Text

Description automatically generated with medium confidence

UNIVERSITY OF HERTFORDSHIRE

Department of Computer Science

MSc Data Science and Analytics

Data Science and Analytics Masters Project

7COM1075-0509

September 2022

EXPLORING MODEL-AGNOSTIC INTERPRETABLE MACHINE LEARNING MODELS ON THE EXAMPLE OF RED WINE QUALITY ASSESSMENT

Name: Ademola Emmanuel Oyagbohun

Student ID: 20022877

Supervisor: Dr. Justin Mifsud

# Abstract

Machine Learning has gained humongous growth and popularity over the past few decades most especially due to the swiftly growing repositories of data and exceptionally powerful computational power available to use complex algorithms to process these data and obtain meaningful and accurate insights, predictions and patterns, but even this as much of a breakthrough as it sounds, it comes with its own fair share of problems, questions, concerns and doubts which arises especially in terms of interpretability and explainability of the so called algorithms because it is also being applied in fields that there is no room for error like medicine. Questions like “If it is so accurate, how did it arrive at this answer?” are being asked.

So far, there is no generic form of measurement for interpretability, scientists and researchers alike have sought out to make complex algorithms more interpretable by creating interpretability frameworks. This research work investigates this framework’s interaction with machine learning algorithms by finding out if there is a form of relationship or dependency between machine learning model accuracy and local interpretability of these same models.

Table of Contents

[Abstract 2](#_Toc114786865)

[Chapter 1: Introduction 7](#_Toc114786866)

[1.1 Background 7](#_Toc114786867)

[1.2 Problem Statement 8](#_Toc114786868)

[1.3 Research Question 8](#_Toc114786869)

[1.4 Aim 8](#_Toc114786870)

[1.5 Objectives 8](#_Toc114786871)

[1.6 Feasibility 8](#_Toc114786872)

[1.7 Research Structure 9](#_Toc114786873)

[Chapter 2: Literature review 10](#_Toc114786874)

[2.1 Introduction 10](#_Toc114786875)

[2.2 Machine learning Models 10](#_Toc114786876)

[2.2.1 Black box Models 10](#_Toc114786877)

[2.2.2 White box Models 11](#_Toc114786878)

[2.2.3 Grey box Models 11](#_Toc114786879)

[2.3 Explainability in Machine Learning 12](#_Toc114786880)

[2.4 Types of Explanations 13](#_Toc114786881)

[2.4.1 Selected Explanations 13](#_Toc114786882)

[2.4.2 Explanations based on the unorthodox 13](#_Toc114786883)

[2.4.3 Truthful Explanations 14](#_Toc114786884)

[2.5 Interpretability in Machine Learning 14](#_Toc114786885)

[2.5.1 Importance of interpretability in Machine Learning 14](#_Toc114786886)

[2.5.2 Interpretability Methods 16](#_Toc114786887)

[2.6 Evaluation Of Interpretability 19](#_Toc114786888)

[Chapter 3: Methodology 20](#_Toc114786889)

[3.1 Introduction 20](#_Toc114786890)

[3.2 Purpose of research 20](#_Toc114786891)

[3.3 Ethical, Professional, Legal and Social issues consideration and mitigation 20](#_Toc114786892)

[3.3.1 Ethical Issues and mitigation 20](#_Toc114786893)

[3.3.2 Professional Issues and mitigation 20](#_Toc114786894)

[3.3.3 Legal Issues and mitigation 20](#_Toc114786895)

[3.3.4 Social Issues and mitigation 20](#_Toc114786896)

[3.4 Tools Used 21](#_Toc114786897)

[3.4.1 Python Programming Language 21](#_Toc114786898)

[3.4.2 Google Colab Environment 21](#_Toc114786899)

[3.5 Dataset 21](#_Toc114786900)

[3.6 Data Loading 22](#_Toc114786901)

[3.6.1 Data Pre-processing 22](#_Toc114786902)

[3.6.2 Data Normalization 23](#_Toc114786903)

[3.7 Training and Testing Split 23](#_Toc114786904)

[3.8 Machine Learning 24](#_Toc114786905)

[3.8.1 Naïve Bayes 24](#_Toc114786906)

[3.8.2 Decision Tree 24](#_Toc114786907)

[3.8.3 Random Forest 24](#_Toc114786908)

[3.8.4 Neural Network 24](#_Toc114786909)

[3.8.5 Deep Neural Network (1 hidden Layer) 25](#_Toc114786910)

[3.8.6 Deep Neural network (2 hidden Layers) 26](#_Toc114786911)

[3.9 Explainability 26](#_Toc114786912)

[3.9.1 LIME & SHAP 26](#_Toc114786913)

[Chapter 4: Analysis and results 30](#_Toc114786914)

[4.1 Introduction 30](#_Toc114786915)

[4.2 Result collation 30](#_Toc114786916)

[4.3 Result Explanation 30](#_Toc114786917)

[Chapter 5: Conclusion, Evaluation and Future Work 34](#_Toc114786918)

[References 35](#_Toc114786919)

[Appendix A: Gantt Chart 40](#_Toc114786920)

[Appendix B : GitHub Repository 41](#_Toc114786921)

[Appendix C: Source code for analysis 42](#_Toc114786922)

[Appendix D: Source code for result 43](#_Toc114786923)

Table of Figures

[Figure 1 Black Box Model Description 10](#_Toc114740800)

[Figure 2 White Box Model Composition 10](#_Toc114740801)

[Figure 3 Architecture of Grey Box Model 11](#_Toc114740802)

[Figure 4 Recommended products that are frequently bought together. 16](#_Toc114740803)

[Figure 5 Real world ideology of explaainable machine learning. 21](#_Toc114740804)

[Figure 6 Data Loading 26](#_Toc114740805)

[Figure 7 Data Pre-processing 27](#_Toc114740806)

[Figure 8 Data Normalization 27](#_Toc114740807)

[Figure 9 Train Test Split 28](#_Toc114740808)

[Figure 10 Naive Bayes Classifier 28](#_Toc114740809)

[Figure 11 Decision Tree 28](#_Toc114740810)

[Figure 12 Random Forest 28](#_Toc114740811)

[Figure 13 Neural Network 29](#_Toc114740812)

[Figure 14 Deep Neural Network (1Hidden Layer) 30](#_Toc114740813)

[Figure 15 Deep Neural Network (2 Hidden Layers) 31](#_Toc114740814)

[Figure 16 LIME & SHAP Implementation 31](#_Toc114740815)

[Figure 17 randomly chosen rows from test data 32](#_Toc114740816)

[Figure 18 Interpretation for Naive Bayes prediction 32](#_Toc114740817)

[Figure 19 Interpretation for Decision Tree Prediction 32](#_Toc114740818)

[Figure 20 Interpretation for Random Forest Prediction 33](#_Toc114740819)

[Figure 21 Interpretation for Neural Network Prediction 33](#_Toc114740820)

[Figure 22 Interpretation for Deep Neural Network Prediction 33](#_Toc114740821)

[Figure 23 Interpretation for Deep Neural Network with two hidden layers Prediction 34](#_Toc114740822)

[Figure 24 Results data table 35](#_Toc114740823)

[Figure 25 Code to generate scatterplots and regression line 36](#_Toc114740824)

[Figure 26 Scatterplots with line of best fit 37](#_Toc114740825)

[Figure 27 Code to generate correlation & correlation matrix 37](#_Toc114740826)

[Figure 28 Correlation matrix heatmap 38](#_Toc114740827)

# Introduction

## Background

As technology keeps advancing and becoming crucial in every aspect of our lives, autonomous technology has rapidly grown. It has been entrusted to making decisions that were previously assigned to humans, decisions ranging from simple book or movie recommendation that does not have much impact when wrong to complex decisions like loan approval(Carvalho et al., 2019), driving a car (Stilgoe, 2018), flying a plane (Auto-pilot), image recognition (Pak and Kim, 2017), and performing diagnostics (Ahmad et al., 2018) that could cause severe damages or loss of human lives if there is a tiny mistake or misconception. These algorithms are built and trained by humans, and due to how complex they are with many hidden layers, they have proved to be very accurate on sample instances (Ribeiro et al., 2016). However, there is still a catch, which is that these algorithms are very opaque, also called black box (Barredo Arrieta et al., 2020), where the procedure, reason, and motive behind a prediction or classification is a mystery to both the creators and users and only metric indicators such as accuracy, root mean square error (RMSE), recall, precision, and the harmonic mean between precision and recall which is also called the F1 score, are used to validate the results of a prediction of a black box model which is typically not enough.

Because of how these autonomous models decisions can entirely affect human lives, there have been increasingly raising questions on how these models make their decisions which is applicable in different domains, like what features are essential that could alter the result of the model prediction in a loan or mortgage application when a user is denied (Carvalho et al., 2019), can the model be entirely trusted in autonomous vehicles and object recognition (Stilgoe, 2018). Because of the ethical, professional and legal issues in healthcare cases, there is a lack of trust in these models because users' trust is directly impacted by how much they can understand and predict the model's behaviour (Ribeiro et al., 2016).

Therefore, there is a need to ultimately find a way to make these models accountable for every decision it makes and subject them to human supervision to decide if there is something wrong or if they can be trusted with delicate tasks because the modes of operation can be verified to work perfectly under either any circumstance or a particular circumstance while observing due procedures and not in breach of any legal, ethical, professional and social laws and conducts.

**There have been numerous research works that have proposed techniques to help reveal the activities and reasoning patterns behind machine learning models. Examples of these approaches are Linear proxy models, Decision Trees, Automatic rule extraction, and Salience mapping** (Gilpin et al., 2018). Although typically, simpler linear models, which are less accurate, are the most explainable (Angelov and Soares, 2019) but there is still a grey area between model accuracy and interpretability.

In this study, I would set a classification target of 0.65 which translate to a of 65% threshold for classifying good and bad wines. i.e. if wine quality is greater than 65% classify as good wine, otherwise, classify as bad wine . The red wine data set will be normalised and trained on six different machine learning classification algorithms, which are Naïve Bayes classifier, Random Forest Classifier, Decision tree, Neural Network based on multi layered perceptron classifier, ,Deep Neural Network based on Keras Sequential and Dense comprising of one hidden layer and epoch of ten, and lastly, another Deep Neural Network with two hidden layers and an epoch of one hundred and fifty. Their respective accuracy will be derived and noted. LIME and SHAP model-agnostic interpretability methods will be applied to the generated classification outcome. The accuracy of the algorithm and the developed interpretation will be investigated to identify if there is a pattern or an impact of accuracy on the outcome of local interpretability. **Often, interpretability and explainability are used interchangeably; although these two concepts are closely related, there are distinct differences between these two terms. According to** (Barredo Arrieta et al., 2020), **interpretability is defined as the ability to provide meaning in understandable terms to a human, which can further be described as referring to the passive characteristics of a model, denoting any action taken with the intent of clarifying the internal functions of the model. Whereas explainability is represented as an active character of a model,** **indicating any activities taken by a model with the intent of clarifying its internal function.**

## **Problem Statement**

**There has not been a concrete mathematical definition for interpretability or explainability, nor a form of metric to measure interpretability or explainability. However, there have been** **attempts to distinguish these concepts and related concepts such as understandability, transparency, and comprehensibility.** (Linardatos et al., 2020). The solution to this problem has always been to use transparent or white box models such as decision tree rules, additive models, or sparse linear models instead of models that are functionally black boxes such as arbitrary neural networks (ANN) (Ribeiro et al., 2016). There are undergoing researches in making black box models more transparent and this is led advent of interpretability methods and frameworks, but they do not fully unravel the mystery behind these black box models. Perhaps investigating more and understanding these interpretability methods and how they interact with machine learning models can help us make black box models completely transparent.

## Research Question

Is there a relationship between machine learning model accuracy and model agnostic local interpretability?

## Aim

This research work is inspired by the need to gain better insights into the interworking of current local interpretability methods, understand how it makes its interpretations and ultimately investigate if factors such as accuracy can influence interpretation results. This project work aims to cover the grey area of investigating if model accuracy can influence local interpretation.

## Objectives

The achieve the research aim and ultimately answer the research question stated above, the following objectives have been highlighted and will serve as a sequential path to actualize the research aims and answer the research question.

1. Determine an ideal classification target on the red wine dataset to classify good and bad wine.
2. Normalise the data and train machine learning models to predict wine quality from the dataset and classify it into appropriate groups (good or bad).
3. Estimate the performance of the trained machine learning models in terms of accuracy.
4. Apply local model-agnostic interpretability techniques to interpret the behaviour of the trained machine learning models and predictions made for specific individuals.
5. Interpret the results of the model-agnostic interpretability techniques and investigate if there is a relationship between accuracy and local interpretability.

## Feasibility

This research work is entirely conceivable within the stipulated time frame as the research involves investigating and exploration using and tweaking only machine learning tools, libraries and frameworks that are readily open source and not making use of custom algorithms or tools for the analysis. The estimated 600 hours of research work feasible and completely utilised during this research work especially due to limited articles and publications relating to machine learning interpretability.

## Research Structure

Chapter 1: This chapter encompasses what the entirety of the project domain is about, comprising of background, the current issues in this field, the aim and objectives, research feasibility and the research question the thesis hopes to answer.

Chapter 2: This chapter encapsulates the different literatures about machine learning interpretability and explainability, the critical explanation of their differences and similarities, several types and levels of interpretability and how it affects us as humans and the society at large.

Chapter 3: This section of the project involved getting the dataset and other technical hands-on work involved in this project using machine learning tools and libraries whilst considering the ethical, legal, professional, and social implications of this research, analysis and results.

Chapter 4: This chapter focuses primarily on explaining the results of the analysis done in the previous chapter, providing the meanings behind the numbers, table, and plots.

Chapter 5: This chapter summarizes the interpretation of the results, critical reflection of the research work and how it relates to the research gap, evaluation and suggestion on how this research work could be improved and potential areas of future research work.

References: This contains list of all materials that was consulted and cited during this research work.

Appendix A: This section contains the Gantt chart illustrating the activity timeline for this project from start to finish.

Appendix B: This section contains the link to the GitHub repository where all the materials, resources and files used in this project and generated from this project can be found.

Appendix C: This section contains the complete source code used in this project.

# Literature review

## Introduction

Machine learning methodologies have been effectively applied to the fields of science, technology, and business as processing power and data availability has rapidly been on the rise. (Jordan and Mitchell, 2015). Through Decision Support systems, Machine Learning can be vital in helping organisations transform the vast amounts of data at their disposal into reasonable insights and pattern discovery that can be put to use to make managerial decisions and maximise business efficiency and productivity(Bohanec et al., 2021). Analysts should be able to understand the predictions model in order to be able to make decisions based on the data processed by machine learning algorithms more quickly, better, and more reliably in order to make the best of the sophisticated analytics capabilities made possible by machine learning (Ayoub et al., 2021).

Alternatively, Some machine learning techniques offer a feature relevance score for every input parameter that is inherently interpreted. On the other hand, the most complex machine learning algorithms that enable higher performance are referred to as black boxes, as they give consumers predictions without explaining how the predictions were made. This lack of interpretability frequently turns into a lack of trust that prevents the acceptance of machine learning, as well as the potential for it to mask difficulties with justice. (Misztal-Radecka and Indurkhya, 2021), putting this factor into consideration, it can be especially harmful in the context of decision support systems (Ebadi et al., 2019) and other sociotechnical systems (Andras et al., 2018). The study of interpretable machine learning clarifies the relationship between input features and predictions of black-box ML models. Model-specific and model-agnostic strategies for machine learning interpretability can be distinguished. The former, sometimes referred to as ad-hoc approaches, are created for a certain category of models. (Molnar et al., 2020). The latter, usually referred to as post-hoc techniques, may be used in addition to any machine learning model (Gilpin et al., 2018).

## Machine learning Models

### Black box Models

A Black Box Model is a model or a system whose internal mechanisms and functionalities are hidden. In machine learning, black boxes are described as the kind of models that cannot be understood by merely glancing at the attributes and/or features (Molnar, 2022). Also, the term black box is used for labelling all those machine learning models that are mathematically ridiculously hard to explain and to be understood by experts in functional domains (Rudin, 2018). These black-box based models can be grouped into the following categories: based on hyperplanes like those used by the support vector machine (SVM) (Ma and Guo, 2014); inspired by the biological neural networks that constitute animal brains (Momeni et al., 2015); based on probabilistic and combinatory logic, like the probabilistic logic networks (PLNs) (Richardson and Domingos, 2006) and those based on the instances (a.k.a. lazy learning) where the function is only approximated locally, like the k-nearest neighbours (Aha et al., 1991), (Altman, 1992). The opposite of a black box is called the White Box. Despite the fact that not all machine learning models are black boxes, model-agnostic methods for interpretability consider them all as such.

Illustration


Figure 1 Black Box Model Description

### White box Models

The terms white box, understandable model, and explainable artificial intelligence (XAI) are used for labelling all those machine learning models providing results associated with their models that are easy to understand by experts in the application domain. Usually, these models provide a good trade-off between accuracy and explainability (Rudin, 2018), (Ridley, 2022). Usually, the terms understandable and interpretable refer to all those models explained to experts in the application area. However, based on the explanation provided (Rudin, 2018), an understandable model refers to those machine learning needs an additional model or other features for explaining to experts in the application area. On the other hand, an interpretable model can explain to experts without using any additional model.

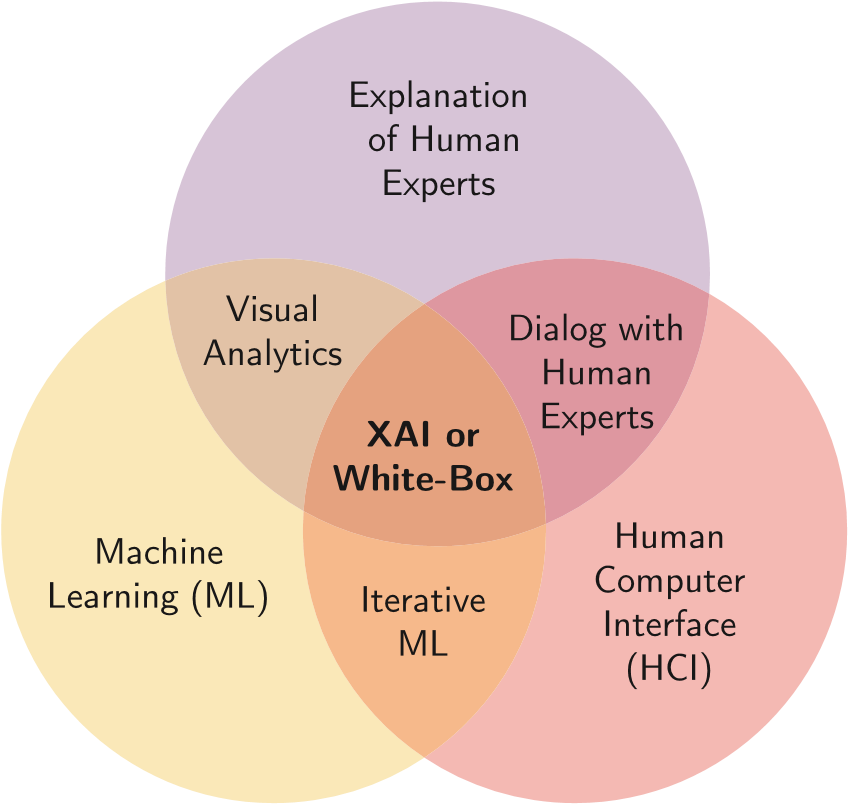


Figure 2 White Box Model Composition (Loyola-Gonzalez, 2019)

### Grey box Models

The distinct combination of white box and black box models results is a grey box model (Bohlin, 2006). A grey box model's primary goal is to create a functional combination of black and white box models in order to mix and reap the benefits of both, thereby creating a more effective global composite model. Generally, any collection of machine learning algorithms with both black and white box models is referred to as a grey box. In recent research, Grau et al. combined White and Black Box machine learning models following a self-labelled methodology to build an accurate, transparent, and interpretable prediction model.

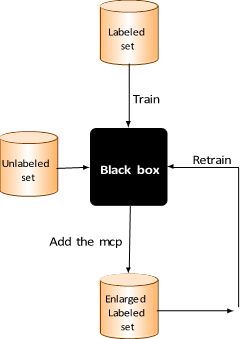
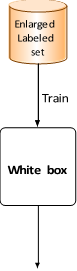


Figure 3 Architecture of Grey Box Model (Pintelas et al., 2020)

The trained model used for predictions

## Explainability in Machine Learning

The cognitive process of describing the results of the operations that make up a machine learning model is referred to as explanation. The explainer can either be human which could be a domain expert or a layperson or a machine trained to perform an explanation (Molnar, 2022).

Explanations are contrastive (Lipton, 1990). Humans sometimes do not ask the exact rationale as to why the outcome of a specific prediction is what it is, or the reason behind an unexpected result, but we tend to be comparative and become intrigued to find out the reason for a certain outcome in contrast to another result. Most time, this situation occurs when the result we aim to achieve is largely deviated from the actual result we eventually got. We often think of hypothetical cases. For example, what would have been the outcome if input a was much larger or smaller?, for the prediction of a home price, a homeowner might be keen to know what factors influenced the price of the house to go higher than they anticipated, or in a case of a loan refusal, the potential borrower might not want to know all the factors that collectively caused the rejection, they would be particular about the specific factors and conditions that need to be changed or improved to make the loan application successful upon reapplying. The gap and unknown between what is currently at hand and what the borrower is aiming for seem to be the main driving force for an explanation. A doctor might ask why a certain drug did not work on a patient; as such, They would require an explanation that compares the results of the patient in their care to another comparable non-responding patient who received the medicine and had success with the medication. Contrastive explanations that compare similar instances are relatively easier to understand than complete explanations (Molnar, 2022). A complete explanation of the doctor's question about why the drug does not work might include: The patient has had the disease for many years, various genes are over-expressed, and the patient's body quickly converts the medication to ineffectual components, whereas a contrastive explanation might be easier to comprehend: The non-responding patient, in contrast to the responding patient, possesses a certain gene combination that reduces the drug's effectiveness. The best explanation emphasises the main differences between the reference item and the object of interest (Molnar, 2022).

What this means for interpretable machine learning is that humans prefer to analyse disparities between a prediction and another instance's prediction rather than seeking a thorough explanation for a prediction which could be a natural or an artificial instance. Because of the need for a point of comparison, developing contrastive explanations depends on the application, and this may be dependent on the data point or instance that needs to be explained and the person receiving the outcome of the explanation. Finding prototypes or archetypes in the data may also be a part of the solution for the automated production of contrastive explanations (Molnar, 2022).

## Types of Explanations

### Selected Explanations

Individuals does not explicitly require clarifications that cover the actuality and entirety of the whole probable causes of an event. We are habituated to choosing a select few causes from various possibilities as the explanation. For example, the increase in the cost of food items can be blamed solely on rising petroleum prices or the decline in stocks of a particular company blamed on a current scandal when there could have been other major contributing factors to these effects.

The Rashomon Effect is the capacity of an event or occurrence to be explained by more than one cause. It is important for models to make good predictions from multiple distinct attributes. The performance of ensemble methods, which include many models with various features and interpretations, is better because the predictions are more reliable and precise when the narratives are averaged, and it also means multiple explanations for a prediction.

For interpretable machine learning, this means making an explanation exceptionally brief and giving minimal causes, even when the actual values are much more complex. With this in mind, the LIME approach performs exceptionally well.

Social explanations: They are a component of dialogue or interaction between the person explaining and the person receiving the explanation. The explanations' nature and content are determined by the social context. For interpretable machine learning, this means focusing on the target audience and the machine learning application's social context. The social component of the machine learning model must be implemented correctly for your particular application, and domain experts from the humanities can assist with recognizing and mitigating any social issues that may arise.

### Explanations based on the unorthodox

We as humans generally concentrate more on unorthodox explanations for occurrences. (Kahnemann and Tversky, 1981). Situations like these have fractional probabilities of occurrence but eventually, happen. Significantly eliminating these abnormal causes can change the entire outcome of an explanation. These unorthodox causes are favourably viewed by people as explanations.

### Truthful Explanations

Realistic validity for good explanations does exist, but alarmingly, this for us is not the most crucial aspect of a valid explanation. For instance, as mentioned earlier, it appears that being selective is more important than being truthful. The list of pertinent factors is covered by an explanation that picks just one or two potential causes. Explanation selectiveness leaves out some of the iotas the truth irrespective of how minimal it might be, just as stated in the example given earlier in selected explanations. For interpretable machine learning, this means that the generated explanation should be very precise and as thorough as possible, which is also referred to as fidelity in machine learning. For humans, the selectivity, contrast, and social component of an explanation are more important than its fidelity.

Good explanations reflect the explainer’s pre-existing views. We, as humans, tend to ignore information that counterfeits beliefs and knowledge that we have previously had. This is known as confirmation bias (Nickerson, 1998). This type of bias does not overlook explanations. Although prior beliefs differ in every individual, also there are shared ideologies too, such as religion, which people tend to disregard or undervalue whenever it is not in accordance with theirs.

For interpretable machine learning, what this means is that it is actually really challenging to include good explanations into machine learning since they are congruent with prior views, which would severely harm prediction performance.

Good explanations are wide-ranging and credible. A relatively general cause that can account for several events may be regarded as a good explanation. This defies the notion that unusual causes produce plausible explanations. I believe that unusual causes outweigh common ones. By definition, abnormal causes are uncommon in the immediate situation and whenever the situation arises when there are no unorthodox events, a generic explanation is thought to be a good one.

## Interpretability in Machine Learning

It is challenging to define interpretability from a mathematical perspective, but alternatively, A non-mathematical definition of interpretability, according to (Miller et al. 2017), is: "Interpretability is the degree to which a human can understand the cause of a decision." Another definition is "Interpretability is the degree to which a human can consistently and independently predict the model's result" (Kim et al., 2016). The more interpretable a machine learning model is, the simpler it is to understand why particular judgments or predictions were made. Ideally, a model is better interpretable than another model if its decisions are more accessible for a human to understand than decisions from the other model that is much more difficult to understand.

Interpretable machine learning is generally the extraction of knowledge from a machine learning model that deals with relationships either contained in data or learned by the model. (Molnar, 2022)

### Importance of interpretability in Machine Learning

Why don't we just accept the model and disregard its reasoning if a machine learning model performs well? The issue is that the majority of real-world jobs cannot be well described by a single statistic, such as classification accuracy (Doshi-Velez and Kim, 2017).

When it comes to interpretability in predictive modelling, there is always a trade-off between knowing just the outcome of a prediction, For example, the probability of a house being burgled into in a certain environment Or knowing the reason why a prediction was made and losing model accuracy in the process of obtaining interpretability. In some circumstances, it suffices to know that the prediction performance on a test dataset was good without caring about the reason why a conclusion was taken. However, in other cases, knowing the reason behind a decision can help you learn more about the problem, the dataset involved and several reasons of factors that may cause the model to fail. A lot of models do not typically need explanations simply because their applications are in harmless situations, where an error wound have little to no major impact, such in a movie recommendation system, or because the method has previously undergone considerable research and evaluation. Interpretability is required because problem formalisation is insufficient (Doshi-Velez and Kim 2017), this signifies that it is not enough for specific problems or tasks to get the prediction. Because a successful forecast only partially resolves your initial problem, the machine learning model is explicitly required to comprehensively describe how it arrived at the prediction. The demand for interpretability and explanations is driven by the following factors. (Doshi-Velez and Kim, 2017).

Scientific discoveries are shrouded in secrecy when opaque machine learning models are applied to study if a machine learning model plainly provides recommendations or decisions without justifications. To promote knowledge and satiate interest in the motivations behind particular predictions or behaviours facilitated by unmanned or smart computers, interpretability and explanations are essential. Humans generally does not, of course, require justifications for every occurrence. The majority of people do not mind if they do not comprehend how computers operate. Unexpected events pique our interest.

Learning and the search for meaning in the world are intimately intertwined. Contradictions or discrepancies between components of our knowledge frameworks should be reconciled. A loan application that is rejected by a machine learning model can come as a complete surprise to the persons applying. They can explain away the discrepancy between what they expected and the eventual outcome by providing some context. The explanations need not entirely explain the circumstance, but they must address the root reason.

Science makes an effort to learn and gather information, yet many issues are solved by using massive datasets and opaque machine learning algorithms. This transition from qualitative to quantitative approaches and machine learning is occurring in many scientific areas. Instead of the data, the model itself naturally transforms to become the source of knowledge. It is possible to retrieve this additional knowledge that the model has captured thanks to interpretability (Molnar et al., 2020).

Machine learning models, by default, incorporate biases from the training set. Machine learning models might then start to bias against underrepresented groups of people on account of this. To find bias in machine learning models, interpretability is a helpful debugging tool. A machine learning model for automatically approving or rejecting credit applications may be biased against a historically marginalised minority, whereas the major objective is to only lend money to those who would eventually pay it back. In this case, because you must not only minimise loan defaults but also refrain from discrimination based on certain demographics, the problem formulation is insufficient. Giving loans in a minimal risk, complying manner is an extra restriction that was not taken into account when the machine learning model was being optimised.

To increase social acceptance, machine learning and algorithms must be incorporated into daily life. People attach thoughts, feelings, intentions, and other things to things.

In order to control social interactions, explanations are used. By establishing a mutual understanding of something, the explainer is able to have an impact on the recipient's behaviour, feelings, and beliefs. A machine could need to influence our emotions and beliefs in order to engage with us. Machines must influence us in order to accomplish their intended task.

Machine learning models cannot be audited and debugged until they are thoroughly understood. Both before and after deployment, and also during research and development, the ability to interpret is helpful., even in low-risk contexts like movie recommendations. An incorrect prediction's interpretation aids in determining its root cause. It provides guidance on how to debug and repair the model. If you are confident that the machine learning model can explain decisions, you can analyse the following traits more easily.

Fairness: ensuring that projections are fair and do not discriminate against underrepresented groups either implicitly or outright. It is simpler for a human to determine whether a decision is based on a learnt demographic for example, gender bias when the decision-making model is interpretable and can explain why it decided that a certain person should not be granted a loan.

Privacy: Ensuring that the data's sensitive information is safely and securely protected.

Reliability: Ensuring that minor input changes do not cause the prediction to shift significantly.

Causality: Ensuring you are picking up just causal relationships.

Trust: It is easier for humans to trust a system that explains its decisions than a black box (Doshi-Velez and Kim 2017).

### Interpretability Methods

#### Algorithm Transparency

The term algorithm transparency is simply the opacity about how a machine learning algorithm trains a machine learning model from the available dataset and also illustrate any other kinds of relationships that can be learned from it (Carvalho et al., 2019). This understanding of the algorithm's fundamental workings pertains to predicted output rather than the particular model that was learned. Only domain knowledge of the machine learning model itself, and not just knowledge of the dataset or trained model, is necessary for algorithm transparency. (Molnar, 2022).

#### Global, Holistic Model Interpretability

This tends to answer the question of how a trained model makes predictions as a whole.

If you can fully understand a model simultaneously, you can say it is interpretable (Lipton, 2017). To explain the global model output, it is required to have the trained machine learning model, good domain knowledge of algorithms and the dataset used in training the model. Recognizing how a machine learning model makes all the decisions based on a holistic perspective of its attributes is necessary for this level of interpretability as well as the learned components, such as weights, other parameters, and structures. Which features are paramount, and what kind of relationship occurs between them? Global model interpretability aims to create an understanding of the distribution of the target outcome based on the feature's importance. Global model interpretability is an incredibly challenging procedure to attain. Typically, Any machine learning algorithm with more features than a handful will probably be too confusing to accommodate the short-term memory of a person. When we attempt to understand a model, we generally put just certain factors into account, such as the weights in linear models.(Molnar et al., 2020).

#### Local Interpretability For One Prediction

This involves diving deeper into a single instance, examining what the model predicts for this instance, and explaining the reason behind the prediction. Considering a single predictive class, the behaviour of the complex model might behave satisfactorily (Molnar et al., 2020). The prediction might locally solely depend linearly on a few features rather than complex dependence on them. For instance, a house's worth might not be linearly correlated with its size. However, if you are concentrating on a single 100-square-meter house, there is the likelihood that your model projection for that data subset varies linearly on the size. By simulating how the anticipated price changes when the size is either increased or decreased by 10 square metres. Therefore, this makes local explanations more accurate most of the time than global explanations.

#### Local Interpretability For Multiple Predictions

Both individual instance explanations, as well as techniques for global model interpretation, can be used to explain model predictions for multiple instances. This occurs by treating the group of instances as a complete dataset then executing global interpretation to the dataset. The particular explanation technique can be applied to every attribute before averaged for the whole group. (Molnar, 2022).

#### Model Agnostic Methods

Distinguishing explanations from the machine learning model, i.e., model-agnostic interpretation methods, has some advantages (Ribeiro et al., 2016). Their flexibility is the most significant advantage of model-agnostic interpretation techniques over model-specific ones. Machine learning developers have the flexibility to implement any machine learning model they like in their work since a generic interpretation method can be applied to any model to still accomplish the same result. A graphical user interface, for example, is one example of something that relies on the interpretation of a machine learning algorithm and eventually becomes independent of the underlying model. (Molnar, 2022). Typically, when completing a task, many distinct types of machine learning models are often considered. Model-agnostic explanations are simpler to utilise when comparing models in terms of interpretability since they may be used in any model.

Utilizing solely interpretable models is an alternative to model-agnostic interpretation approaches. However, this frequently has the considerable drawback of losing predictive performance in comparison to other machine learning models and limiting your model selection. Utilizing model-specific interpretation techniques is an additional choice. This has the problem of limiting you to a single model and switching to another model will be difficult.

Model flexibility: This simply means an interpretation method can work with virtually any machine learning model, from simpler models like decision trees and naive Bayes to more complex models.

Explanation flexibility: This means that there is no limitation to the form of explanation that can be generated.

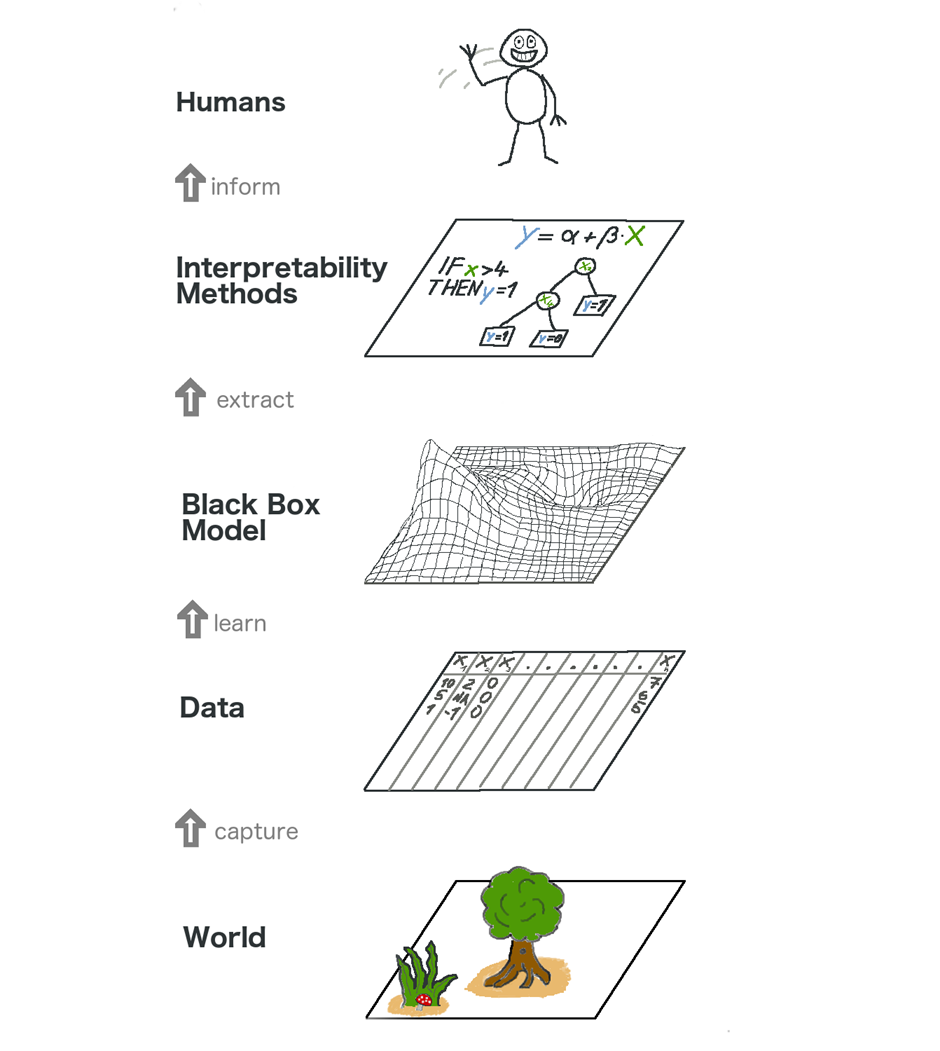
Representation flexibility: This indicates that the explanation system may use a different feature representation from the model it is describing (Molnar, 2022.

Figure 5 Real world ideology of explainable machine learning (Molnar, 2022).

The first layer is the world. Which is everything that we can see, feel and touch. It is typically represented nondigitally, it could be anything like a puddle of water or a stray cat walking on the roof. This comprises of natural things and data can only be derived at this layer through perception or observation.

The second layer is the Data layer which comprises anything that has computer encoding ranging from images, texts, audio or tabular data. Many of these data are generated from taking measurements and recordings from the world layer. The transition between world and data level involves the conversion of nondigital perceptions to digital readings. The activities that takes place at this stage sometimes is categorized as the of the most vital stages if not the most vital, because no scientific or mathematical observations and insights can occur in the first instance if there is no way to acquire the necessary data.

We get the Black Box Model layer by executing computer aided complex analytics on the data from the data layer, and these analysis helps to gain deeper insights of the environment around us, discover previously hidden patterns and finally make predictions about the unknows.

The Interpretability Methods layer sits above the Black Box Model layer and aids in dealing with the opaque nature of machine learning models, thus providing answers to unclear outcomes of predictions or decisions.

A Human occupies the last layer, and this involves a human, either a domain expert or not, putting the derived explanation to use. This is the ultimate endpoint of any machine learning model, and these human use of the insights derived from the models could be used to make managerial decisions and in the process improving human standard of life ehen put to good use.

## Evaluation Of Interpretability

There has not been any valid general agreement about the actual or mathematical definitions of interpretability in machine learning, and neither has there been a specific metric to measure its performance or truthfulness. However, There has been some preliminary research around this and an effort to develop some methods for evaluation, as described below (Doshi-Velez and Kim, 2017).

Application level evaluation: This process simply involves domain experts personally testing the algorithm with either real world scenarios or hypothetical scenarios to examine the algorithm’s decision making process, prediction or course of action. The standard for determining this is typically dependent on how well a person would be able to justify the same choice.

Human level evaluation: This is rather an abridged interpretation. The difference between these experiments is that they are not conducted with domain experts but with laypersons. This makes experiments typically cheaper and much easier to find volunteers for testing. This form of evaluation is typically dependent on feedback to examine its level of performance.

Function level evaluation: At this level of evaluation, humans are generally not required. The process works in a way that the model would have been initially evaluated by a human at the human level, then at this stage the model uses a form of reinforcement leaning to actually evaluate itself whenever required using the human level evaluation metrics as a threshold.

# Methodology

## Introduction

This chapter highlights the processes and methods that were adopted in this research work while also considering the likely implications of this research work. Interpretability in machine learning is still a grey area, with a lot of research work still going on. In this section, I aim to accomplish my research objectives by training six machine learning algorithms on the dataset, then using the trained model to perform predictions on the test data and generate accuracy. Furthermore, use LIME and SHAP local interpretability on five random rows of the prediction data to discover insights behind predictions, collate results and compare them against accuracy and ultimately provide an answer to my research question.

## Purpose of research

As stated earlier in this study, the primary purpose of this research work is to investigate and establish if algorithmic accuracy naturally plays a part in local model-agnostic machine learning interpretability or not using the red wine dataset.

## Ethical, Professional, Legal and Social issues consideration and mitigation

### Ethical Issues and mitigation

**The ethical issue related to this work that this research project hopes to mitigate is classified under Risk Zone 7 of the EthicalOS Toolkit: Implicit Trust and User Understanding and Risk Zone 4: Machine Ethics and Algorithmic Bias, as this project** **researches to better understand the interpretability of models and help curtail bias and discrepancies of machine learning models by investigating if model accuracy influences interpretability results.**

**Moreover, I am fully aware of the university procedure regarding ethics in a research project. I acknowledge that I dd not involve any human subjects or made use of any human-related data, be it personal or sensitive, thereby exempting myself from applying for ethics approval.**

### Professional Issues and mitigation

**Using the BCS code of conduct, there is no specific breach of professional conduct. Nonetheless, I am constantly making sure I do not breach any conduct, especially BCS Principle 2:** PROFESSIONAL COMPETENCE AND INTEGRITY(The British Computer Society Code of Conduct, 2016)

### Legal Issues and mitigation

**As human subjects or data are not in use in this project, there are no legal issues, nor are there any breaches of privacy and discrimination as stated in the laws listed below:**

* **Data Protection (GDPR 2018),**
* **Racial Discrimination (Civil Right Acts of 1964)**
* **Sex Discrimination (Equal Pay Act of 1963,** Civil Right Acts of 1964)
* Citizenship Discrimination (Immigration and Reform Control Act)
* Age Discrimination (Age Discrimination Employment Act of 1967)
* Disability Status Discrimination (Rehabilitation Act of 1973)

Intellectual theft likewise collusion is also a major legal challenge especially when performing research such as this, to mitigate this issue, all materials used in this project are duly referenced and acknowledged and I attest that this research wok is original and singlehandedly done my myself alone.

### Social Issues and mitigation

**Social dilemmas exist in most cases where individuals, organizations and society are in a relational triangle around decisions that affect the society at large** (Strümke et al., 2021).

Social issues are remarkably similar to consequentialist ethical issues, where the interest of the larger society. These issues typically affect developers and organizations when the best outcome for everybody would be achieved if there is a specific pattern of behaviour. Implementing these behaviours can lead to certain drawbacks like a breach of legislation or a professional code of conduct. Unlike ethical, professional, and legal issues, there is no metric to classify social issues, which can lead to a severe social dilemma, especially for more prominent organizations.

This project does not pose any form of social issue because it does not contain any human subject nor have a targeted result at a group of people or individuals.

Although, the result of this research project might pave the way for further research that might be applied to humans to help mitigate social issues.

## Tools Used

### Python Programming Language

Python programming language is the primary programming language that the technical aspects of this research work is based on. This programming language was chosen because it is a high level, general purpose programming language with vast majority of machine learning and artificial intelligence algorithm as well as very efficient statistical and analytical frameworks that makes the tasks of data pre-processing, preparation, normalisation and visualisation less tedious and more efficient. Also, python possesses the ability to efficiently deal with enormous amounts of data and running complex algorithms while using as little computational power as possible. All these features make python a very suitable programming language for this research work.

### Google Colab Environment

Google Colab is a notebook environment created by Google Research in 2017. It is a free cloud based notebook runtime environment that allows users run python codes. This platform was chosen as the runtime environment in executing the codes and algorithms in this research work because it eliminates any redundancy that may arise due to low computational capacity, it requires no setup on a local machine, provides automatic backup of files in case of unintended deletion or loss, and most importantly, it contains a repository of almost any algorithm or framework without having to manually set it up which makes it the choice for executing the codes associated with this research work.

## Dataset

The dataset used in this research work is related to red vinho Verde wine samples from the north of Portugal. The dataset is available publicly for research purposes and was obtained from [UCI Machine Learning Repository: Wine Quality Data Set](https://archive.ics.uci.edu/ml/datasets/wine+quality) and contains the following attributes:

Input variables (based on physicochemical tests):  
1 - fixed acidity  
2 - volatile acidity  
3 - citric acid  
4 - residual sugar  
5 - chlorides  
6 - free sulfur dioxide  
7 - total sulfur dioxide  
8 - density  
9 - pH  
10 - sulphates  
11 - alcohol  
Output variable (based on sensory data):  
12 - quality

This dataset consists of 1599 rows and 12 columns. [Cortez et al., 2009].

## Data Loading

To successfully perform the research work, the dataset first needs to be loaded into the Google Colab notebook environment, for this loading to take place, some python libraries have to be imported to enable the processing of the dataset which is in csv format into a data frame.

The required libraries were imported but Python Pandas library specifically does the job of loading the data into a data frame which makes it easier to index, manipulate and pre-process within python. The code to import the libraries and load the dataset as well as a quick glance of the first few rows of the dataset is shown below.

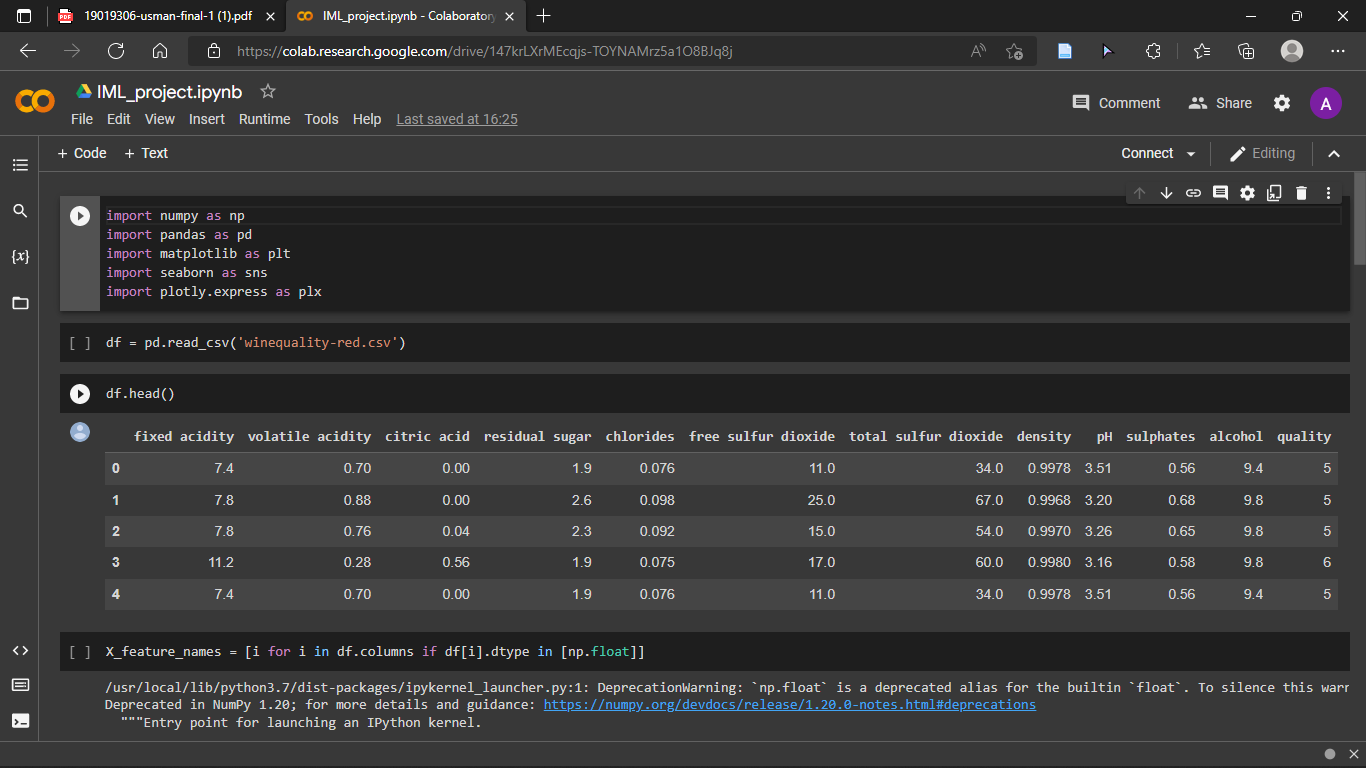


Figure 6 Data Loading

### Data Pre-processing

Due to the nature of the dataset and what I am trying to achieve in this research work, data pre-processing needs to be done. This process is simply done to transform the dataset into a state whereby it is well structured, complete, without any errors or missing values and ultimately suitable for the kind of analysis to be performed. On this dataset, there were no missing values, and it was well structured, the only prepossessing task that was done on this dataset was to distinguish the input and output variables and separate them since this research is based on prediction. The image below shows the code that was used for the pre-processing.

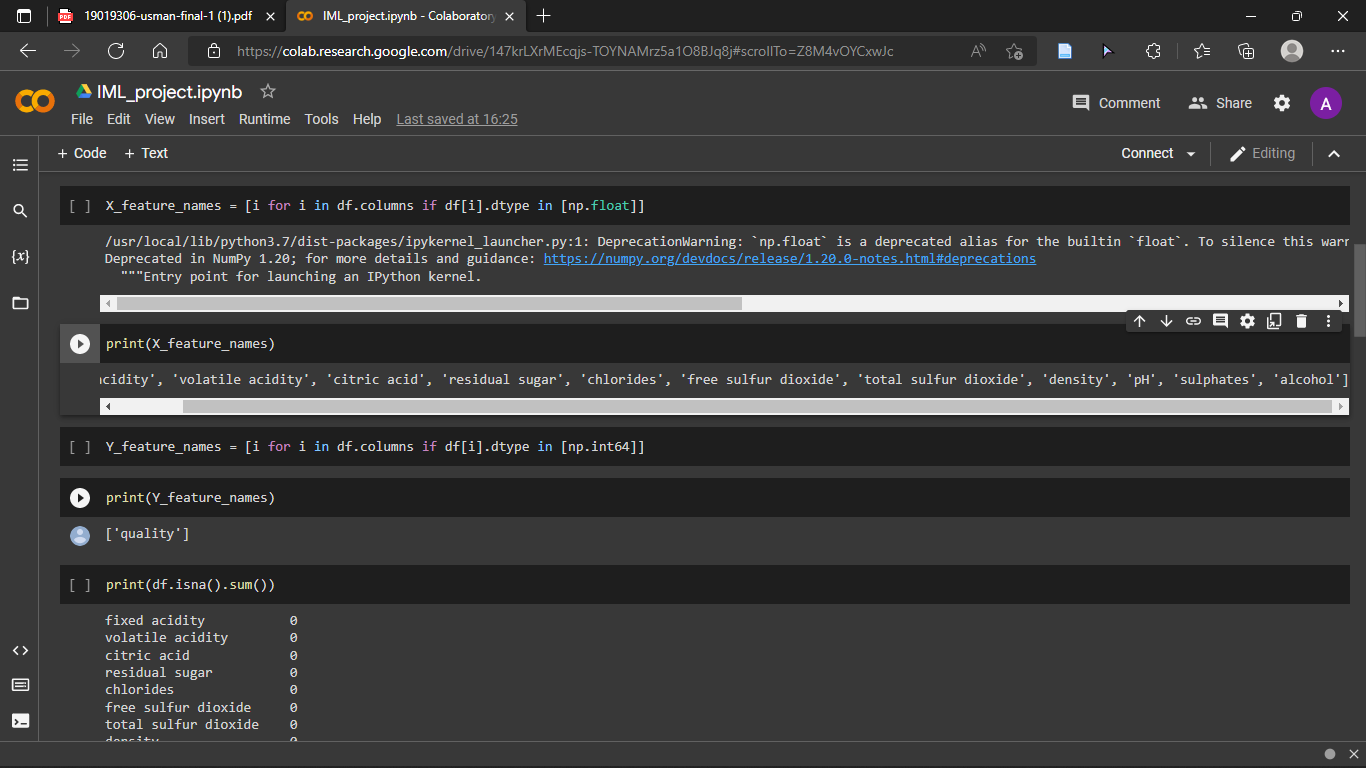


Figure 7 Data Pre-processing

### Data Normalization

The aim of data normalization is to effectively reduce redundancy and standard deviation without losing any information. This process also helps to eliminate bias when performing analysis. In this section before normalizing, I set a threshold of 6.5 for classifying good or bad wine based on quality, then proceeding to normalize the data thereby making it ready for model training and classification.

A screenshot of a computer

Description automatically generated

Figure 8 Data Normalization

## Training and Testing Split

This feature is especially important when training a machine learning model, and it involves splitting the data randomly and using a larger chunk of it to train models and the other part to test the trained models and see how they perform. In this research, I used a 70:30 ratio, that is, 70% of the whole normalized data for training the model and the remaining 30% for testing the model performance. See the image below for the code used to perform the training and testing split.

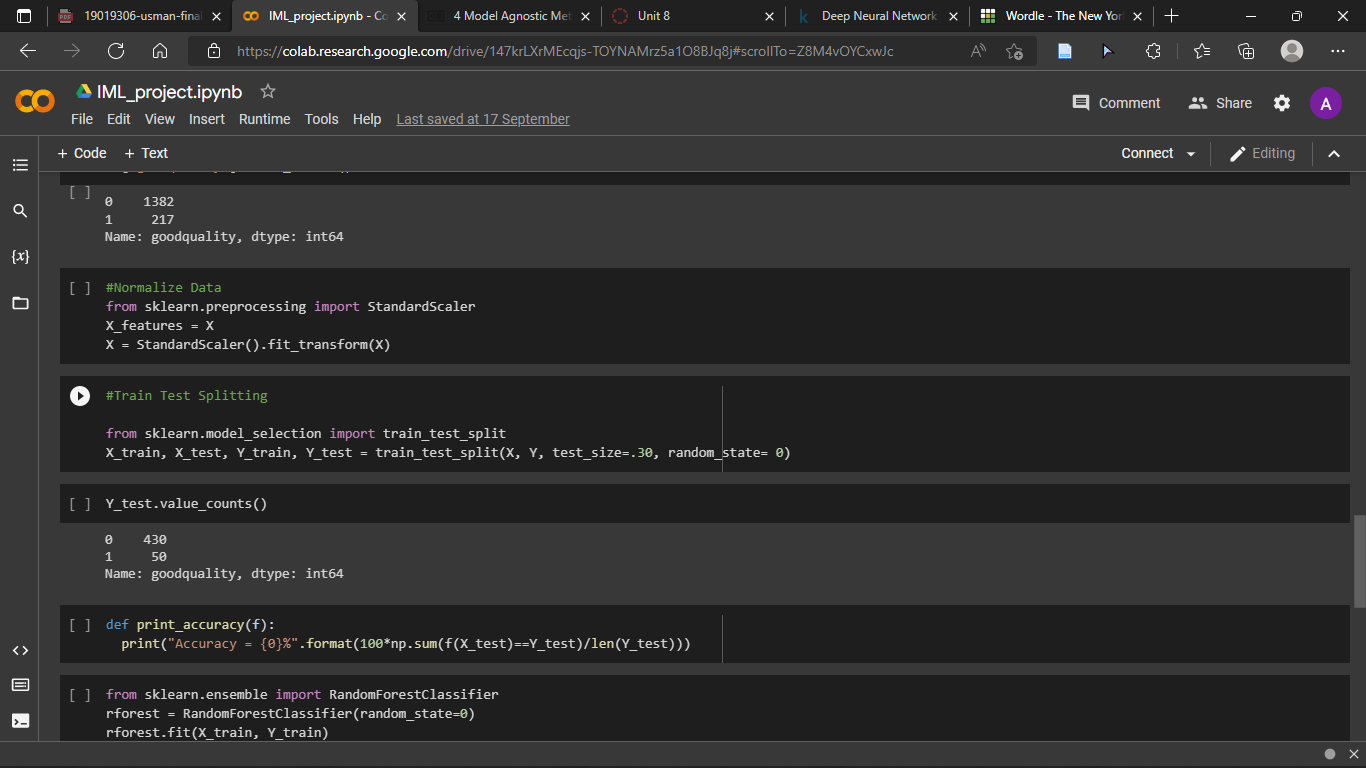


Figure 9 Train Test Split

## Machine Learning

In this section, I trained six algorithms in order of complexity on the already normalized dataset and obtained their performance accuracy.

### Naïve Bayes

Naïve Bayes classifier was trained on the dataset, the classifier was used with its default parameter values and no changes were made.

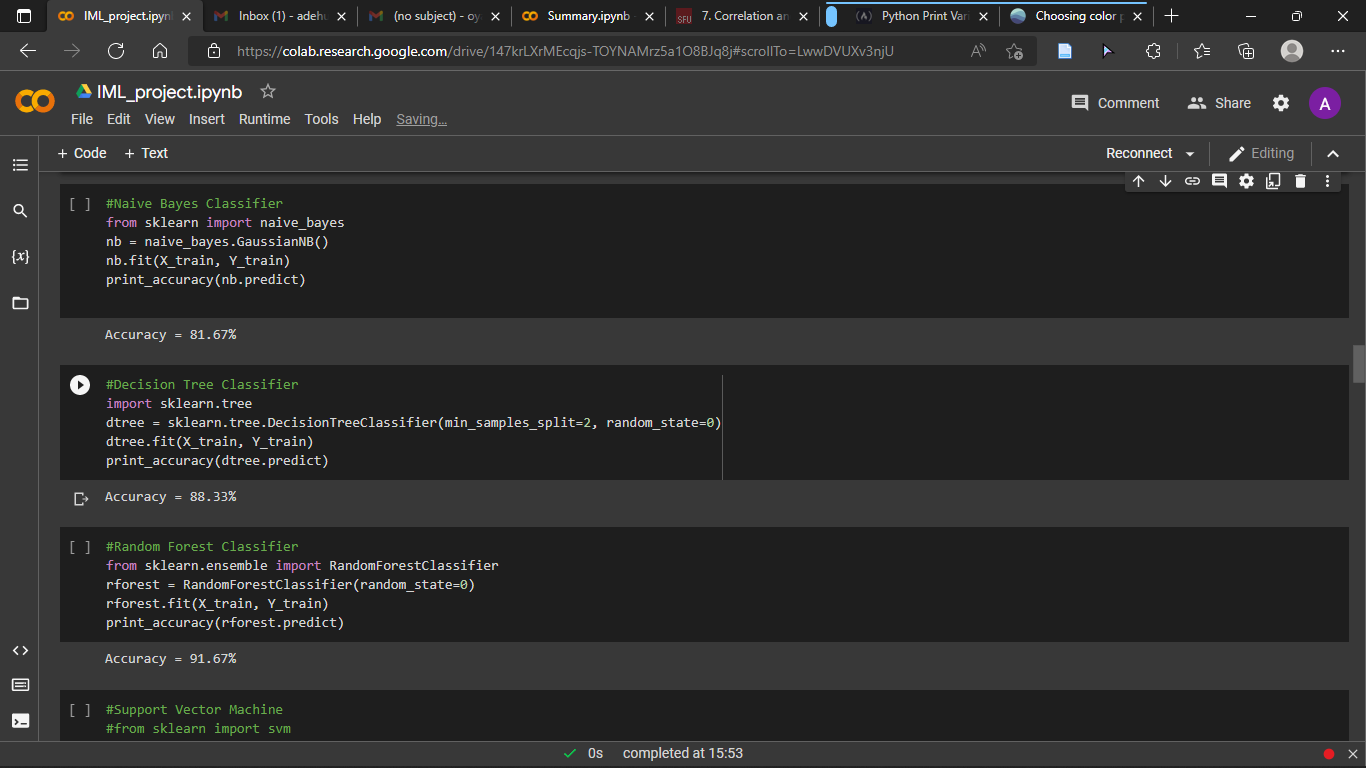


Figure 10 Naive Bayes Classifier

### Decision Tree

The Decision Tree algorithm was used in training the dataset, the parameter min\_sample\_split is set to two to ensure that there are at least two samples required to split an internal node, and random\_state is set to zero because the normalised data had already been randomised. The prediction accuracy was also generated.

A screenshot of a computer

Description automatically generated

Figure 11 Decision Tree

### Random Forest

Random Forest Classifier was applied in training the dataset, there were no parameter changes as the default values were used except random\_state which had been applied to the normalized data prior to classification and the accuracy was generated.

A screenshot of a computer

Description automatically generated

Figure 12 Random Forest

### Neural Network

This is a simple neural network model from MLPClassifier which is Multi-Layer Perception Classifier which basically relies on an underlying neural network to perform classification tasks. The following parameters were set for this model:

Activation = relu

Hidden\_layer\_sizes = (100,2)

Max\_iter = 20000

Random\_state = 40

Learning\_rate = constant

See the figure below for the code to execute the MLPclassifier neural network

A screenshot of a computer

Description automatically generated

Figure 13 Neural Network

### Deep Neural Network (1 hidden Layer)

This is another neural network that is more structured than the MLPclassifier, where Keras dense and sequential are used to manually structure and define the whole neural network model as well as each layer of the model. I made use of only one hidden layer sigmoid activation in this model. While compiling and fitting the model, I opt to use the following parameters:

Loss = binary\_crossentropy

Optimizer = adam

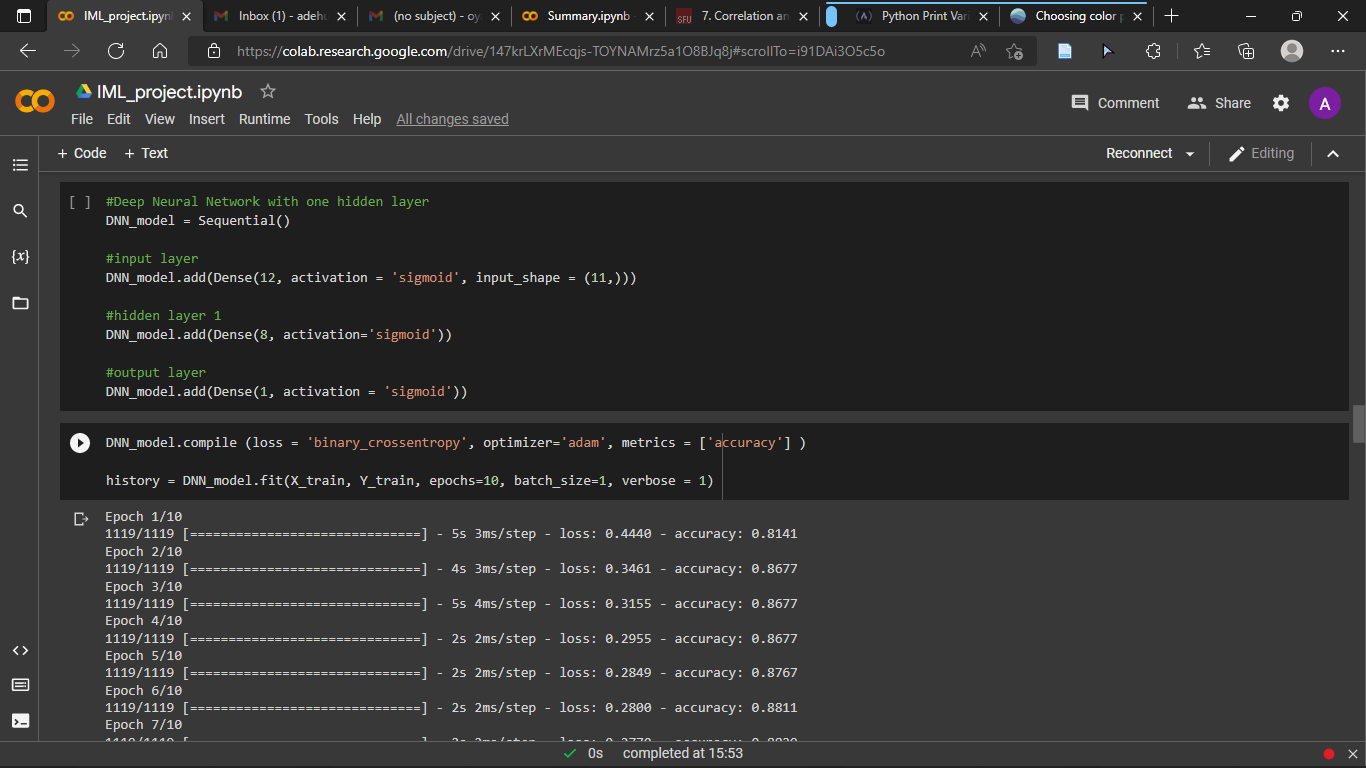
Metrics = accuracy

Epochs = 10

Batch\_size = 1

Verbose = 1

See the figures below for the code to execute the Deep Neural Network



A screenshot of a computer

Description automatically generated

Figure 14 Deep Neural Network (1Hidden Layer)

### Deep Neural network (2 hidden Layers)

This is almost the same as the deep neural network above, and the only difference is that this model contains two hidden layers and I used relu activation for the input and hidden layer while using sigmoid activation for the output layer in this model. While compiling and fitting the model, I opt to use the following parameters:

Loss = binary\_crossentropy

Optimizer = adam

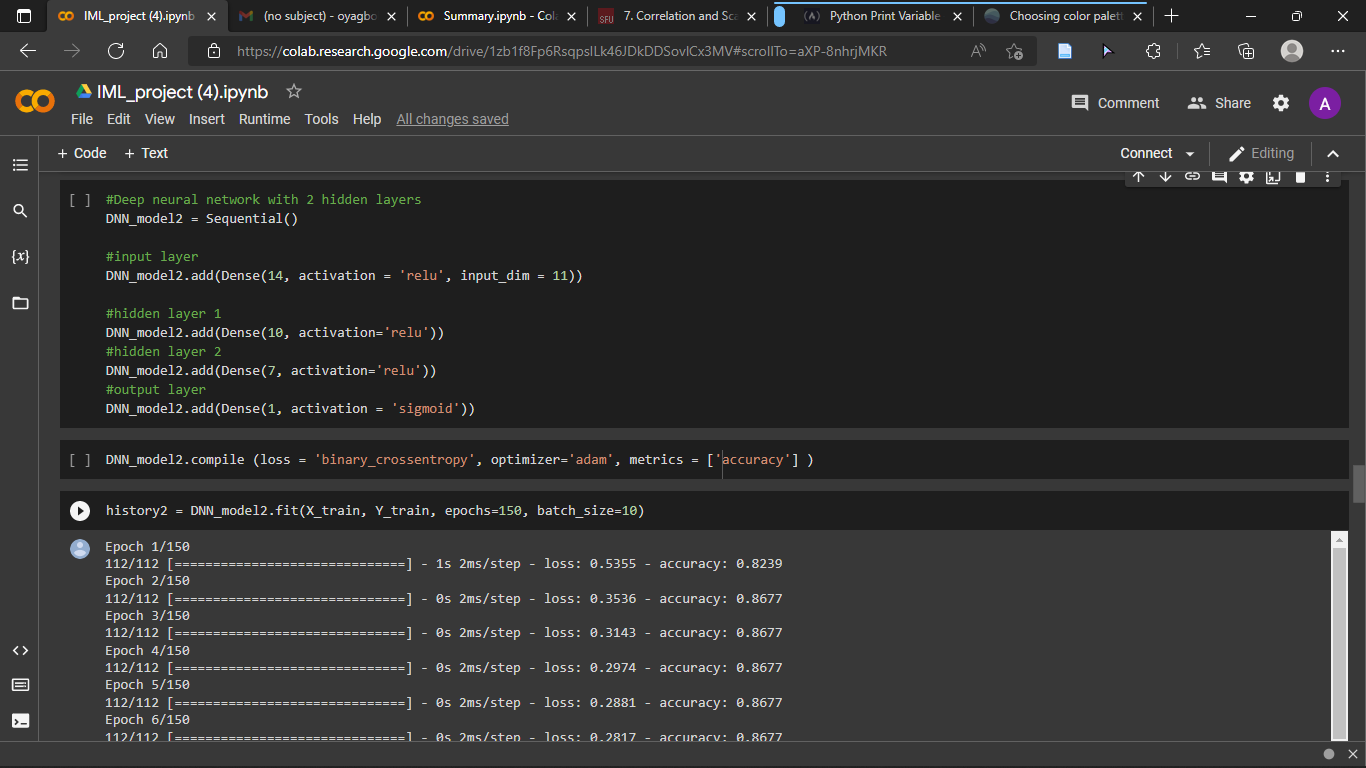
Metrics = accuracy

Epochs = 10

Batch\_size = 1

Verbose = 1

See the figures below for the code to execute the Deep Neural Network



A screenshot of a computer

Description automatically generated

Figure 15 Deep Neural Network (2 Hidden Layers)

## Explainability

### LIME & SHAP

LIME – Local Interpretable Model-agnostic Explanations is a machine learning interpretability model whose goal is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier (Ribeiro et al., 2016). SHAP – Shapley Additive exPlanations also shares the same goal as LIME which is to establish an interpretable explanation for classification models. In this research, I applied LIME and SHAP to the different prediction models initially generated to provide an explanation of the predictions.

The randomly chosen rows in the predicted data are rows 10, 59, 82, 120, and 200. I used the LIME and SHAP interpretability methods to analyse the predicted results for these rows of data on each algorithm to better understand the metrics behind the predictions and to enable me further the research. The figures below show the step-by-step implementation of LIME and SHAP on each algorithm for each of the five selected rows of data.

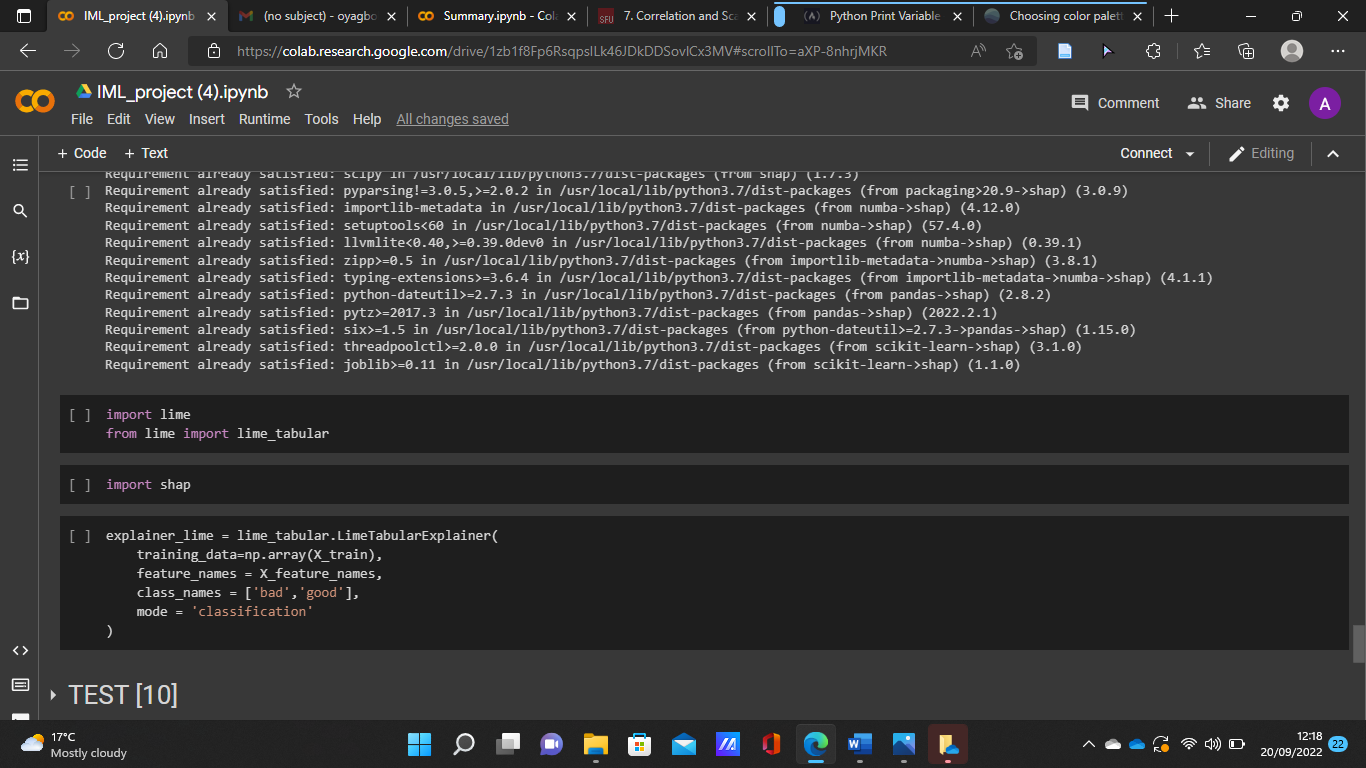


Figure 16 LIME & SHAP Implementation

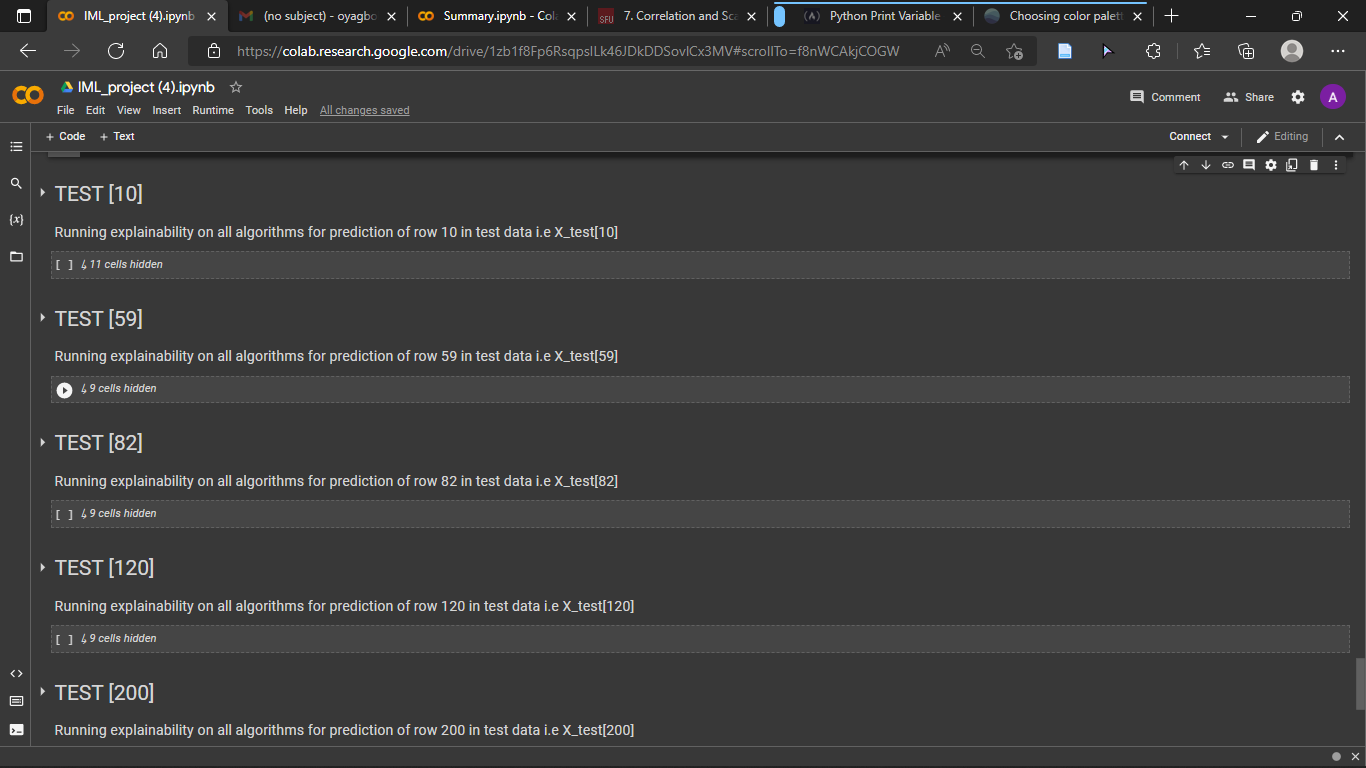


Figure 17 randomly chosen rows from test data

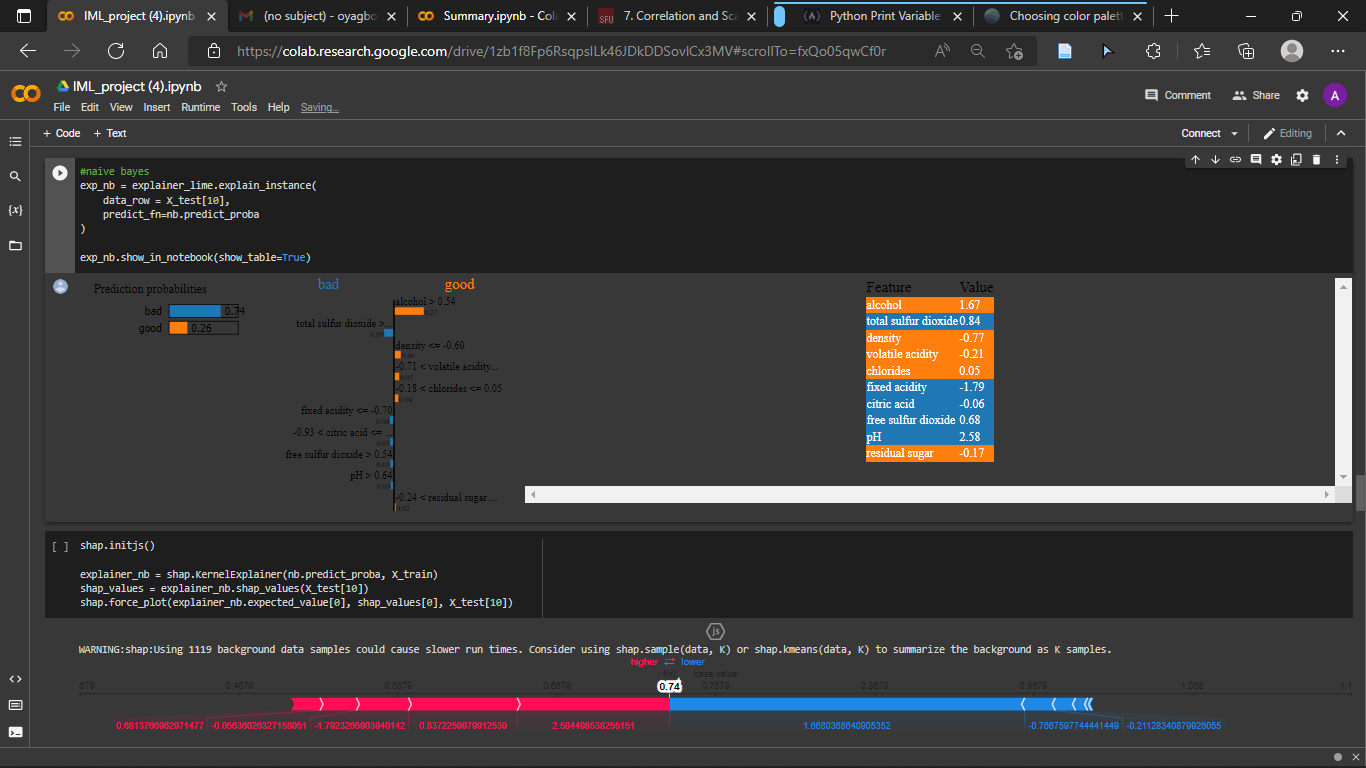


Figure 18 Interpretation for Naive Bayes prediction

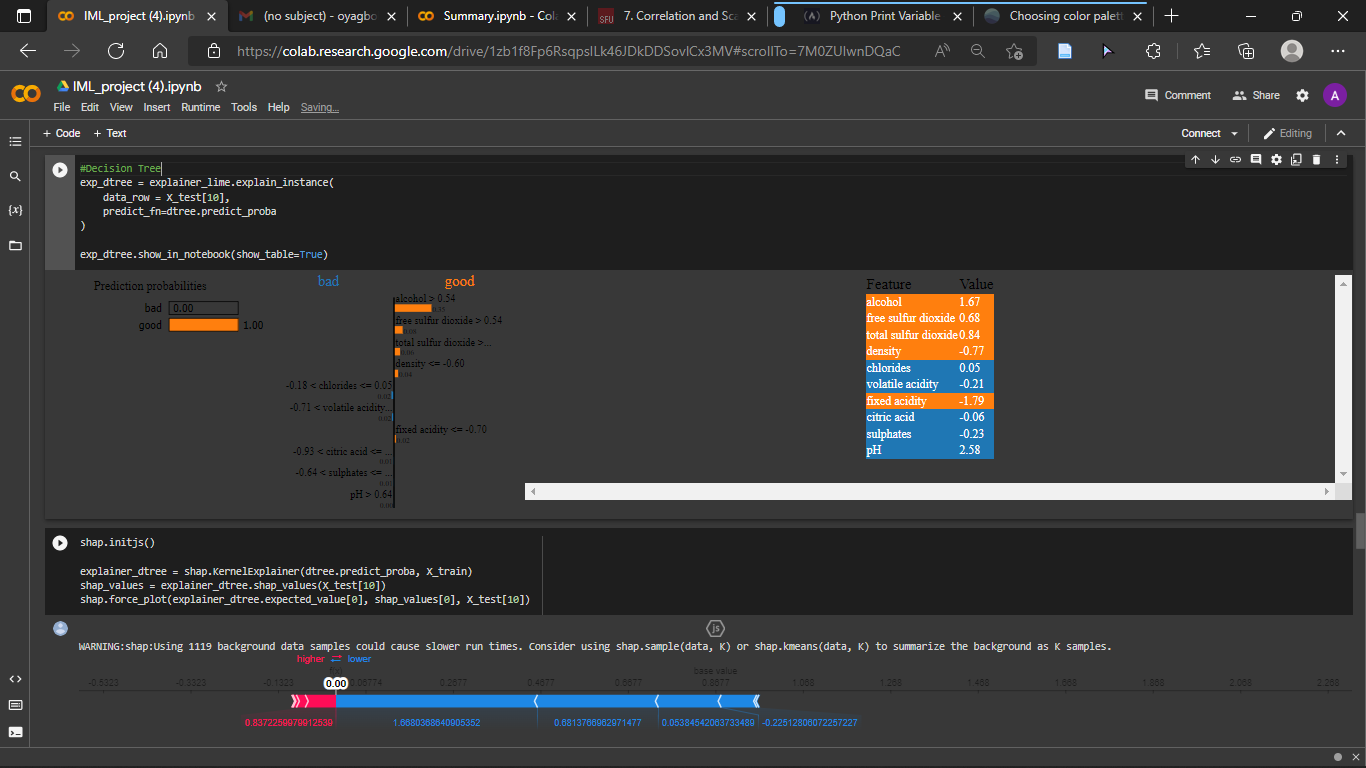


Figure 19 Interpretation for Decision Tree Prediction

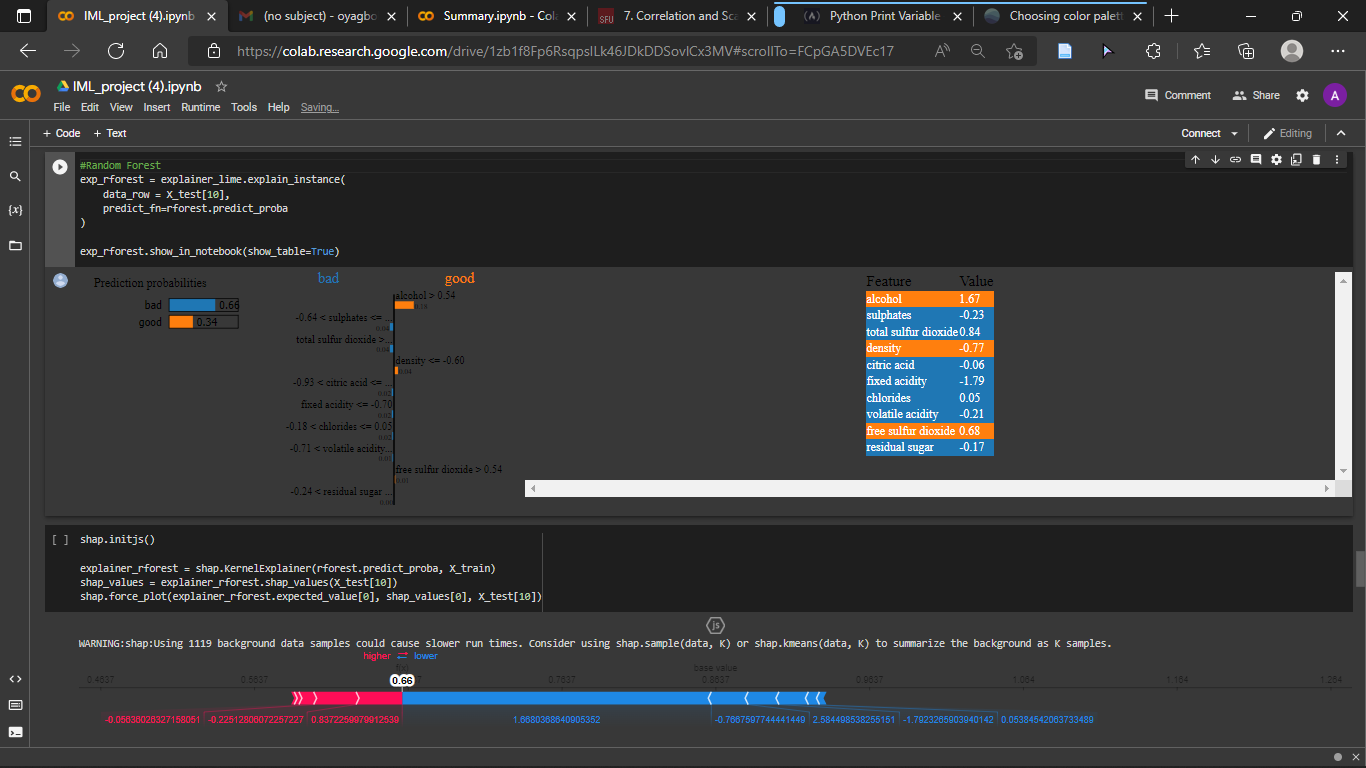


Figure 20 Interpretation for Random Forest Prediction

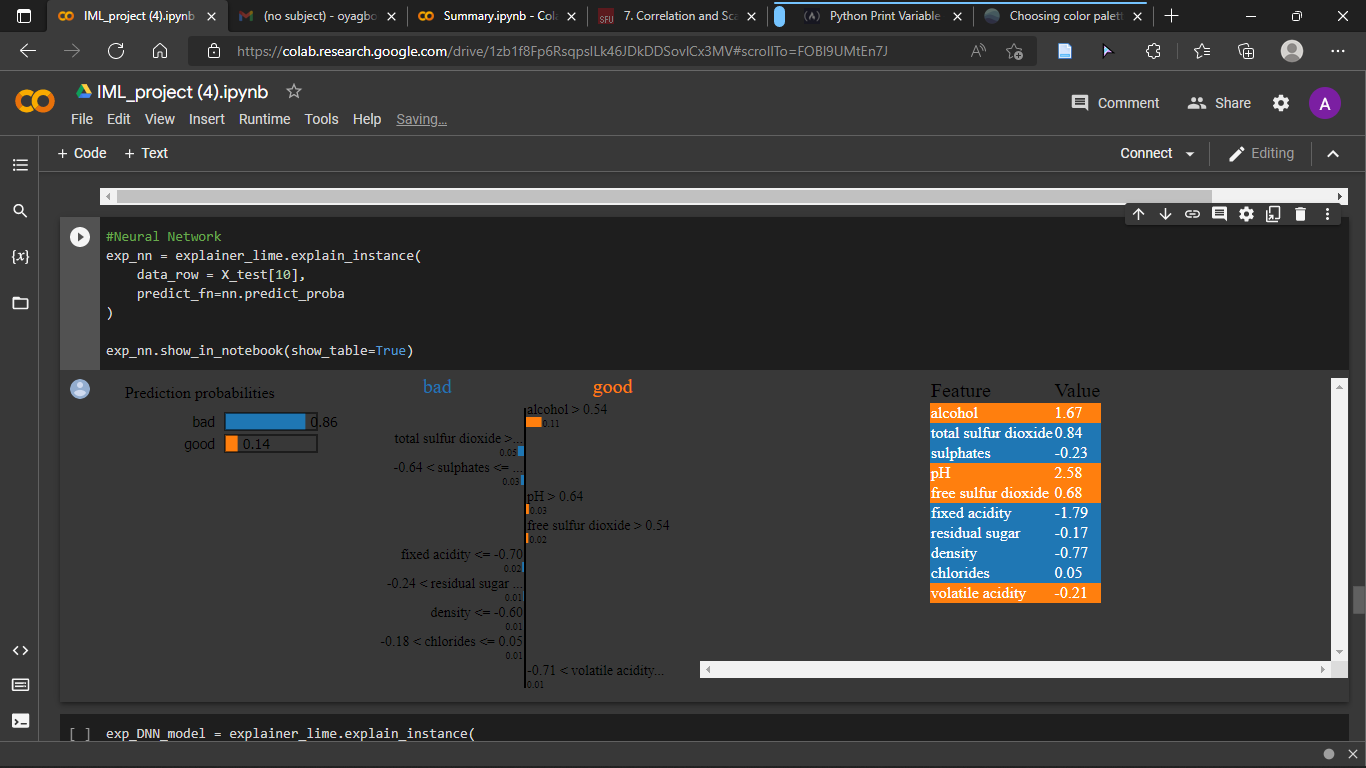


Figure 21 Interpretation for Neural Network Prediction

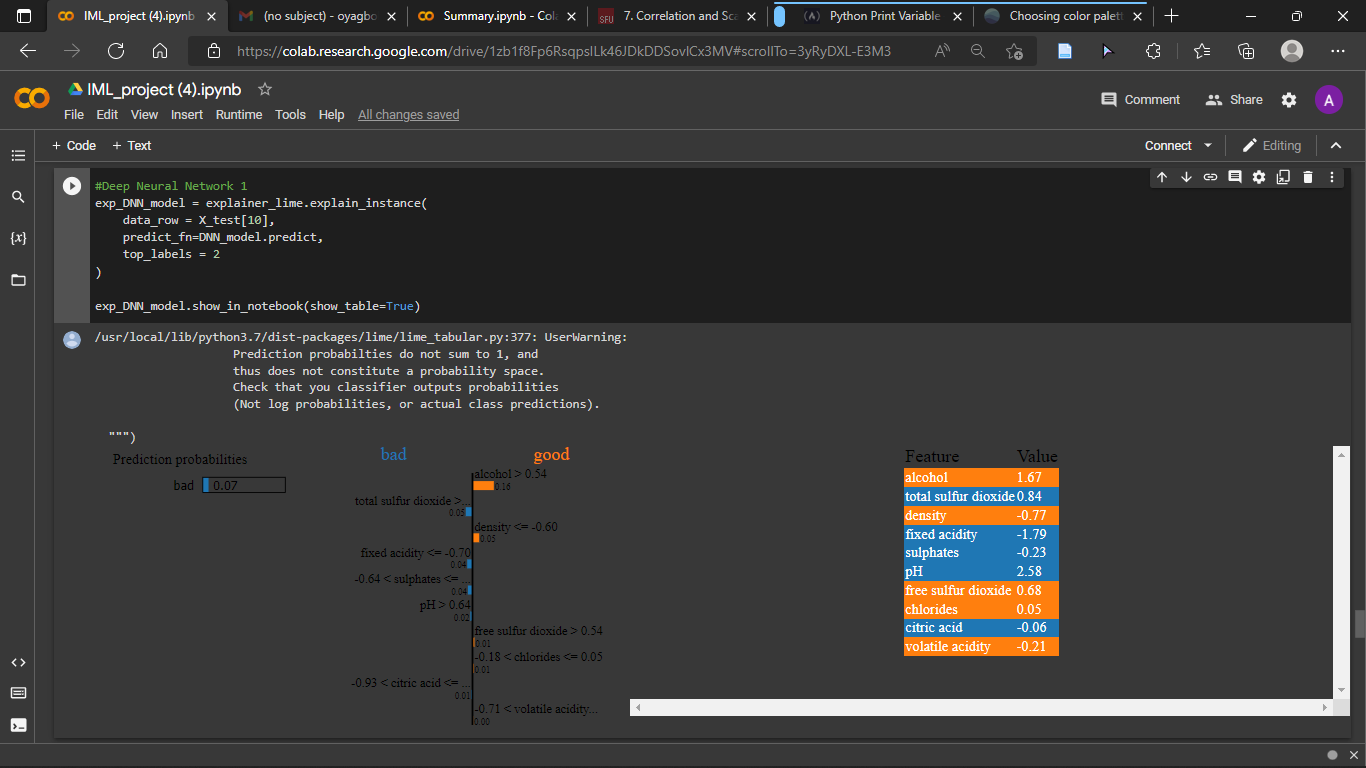


Figure 22 Interpretation for Deep Neural Network Prediction

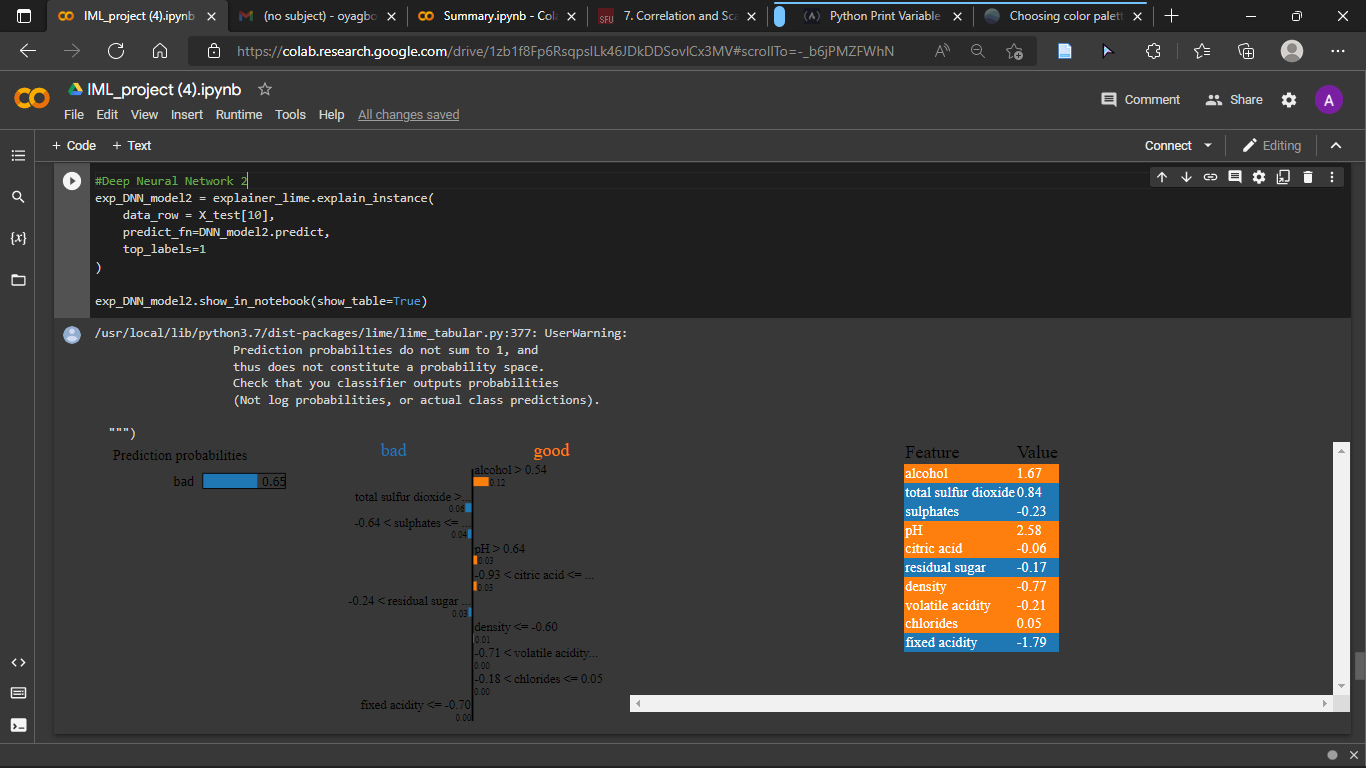


Figure 23 Interpretation for Deep Neural Network with two hidden layers Prediction

# Analysis and results

## Introduction

This section of this research work is dedicated to demonstrating the data analysis performed in retrospect to achieving the research aim and providing an answer to the research question, explaining the results and presenting appropriate visualisation to depict the analysis done.

## Result collation

After executing the interpretability methods LIME and SHAP on the trained models for the randomly selected data points, the results were compiled into a csv data table. The data content of the table includes model accuracy and prediction of how good the wine in each selected data point is for each of the algorithms.

0 - Naïve Bayes classifier

1 – Decision Tree classifier

2 – Random Forest classifier

3 – Neural Network (MLPClassifier)

4 – Deep Neural Network (1 hidden layer)

5 – Deep Neural Network (2 hidden layers)

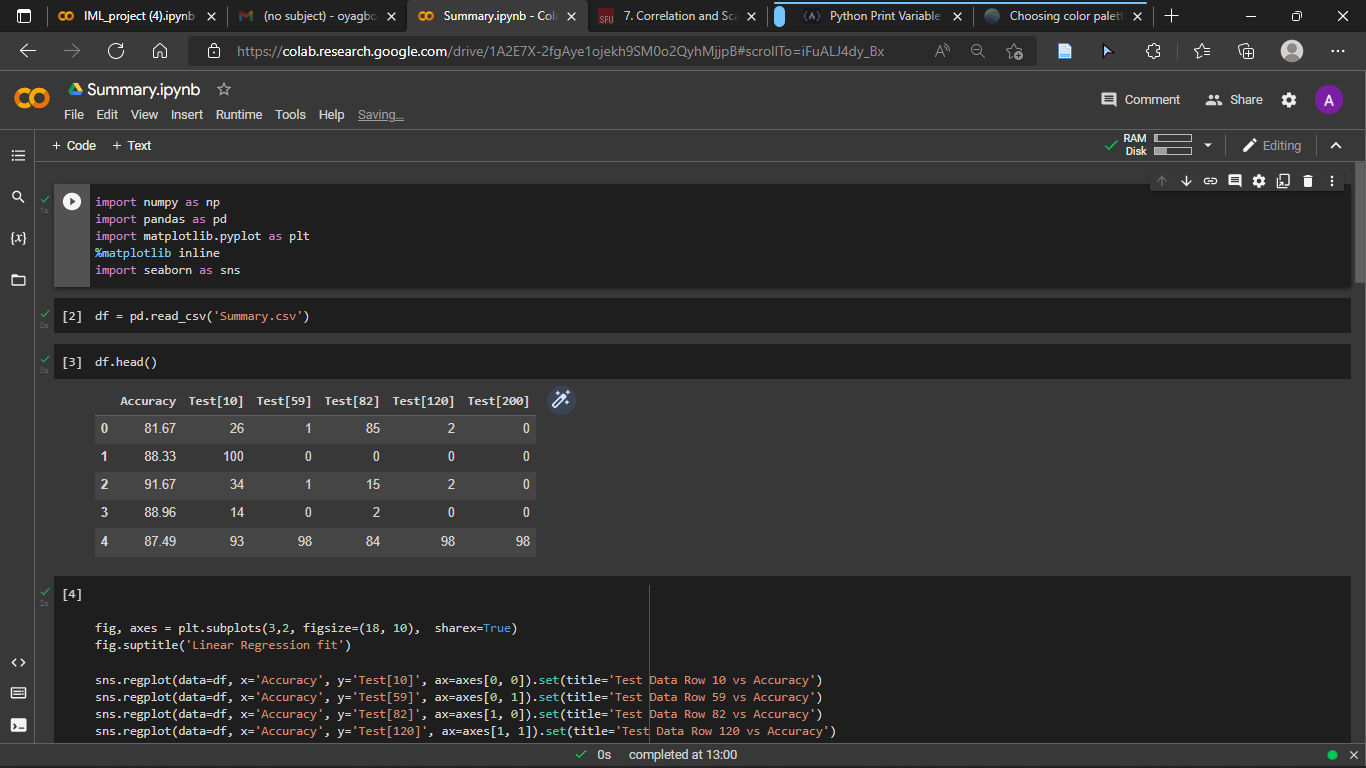


Figure 24 Results data table

## Result Explanation

In order to gain relevant insight into the result data previously generated and to also determine whether there a relationship exists between each models’ accuracy and local interpretability performance; a scatterplot was deployed as well as a linear regression line (also called line of best fit). This process works by plotting the graph of two features and finding a line that best fits the data points in the plot linearly, and the result or interpretation to show relationship between two features in this plot is solely dependent on how well the data points aligns with the line of best fit to determine their level of correlation, and also the direction and position at which the line is, also putting into consideration the start and end point of the line to determine if it is a positive or negative correlation. In this research work, I plot the values of model accuracy on the x-axis against the interpretability performance of the models for each of the selected data points on the y-axis. The image below shows the code used to generate this plots.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 25 Code to generate scatterplots and regression line

Accordingly, there is a shared similarity between all the plots generated, and this is that there is little to no correlation between the two features which are model accuracy and interpretability performance judging from the plots alone.

Comprehensively, on the first data point (Test[10]), the datapoints are heavily dispersed and separated from the linear regression line and also, the linear regression line appears to be on a downward slope, this interprets that there is an exceptionally low and negative correlation between these two features. Therefore, we can ascertain that on this data point, there is no relationship between model accuracy and interpretability performance.

On the second data point (Test[59]), the data points are also likewise not appearing to be clustered enough to show a relationship exists between them, but the linear regression line does have an upward direction of movement, this translates that on this data point, the relationship between the accuracy and interpretability performance is positive, but it is a very weak and negligible relationship that exists between them.

On the third data point (Test[82]), the data points seem to be the most scattered amongst the plots, which is already a giveaway point that very little to no relationship at all exists between these two features, and also, the linear regression line appears to be totally horizontal with a slight downward tilt, this indicates a very little albeit negative relationship between the two instances

On the forth data point (Test[120]), the data points are much more closely clustered in contrast to the previous plots, and it appears close to aligning with the linear regression line which has as upward momentum indicating positivity, this means that a relationship exists between the two attributes in this data instance but unfortunately, the relationship is not strong enough. Therefore, we can ascertain that there is a weak relationship between the two attributes in that data point.

On the fifth and final data point (Test[200]), the data points are very much similar to the forth data point previously discussed, which also indicates that there seem to be a relationship building up between model accuracy and interpretability performance but not strong enough to acknowledge that there is a tangible relationship existing between them. The regression line also have an upward tilt to it, indicating that there is a very weak positive correlation between the attributes on this data point.

The figures below shows the generated plots explained above.

Graphical user interface

Description automatically generated with medium confidence

Figure 26 Scatterplots with line of best fit

In order to give a clearer understanding about the figures above, the Pearson correlation coefficient was used to generate the actual correlation values for each pair of model accuracy against the interpretability performance for each of the chosen data point and then a correlation matrix was also generated to show how all the attributes individually pair against themselves and each other and back up the plots above.A screenshot of a computer

Description automatically generated

Figure 27 Code to generate correlation & correlation matrix

Shown below is the heatmap of the correlation matrix to visualise how the values contrasts and pair up against each other graphically.

A picture containing chart

Description automatically generated

Figure 28 Correlation matrix heatmap

# Conclusion, Evaluation and Future Work

Machine learning interpretability likewise explainability is a research area that is being explored in the quest for knowledge and to quench scientific curiosity behind complex machine learning models. Despite this, there are still quite a large number of grey areas that are seemingly yet to be explored.

Scientists, researches, analysts, statisticians and other professionals with domain knowledge about machine learning collectively believe the less complex an algorithm is, the easier it is to interpret and explain. I believe this is particularly true, because for example, personally, I can easily see a decision tree structure and can calculate and understand the working mechanisms like the reason for the root nodes and child nodes as long as the dimensionality is relatively low, else the process can be tedious, but this is not the case for seemingly complex algorithms and classifiers, whereas these complex classifiers produces the most accurate results, this had led to the drive in research work to make machine learning more interpretable and progress is being made with the advent of interpretable methods of which some of them were employed in this research work.

Currently, there is no specific metric to measure the correctness of machine generated interpretability for these complex models, however, this research work aims to answer the question “Is there a relationship between machine learning model accuracy and local interpretability?”, to successfully answer this research question, the research objectives were followed and achieved through a chronological approach, the red wine dataset was downloaded from UCI website ([UCI Machine Learning Repository: Wine Quality Data Set](https://archive.ics.uci.edu/ml/datasets/wine+quality)) and was imported into a data frame and normalised, then six machine learning models were trained on the data to derive their performance, interpretation was done for five random data points from the predicted data and linear regression line as well as Pearson correlation coefficient was used to determine the relationship that exists between accuracy and interpretability.

This research work focuses solely on establishing if a tangible relationship exists between model accuracy and local interpretability and if for a fact accuracy can influence local interpretability. Ultimately, the findings of the researcher are that there is little to no direct relationship between accuracy and local interpretability and I believe that model accuracy have almost no influence in result of local interpretability.

This research work can further be improved from different perspectives such as employing global interpretability to determine influence of accuracy, or employing different datasets or models. Likewise, investigation could also be done if other external factors such as standard deviation or dimensionality reduction could have significant impact on interpretability. Furthermore, further research can be done on feature interpretation to determine why some features have more positive or negative weights regarding predictions in complex algorithms like deep neural networks.

# References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Mach Learn 6, 37–66. <https://doi.org/10.1007/BF00153759>

Ahmad, M.A., Eckert, C., Teredesai, A., 2018. Interpretable Machine Learning in Healthcare, in: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. Presented at the BCB ’18: 9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, ACM, Washington DC USA, pp. 559–560. <https://doi.org/10.1145/3233547.3233667>

Altman, N.S., 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician 46, 175–185. <https://doi.org/10.1080/00031305.1992.10475879>

Andras, P., Esterle, L., Guckert, M., Han, T.A., Lewis, P.R., Milanovic, K., Payne, T., Perret, C., Pitt, J., Powers, S.T., Urquhart, N., Wells, S., 2018. Trusting Intelligent Machines: Deepening Trust Within Socio-Technical Systems. IEEE Technol. Soc. Mag. 37, 76–83. <https://doi.org/10.1109/MTS.2018.2876107>

Angelov, P., Soares, E., 2019. Towards Explainable Deep Neural Networks (xDNN).

Ayoub, J., Yang, X.J., Zhou, F., 2021. Combat COVID-19 infodemic using explainable natural language processing models. Information Processing & Management 58, 102569. <https://doi.org/10.1016/j.ipm.2021.102569>

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020a. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020b. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>

**BCS code of conduct**. 2016. *The British Computer Society Code of Conduct*. [online] Available at: <<http://www.bcs.org/upload/pdf/conduct.pdf>> [Accessed 30 September 2022].

Bohanec, M., Tartarisco, G., Marino, F., Pioggia, G., Puddu, P.E., Schiariti, M.S., Baert, A., Pardaens, S., Clays, E., Vodopija, A., Luštrek, M., 2021. HeartMan DSS: A decision support system for self-management of congestive heart failure. Expert Systems with Applications 186, 115688. <https://doi.org/10.1016/j.eswa.2021.115688>

Bohlin, T., 2006. Practical grey-box process identification: theory and applications, Advances in industrial control. Springer, London.

Cai, B., Kong, X., Liu, Y., Lin, J., Yuan, X., Xu, H., Ji, R., 2019. Application of Bayesian Networks in Reliability Evaluation. IEEE Trans. Ind. Inf. 15, 2146–2157. <https://doi.org/10.1109/TII.2018.2858281>

Carvalho, D.V., Pereira, E.M., Cardoso, J.S., 2019a. Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics 8, 832. <https://doi.org/10.3390/electronics8080832>

Dong, G., 2019. Exploiting the Power of Group Differences: Using Patterns to Solve Data Analysis Problems. Springer International Publishing, Cham. <https://doi.org/10.1007/978-3-031-01913-5>

Doshi-Velez, F., Kim, B., 2017. Towards A Rigorous Science of Interpretable Machine Learning.

Ebadi, A., Gauthier, Y., Tremblay, S., Paul, P., 2019. How can Automated Machine Learning Help Business Data Science Teams?, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). Presented at the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, Boca Raton, FL, USA, pp. 1186–1191. <https://doi.org/10.1109/ICMLA.2019.00196>

**Ethical OS Toolkit**. 2018. *The Ethical OS Toolkit* [online] Available at: <<http://www.ethicalos.org>> [Accessed 20 September 2022].

Fiok, K., Karwowski, W., Gutierrez, E., Wilamowski, M., 2021. Analysis of sentiment in tweets addressed to a single domain-specific Twitter account: Comparison of model performance and explainability of predictions. Expert Systems with Applications 186, 115771. <https://doi.org/10.1016/j.eswa.2021.115771>

Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L., 2018a. Explaining Explanations: An Overview of Interpretability of Machine Learning, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). Presented at the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, Turin, Italy, pp. 80–89. <https://doi.org/10.1109/DSAA.2018.00018>

Guerrero, J.I., Miró-Amarante, G., Martín, A., 2022. Decision support system in health care building design based on case-based reasoning and reinforcement learning. Expert Systems with Applications 187, 116037. <https://doi.org/10.1016/j.eswa.2021.116037>

H, A., R, A., 2019. Artificial Intelligence and Machine Learning for Enterprise Management, in: 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT). Presented at the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, Tirunelveli, India, pp. 1265–1269. <https://doi.org/10.1109/ICSSIT46314.2019.8987964>

Harb, J.G.D., Ebeling, R., Becker, K., 2020. A framework to analyze the emotional reactions to mass violent events on Twitter and influential factors. Information Processing & Management 57, 102372. <https://doi.org/10.1016/j.ipm.2020.102372>

Jordan, M.I., Mitchell, T.M., 2015a. Machine learning: Trends, perspectives, and prospects. Science 349, 255–260. <https://doi.org/10.1126/science.aaa8415>

Kang, Z., Catal, C., Tekinerdogan, B., 2020. Machine learning applications in production lines: A systematic literature review. Computers & Industrial Engineering 149, 106773. <https://doi.org/10.1016/j.cie.2020.106773>

Kim, B., Khanna, R., Koyejo, O.O., 2016. Examples are not enough, learn to criticize! Criticism for Interpretability, in: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc.

Lee, I., Shin, Y.J., 2020. Machine learning for enterprises: Applications, algorithm selection, and challenges. Business Horizons 63, 157–170. <https://doi.org/10.1016/j.bushor.2019.10.005>

Li, Y., Yang, L., Yang, B., Wang, N., Wu, T., 2019. Application of interpretable machine learning models for the intelligent decision. Neurocomputing 333, 273–283. <https://doi.org/10.1016/j.neucom.2018.12.012>

Li, Z., Zhang, Q., Du, X., Ma, Y., Wang, S., 2021. Social media rumor refutation effectiveness: Evaluation, modelling and enhancement. Information Processing & Management 58, 102420. <https://doi.org/10.1016/j.ipm.2020.102420>

Linardatos, P., Papastefanopoulos, V., Kotsiantis, S., 2020. Explainable AI: A Review of Machine Learning Interpretability Methods. Entropy 23, 18. <https://doi.org/10.3390/e23010018>

Lipton, P., 1990. Contrastive Explanation. Roy. Inst. Philos. Suppl. 27, 247–266. <https://doi.org/10.1017/S1358246100005130>

Lipton, Z.C., 2017. The Mythos of Model Interpretability.

Loyola-Gonzalez, O., 2019. Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. IEEE Access 7, 154096–154113. <https://doi.org/10.1109/ACCESS.2019.2949286>

Ma, L., Sun, B., 2020. Machine learning and AI in marketing – Connecting computing power to human insights. International Journal of Research in Marketing 37, 481–504. <https://doi.org/10.1016/j.ijresmar.2020.04.005>

Ma, Y., Guo, G. (Eds.), 2014. Support vector machines applications. Springer, Cham [Switzerland] ; New York. <https://doi.org/10.1007/978-3-319-02300-7>

Misztal-Radecka, J., Indurkhya, B., 2021. Bias-Aware Hierarchical Clustering for detecting the discriminated groups of users in recommendation systems. Information Processing & Management 58, 102519. <https://doi.org/10.1016/j.ipm.2021.102519>

Molnar, C., 2022. Interpretable machine learning: a guide for making black box models explainable, Second edition. ed. Christoph Molnar, Munich, Germany.

Molnar, C., Casalicchio, G., Bischl, B., 2020. Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges, in: Koprinska, I., Kamp, M., Appice, A., Loglisci, C., Antonie, L., Zimmermann, A., Guidotti, R., Özgöbek, Ö., Ribeiro, R.P., Gavaldà, R., Gama, J., Adilova, L., Krishnamurthy, Y., Ferreira, P.M., Malerba, D., Medeiros, I., Ceci, M., Manco, G., Masciari, E., Ras, Z.W., Christen, P., Ntoutsi, E., Schubert, E., Zimek, A., Monreale, A., Biecek, P., Rinzivillo, S., Kille, B., Lommatzsch, A., Gulla, J.A. (Eds.), ECML PKDD 2020 Workshops, Communications in Computer and Information Science. Springer International Publishing, Cham, pp. 417–431. <https://doi.org/10.1007/978-3-030-65965-3_28>

Momeni, E., Nazir, R., Jahed Armaghani, D., Maizir, H., 2015. Application of Artificial Neural Network for Predicting Shaft and Tip Resistances of Concrete Piles. Earth sci. res. j. 19, 85–93. <https://doi.org/10.15446/esrj.v19n1.38712>

Nickerson, R.S., 1998. Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. Review of General Psychology 2, 175–220. <https://doi.org/10.1037/1089-2680.2.2.175>

Pak, M., Kim, S., 2017. A review of deep learning in image recognition, in: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT). Presented at the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), IEEE, Kuta Bali, pp. 1–3. <https://doi.org/10.1109/CAIPT.2017.8320684>

Panigutti, C., Perotti, A., Panisson, A., Bajardi, P., Pedreschi, D., 2021. FairLens: Auditing black-box clinical decision support systems. Information Processing & Management 58, 102657. <https://doi.org/10.1016/j.ipm.2021.102657>

Pintelas, E., Livieris, I.E., Pintelas, P., 2020a. A Grey-Box Ensemble Model Exploiting Black-Box Accuracy and White-Box Intrinsic Interpretability. Algorithms 13, 17. <https://doi.org/10.3390/a13010017>

Przybyła, P., Soto, A.J., 2021. When classification accuracy is not enough: Explaining news credibility assessment. Information Processing & Management 58, 102653. <https://doi.org/10.1016/j.ipm.2021.102653>

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. Model-Agnostic Interpretability of Machine Learning. <https://doi.org/10.48550/ARXIV.1606.05386>

Richardson, M., Domingos, P., 2006. Markov logic networks. Mach Learn 62, 107–136. <https://doi.org/10.1007/s10994-006-5833-1>

Rico-Juan, J.R., Taltavull de La Paz, P., 2021. Machine learning with explainability or spatial hedonics tools? An analysis of the asking prices in the housing market in Alicante, Spain. Expert Systems with Applications 171, 114590. <https://doi.org/10.1016/j.eswa.2021.114590>

Ridley, M., 2022. Explainable Artificial Intelligence (XAI). ITAL 41. <https://doi.org/10.6017/ital.v41i2.14683>

Rudin, C., 2018. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. <https://doi.org/10.48550/ARXIV.1811.10154>

Samek, W., Wiegand, T., Müller, K.-R., 2017. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models.

Smiti, A., 2020. When machine learning meets medical world: Current status and future challenges. Computer Science Review 37, 100280. <https://doi.org/10.1016/j.cosrev.2020.100280>

Stilgoe, J., 2018. Machine learning, social learning and the governance of self-driving cars. Soc Stud Sci 48, 25–56. <https://doi.org/10.1177/0306312717741687>

Štrumbelj, E., Kononenko, I., 2011. A General Method for Visualizing and Explaining Black-Box Regression Models, in: Dobnikar, A., Lotrič, U., Šter, B. (Eds.), Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 21–30. <https://doi.org/10.1007/978-3-642-20267-4_3>

Strümke, I., Slavkovik, M., Madai, V.I., 2021. The social dilemma in artificial intelligence development and why we have to solve it. AI Ethics. <https://doi.org/10.1007/s43681-021-00120-w>

van Klompenburg, T., Kassahun, A., Catal, C., 2020. Crop yield prediction using machine learning: A systematic literature review. Computers and Electronics in Agriculture 177, 105709. <https://doi.org/10.1016/j.compag.2020.105709>

West, D., West, V., 2000. Model selection for a medical diagnostic decision support system: a breast cancer detection case. Artificial Intelligence in Medicine 20, 183–204. <https://doi.org/10.1016/S0933-3657(00)00063-4>

Yu, G., Chen, Z., Wu, J., Tan, Y., 2021. Medical decision support system for cancer treatment in precision medicine in developing countries. Expert Systems with Applications 186, 115725. <https://doi.org/10.1016/j.eswa.2021.115725>

Dataset Source: UCI Machine Learning Repository - Wine Quality Data Set, n.d. URL <https://archive.ics.uci.edu/ml/datasets/wine+quality> (accessed 9.20.22).

# Appendix A: Gantt Chart

Gantt Chart Timeline

Description automatically generatedGantt Chart For Project Timeline

# Appendix B : GitHub Repository

[Adeboy1876/Model-agnostic-IML: This repository contains the codes for investigating if model accuracy influences the interpretability of the model (github.com)](https://github.com/Adeboy1876/Model-agnostic-IML)

# Appendix C: Source code for analysis

import numpy as np

import pandas as pd

import matplotlib as plt

import seaborn as sns

#read csv file

df = pd.read\_csv('winequality-red.csv')

df.head()

#separate target column from the rest od data

X\_feature\_names = [i for i in df.columns if df[i].dtype in [np.float]]

print(X\_feature\_names)

['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']

Y\_feature\_names = [i for i in df.columns if df[i].dtype in [np.int64]]

print(Y\_feature\_names)

['quality']

#checking for null columns

print(df.isna().sum())

corr = df.corr()

plt.pyplot.subplots(figsize=(20,10))

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot = True, cmap=sns.diverging\_palette(220,20, as\_cmap=True))

#set value 1 for all tiems in quality column greater than 8 (good) otherwise, set as 0 (bad)

df['goodquality'] = [1 if x>=6.5 else 0 for x in df['quality']]

#set Y as good quality column, which will be used for classification

X = df.drop(['quality','goodquality'], axis = 1)

Y = df['goodquality']

df['goodquality'].value\_counts()

0 1382 1 217 Name: goodquality, dtype: int64

#Normalize Data

from sklearn.preprocessing import StandardScaler

X\_features = X

X = StandardScaler().fit\_transform(X)

#Train Test Splitting

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=.30, random\_state= 40)

Y\_test.value\_counts()

0 411 1 69 Name: goodquality, dtype: int64

#Function to print model accuracy

def print\_accuracy(f):

  print('Accuracy = {:.2f}%'.format(100\*np.sum(f(X\_test)==Y\_test)/len(Y\_test)))

#Naive Bayes Classifier

from sklearn import naive\_bayes

nb = naive\_bayes.GaussianNB()

nb.fit(X\_train, Y\_train)

print\_accuracy(nb.predict)

Accuracy = 81.67%

#Decision Tree Classifier

import sklearn.tree

dtree = sklearn.tree.DecisionTreeClassifier(min\_samples\_split=2, random\_state=0)

dtree.fit(X\_train, Y\_train)

print\_accuracy(dtree.predict)

Accuracy = 88.33%

#Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier

rforest = RandomForestClassifier(random\_state=0)

rforest.fit(X\_train, Y\_train)

print\_accuracy(rforest.predict)

Accuracy = 91.67%

#Neural Network

from sklearn.neural\_network import MLPClassifier

nn = MLPClassifier(activation='relu', hidden\_layer\_sizes=(100,2),max\_iter=20000, random\_state=40, learning\_rate = 'constant')

nn.fit(X\_train, Y\_train)

print\_accuracy(nn.predict)

Accuracy = 88.96%

from tensorflow import keras

from keras.models import Sequential

from keras.layers import Dense

#Deep Neural Network with one hidden layer

DNN\_model = Sequential()

#input layer

DNN\_model.add(Dense(12, activation = 'sigmoid', input\_shape = (11,)))

#hidden layer 1

DNN\_model.add(Dense(8, activation='sigmoid'))

#output layer

DNN\_model.add(Dense(1, activation = 'sigmoid'))

DNN\_model.compile (loss = 'binary\_crossentropy', optimizer='adam', metrics = ['accuracy'] )

history = DNN\_model.fit(X\_train, Y\_train, epochs=10, batch\_size=1, verbose = 1)

loss, acc = DNN\_model.evaluate(X\_train, Y\_train, verbose = 0)

print('Accuracy: %.2f' % (acc \* 100))

Accuracy: 88.65

#Deep neural network with 2 hidden layers

DNN\_model2 = Sequential()

#input layer

DNN\_model2.add(Dense(14, activation = 'relu', input\_dim = 11))

#hidden layer 1

DNN\_model2.add(Dense(10, activation='relu'))

#hidden layer 2

DNN\_model2.add(Dense(7, activation='relu'))

#output layer

DNN\_model2.add(Dense(1, activation = 'sigmoid'))

DNN\_model2.compile (loss = 'binary\_crossentropy', optimizer='adam', metrics = ['accuracy'] )

history2 = DNN\_model2.fit(X\_train, Y\_train, epochs=150, batch\_size=10)

loss2, acc2 = DNN\_model2.evaluate(X\_train, Y\_train)

print('Accuracy: %.2f' % (acc2 \* 100))

Accuracy: 97.32

!pip install lime

!pip install shap

import lime

from lime import lime\_tabular

import shap

explainer\_lime = lime\_tabular.LimeTabularExplainer(

    training\_data=np.array(X\_train),

    feature\_names = X\_feature\_names,

    class\_names = ['bad','good'],

    mode = 'classification'

)

# TEST [10]

#### Running explainability on all algorithms for prediction of row 10 in test data i.e X\_test[10]

#naive bayes

exp\_nb = explainer\_lime.explain\_instance(

    data\_row = X\_test[10],

    predict\_fn=nb.predict\_proba

)

exp\_nb.show\_in\_notebook(show\_table=True)

#shap

shap.initjs()

explainer\_nb = shap.KernelExplainer(nb.predict\_proba, X\_train)

shap\_values = explainer\_nb.shap\_values(X\_test[10])

shap.force\_plot(explainer\_nb.expected\_value[0], shap\_values[0], X\_test[10])

#Decision Tree

exp\_dtree = explainer\_lime.explain\_instance(

    data\_row = X\_test[10],

    predict\_fn=dtree.predict\_proba

)

exp\_dtree.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_dtree = shap.KernelExplainer(dtree.predict\_proba, X\_train)

shap\_values = explainer\_dtree.shap\_values(X\_test[10])

shap.force\_plot(explainer\_dtree.expected\_value[0], shap\_values[0], X\_test[10])

#Random Forest

exp\_rforest = explainer\_lime.explain\_instance(

    data\_row = X\_test[10],

    predict\_fn=rforest.predict\_proba

)

exp\_rforest.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_rforest = shap.KernelExplainer(rforest.predict\_proba, X\_train)

shap\_values = explainer\_rforest.shap\_values(X\_test[10])

shap.force\_plot(explainer\_rforest.expected\_value[0], shap\_values[0], X\_test[10])

#Neural Network

exp\_nn = explainer\_lime.explain\_instance(

    data\_row = X\_test[10],

    predict\_fn=nn.predict\_proba

)

exp\_nn.show\_in\_notebook(show\_table=True)

#Deep Neural Network 1

exp\_DNN\_model = explainer\_lime.explain\_instance(

    data\_row = X\_test[10],

    predict\_fn=DNN\_model.predict,

    top\_labels = 2

)

exp\_DNN\_model.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_DNN\_model = shap.KernelExplainer(DNN\_model.predict, X\_train)

shap\_values = explainer\_DNN\_model.shap\_values(X\_test[10])

shap.force\_plot(explainer\_DNN\_model.expected\_value[0], shap\_values[0], X\_test[10])

#Deep Neural Network 2

exp\_DNN\_model2 = explainer\_lime.explain\_instance(

    data\_row = X\_test[10],

    predict\_fn=DNN\_model2.predict,

    top\_labels=1

)

exp\_DNN\_model2.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_DNN\_model2 = shap.KernelExplainer(DNN\_model2.predict, X\_train)

shap\_values = explainer\_DNN\_model2.shap\_values(X\_test[10])

shap.force\_plot(explainer\_DNN\_model2.expected\_value[0], shap\_values[0], X\_test[10])

# TEST [59]

#### Running explainability on all algorithms for prediction of row 59 in test data i.e X\_test[59]

exp\_nb\_59 = explainer\_lime.explain\_instance(

    data\_row = X\_test[59],

    predict\_fn=nb.predict\_proba

)

exp\_nb\_59.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_nb\_59 = shap.KernelExplainer(nb.predict\_proba, X\_train)

shap\_values = explainer\_nb\_59.shap\_values(X\_test[59])

shap.force\_plot(explainer\_nb\_59.expected\_value[0], shap\_values[0], X\_test[59])

exp\_dtree\_59 = explainer\_lime.explain\_instance(

    data\_row = X\_test[59],

    predict\_fn=dtree.predict\_proba

)

exp\_dtree\_59.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_dtree\_59 = shap.KernelExplainer(dtree.predict\_proba, X\_train)

shap\_values = explainer\_dtree\_59.shap\_values(X\_test[59])

shap.force\_plot(explainer\_dtree\_59.expected\_value[0], shap\_values[0], X\_test[59])

exp\_rforest\_59 = explainer\_lime.explain\_instance(

data\_row = X\_test[59],

predict\_fn=rforest.predict\_proba

)

exp\_rforest\_59.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_rforest\_59 = shap.KernelExplainer(rforest.predict\_proba, X\_train)

shap\_values = explainer\_rforest\_59.shap\_values(X\_test[59])

shap.force\_plot(explainer\_rforest\_59.expected\_value[0], shap\_values[0], X\_test[59])

exp\_nn\_59 = explainer\_lime.explain\_instance(

    data\_row = X\_test[59],

    predict\_fn=nn.predict\_proba

)

exp\_nn\_59.show\_in\_notebook(show\_table=True)

exp\_DNN\_model\_59 = explainer\_lime.explain\_instance(

    data\_row = X\_test[59],

    predict\_fn=DNN\_model.predict,

    top\_labels = 2

)

exp\_DNN\_model\_59.show\_in\_notebook(show\_table=True)

exp\_DNN\_model2\_59 = explainer\_lime.explain\_instance(

    data\_row = X\_test[59],

    predict\_fn=DNN\_model2.predict,

    top\_labels = 2

)

exp\_DNN\_model2\_59.show\_in\_notebook(show\_table=True)

# TEST [82]

#### Running explainability on all algorithms for prediction of row 82 in test data i.e X\_test[82]

exp\_nb\_82 = explainer\_lime.explain\_instance(

    data\_row = X\_test[82],

    predict\_fn=nb.predict\_proba

)

exp\_nb\_82.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_nb\_82 = shap.KernelExplainer(nb.predict\_proba, X\_train)

shap\_values = explainer\_nb\_82.shap\_values(X\_test[82])

shap.force\_plot(explainer\_nb\_82.expected\_value[0], shap\_values[0], X\_test[82])

exp\_dtree\_82 = explainer\_lime.explain\_instance(

    data\_row = X\_test[82],

    predict\_fn=dtree.predict\_proba

)

exp\_dtree\_82.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_dtree\_82 = shap.KernelExplainer(dtree.predict\_proba, X\_train)

shap\_values = explainer\_dtree\_82.shap\_values(X\_test[82])

shap.force\_plot(explainer\_dtree\_82.expected\_value[0], shap\_values[0], X\_test[82])

exp\_rforest\_82 = explainer\_lime.explain\_instance(

    data\_row = X\_test[82],

    predict\_fn=rforest.predict\_proba

)

exp\_rforest\_82.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_rforest\_82 = shap.KernelExplainer(rforest.predict\_proba, X\_train)

shap\_values = explainer\_rforest\_82.shap\_values(X\_test[82])

shap.force\_plot(explainer\_rforest\_82.expected\_value[0], shap\_values[0], X\_test[82])

exp\_nn\_82 = explainer\_lime.explain\_instance(

    data\_row = X\_test[82],

    predict\_fn=nn.predict\_proba

)

exp\_nn\_82.show\_in\_notebook(show\_table=True)

exp\_DNN\_model\_82 = explainer\_lime.explain\_instance(

    data\_row = X\_test[82],

    predict\_fn=DNN\_model.predict,

    top\_labels = 2

)

exp\_DNN\_model\_82.show\_in\_notebook(show\_table=True

)

exp\_DNN\_model2\_82 = explainer\_lime.explain\_instance(

    data\_row = X\_test[82],

    predict\_fn=DNN\_model2.predict,

    top\_labels = 2

)

exp\_DNN\_model2\_82.show\_in\_notebook(show\_table=True)

# TEST [120]

#### Running explainability on all algorithms for prediction of row 120 in test data i.e X\_test[120]

exp\_nb\_120 = explainer\_lime.explain\_instance(

    data\_row = X\_test[120],

    predict\_fn=nb.predict\_proba

)

exp\_nb\_120.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_nb\_120 = shap.KernelExplainer(nb.predict\_proba, X\_train)

shap\_values = explainer\_nb\_120.shap\_values(X\_test[120])

shap.force\_plot(explainer\_nb\_120.expected\_value[0], shap\_values[0], X\_test[120])

exp\_dtree\_120 = explainer\_lime.explain\_instance(

    data\_row = X\_test[120],

    predict\_fn=dtree.predict\_proba

)

exp\_dtree\_120.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_dtree\_120 = shap.KernelExplainer(dtree.predict\_proba, X\_train)

shap\_values = explainer\_dtree\_120.shap\_values(X\_test[120])

shap.force\_plot(explainer\_dtree\_120.expected\_value[0], shap\_values[0], X\_test[120])

exp\_rforest\_120 = explainer\_lime.explain\_instance(

    data\_row = X\_test[120],

    predict\_fn=rforest.predict\_proba

)

exp\_rforest\_120.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_rforest\_120 = shap.KernelExplainer(rforest.predict\_proba, X\_train)

shap\_values = explainer\_rforest\_120.shap\_values(X\_test[120])

shap.force\_plot(explainer\_rforest\_120.expected\_value[0], shap\_values[0], X\_test[120])

exp\_nn\_120 = explainer\_lime.explain\_instance(

    data\_row = X\_test[120],

    predict\_fn=nn.predict\_proba

)

exp\_nn\_120.show\_in\_notebook(show\_table=True)

exp\_DNN\_model\_120 = explainer\_lime.explain\_instance(

    data\_row = X\_test[120],

    predict\_fn=DNN\_model.predict,

    top\_labels = 2

)

exp\_DNN\_model\_120.show\_in\_notebook(show\_table=True)

exp\_DNN\_model2\_120 = explainer\_lime.explain\_instance(

    data\_row = X\_test[120],

    predict\_fn=DNN\_model2.predict,

    top\_labels = 2

)

exp\_DNN\_model2\_120.show\_in\_notebook(show\_table=True)

# TEST [200]

#### Running explainability on all algorithms for prediction of row 200 in test data i.e X\_test[200]

exp\_nb\_200 = explainer\_lime.explain\_instance(

    data\_row = X\_test[200],

    predict\_fn=nb.predict\_proba

)

exp\_nb\_200.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_nb\_200 = shap.KernelExplainer(nb.predict\_proba, X\_train)

shap\_values = explainer\_nb\_200.shap\_values(X\_test[200])

shap.force\_plot(explainer\_nb\_200.expected\_value[0], shap\_values[0], X\_test[200])

exp\_dtree\_200 = explainer\_lime.explain\_instance(

    data\_row = X\_test[200],

    predict\_fn=dtree.predict\_proba

)

exp\_dtree\_200.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_dtree\_200 = shap.KernelExplainer(dtree.predict\_proba, X\_train)

shap\_values = explainer\_dtree\_200.shap\_values(X\_test[200])

shap.force\_plot(explainer\_dtree\_200.expected\_value[0], shap\_values[0], X\_test[200])

exp\_rforest\_200 = explainer\_lime.explain\_instance(

    data\_row = X\_test[200],

    predict\_fn=rforest.predict\_proba

)

exp\_rforest\_200.show\_in\_notebook(show\_table=True)

shap.initjs()

explainer\_rforest\_200 = shap.KernelExplainer(rforest.predict\_proba, X\_train)

shap\_values = explainer\_rforest\_200.shap\_values(X\_test[200])

shap.force\_plot(explainer\_rforest\_200.expected\_value[0], shap\_values[0], X\_test[200])

exp\_nn\_200 = explainer\_lime.explain\_instance(

    data\_row = X\_test[200],

    predict\_fn=nn.predict\_proba

)

exp\_nn\_200.show\_in\_notebook(show\_table=True)

exp\_DNN\_model\_200 = explainer\_lime.explain\_instance(

    data\_row = X\_test[200],

    predict\_fn=DNN\_model.predict,

    top\_labels = 2

)

exp\_DNN\_model\_200.show\_in\_notebook(show\_table=True)

exp\_DNN\_model2\_200 = explainer\_lime.explain\_instance(

    data\_row = X\_test[200],

    predict\_fn=DNN\_model.predict,

    top\_labels = 2

)

exp\_DNN\_model2\_200.show\_in\_notebook(show\_table=True)

# Appendix D: Source code for result

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

df = pd.read\_csv('Summary.csv')

df.head()

fig, axes = plt.subplots(2,3, figsize=(20, 10),  sharex=True)

fig.suptitle('Linear Regression fit')

sns.regplot(data=df, x='Accuracy', y='Test[10]', ax=axes[0, 0]).set(title='Test Data Row 10 vs Accuracy')

sns.regplot(data=df, x='Accuracy', y='Test[59]', ax=axes[0, 1]).set(title='Test Data Row 59 vs Accuracy')

sns.regplot(data=df, x='Accuracy', y='Test[82]', ax=axes[0, 2]).set(title='Test Data Row 82 vs Accuracy')

sns.regplot(data=df, x='Accuracy', y='Test[120]', ax=axes[1, 0]).set(title='Test Data Row 120 vs Accuracy')

sns.regplot(data=df, x='Accuracy', y='Test[200]', ax=axes[1, 1]).set(title='Test Data Row 200 vs Accuracy')

axes.flat[-1].set\_visible(False)

from scipy import stats

corr1 = stats.pearsonr(df['Accuracy'], df['Test[10]'])

corr2 = stats.pearsonr(df['Accuracy'], df['Test[59]'])

corr3 = stats.pearsonr(df['Accuracy'], df['Test[82]'])

corr4 = stats.pearsonr(df['Accuracy'], df['Test[120]'])

corr5 = stats.pearsonr(df['Accuracy'], df['Test[200]'])

print('Correlation Coefficient between Accuracy and Test[10] = {}'.format(corr1))

print('Correlation Coefficient between Accuracy and Test[59] = {}'.format(corr2))

print('Correlation Coefficient between Accuracy and Test[82] = {}'.format(corr3))

print('Correlation Coefficient between Accuracy and Test[120] = {}'.format(corr4))

print('Correlation Coefficient between Accuracy and Test[200] = {}'.format(corr5))

Correlation Coefficient between Accuracy and Test[10] = (-0.08656027610421844, 0.8704838701290118)

Correlation Coefficient between Accuracy and Test[59] = (0.4621136039414775, 0.35617153915180183)

Correlation Coefficient between Accuracy and Test[82] = (-0.01929320276638319, 0.9710637865824102)

Correlation Coefficient between Accuracy and Test[120] = (0.46410238930011, 0.3538281613570704)

Correlation Coefficient between Accuracy and Test[200] = (0.460070232420421, 0.3585849465432724)

cormat = df.corr()

round(cormat,2)

|  | **Accuracy** | **Test[10]** | **Test[59]** | **Test[82]** | **Test[120]** | **Test[200]** |
| --- | --- | --- | --- | --- | --- | --- |
| **Accuracy** | 1.00 | -0.09 | 0.46 | -0.02 | 0.46 | 0.46 |
| **Test[10]** | -0.09 | 1.00 | 0.28 | -0.03 | 0.28 | 0.29 |
| **Test[59]** | 0.46 | 0.28 | 1.00 | 0.73 | 1.00 | 1.00 |
| **Test[82]** | -0.02 | -0.03 | 0.73 | 1.00 | 0.73 | 0.73 |
| **Test[120]** | 0.46 | 0.28 | 1.00 | 0.73 | 1.00 | 1.00 |
| **Test[200]** | 0.46 | 0.29 | 1.00 | 0.73 | 1.00 | 1.00 |

corr = df.corr()

plt.subplots(figsize=(15,7))

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annnot = True, cmap=sns.diverging\_palette(220,20, as\_cmap=True))