minor variations can discernibly differentiate sub-species or breeds. Crafting modifications to such images that are perceptibly subtle, yet sufficiently influential to alter classification without violating the inherent visual logic of the animal represented, is an intricate balancing act. The task becomes twofold: to not only alter critical, model-influencing features but to do so in a manner that retains the inherent believability and naturalism of the image.

In the experimental framework, the swapping of superpixels between two images fundamentally involves selecting and exchanging localized, coherent clusters of pixels (superpixels) between them. To illustrate, two images from different classes (e.g., different animal species). A superpixel from the first image, which represents a discernible region (such as a particular marking or feature), is identified and swapped with a superpixel from the second image. The resultant image thereby embodies a synthetic melding of features from both original images.

Initially, the model is loaded and image preprocessing is conducted using torchvision's transformation capabilities, ensuring images are of the appropriate format and normalization for model input. A mechanism to load images and their corresponding class labels from a directory is instan-

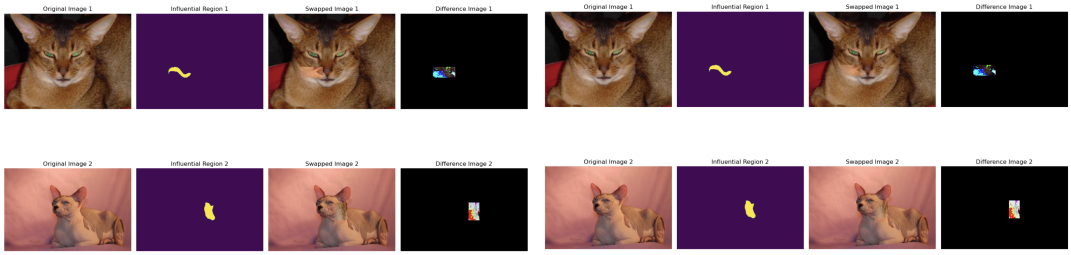

Figure 8.2: Developed Counterfactual images with and without gradient masking

tiated, producing a dataset and corresponding mappings between class names and indices. The core functionality unfolds within a series of defined functions: get_superpixels(image, n_segments) obtains superpixels of the image using SLIC segmentation; get_most_influential_superpixel(image, model, original_class, segments) determines the superpixel, which upon alteration, most significantly impacts the model's prediction; change_prediction(image_path, model) iteratively identifies and modifies influential superpixels until a change in model prediction is observed; and get_counterfactual_explanation(original_image_path, counterfactual_image) yields an image illustrating the perturbation effect by subtracting pixel values of the original and counterfactual images.

Moreover, the algorithm engages further sophistication by integrating bounding boxes around identified superpixels and implements alpha blending to smoothly integrate superpixels from two different images, facilitating the seamless swapping of these regions to create a nuanced, visually-coherent resultant image. This is achieved by a mask, which generates a circular gradient to achieve smooth transitions during swapping, and fading of the pixel, which conducts the actual swapping while preserving the aesthetic consistency of the resultant images. Finally, methodically organizing the original , swapped, and difference images for concise, comprehensive visual comparison, aiding in evaluating the qualitative effect of the superpixel swapping strategy. This methodology, blending sophisticated superpixel identification and swapping, offers a nuanced approach to comprehending model behavior and exploring counterfactual scenarios in image classification.

The difficulty further escalates due to the need to maintain a delicate equilibrium between generating an image that is sufficiently different to yield an alternative classification (influence) while ensuring that the alterations are realistic and credible (authenticity). Manipulating superpixels to create a believable counterfactual can sometimes result in images that, while theoretically plausible, might not resonate with real-world, empirically observed variations within the animal kingdom. For example, altering the coloration of a particular bird species to influence classification might result in a hue that is not naturally occurring, thereby challenging the ecological validity of the counterfactual scenario generated.

From a computational and algorithmic perspective, ensuring the coherent modification of superpixels that respect the intricate patterns and textures within animal images demands advanced optimization strategies. These strategies must be capable of navigating through the high-dimensional space of potential image alterations, identifying pathways of modification that not only influence model outcomes but also adhere to the visual coherence and plausibility of the resultant image. Developing algorithms that can respect the multitude of unspoken rules regarding animal appearance, texture, and coloration, especially across a diverse dataset like OxfordPetIII, brings about complex computational and modeling challenges.

Implementing counterfactual visual explanations, specifically through superpixel perturba-

tion, introduces a host of challenges and nuanced complexities. The inherent diversity and granularity found in animal images, from various textures to color patterns, generate a multi-faceted environment wherein subtle variations can significantly impact classification outputs. Ensuring that identified and modified superpixels authentically represent crucial decision-making aspects within the CNN, without inadvertently introducing misrepresentative or artificial attributes, becomes a critical yet intricate endeavor. This becomes particularly complex when attempting to maintain the naturalistic integrity of animal images while performing subtle modifications to derive counterfactual scenarios.

8.1.3 Grad-Cam Picture

Implementing the Grad-CAM technique on animal images from the OxfordPetIII dataset clarified the Convolutional Neural Networks (CNNs) classification by pinpointing vital image regions via heatmaps. These visual guides showcased the model's focus, effectively illuminating its classification rationale and strategy.

Gradient-weighted Class Activation Mapping (Grad-CAM) facilitates an understanding of the critical regions within input images that inform classification decisions made by the Convolutional Neural Network (CNN). The primary utility of Grad-CAM is to formulate a localization map, which emphasizes those regions of the image that are crucial for the model's predictive output. It adeptly creates a visualization that highlights the influential areas impacting the model's predictive decision, thereby serving as a robust tool for visual interpretability.

In the context of the provided code snippet, Grad-CAM is utilized to visualize class activation maps for a collection of images, leveraging a pre-trained model. The initial step involves retrieving a pre-trained constructed model that performs the classification of the images. This step negates the necessity of retraining the model, thereby ensuring its availability for immediate application. The images are subsequently loaded and undergo a series of preprocessing steps through specifically defined functions. These functions perform a multitude of actions, such as converting the images to RGB, resizing, cropping, and normalization, thus prepping them for model input. Additionally, the images and their respective class labels, extracted from their filenames, are stored for ensuing use.

The selection of a specific class of interest is undertaken using the variable chosen_class_name. This variable facilitates the procurement of the corresponding image path, following which the image is loaded and preprocessed, preparing it for input into the model. Utilizing Grad-CAM, instantiated from the torchcam.methods module, the class activation map is generated. This involves designating a target_layer - a particular convolutional layer within the model. The essence of Grad-CAM revolves around computing the score's gradient for the chosen class, relative to the feature maps emanating from the last convolutional layer. This is followed by a weighted combination of these maps, subjected to ReLU activation, to produce a coarse localization map that underscores the image regions instrumental in predicting the class.

The final stride in the application of Grad-CAM involves the visualization of the original image in conjunction with the computed activation map, facilitated by matplotlib. This may involve an additional step of resizing and normalizing the activation map to ensure spatial consistency and a coherent overlay on the original image. The resulting visualization adeptly illuminates the spatial regions within the image that predominantly influenced the model's classification decision, serving as an invaluable asset for enhancing interpretability and informing model development and diagnostic processes.

However, employing Grad-CAM brought challenges, particularly in validating the accuracy of the generated heatmaps. Ensuring that these visualizations authentically represented the model's focus areas without being sidetracked by unrelated image components or backgrounds

was pivotal to avoid misinterpreting model activity. Further, discerning subtle, vital features, especially in distinguishing alike animal species, was sometimes obfuscated due to the heatmaps' occasionally coarse granularity.

Figure 8.3: Image obfuscated with coarse granularity

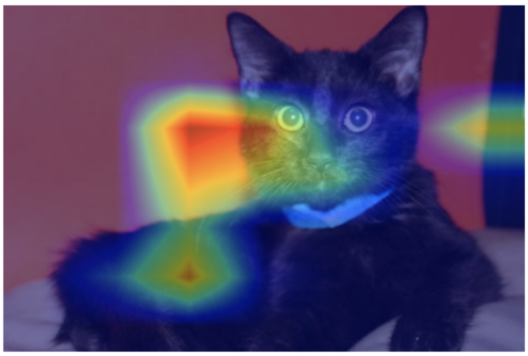

Figure 8.4: Image of black cat with heatmap showing important regions

Deploying Grad-CAM, despite its challenges, brings to light several undeniable advantages. It unveils the otherwise hidden layers of model decision-making, providing a visual conduit through which we can begin to decipher and discuss model behaviors and priorities. This visual demystifies the black-box nature of CNNs, making them more accessible and digestible to non-expert stakeholders and facilitating more transparent discussions about model outputs. Moreover, the visualizations generated by Grad-CAM serve as a valuable diagnostic tool, allowing data scientists to identify and understand areas where the model's attention is either adequately focused or potentially misdirected, thereby opening avenues for further model refinement and optimization. Consequently, Grad-CAM becomes a bridge, connecting complex model mechanics with tangible, accessible visual representations, enhancing both interpretability and ongoing model development.

Despite these hurdles, using Grad-CAM enhanced the understanding of CNN classifications by visually representing the model's focal points during decision-making. The derived heatmaps not only spotlight key image areas but also underscore the need for further refinement and

validation of the technique, reinforcing its potential to fortify model interpretability in subsequent applications. The critical task ahead is to certify that these visual aids are precise, stable, and a true mirror of the model's internal computations and focus, ensuring robust interpretability in real-world scenarios.

8.1.4 Rule Extraction using Activation Values

In the realm of machine learning, decision trees have been heralded for their innate transparency and interpretability, qualities particularly in the extraction of actionable rules from constructed models. As a component of Explainable Artificial Intelligence (XAI), rule extraction using decision trees in the OxfordPetIII animal image analysis shows a more coherent and solid foundation on comprehending model actions. Thus helping to clarify model actions and estimations on image classifications.

Embarking on rule extraction, the CNN model designed for animal image classification within the OxfordPetIII dataset was dissected to derive discernible and actionable rules. Here, the decision tree acted as an intuitive mapping tool, converting complex layers of the neural network into a set of rules which could be readily interpreted, even by non-experts. However, challenges burgeoned when confronting the nuanced and diverse image attributes present in animal images - ensuring the extracted rules accurately mirrored the delicate balance of features, such as fur textures, color patterns, and anatomical shapes, that the CNN model factored into its classifications.

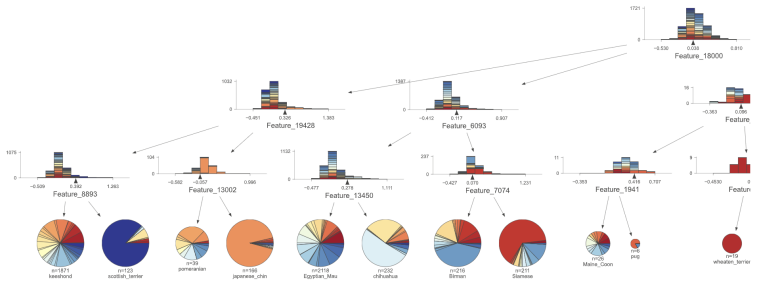

Figure 8.5: Portion of the constructed Decision Tree using Dtreeviz

However, this method has an issue accurately portraying image features unlike its other tabular and text counterparts. This is due to the inherent application and analysis of image models. In this case a model continually learns and expand the learned features meaning that each base feature changes depending on the run and different section of the image making it difficult to label a rule. For example how is this techniques supposed to identify and label in the model a cats eye vs a dogs eye.

Chapter 9

Applied XAI Techniques on Text Data

9.1 Applied XAI techniques on Text Data

9.1.1 Rule Extraction on Text Data

Rule extraction in the context of text data, especially regarding spam detection using LSTM models, leverages an intricate understanding of how textual sequences and their constitutive elements are processed and weighed to ascertain predictive outcomes. By extending this concept towards rule extraction using decision trees, we delve into a systematic and hierarchical approach that seeks to encapsulate the learned predictive pathways of the LSTM model into an interpretable, rule-based framework.

Utilizing decision trees for rule extraction from an LSTM model trained on a 'Spam' or 'Not Spam' dataset enables an understanding of how various words or sequences within the text data influence the spam classification outcome. This essentially involves translating the learnt sequential and temporal dependencies captured by the LSTM during training into a set of hierarchical decision rules that can be visualized and interpreted through a decision tree. Each node in the tree represents a decision based on a particular word or sequence of words, guiding the pathway down the tree until a classification decision (i.e., 'Spam' or 'Not Spam') is reached.

Implementing rule extraction via decision trees on an LSTM model trained for text classification, particularly for spam detection, involves a complex, layered process. The model inherently captures and processes sequential dependencies within the text data, navigating through the textual elements and their temporal relationships to formulate its predictions. Herein lies a challenge as LSTM models, with their recurrent and memory-driven architecture, inherently operate in a non-linear, dynamic, and temporally-dependent manner. Translating these rich, sequential learnings into a static, hierarchical decision tree involves distilling and approximating the nuanced, temporally-contingent pathways into a set of discrete, interpretable decision rules.

In practical terms, other techniques like LIME, as discussed in previous sections, to first understand how different words or phrases influence the predictive outcomes. Subsequently, the insights gained from such explanatory models can inform the construction of decision trees, wherein words or phrases that are highly influential in the model's predictive processes are deployed as decision nodes in the tree, guiding the hierarchical, rule-based decision-making process towards classifying texts as 'Spam' or 'Not Spam'. The decision tree thus becomes a simplified, interpretable approximation of the LSTM's predictive logic, delineating clear, understandable

rules that approximate how the LSTM navigates through the textual data to ascertain its predictions.

While the decision tree provides interpretability and simplicity in understanding the model's decision-making process, it's pivotal to acknowledge the potential loss of predictive accuracy and nuance. The LSTM's capability to understand and process complex temporal and sequential dependencies within the text may not be fully represented within the decision tree, thereby creating a trade-off between interpretability and predictive richness. Nonetheless, in applications where model interpretability is paramount, such as ensuring fairness, transparency, and accountability in spam detection, rule extraction through decision trees presents a valuable approach to demystify the complex, nuanced decision pathways of LSTM models, enhancing their accessibility, scrutiny, and ethical deployment in real-world scenarios.

9.1.2 Attribution values using Lime

A dataset comprising of emails, spam, and their corresponding labels are imported from a CSV file. After loading, the textual data undergoes tokenization and is converted into numerical form through a created mapping which accommodates varied sequence lengths and integrates special tokens for padding and unrecognized words. The defined TextDataset class facilitates the convenient management and conversion of text data into a format amenable to PyTorch's computational demands and batch processing during model training. Furthermore, this processed dataset is judiciously partitioned into training, validation, and test subsets, ensuring a robust framework for subsequent model training and evaluation phases.

Following data processing, the script addresses the model deployment aspect, wherein the constructed pre-trained LSTM model is loaded from a .dill file, subsequently being positioned into an evaluation mode, ready to make spam categorization predictions on incoming text messages. A pivotal function, predict_proba, is defined which performs the role of a bridge between raw text inputs and the numerical output. This function is tasked with taking raw text inputs, transforming them into a numerical format via tokenization and mapping, and then forwarding them through the model procuring the predictions. These output predictions are translated into a probability format which is crucial for compatibility with the LIME interpretability framework in ensuing stages.

The last step is when the predictions are used to demystify the predictive decisions made by the LSTM model through the utilization of the LIME (Local Interpretable Model-agnostic Explanations) interpretability framework. LIME is tasked with generating perturbations of the input data, obtaining predictions on these perturbed instances from the LSTM model, and then constructing a locally faithful interpretable model that approximates the decision boundary of the original model. By utilizing a test instance and employing the explain_instance method from LIME, the script endeavors to render transparent the model's prediction mechanism, particularly illuminating the contributions of individual features (words or phrases) towards the final prediction. The explanatory results, zeroing in on 10 pivotal features, are visualized to facilitate user comprehension, ensuring that the model's decision-making mechanism is not an inscrutable "black-box" but rather a transparent and interpretable entity, notwithstanding the intrinsic challenges related to maintaining coherency and precision in explanatory models for natural language processing tasks. The goal is to illustrate the impact of altering a textual instance on the model's predictive logic, by conducting an insightful comparison between the explanations of an original and a modified text instance. A specified index, directs us towards the initial test instance that we desire to investigate and potentially modify. The original instance is retrieved directly from the data, while the modified instance is derived by excluding specific words, such as "camera" and "a", from the original text. This is followed by defining a new prediction probability func-

tion that adapts to this modified data. Employing LIME, explanations for both original and modified instances are generated by invoking the explain_instance method, which illuminates the words or phrases that predominantly influence the model's predictive decisions for both instances. Subsequently, these explanations are printed and visualized to facilitate an intuitive understanding and comparison of the model's reasoning behind its predictions in the context of the two instances. Through this, users or model developers can observe how the absence or presence of specific words (like "camera" or "a") alters the model's interpretative reasoning and prediction, thereby providing insights into the model's sensitivity and stability concerning input variations and aiding in comprehending how certain textual elements influence predictive outcomes. Upon integrating the provided LIME output into the discussion, it introduces a nuanced

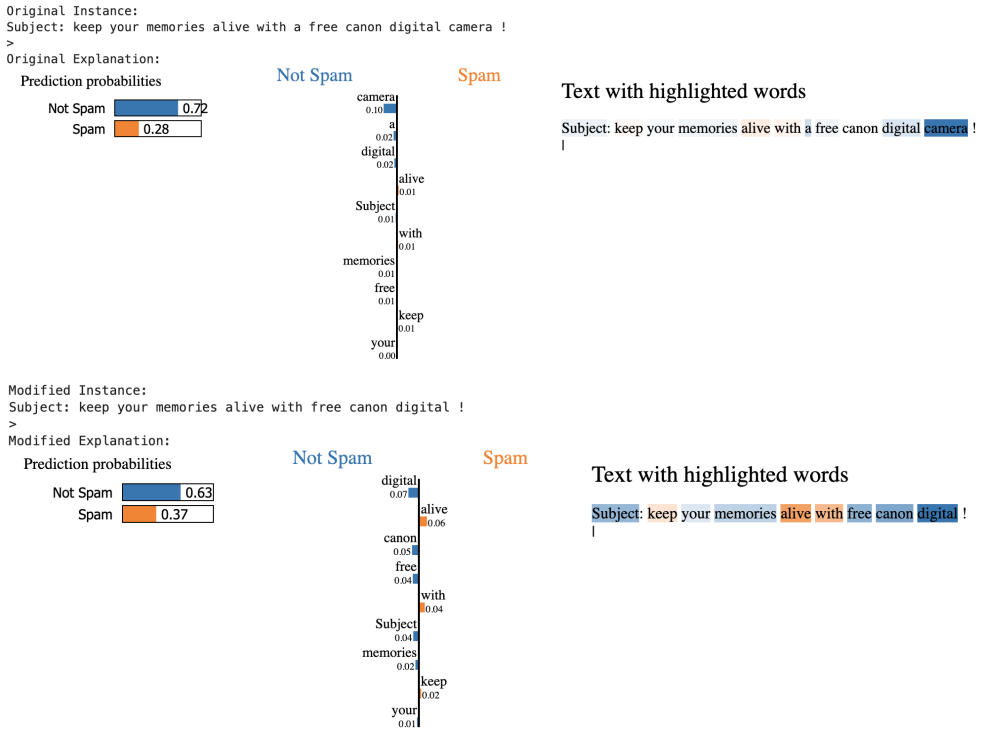

Figure 9.1: Lime Attribution scoring based on specific words in email

layer to the understanding and interpretation of LSTM models in text classification tasks. The original instance, "Subject: keep your memories alive with a free canon digital camera !", was classified with a 0.72 probability of being "Not Spam" and a 0.28 probability of being "Spam". Intriguingly, LIME highlighted "camera" as an impactful word with a 0.10 weight towards the prediction. Post-modification, where the words "camera" and "a" are removed to produce "Subject: keep your memories alive with free canon digital !", the predictive probability shifted to 0.63 for "Not Spam" and 0.37 for "Spam", underscoring a non-trivial impact on model interpretability and outcomes based on textual variations. This tangible example mirrors the complexities and nuances faced when employing techniques like LIME for model interpretability in textual contexts. The shifting prediction probabilities and influential weights assigned to certain words, such as "camera" in the original instance, underline the potential for specific textual elements to skew model decisions, which can be intricately understood and scrutinized through LIME's

explanatory lens. The word "camera" in the original text significantly influenced the model's prediction, implying a sensitivity to specific terms, which may illuminate potential biases or dependencies within the model's layers. The modification of text, albeit subtle, also altered the model's predictive confidence, which is crucial for developers to comprehend and potentially mitigate against unjustifiable word sensitivities or biases, ensuring that the model's predictions are not only accurate but also robust and justifiable across varied instances. This exemplifies the crucial impact and benefit of employing LIME in network models when facilitating an enriched understanding of how sequential and textual data are navigated, processed, and weighted within theneural networks. It offers a pathway to unravel the complexities of the models predictions, especially in a domain like spam detection, where textual intricacies, slang, and varying structures can profoundly influence predictive outcomes. This, consequently, guides towards more insightful, ethically grounded, and reliable model development, tuning, and deployment, ensuring that critical terms or textual structures do not unduly or unintentionally sway predictions, and that models generalize effectively and transparently across diverse textual instances. In turn, developers and users alike are empowered with an enhanced understanding and trust in the model's predictive capacities, anchoring the LSTM in a foundation of comprehensibility and reliability amidst its complex, recurrent processing pathways. Navigating through the realms of interpretability and visualization with such models, especially when dealing with text data like spam detection, presents a multifaceted challenge and opportunity in machine learning. The complexity and opacity of LSTMs, primarily due to their ability to maintain hidden states and memory cells that decipher long-term dependencies in data, catapult them into a domain that is inherently intricate and hard to interpret. The recurrence, hidden activations, and transformations within the LSTM architecture are formidable to distil into a simpler, rule-based representation or a decision tree, which are typically more interpretable and transparent. Consequently, the transition from high-dimensional, temporal dependencies to a 2D/3D visual space or simple rule sets is fraught with difficulties, potentially leading to the loss or misrepresentation of essential information, especially the nuanced interactions and sequential dependencies identified by the LSTM.

Moreover, the visualization of the model is not straightforward due to the temporal and non-linear nature of their decision-making mechanics. The temporal dependencies embedded within LSTMs, involving memory cells and evolving hidden states, and their inherent non-linear transformations add to the complexity of crafting an accurate, comprehensible visual representation. Thus, these models require a strategic balance between simplification for the sake of interpretability and the retention of predictive accuracy and detail in representation. Ethically and practically, it is imperative that LSTMs maintain a degree of transparency and interpretability to ensure accountability and fairness in their deployment, particularly in contexts like spam detection where a misclassification could yield significant implications. The endeavor towards interpretability, despite the challenges, promises not only ethical deployment but also the enhancement of user trust and model adoption by demystifying complex mechanics, ensuring that the pathways leading to predictions are transparent and scrutable to both domain experts and laypersons alike. Consequently, the realm of interpreting LSTM models in text data is as much about navigating through computational and visual complexities as it is about ensuring ethical, comprehensible, and trustworthy AI practices in practical deployments.

9.1.3 Text Counterfactuals

Text Counterfactuals, particularly those generated through Natural Language Processing (NLP) augmenters, play a crucial role in analyzing and improving model interpretability in various NLP applications. Given the provided code snippets, an NLP augmenter is employed to generate

textual counterfactuals, which are essentially alternate versions of original text inputs. In this context, an augmentation model based on BERT ("bert-base-uncased") is utilized to create these counterfactuals by substituting words within the original text, while maintaining grammatical and, to an extent, semantic coherence. The primary objective here is to produce variations of the original text that could potentially yield different predictions when fed into a model, subsequently offering insights into the model's decision-making process and potential biases.

The intricacies and challenges in implementing Text Counterfactuals are multifaceted. Firstly, maintaining semantic coherence while altering the original text's meaning is a precarious balancing act. The substitutions, while being adequately diverse to explore the model's behavior under various inputs, should still retain logical and grammatical coherence to ensure the resulting counterfactuals are realistic and interpretable. Furthermore, identifying the most pertinent words or phrases to alter, ensuring that the changes are substantial enough to emphasize the model's sensitivities and biases, presents another challenge. In some instances, minor text modifications might not considerably impact model predictions, thus requiring more substantial, yet still semantically valid, alterations.

A primary focus was on generating textual counterfactuals for alternate versions of a given text, while also utilizing the nlpaug library for natural language processing (NLP) and pandas for data manipulation. The textual input was used to generate a predefined number of counterfactual instances by substituting words, underpinned by the BERT ('bert-base-uncased') model. These substitutions ensure the generation of semantically and contextually pertinent textual variations. The main execution block engages with a dataset, assuming it is a CSV file with a textual column, iterating through its entries, and leveraging the aforementioned function to create and display the counterfactuals alongside the original text. Primarily, this code could be leveraged for creating additional data for model training (data augmentation), testing model robustness against varied input, and exploring potential biases or sensitivities in model responses to specific textual alterations, thereby potentially aiding in enhancing model interpretability and robustness in diverse NLP applications and scenarios

Proceeding with BERT, a pre-trained language model, to generate variations (counterfactuals) of the original text by substituting words, with the aim of preserving semantic coherence. However, the potential occurrence of negative variations poses certain challenges and considerations. For instance, a statement like "The weather is sunny" might be counterfactually altered to "The weather is cloudy", changing the assertion's inherent sentiment or factual representation. Such negative variations can be pivotal for model training and testing in tasks like sentiment analysis or fact-verification, allowing for an exploration of model reliability and sensitivity across different textual context.

Creating counterfactuals for text, especially when utilizing automated methods like the one demonstrated in the provided code, brings forth a myriad of complexities and considerations, particularly concerning interpretability and semantic coherence. Firstly, ensuring that the generated counterfactuals retain a semblance of logical and contextual coherence with the original text is challenging. The code uses BERT, a transformer-based model, to substitute words in an attempt to generate varied yet contextually related statements. However, language is intricate, and even slight modifications can alter meaning, introduce ambiguity, or generate syntactically incorrect sentences, all of which impact the interpretability and usability of the counterfactuals.

In the context of model interpretability, such counterfactual examples can be valuable to understand how well the model generalizes across varied inputs and to probe potential vulnerabilities or biases. Nevertheless, the difficulty arises in ensuring that these counterfactuals are genuinely representative of plausible alternative scenarios and not just arbitrary or nonsensical variations. Evaluating and ensuring the quality, relevance, and appropriateness of generated alternatives is non-trivial and introduces a layer of complexity to model assessment and devel-

opment. Additionally, ethical considerations emerge, especially if counterfactuals inadvertently introduce or perpetuate harmful stereotypes or biases.

Moreover, the impact on interpretability is twofold. While providing diverse inputs to evaluate and enhance model robustness, it also mandates careful scrutiny to ensure that counterfactuals are meaningful and are steering model development and evaluation in a constructive and ethical manner. Thus, the use of text counterfactuals, while potentially illuminating in understanding model dynamics, requires cautious, context-aware application and evaluation to genuinely enhance interpretability and ethical model use.

Chapter 10

Applied XAI Techniques on Tabular Data

10.1 Applied XAI Techniques on Tabular Data

10.1.1 Rule Extraction on Tabular Data

In the comprehensive analysis performed on the wine datasets, the robust data preparation and modeling steps have played pivotal roles in establishing a framework designed to heighten model interpretability and decision clarity. Initially, data from two distinctive wine types, red and white, were imported and methodically concatenated into a singular data frame. This amalgamation not only unified the data but also facilitated the introduction of a 'color' attribute, specifying the wine type, thereby enriching the data context. This data preparation involving careful concatenation, transformation, and encoding of categorical variables was elemental in furnishing a dataset ready for model deployment.

Addressing the apparent class imbalance, discernible in both 'quality' and 'color' attributes, was resolved through over-sampling and under-sampling strategies, respectively. While over-sampling was devised to balance the 'quality' class distribution equitably, under-sampling was employed to harmonize the occurrences of both wine types, fortifying the model against potential bias and ensuring a more generalized performance. This intricate balancing act, albeit essential, introduced a series of complexities, particularly risking the introduction of bias through over-sampling and potential loss of information through under-sampling. Nevertheless, the steps were vital to curb the model's tendency to skew toward the majority class, enhancing its predictive reliability and interpretability.

After, Decision Trees were strategically selected, owing to their inherent interpretable nature and the capability to furnish transparent decision-making insights. The employment of this algorithm, potentially visualized via tools like dtreeviz, highlighted an emphasis on maintaining transparency in how predictions were articulated by allowing observers to visually track decision paths and comprehend how input features influenced predictive outcomes. This transparency not only elevated the interpretability of the model but also brought to light the features pivotal in determining the results, thereby offering tangible insights into the influential variables dictating wine quality and type.

However, navigating the relationship between model simplicity and accuracy, specific difficulties and counteractive impacts emerged. The transparency and ease of interpretation of Decision Trees occasionally veered towards oversimplification of the model, potentially undermining its

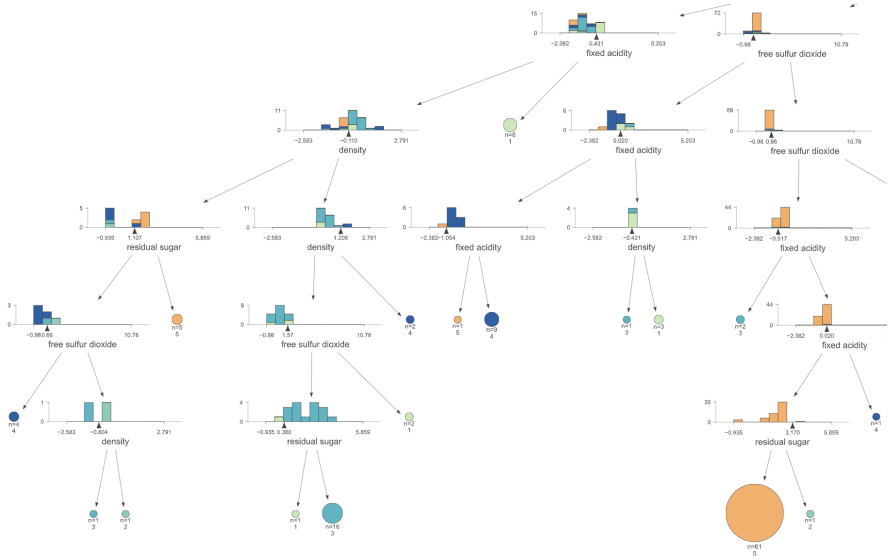

Figure 10.1: Decision Tree for Tabular Data Showing branches based on wine information

ability to capture and represent more nuanced relationships within the data. Additionally, the sensitivity of Decision Trees to data variations posed a challenge, as small perturbations could precipitate different decision splits, thereby introducing variability into the ruleset and, by extension, the interpretability of the model.

In conclusion, the amalgamation of astute data preparation, strategic class balancing, and adopting inherently interpretable models like Decision Trees coalesced to forge a pathway toward enhanced model interpretability and decision clarity. Despite its intrinsic challenges, the methodological approach predominantly offered a visible and traceable model decision-making process, which elevated the comprehensibility and trust in the model's predictive prowess.

10.1.2 Tabular Counterfactual

In the initial stages, crucial libraries such as pandas, numpy, imblearn, dice_ml, and more are imported, establishing a foundation for data handling, machine learning, and imbalanced data mitigation. The data preparation phase is characterized by concatenating red and white wine datasets into a unified DataFrame and incorporating a 'color' attribute to differentiate wine types, alongside ensuring numerical columns are cast to float type.

Addressing the data imbalance is rectified through the adept application of the RandomOverSampler and RandomUnderSampler from imblearn, aimed at mitigating biases in the 'quality' and 'color' attributes, respectively, thereby enhancing the model's predictive generalizability and interpretability. This nuanced approach to balancing ensures that both features are considerately managed to prevent model bias towards prevalent classes. Additionally, the 'color' attribute is label-encoded to facilitate computational processing, converting categorical values to a numerical format.

In the model utilization phase, a pre-constructed and trained model, wine_model.dill, is loaded and utilized to predict a query instance extracted from the balanced dataset. If developed using PyTorch, the model is switched to evaluation mode to ensure consistent predictive performance.

While mitigating imbalance, the undersampling technique may introduce data loss, excluding potentially insightful instances from the modeling process. Additionally, counterfactual explanations, although insightful, may weave a complex web of interpretative challenges, especially when dealing with numerous or correlated features. The computational demands of generating counterfactuals and maintaining consistent and reliable results across different model backends (PyTorch and Scikit-learn) also present potential difficulties. Consequently, while the code offers a structured means to navigate through model interpretation and validation, it does so by navigating through the complexities and challenges that intertwine data integrity and explanatory interpretability.

Original Data

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	color	quality
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	0	0

Table 10.1: Original Data

Diverse Counterfactual set (new outcome: 5)

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	color	quality
-	-	-	-	-	-	7.0	0.989	3.85	-	-	1.0	5.0
-	-	-	20.8	-	-	28.1	-	3.24	-	-	1.0	5.0
-	-	-	22.9	-	-	-	-	-	1.98	13.9	-	5.0
-	1.22	-	25.5	-	-	-	-	3.51	-	13.7	-	5.0

Table 10.2: Diverse Counterfactual Set

The provided output illustrates the model's initial prediction for a particular instance of wine and offers ten diverse counterfactual examples. Initially, the wine sample, characterized by features such as fixed acidity of 7.4, volatile acidity of 0.7, residual sugar of 1.9, etc., is predicted by the model to belong to the quality class 0. This scenario might be problematic for a vintner aiming for higher-quality production. The generated counterfactuals proactively tackle this by suggesting modifications to the original instance that could uplift the predicted quality to a preferable class, namely 5.

Delving into the counterfactuals, alterations in various features, like 'total sulfur dioxide,' 'density,' 'pH,' 'volatile acidity,' 'residual sugar,' and 'alcohol,' across different instances, evidently influence the model's prediction. For example, in the first counterfactual, merely adjusting the 'total sulfur dioxide' to 7.0, 'density' to 0.989, and 'pH' to 3.85 (while changing 'color' to 1.0) prompts the model to predict the more favorable quality class 5. When evaluated, these alternatives can help winemakers experiment with and understand how certain modifications in the wine-making process (like altering acidity or sugar levels) might significantly impact the resulting wine quality, according to the model's learned patterns. Moreover, it could direct the model developers towards understanding its sensitivities and biases, fostering further refinement and optimization of the predictive model by aligning it more closely with real-world, practical applications and expectations.

10.1.3 Partial Dependence Plots and Variable Effect Curve

Partial Dependence Plots (PDPs) and variable importance through permutation (VECs) are potent instruments for gauging and visualizing the impact of a feature, or a set of features, on

a predictive model's output. Both mechanisms are pivotal for unraveling the relationships and dependencies encapsulated within complex models, particularly in domains where understanding the model's decision-making pathway is paramount, such as healthcare or finance.

A neural network model is first utilized to predict wine quality based on numerous chemical attributes and wine type (red or white). PDP is then applied via a surrogate model (a RandomForestRegressor) trained on the neural network's predictions to unveil the relationship between a feature—say, "alcohol" content—and the predicted quality of the wine. It grants a graphical representation that depicts how alterations in a particular feature value influence the mean prediction of the model, holding all other features constant. A line in the PDP graph thus illustrates how varying alcohol levels impact the anticipated wine quality, furnishing interpretable insights into feature-effect relationships.

However, despite their efficacy in furnishing intelligibility, PDP and VEC are full of challenges. One prominent difficulty pertains to their performance in the presence of interaction effects among features. PDPs, in particular, may provide misleading insights when strong interactions between components are prevalent since they assume that the features are independent. Additionally, in high-dimensional spaces or with numerous categorical variables, PDPs can become computationally intensive and potentially obfuscate more than elucidate due to the sheer complexity of the resulting plots. Their simplicity can also be a double-edged sword, as they condense multi-dimensional data into a two-dimensional plot, which might sometimes obscure the underlying complexity or high-order interactions.

Furthermore, the interpretability offered by PDPs and VEC in model scrutiny is vital for model validation, especially in regulated industries. Their implementation aids in ascertaining that a model's predictions adhere to expected behavioral patterns and that the model comprehensively understands the underlying data-generating process. For instance, if a PDP indicates that an increase in alcohol drastically enhances predicted wine quality beyond logical limits, it could signal potential overfitting, data leakage, or other model inadequacies. Consequently, the intelligibility they provide only facilitates more transparent communication with non-technical stakeholders but also acts as a diagnostic tool for model assessment and validation.

PDP specifically gauges and visualizes the univariate marginal effect of a feature on the predicted outcome, holding other features constant. The PDP curve represents the mean predicted outcome as a function of a specific feature (for instance, the "alcohol" feature in the provided code snippet). A flat curve indicates that the feature has a negligible impact on the prediction, whereas a steep curve signals that variations in the feature value significantly influence the prediction. Ascending slopes suggest positive correlations, while descending ones imply inverse relationships. These interpretations can become vital in various domains, aiding experts in comprehending how alterations in certain variables, like the alcohol level in wine, can potentially enhance or diminish the predicted quality.

Below the constructed PDP charts with chlorides, citric acid, and density, display non-linear relationships in their respective PDPs. For instance, the initial decline and subsequent escalation in the PDP of chlorides might suggest a dual effect on the predicted outcome – potentially a measure like taste or quality in wine. In this scenario, lower and upper extremes of chloride levels could be associated with distinct flavor profiles, each appealing and unappealing in its own right, creating a U-shaped influence on the outcome. Citric acid and density, following a similar PDP trajectory, hint at a commonality where median values might be less conducive to positive predictions compared to their respective lower and higher extremes. These non-linear relationships serve to highlight the often complex and multi-faceted roles that individual features play in influencing predictive outcomes.

Conversely, considering the PDPs of fixed acidity and alcohol, which depict mostly positive and linear relationships, a different aspect of feature influence is brought to the fore. The

consistent, linear increase of the prediction with the values of fixed acidity and alcohol may be representative of a direct, proportionate relationship between these features and the model's outcome. In contexts such as wine quality prediction, a gradual increase in alcohol may correlate with a consistent enhancement in certain desirable characteristics, thereby progressively elevating predicted quality. Similarly, the positive linearity with fixed acidity might indicate a steadily enhancing effect on the output metric.

Since PDPs can illustrate both linear and non-linear relationships, they offer a nuanced view of the impact a feature exerts across varied data points. However, PDPs inherently assume no interactions between the features, which can be a considerable limitation, especially when the influential feature is correlated with others. This could lead to erroneous or overly simplistic interpretations, particularly when the isolated impact of a single feature does not authentically represent its effect within the broader model.

VEC or permutation feature importance, on the other hand, offers a different lens through which feature impact is scrutinized. It involves permuting the values of a specific feature and observing the resultingg impact on model accuracy. A substantial decrease in model performance upon permuting a feature implies its high importance. Understanding the ranking of variable importance can be crucial for model refinement and for prioritizing features in exploratory data analyses and feature engineering phases. It enables domain experts and data scientists to hone their focus onto aspects that significantly drive predictions, thereby ensuring resource and attention allocation that aligns with impactful variables.

However, it's crucial to note that VEC also has limitations; primarily, it might not accurately reflect feature importance in the presence of correlated variables or in the presence of complex interactive effects. Moreover, it doesn't provide a direct view into the nature of the relationship (e.g., whether it's linear, non-linear, or threshold-based) between the feature and the target variable.

The Variable Effect Curves for sulfur dioxide, volatile acidity, chlorides, and density in our model becomes imperative for making informed decisions. Initially, these variables exhibit a low impact on the model, suggesting that within a particular range, their alterations do not considerably influence the predictive outcome. A following subtle decline indicates a potentially perplexing region wherein their variations might slightly misalign with the model's predictions, perhaps implying an area where the model fails to capture the underlying patterns accurately. Then, a subsequent sharp ascent towards the top-right corner marks a drastic shift, highlighting that beyond a certain threshold, these variables become critically influential in guiding the predictions. Where as those that initially start at the top left and then decrease rightwards show the opposite, a decrease in importance. The stark change from inconsequential to highly impactful status of these variables shows the importance of managing them . It also shows how understanding why such a differential impact is observed and how it can be leveraged or mitigated to harness reliable, consistent predictions from the model, ensuring it behaves in a manner that aligns with empirical knowledge and practical utility. This would aid in deploying the model effectively, especially in scenarios where precision and reliability are paramount, thereby navigating through the complexity of predictive analytics with informed assurance.

In application scenarios, the information offered by PDPs and VEC plays a crucial role not just in interpreting model predictions, but also in aligning the model with domain knowledge and logical expectations. For instance, in the wine quality prediction scenario, if PDPs indicate a seemingly illogical or non-intuitive impact of certain chemical properties on quality, or if VEC indicates an unexpected ranking of feature importance, these discrepancies require a model and data audit to ascertain the root causes and ensure that the model genuinely encapsulates the underlying data patterns and domain logic. Thus, the interpretability and insights offered by PDP and VEC can provide a broader conceptualization of model validation, audit, and refinement, thereby bolstering robust and reliable predictive analytics.

A pre-trained neural network model, serves as a foundation, predicting wine quality based on several features. Subsequently, a surrogate model, trained using a Random Forest Regressor, is applied, employing the same features but taking the neural network's predictions as its target. This establishes a scenario wherein the surrogate model endeavors to approximate the predictions of the inherently more complex and opaque neural network. Partial Dependence Plots (PDPs) derived from the surrogate model become a tool to visualize the relationship between individual features and the predicted outcome, embedding a tangible interpretative layer over the machine-learning model's predictions.

Moreover, surrogate models play a pivotal role in deciphering the complex behavioral dynamics of black-box models like neural networks. Through PDPs, the surrogate model shows how fluctuations in individual features correlate with alterations in the predicted outcome, offering a

human-interpretable narrative of model behavior across diverse feature landscapes. It's crucial to underline, however, that these interpretations, while valuable, are approximations and may not accurately reflect the true underlying mechanisms of the original model.

One of the notable implications of surrogate models and PDPs lies in providing insights into the influence of individual features on predictions. For instance, discerning how variables such as alcohol content or acidity might influence wine quality, allows domain experts to develop a nuanced understanding of influential factors, crafting a foundation for practical, data-driven decision-making.

Validating and establishing trust in machine learning models is another vital aspect where surrogate models prove to be instrumental. If the surrogate model, despite its relative simplicity, can closely approximate the complex neural network's predictions, it may validate and corroborate the original model's predictive prowess, also serving as a rudimentary form of model validation.

However, while surrogate models offer a trove of insights and interpretability, navigating through their limitations requires a judicious approach. Surrogate models provide approximations and may not fully capture the nuanced behaviors and decision boundaries of the original model. Furthermore, PDPs, while enlightening, might fail to illuminate interactions between features and could potentially present biased interpretations when correlated features are present.

In summation, surrogate models applied to wine quality prediction, show a pathway towards enhancing model interpretability and practical insights in leveraging machine-learning models. To move towards more robust, interpretable AI requires carful navigation through the capabilities and limitations inherent in these methodologies, ensuring that insights derived are not merely informative but are representative and valid within the broader context of model behavior and data dynamics. This balance between insightful interpretability and cautious acknowledgment of limitations is paramount in advancing responsible and effective applications of AI in diverse fields.

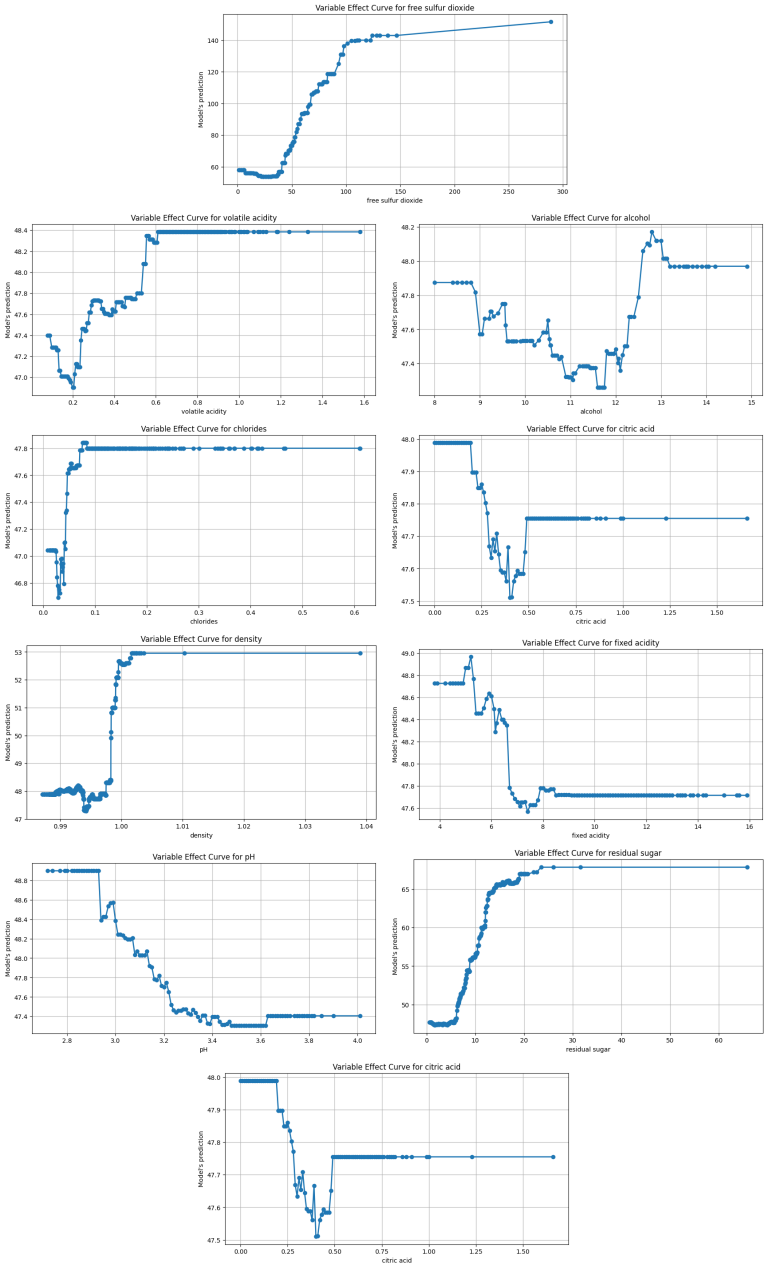

Figure 10.2: VEC Plots of Wine Data

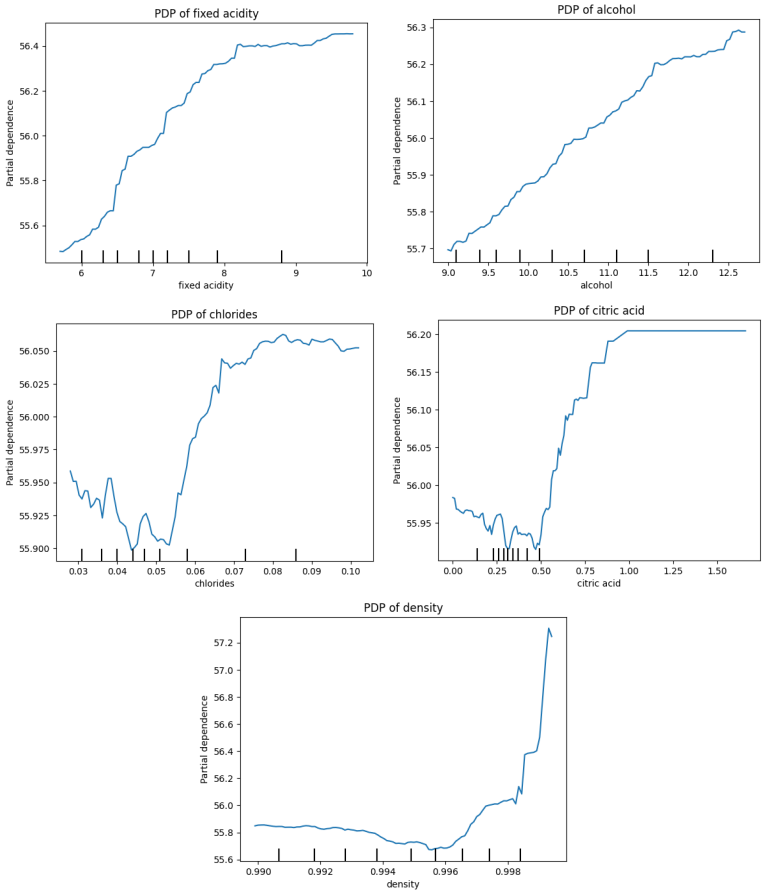

Figure 10.3: Charts of the Partial Dependancy Plot on Wine Data

Conclusions

From the techniques above, a conclusion can be made that despite attempts to aid in interpretability, there are some actions that once simplified become increasingly difficult to understand. However, there are some techniques that based on the inherent data become easier to comprehend.

In terms of Image data, most of the techniques highlighted aspects of the model and data that are beneficial towards appropriate utilization of the model. For instance, grad-cam show important regions that the model deemed to be vital in providing a classification. This can be contrasted against Activation Maximization which is a slightly similar technique, but attempts to use information the model learned to produce an image of a specified class. The Activation Maximization despite using an image as a prior is still difficult to understand or discern specific features in the image.

The other two image techniques faced similar difficulty but clarified there own unique views of the model to improve interpretability.The extraction of rules from the model using a surrogate, produce a large tree with 37 classes that drastically increases the complexity of the produced rules. Thus incurring an even larger chart that can be difficult to follow. In an attempt to counteract this dTreeviz was used to apply charts on the nodes of the constructed tree. While this aided in the highlighting the decisions on the terminal node and it induced more confusion as the large amount of data was unable to appropriately captured in the bar charts representing the branches. As specific parts of the convoluted images are altered and learned its extremely difficult to not only produce parts of the image to help know which feature determines which branch to take, but is extremely difficult as parts of image aren't labeled. Making it difficult for simplified surrogates to produce results on which part of an image is a cat ear or a dog ear.

Counterfactuals, were unique as the technique applied took information from specific regions and swapped them. This often makes images that appear unnatural, but test the discernment of the model through slight perturbation helping people limit the bounds of why such a decision was made and which features of the image are influential of the whole.

Text data, is treated drastically different as it has a temporal aspect that help combine meaning to each previous phrase and sentence. Looking at the Rule extraction, it was easier to provide methods that aided in the comprehension of the splits and reasoning behind it. Looking at the, text rule extraction, it becomes easy to see the words are a lot simpler to use to represent each node split. This is because words, unlike images, are easy to represent on their own and provide impactful information from the past information, images and their features on their own provide little to no information on what is being focused on until higher level features are constructed.

This aspect similarly improves the comprehensibility of the other techniques, for instance when using LIME to look at attribution scores for each email, it provides an aid to show which words tend to be more related to spam emails compared to other within that same instance. This provides support on a models decision and can give a view of the inner-workings of what

influence the decision when determining spam.

In terms of counterfactuals, they were slightly difficult to interpret as there a many ways to approach the same meaning and understanding the for certain sentences. Looking through, the outputs we cant see that there are also certain tones that are used when writing in specific scenarios the bert and the augmenter were unable to fully grasp. For example, in terms of a words that would make sense logically but would never be used in an email is "ain't" which when used to perform negations add layers of more complexity that without checking and fixing make the model unable to perform or generalize appropriately. Especially when in the context of the classification of Spam or ham, should be able to discern the best constructed spam email to protect individuals.

Tabular data while the most structured compared to the others lacked in terms of the quality of techniques as the image and text data as each attribute may be similar but can be from different ranges, types, and may have certain rules. These rules could be specific ordinal values, or label representations for text or non-numerical values.

Comparing the tabular rule extraction compared to the image and text, the tabular is most similar to the text and the attributes of the table can be easily represented similar to each word in a sentence. However, unlike the text data, tabular data is easier for the surrogate model to interpret as long as each instance of an attribute can is numeric or can be represented as such. Thus providing some view on the rules and interactions of specific values and columns that are less complex than try to construct meaning from the various word that make a sentence, which combined with other sentences make an email.

The PDPs and VECs also provide their own view on the internal workings of the models and its features. PDPs depict how changes in a feature's values correlate with changes in the average predicted outcome. They display both linear and non-linear relationships between the feature and the outcome, disregarding the influence of other variables. Meaning that they show individually how each feature influences the model and the predictions. While VECs show the importance of a feature through the disruption of feature values. Thus providing easy to understand visuals on how a specific feature like alcohol would influence predictions, while VECs show the brevity or importance of the feature in the model. Together both of these visuals provide more comprehensive information than if they were used individually.

The analysis showed the varied techniques' effectiveness in interpretability across different data types - image, text, and tabular data. For image data, techniques like grad-cam highlighted vital regions for classification, while Activation Maximization and similar methods offered somewhat clouded insights into feature importance, partially due to the inherent complexity and unlabeled aspects of image parts. In contrast, text data permitted clearer interpretability through Rule Extraction and LIME, with words providing distinct, understandable node splits and visible attribution scores for elements like spam email identification. However, subtleties in language, like unconventional word choices, presented their challenges. Tabular data, possessing structured yet diverse attribute types, allowed for surrogate model interpretability when attributes were numeric or representable as such. Moving forward, the synthesis of insights from Partial Dependence Plots (PDPs) and Variable Effect Curves (VECs) can provide a robust lens through which feature influence and importance in models are viewed. The following steps involve refining these techniques, addressing identified limitations (such as the interpretability of image features and nuances in text data), and exploring how the amalgamation of multiple methods might furnish more robust and nuanced insights into model interpretability across varied data types. Further exploration into models capable of intuitively handling different data types and techniques that could effectively navigate their respective challenges might also be pivotal.

List of Figures

List of Tables

Bibliography

Russell, S. J., Norvig, P. (2016). Artificial intelligence: A modern approach. Pearson Education Limited.

Mitchell, T. M. (1997). Machine learning. McGraw Hill.

Castelvecchi, D. (2016). Can we open the black box of AI? Nature News, 538(7623), 20.

Gunning, D. (2017). Explainable Artificial Intelligence (XAI). Defense Advanced Research Projects Agency (DARPA).

Holzinger, A., Biemann, C., Pattichis, C. S., Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? arXiv preprint arXiv:1712.09923.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A. (2019). A survey on bias and fairness in machine learning. arXiv preprint arXiv:1908.09635.

Buolamwini, J., Gebru, T. (2018). Gender Shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of the 1st Conference on Fairness, Accountability, and Transparency, PMLR 81, 77-91.

Dignum, V. (2018). Ethics in artificial intelligence: Introduction to the special issue. Ethics and Information Technology, 20(1), 1-3.

Ananny, M., Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. New Media Society, 20(3), 973-989.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... Dean, J. (2019). A guide to deep learning in healthcare. Nature Medicine, 25(1), 24-29.

Zhang, D., Xiao, R. (2019). Machine learning applications in finance. In Financial Signal Processing and Machine Learning (pp. 251-273). Wiley-IEEE Press. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

Shortliffe, E. H., Sepúlveda, M. J. (2018). Clinical decision support in the era of artificial intelligence. Jama, 320(21), 2199-2200.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.

Boddington, P. (2017). Towards a code of ethics for artificial intelligence. Springer.

Buolamwini, J., Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of the 1st Conference on Fairness, Accountability and Transparency, in PMLR 81:77-91 LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.

Chen, J. H., Asch, S. M. (2017). Machine Learning and Prediction in Medicine — Beyond

the Peak of Inflated Expectations. New England Journal of Medicine, 376(26), 2507–2509.

Barocas, S., Selbst, A. D. (2016). Big data's disparate impact. California Law Review, 671.

Pasquale, F. (2015). The black box society. Harvard University Press.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Lipton, Z. C. (2016). The mythos of model interpretability. arXiv preprint arXiv:1606.03490.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1721-1730).

Doshi-Velez, F., Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Holzinger, A., Langs, G., Denk, H., Zatloukal, K., Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(4), e1312.

Goodman, B., Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". AI magazine, 38(3), 50-57.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).

Lundberg, S. M., Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in neural information processing systems (pp. 4765-4774).

Sundararajan, M., Taly, A., Yan, Q. (2017). Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3319-3328).

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado González, A., ... Herrera, F. (2019). Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. Information Fusion, 58. https://doi.org/10.1016/j.inffus.2019.12.012

Wachter, S., Mittelstadt, B., Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harvard Journal of Law Technology, 30(2), 1-37.

Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics, 24(1), 44-65. https://doi.org/10.1080/10618600.2014.907095

Ribeiro, M. T., Singh, S., Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144. https://doi.org/10.1145/2939672.2939778

Lundberg, S. M., Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H. (2015). Understanding neural networks through deep visualization. ICML Deep Learning Workshop, 2015, 1-8.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision, 618-626. https://doi.org/10.1109/ICCV.2017.74

Ghorbani, A., Abid, A., Zou, J. (2019). Interpretation of neural networks is fragile. Proceedings of the AAAI Conference on Artificial Intelligence, 33, 3681-3688. https://doi.org/10.1609/aaai.v33i01.33013681

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. https://arxiv.org/pdf/1706.07269.pdf

Kizilcec, R. F. (2016). How much information?: Effects of transparency on trust in an algorithmic interface. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 2390-2395). https://doi.org/xx.xxxxx [Replace with actual DOI if available]

Dietvorst, B. J., Simmons, J. P., Massey, C. (2015). Algorithm aversion: People erroneously avoid algorithms after seeing them err. Journal of Experimental Psychology: General, 144(1), 114. https://doi.org/10.1037/xge0000033

Doshi-Velez, F., Kim, B. (2016) . Towards a rigorous science of interpretable machine learning. arXiv https://arxiv.org/pdf/1702.08608.pdf

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A. (2016). Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2921-2929).

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 618-626).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Lundberg, S. M., Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (pp. 4765-4774).

Carvalho, D. V., Pereira, E. M., Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. Electronics, 8(8), 832. https://doi.org/10.3390/electronics8080832

Bengio, Y., Courville, A., Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

Zhang, W., Yang, J. (2018). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.

Lipton, Z. C. (2018). The mythos of model interpretability. arXiv preprint arXiv:1606.03490.

Gramegna, A., Giudici, P. (2021). SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk. Frontiers in Artificial Intelligence, 4. https://doi.org/10.3389/frai.2021.752558

Olah, C., Mordvintsev, A., Schubert, L. (2017). Feature visualization. Distill.

Klimt, B., Yang, Y. (2004). Enron-Spam datasets. Retrieved from https://www.cs.cmu.edu/ enron/

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547-553.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan (2020). Explaining machine learning classifiers through diverse counterfactual explanations. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency.

O. M. Parkhi, A. Vedaldi, A. Zisserman, C. V. Jawahar Cats and Dogs IEEE Conference on Computer Vision and Pattern Recognition, 2012

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S. (2019). Counterfactual visual explanations. https://arxiv.org/pdf/1904.07451.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T. (2016). Generating Visual Explanations. [Preprint]. arXiv:1603.08507. https://arxiv.org/abs/1603.08507

Zeiler, M. D., Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. [Preprint]. arXiv:1311.2901. https://doi.org/10.48550/arXiv.1311.2901

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H. (2015). Understanding Neural Networks Through Deep Visualization. [Preprint]. arXiv:1506.06579. https://doi.org/10.48550/arXiv.1506.06579

Zintgraf, L. M., Cohen, T. S., Adel, T., Welling, M. (2017). Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. [Preprint]. arXiv:1702.04595. https://doi.org/10.48550/arXiv.1702.045

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. [Preprint]. arXiv:1912.01703. https://doi.org/10.48550/arXiv.1912.01703

Inglis, A., Parnell, A., Hurley, C. B. (Year). Visualizing Variable Importance and Variable Interaction Effects in Machine Learning Models. Hamilton Institute, Maynooth University; Hamilton Institute, Insight Centre for Data Analytics, Maynooth University; Department of Mathematics and Statistics, Maynooth University. https://arxiv.org/pdf/2108.04310.pdf

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys (CSUR), 51(5), Article 93. https://doi.org/10.1145/3236009