Towards Scrutable Decision Tree-based User Model utilising Interactive and Interpretable Machine Learning (SUM-IML)

A thesis submitted to
University of Trinity College Dublin

Dima Saber Mahmoud

ADAPT Centre
School of Computer Science and Statistics,
Trinity College Dublin,
Dublin, Ireland.

Supervised by
Prof. Owen Conlan

27st May, 2021

DECLARATION

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Dima Saber Mahmoud

27th May 2021.

Permission to Lend and/or Copy

I, Dima Saber Mahmoud, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Dima Saber Mahmoud

27th May 2021.

ACKNOWLEDGEMENT

This research is enhanced by ADAPT Centre at Trinity College Dublin. ADAPT Centre for Digital Content Technologies is funded under support from Science Foundation Ireland (SFI) Research Centres Programme (Grant 13/RC/2106) and co-funded under the European Regional Development Fund. Special thanks to Prof. Owen Conlan for his continuous guidance, and advice.

TABLE OF CONTENTS

Table of Figures

TABLE OF TABLES

ABSTRACT

Personalising user models has gained considerable attention in recent literature. In an information-rich environment, it is crucial not only to provide the information at any time, at any place, and in any form, but to also minimise information overload for the user and ease their ability to access relevant information in their user model. Supplying tailored information and providing offerings that suit each user's interests may be enhanced by involving the user.

Recently, much research has been concerned with employing Machine Learning for user modelling. Machine Learning (ML) tries to mimic and predict a user’s activities, however, it cannot model the user themselves. Users can benefit from involvement in the modelling process by incorporating their input as considerations in modelling through employing interactive Machine Learning or human-in-the-loop controls. Such user involvement needs the user to understand the model behaviour to ensure their inputs are effective. This can be achieved by utilising Machine Learning interpretability techniques. This work proposes an interpretation of the model to the user in order to provide the user with better understanding for the model behaviour.

In this study, the proposed approach, termed SUM-IML (Scrutable User Modelling using Interactive and Interpretable Machine Learning)., implements model scrutability by combining the benefits of interactive ML as well as Interpretable ML in user modelling. This thesis presents the research question driving this work, a state-of-the-art review of user modelling, scrutability, interactive, interpretable Machine Learning literature and the evaluation methodologies required. It then presents two related experiments that demonstrate the exploration of research question through their results and the conclusion.

1. INTRODUCTION

1.1. MOTIVATION

Generally, A user model is a set of personal information related to a specific user, serving as a representation of the user in a system. As user modelling is primarily based on personal information, user control over this model is considered a crucial factor (Assad et al. 2007). Individual understanding and control over the modelling process is important as it can empower the user to maintain and manage what is modelled about them. A model is described as a *scrutable model* when it allows the user to interrogate it. This ensures people can control their personal information and understand how the model uses this information. (Assad et al. 2007).

Recently, much research has been concerned with employing Machine Learning (ML) in user modelling. The Internet evolution has motivated a surge of research in this field (Kelleher, Mac Namee, and D'Arcy 2015). By closely investigating how interactions affect user interests, the early studies of M. E. Muller (Muller 2004) showed the importance of the user's need to be in charge of the model. Therefore, arguing based on logic, it is thus necessary to provide users with an interpretable system. This is a gap in the state-of-the-art, as the user's need to have a scrutable user model is currently not implemented in the case of ML-driven user model.

Fundamentally, Machine Learning tries to simulate and predict the user's performance; however, it cannot model the user themselves. Through ML we can build systems that can provide users with suggestions and predictions with high accuracy as it can learn from past data (typically the user's interaction history) (Dietterich 1997). However, there may be several present and future user interests that are not represented in this data as it is incomplete, or the user has not expressed this interest in the digital data being analysed. Thus, involving the user in the modelling process can be considered a highly valuable input in this context. However, to date, such user input towards ML-driven user modelling has received little attention (Amershi et al. 2014).

This research will examine two Machine Learning related approaches. The first approach is to take advantage of user involvement in the modelling process and considering their input in building the model (Billsus and Pazzani 2000). Engaging the user in the Machine Learning training process is referred to as interactive Machine Learning (Fails and Olsen 2003) (Amershi et al. 2014) or human-in-the-loop control. This study explores the approach of employing Machine Learning – interactive Machine Learning specifically – to implement the user control part of model scrutability.

The second approach examined is Machine Learning interpretability. Explaining model behaviour and its output prediction is an important aspect in getting users to use ML-driven model effectively. In this case, Miller (Miller 2017) provides a simplistic definition of Interpretability as: the degree to which a human can understand the cause of a decision. This can be achieved within our approach by explaining a prediction, or in other words, presenting a qualitative understanding of the relationship between the instance components (Marco Tulio Ribeiro, Singh, and Guestrin 2016). By employing Machine Learning interpretability approach, we can implement the understandability part of the model scrutability.

1.2. **Research Question**

As we mentioned earlier, model scrutability is found where a user can interrogate their user model. A user model is a set of personal data related to a specific user and it is the basis of all the adaptive changes in system behaviour (Zemmouri and Benslimane 2015). So, it is beneficial for the model to be able to dynamically change to accommodate the changes in user interests. This helps in meeting any shifts in the user's needs (Fischer 2001)(Hothi and Hall 1998) (Girolami and Kabán 2003).

Using Machine Learning in user modelling is a powerful tool for transforming data into computational models (Billsus and Pazzani 2000). We can use it in implementing an efficient smart system as it can learn from the users' behaviour and interactions (Amershi et al. 2014). For this, user's history can be collected and represented in the system.

The challenges usually occur when trying to maintain model scrutability in the case of using ML. The main issues addressed here are: (1) when there are changes in these preferences and these updates are not reflected in the data, e.g. a users' preferred restaurants may change once they have children. Thus, involving the user in the modelling process can be considered an essential input to tune the ML process by re-learning from user feedback; (2) the other important problem is user understanding of the model regarding behaviour and output.

Is it feasible to implement model scrutability when employing – Decision Tree based – Machine Learning in User Modelling?

This research proposes a novel approach to enable scrutability in Decision Tree- based user-model. This proposal comprises the following objectives: (1) Supplementing ML intelligence with user control; and (2) Demonstrating the behaviour of ML models – in a human understandable fashion.

The two major aspects of this research are:

1) The role of the user in system re-learning which is taking user feedback as an input in the modelling loop. Primarily, this concerns the crucial point of how to blend the Machine Learning intelligence and user control. This is in addition to keeping the balance between user engagement and minimising the burden of user control over the model.

2) Explaining the model to the user. Interpreting the model behaviour and its output enables the user to use the model effectively. In other words, the target is to provide a qualitative explanation of the relationship between model components to the user.

Investigation of the research question is outlined by the following objectives:

- To analyse the state-of-the-art in terms of approaches and methodologies for user modelling and scrutability.
- To analyse the state-of-the-art in terms of approaches and methodologies for interactive and interpretable Machine Learning.
- To analyse the state-of-the-art for evaluation techniques for both ML modelling and interpretable ML.
- To propose an approach to support the research question (applying interactive and interpretable ML to build a controllable and understandable model) to maintain the model scrutability.
- To develop and implement a model as a proof of concept.
- To design and execute experimentation and evaluation methodology.

1.3. CONTRIBUTIONS

The novel contribution in this work is proposing a new approach called SUM-IML (Scrutable User Modelling using Interactive and

Interpretable Machine Learning). The idea of this approach is to combine interactive and interpretable Machine Learning to implement a scrutable user model. Both are powerful tools to raise the understanding between the user and the model.

The scrutability of the user model can be maintained by explaining the model and by using the user involvement in building the algorithm and the entire modelling process (human-in-the-loop). Interactive and interpretable ML demonstrate particular potential to solve such problems.

1.4. RESEARCH APPROACH

The strategy of this study is to start by building a case study to implement a scrutable user model-which is discussed in detail in section 3.2 that demonstrates the concept of SUM-IML. The case study builds a user model that predicts the importance of an incoming email to users. The model leverages user understandability and control over the Machine Learning algorithm in different ways.

The case study goes through several examinations in order to evaluate the impact of engaging the user in the modelling process on the results. This is in addition to examining user understanding. This starts with building a personalized ML model that reads user's email messages and learns user preferences. From this, the model can then predict user interest in an incoming email message.

The target of experimentation is implementing the two parts of model scrutability – user input and user understanding. The first experiment is maintaining user input in the model. This is implemented by simulating user feedback about the model prediction. The second experiment is concerned with building an interpretation model that can explain the user model behaviour. Thus, both parts of scrutability are maintained.

1.5. THESIS OVERVIEW

Chapter two represents a review of the state-of-the-art of various components involved in this research including Personalisation, User Modelling, Scrutability, Interactive and Interpretable Machine Learning. Chapter three Shows the design of the proposed SUM-IML approach as well as the design components. The fourth chapter shows the details of the experiments conducted and their results. The fifth is the last chapter which is the conclusion and future work. This is followed by references and appendices.

2. STATE OF THE ART

This section provides a review of the literature. The focus of this review is the studies that targeted the field of user modelling in general as well as user modelling using Machine Learning (ML). First, an overview is given about user modelling, as it is the foundation field that the study is built upon. Following this, the model scrutabilty will be reviewed, and current state-of-the-art interactive ML and interpretable ML will be elaborated on. Finally, the evaluation of both techniques will be discussed.

Fundamentally, the core objective of personalising a user model is to provide users with experiences fitting their specific backgrounds (Y. Yang et al., 2005). Due to the overwhelming amount of information available, there is a significant challenge in not only making information available anytime, anywhere, and in any form but also in precisely specifying the 'right' information at the 'right' time and in the 'right' form. Once information is collected about a certain user, the system can evaluate that data using a preset analytical algorithm and then personalise it to meet the user's needs (O'Keeffe et al., 2012). The user profile affects how information and functions are displayed by highlighting only relevant aspects and thus hiding information that is not needed by the user.

The main challenge is to make systems capable of providing users with experiences fitting their interests and preferences. The massive amount of data that is represented in the user history (preferences and interactions) is what constitutes this challenge.

In this chapter, we discuss the main topics related to this research. The first section will be about user modelling, then the next discusses scrutability in user modelling. Sections 3 and 4 explain ML learning techniques and the corresponding evaluation methodologies.

2.1. USER MODELLING

A model is defined as 'an abstract representation of something that exists in the real world' (Koch, 2001). A user model is 'the presentation of a mental state (such as knowledge, preference, background and experience) related to a context in the real world' (Bra, n.d.). Thus it can be said that a user model is a set of personal data related to a specific user. It is the basis of all the adaptive changes in system behaviour (Zemmouri & Benslimane, 2015). Saying the right thing at the right time in the right way is the main concept of user modelling (Fischer, 2001; Petrelli & Convertino, 2014).

This research is concerned mainly with capturing the 'right' thing. Selecting which data is used and employed in the model depends on the goal of the application and the output required. It can include personal information (Kobsa, 2001) such as users' names, ages, interests, skills, knowledge, preferences and dislikes, or even data about their behaviour and their interactions with different systems. There are different designs in user modelling, though a mixture of them is usually beneficial.

- Static user models: Static models are the primary type of user model. As the name indicates, the model is always static and does not change; once data is collected they are normally not changed again. Changes in user's information do not affect the model and no learning algorithms can be used to alter the model (Fischer, 2001; Hothi & Hall, 1998).

- Dynamic user models: Dynamic models allow a more active representation of users' preferences. This type of model can adapt to changes in the users' interests or their interactions with the model. This dynamic adaptation helps in meeting the different needs of the users (Fischer, 2001; Hothi & Hall, 1998; Girolami & Kabán, 2003).

- Static and Dynamic (SaD) user models: We can easily imagine SaD as a hybrid modelling technique. It can be more common to allow the modeller to move from using just a one-level standard static user model, which uses static unchangeable information, to more thorough two-level models. The result is a static model with a dynamic model overlaid, which accommodates the user's preference alterations and various interactions with the system (Hothi & Hall, 1998).

There is another methodology to categorise user modelling, which considers profile management. There are three main mechanisms for achieving user modelling and profile management (Fan & Poole, 2006). However, the hybrid approach was added and followed afterwards (McBurney et al., 2009; Godoy & Amandi, 2005).

- Explicit modelling: The user is the main active component in the modelling process. It depends completely on them to set their personal information and manage their interests, which assures data precision. However, managing an entire set of preferences manually means putting the burden on the user to carry out preference management responsibilities (Joerding, 1999). In other words, the user must update their profile whenever new content or new services are encountered. This is the only behaviour available to keep users' interests up to date.

  This approach engages the user in the whole task and places the onus on the user. This is the main shortcoming of this mechanism as it undermines the strength of user modelling. It can often lead to a sparse preference set and hence an inaccurate model (McBurney et al., 2009).

- Implicit modelling: This approach is considered the other extreme in the user modelling process. It primarily uses various techniques for monitoring and learning user preferences without engaging the user directly. It employs learning techniques as a substitute for user control of the system. The system tends to maintain the user profile on behalf of the user, which depends mainly on the intelligence of the system. This may affect the information accuracy and, accordingly, the model performance (Rich, 1983).

- Hybrid implicit and explicit modelling: This methodology combines the advantages of the above-mentioned approaches (implicit and explicit user modelling) to help overcome their limitations.

The benefit of the hybrid approach is the minimal burden on the user; however, care must be taken to provide some method of user control. Without such functionality, the user cannot alter system behaviour to reflect new situations or behaviours in a rapid way. Therefore, for more successful systems, a hybrid approach built on implicit modelling must be employed where possible, but it should also provide a mechanism through which the user can manually manipulate their preferences and take final control (McBurney et al., 2009; Niederée et al., 2004; Potey, 2014).

2.2. **SCRUTABILITY**

One classifying technique for personalisation systems lies in describing them with respect to the system controllability, that is, as controlled vs. uncontrolled models. Controlled personalisation makes the user take control of the system. On the other hand, uncontrolled personalisation would not allow the user to influence the adaptation process (García Barrios et al., 2005).

The main concept of scrutability lies in ensuring users have the capability to control their personal information and its use in a personalisation environment (Assad et al., 2007). A model is described as scrutable whenever it enables the user to examine it in order to determine exactly what is modelled. Users can investigate the data as well as the processes and functions that use that data (Holden & Kay, 1999).

The actions of the user to directly or indirectly alter their model give them control over how the system executes, leading to a more adaptable and scrutable environment (Staikopoulos et al., 2012). Therefore, scrutable control over user models may have the potential to improve personalisation services (Kay, 2006). Gaining a better understanding of how the system behaves can enhance the synergy between the system and the user, where the user can directly experiment with the system to determine what happens (Lum, 2007). Scrutability allows users to check the correctness and validation of their model. Because individual user preferences change over time, it is important to allow users the ability to correct any inconsistencies they see in their model (Lum, 2007).

There are a number of examples in existing research of involving the user and supporting their control over the user model, for example, Kay et al. (2002); Hampson et al. (n.d.); Assad et al. (2007); Roll et al. (2005); and Müller (2004).

These are, briefly, some of the motivations behind this study (Kay, 2006):

- The user’s right to see the personal information the computer holds about them in a user model.
- Enabling users to correct the model’s mistakes.
- Enabling users to have control over the user model.

- Giving users the option to know the way that the model performs the personalisation.
- Supporting users to be more self-aware and avoid self-deception, because their user model mirrors their real actions.
- Motivating users to share user model data because they trust its meaning and use.

It is important that systems allow users to have an active role in the way their personal data is used. One way to achieve this is to design the user model in such a way that the user can, but is not mandated to, directly interact with and inspect the model, possibly correcting incorrect or inappropriate data. This interaction includes the user maintaining control of both what is modelled about them and how it is used.

2.3. **MACHINE LEARNING**

The focus of this study is the Decision Tree (DT) based Machine Learning model. In the coming two sections we are going to discuss two Machine Learning (ML) techniques: interactive ML and interpretable ML. The discussion will be in the context of DT-based models.

2.3.1. INTERACTIVE MACHINE LEARNING

Originally the term *Machine Learning* was defined as the ‘artificial generation of knowledge from experience’ (Holzinger, 2016). It can be used efficiently in modelling real-life data and converting it into computational models. However, in these models, users – who are practically the domain experts – are hardly involved in the development process. Applications or models that require user involvement are not usually satisfied with employing ML (Holzinger, 2016).

In Automatic (or classic) Machine Learning (AML; Holzinger, 2016), the work goes through iterations that each includes: data gathering by developers, selecting features to represent data, data preprocessing, data transformation, model training, iteratively tuning the parameters of the modelling algorithm, tweaking features, and then model evaluation. According to the results of the model evaluation phase, the developer or the practitioner decides whether the model will go into further iterations. These iterations include many of the steps above. Furthermore, the modelling process depends on an iterative investigation in the domain space. This is mainly applied by the technical developers, keeping the user out of the loop (Amershi et al., 2014).

Generally, the ultimate goal of Machine Learning models is to develop algorithms/systems which can automatically learn from data. This learning process includes extracting knowledge and making predictions and decisions without human intervention.

One of the disadvantages of such black-box approaches is that they are resource intensive and data hungry. Additionally, black-box approaches have the least reliable approach when it comes to safety and critical domains (Holzinger, 2016). The other enormous drawback of black-box approaches is lack of transparency, as users often cannot tell why a decision has been made. This does not foster trust and acceptance among end-users (Holzinger, 2016). Trust between the user and a system may be of particular importance when that system is tailoring its behaviour based on the personal information of that user, as is the case for personalisation systems.

One of the drawbacks of most ML-driven models is that users require in-depth knowledge of artificial intelligence and statistics to use them effectively. As such, one of the most sought-after goals in Machine Learning is to allow non-experts to interactively train Machine

Learning algorithms. To raise the understanding between the user or the agent and the practitioners, a new methodology was cultivated. This is referred to as *interactive Machine Learning* (Gutzwiller & Reeder, 2017).

Interactive ML is defined as ‘algorithms which interact with agents and can optimise their learning behaviour through this interaction, where the agents can be humans’ (Holzinger, 2016; Holzinger et al., 2017). Hence, interactive Machine Learning puts the ‘human in the loop’ to enable what neither a human nor a computer could do on their own, and the human expert is seen as an agent directly involved in the system evolution (Holzinger, 2016).

In contrasting the dynamicity of this system with classic ML, interactive ML demonstrates more rapidity in updating models with user changes. This is because the system changes itself automatically in response to user input in the modelling process (Amershi et al., 2014). The rapid pace of this loop (updating the model and getting the subsequent results) enables the user to interactively examine the consequences of their modifications (Amershi et al., 2014). In other words, we can say that this improves the model scrutability and makes it more user understandable and adaptable.

Since we can integrate human-in-the-loop (or the involvement of a human) directly into the algorithm, interactive ML approaches can be of particular interest to solve problems where we lack big datasets, deal with complex data and/or rare events, or where automatic ML suffers from insufficient training samples (Holzinger et al., 2017).

In a study by R. S. Gutzwiller and J. Reeder, sixty-six percent of participants chose the interactive ML plans over black-box ones, and their trust in these choices overall was moderate to high. When given the choice between the two model types, it was found that users believed interactive ML generated better behaviours than black-box models. This demonstrates that a truly glass-box interactive ML with a human-in-the-loop intelligent system is required to understand the user's preferences and to be able to discriminate between relevant and irrelevant features, just as we humans can do (Gutzwiller & Reeder, 2017).

Interactive Machine Learning seeks to enable humans to directly interact with Machine Learning algorithms by way of online feedback or demonstrations of behaviours (Harrison & Riedl, 2016). There are several examples of involving the user in different stages of Machine Learning modelling. Most studies are more concerned with engaging the user by retrieving their feedback (Holzinger, 2016; Rahman et al., 2007; Jones et al., 2009).

Such interactive Machine Learning approaches have made advances and had practical successes in many different application domains, for example, health informatics (Holzinger, 2016), medical image retrieval (Rahman et al., 2007), image processing (Fails & Olsen, 2003), autonomous robots (Gutzwiller & Reeder, 2017), cyber-physical systems (Schirner et al., 2013), image segmentation, gesture-based music (Amershi et al., 2014), and many other industrial applications.

2.3.2. Interpretable Machine Learning

Generally, an explanation is an answer to a 'why' question (Miller, 2017). For example, 'why did the insurance company refuse to insure my car?' This kind of inquiry can be answered by interpretable Machine Learning (Molnar, n.d.). Here, we are using this rather simple definition of Machine Learning interpretability from Miller (2017): 'It is the degree to which a human can understand the cause of a decision'. If the user can comprehend an ML-driven model, then this model can be referred to as an interpretable model (Lipton, 2016).

Explaining a prediction means mainly presenting textual and/or visual artefacts. These artefacts are able to interpret the model behaviour by delivering a qualitative illustration of the relationship between the instance components and the model predictions. Interpreting a model and explaining the output prediction are important in gaining human trust, which is what allows them to use Machine Learning models effectively (Ribeiro et al., 2016). Understanding data can be considered critical, as can understanding the model (Van-Belle & Lisboa, 2013) and its internals. The goal is to deliver all this information to humans in an understandable manner (Gilpin et al., 2018). The better this explanation, the easier it is for a user to comprehend why certain decisions (predictions) were made (Tamagnini et al., 2017).

The concept of interpretability is applied in many studies, such as research into clinical and biomedical decision support systems (Van Belle et al., 2012; Zycinski et al., 2012). A subset of ML learning algorithms is used for implementing the interpretable model, and common types in this group are linear regression models, logistic regression models, and Decision Trees. This is the straightforward way to implement interpretability (Molnar, n.d.); however, there are other approaches that could be used.

Some studies have implemented interpretability by using more than one ML algorithm, and the results of these different algorithms give the sense of data (Kernel methods for interpretable Machine Learning of order parameters). Other implementations were done by involving the user in the model design and implementation (Abras et al., 2004), but this needs the user to be an expert in the domain.

Other work has aimed to produce interpretable predictive models by building decision lists (series of if-then statements), and this method can describe a high-dimensional feature space (Letham et al., 2015).

However, this approach would only work effectively where the number of features is low, hence it is not feasible in the case of text classification problems.

Local or Global?

In Machine Learning interpretability, methods can be classified according to several criteria. This classification method is concerned with the locality of the explanation – whether it is local or global (Tamagnini et al., 2017; Doshi-Velez & Kim, 2017). Does the interpretation output explain a single instance of data (prediction) or the whole data (entire model behaviour)?

Local Interpretability

In essence, local interpretability entails the reasons for a single instance (Kim et al., 2018). For building an explanation for a specific decision, the interpretation model can zoom in on this instance and examine what kind of prediction the model makes for this input and why this was done (Doshi-Velez & Kim, 2017). We can use the example of apartment price, which might not depend linearly on the apartment's size. By taking a specific apartment of 100 square meters, we can find that the value changes by increasing or decreasing by 10 square meters and there are other factors that affect the price decision. The local distribution of the target variable may act more effectively. It may be derived linearly or monotonically on one or a number of features rather than having a complex dependence on the features. This is the reason for considering local explanations to be more accurate than global explanations (Molnar, n.d.).

Global (Holistic) Interpretability

Generally, holistic interpretability implies illustrating the behaviour of the model as a whole (Doshi-Velez & Kim, 2017). Explaining the model globally needs two things: knowledge about the algorithm, and the data. The main goal of this level of interpretability is understanding how the model works as a whole, taking into consideration features and each of the learned components like weights, parameters, and structures (Molnar, n.d.).

Moreover, interpreting the decision for multiple instances can be explained by either using methods for global model interpretability or single instance explanations. The global methods can be applied by taking the group of instances, treating them as if they were a complete dataset, and using the global methods on this subset. The single explanation method can be used on each instance and listed or aggregated afterwards for the whole group (Ruping, 2006).

Model-specific or Model-agnostic?

In this classification method, ML models are classified according the relation between the real model and the explanation model. Does the interpretation output demonstrate the existing model and its components, or does it see it as a black-box and explain its behaviour separately?

Model-agnostic explanation.

This approach creates a separate model to provide interpretation for an existing one (Du et al., 2019). It does not have control over the details of the ML model when creating the explanation. Rather, it sees the model as a black-box and depends solely on the model prediction (Plumb et. al, 2018).

Ideally, it analyses feature input and output pairs. Such interpretation is constructed after the model has been trained (post hoc). This type of explanation is feasible in several Machine Learning models (Plumb et al., 2018; Ribeiro et al., n.d.).

Model-specific explanation.

This explanation method is limited to specific model classes (Plumb, Molitor et al., 2018). The interpretation here is achieved by building a self-explanatory model that can include interpretability in its structures (Du et al., 2019). The interpretation here is built mainly by exploiting model-specific properties (Plumb et al., 2018). It usually concludes the explanation depending on the structural design and internal parameters of the model (Du et. al, 2019). This is also called a *white-box* interpretation. Using a white-box method eases the user incorporation and feedback in systems (Ribeiro et al., n.d.).

The family of this type of model includes Decision Trees, rule-based models, linear models, and so on. Inherently interpretable models have an accurate and undistorted explanation; however, this may influence the model performance (Du et. al, 2019).

2.4. **EVALUATION**

Generally, in the context of a Machine Learning-driven model, the evaluation usually assesses the model performance. However, this research is concerned with two aspects: the model performance and the model interpretation. The assessment of each of them depends completely on the nature of the model. This step will be discussed in detail in the following sections.

2.4.1. EVALUATION OF MACHINE LEARNING MODEL

This research is concerned with a specific ML modelling algorithm, Decision Tree (DT)-based modelling. When it comes to the DT-based model, the first question that arises is how accurate the model is, especially when it works on forthcoming cases. DT model evaluation is essential because one should be certain that the output decision will be reliable and efficient. Before mentioning what categories of measures there are, the metrics used for performance evaluation need to be discussed. A metric for DT performance can have more than one meaning. Sometimes the performance is assessed by speed and in other cases by the grown tree size; however, in most cases, the model is measured by accuracy-based metrics (Baykara, 2015). Below are some metrics that are used and their definitions (Han et al., 2011).

- Accuracy-based: These are different measures that can demonstrate the performance of a model. Since accuracy-based metrics give the most realistic and calculable results, they have dominated the evaluation techniques.
- Speed: This is also known as computational cost: how much it costs during construction and use of the model.
- Robustness: This measures the model reliability as well as the correctness of the resulting predictions, especially when the model encounters either noisy data or data with missing values.
- Scalability: This measures how well the model performs when given large amounts of data.

Only the accuracy-based measures are going to be explained, since these are the measures that are going to be used in this research. There are four important terms that we need to start with in order to understand some concepts in the evaluation metrics (Han et al., 2011; Baykara, 2015).

1. *True Positives* (TP): The cases in which we predicted YES, and the actual output was also YES.

2. *True Negatives* (TN): The cases in which we predicted NO, and the actual output was NO.
3. *False Positives* (FP): The cases in which we predicted YES, and the actual output was NO.
4. *False Negatives* (FN): The cases in which we predicted NO, and the actual output was YES.

There are several accuracy-based validation techniques. In the context of this study, we are going to explain some of them: confusion matrix, accuracy, precision, and recall.

Confusion Matrix

Accuracy-based measurements are formed on top of the confusion matrix. It is also known as the coincidence matrix, classification matrix, or contingency matrix. It is a matrix that describes the complete performance of the model (Han et al., 2011; Baykara, 2015). The confusion matrix (shown in Table 2-1) is simply a table m by m where each column and row shows how many instances of some class were labelled as another class. These labelled classes can be the same class as themselves or another class. The numbers along the diagonal from the upper-left corner to the lower-right represent the correctly classified number of instances. The number m is directly proportional to the number of classes there are in the dataset. All the accuracy-based measures are based on this matrix and derived from it (Baykara, 2015).

Table 2-1
Confusion matrix

	Classified Positive	Classified Negative
Actual positive	TP	FN
Actual negative	FP	TN

Accuracy

In Machine Learning, the term *accuracy* usually means classification accuracy. It is the ratio of the number of correct predictions to the total number of input samples. Equation (1) shows how to calculate accuracy (Sahli, 2020).

$$(1) \qquad \text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

Precision

Precision is the number of correct positive results divided by the number of positive results predicted by the classifier, as shown in equation (2). The number of positive results predicted by the classifier is the summation of correct positive results and false positive results (Han et al., 2011; Baykara, 2015).

$$(2) \qquad \text{Precision} = TP/(TP+FP)$$

Recall

Recall is also known as the true positive rate or hit rate. Recall is the number of correct positive results divided by the number of all positive instances, that is, all samples that should have been identified as positive (Flach, 2003). This is shown in equation (3).

$$(3) \qquad \text{Recall} = TP/(TP+FN)$$

2.4.2. Evaluation of Machine Learning Interpretation

Principally, Machine Learning interpretation offers a human-understandable explanation for model behaviour as well as for the relationship between the instance components and the model results. (Ribeiro et al., 2016). Though the research covered interpretable ML, including techniques, procedures and application, the visions on the interpretable ML evaluation perspective are still rather limited (F. Yang et al., 2019).

Unlike conventional ML evaluation that mainly relies on model performance, interpretable ML evaluation focuses on the quality of the explanations. The quality of explanation cannot be easily handled and benchmarked (Wohlin & Andrews, 2001), since ML interpretability is a very subjective concept and a domain-specific notion; therefore, there is no specific definition for the evaluation. In other words, it is necessary to study the model domain for each specific problem (Carvalho et al., 2019). As per the state of the art, there is no rigid description for interpretation evaluation methods, although the research community has begun to work on this (Murdoch et al., 2019).

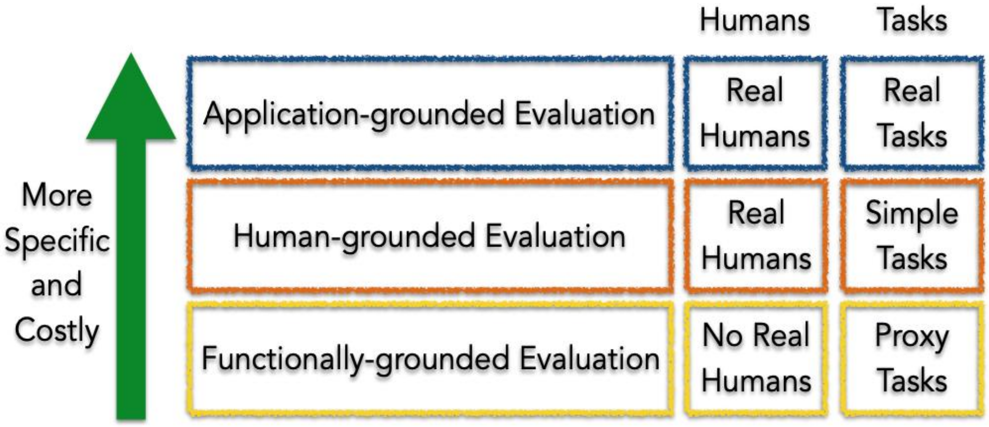
Some model classes are generally amenable to human understanding or, in other words, inherently interpretable, for example, Decision Trees, rule lists, and decision sets (Freitas, 2014; Lage et al., 2018). Within these model classes, there likely still exist some models that are easier for humans to utilise than others, such as shorter Decision Trees rather than longer ones (Lage et al., 2018; Maimon & Rokach, 2014).

Generally in ML, there exists a taxonomy of model evaluation, so the model interpretation and its evaluation should match the corresponding contribution (Doshi-Velez & Kim, 2018). There is more than one categorisation system for interpretation evaluation. Doshi-

Velez and Kim (2018) mentioned a categorisation methodology that will be considered in this study. There are three interpretation evaluation methods: application-grounded, human-grounded, and functionally-grounded. Figure 2-1 shows that these three groups range from task-relevant to all-purpose. Here they are discussed.

Figure 2-1

Taxonomy of evaluation approaches for interpretability (Doshi-Velez & Kim, 2018)



Application-grounded Evaluation:

The main concept is performing a real task experiment and evaluating it using domain experts (humans). This evaluates the quality of an explanation in the context of its end task. This type of experimentation may be conducted either with the exact application task or with a simpler or a partial task. This totally depends on the application. Though this type of evaluation is not an easy metric, it instantly assesses the system and hence its performance concerning the main objective of the system. This results in strong evidence of application success (Ribeiro et al., 2016; Doshi-Velez & Kim, 2017; Molnar, n.d.).

Human-grounded Evaluation

This evaluation is achieved by building simpler experiments that are evaluated by humans. These experiments should maintain the essence of the target application. This evaluation type is appropriate in cases where the real task is challenging. The human here can be a layman user. This opens the experiment to a bigger number of target users and incurs less expense, since there is no need to pay domain experts. Generally, the human-grounded evaluation technique is appealing when it comes to assessing the concepts of the quality of an explanation (Doshi-Velez & Kim, 2018). Ideally, the approach here relies only on the quality of the explanation regardless of whether 1) this interpretation is the model itself, 2) it is a post hoc interpretation of a black-box model, or 3) the model output is correct (Molnar, n.d.).

Functionally-grounded Evaluation

This evaluation technique does not include human involvement; it uses a proxy for explanation quality (Doshi-Velez & Kim, 2018). This type of evaluation is suitable if subject experiments need time and money. It is also appealing if there is a class of models or regularisers that were validated before (Ribeiro et al., 2016; Doshi-Velez & Kim, 2017; Molnar, n.d.).

In the case of human-grounded or application-grounded evaluations, a subjective evaluation can be conducted. Subjective metrics mainly rely on the knowledge and expertise of the humans involved. The key advantage of this type of evaluation is that it is easier to conduct – whether through interviews or questionnaires – however, it can be less exact and more difficult to draw conclusions from (Wohlin et al., 2000). This will be discussed in more detail in the next section.

2.4.3. SURVEY DEVELOPMENT

A survey is defined by Groves et al. (2011) as 'a systematic method for gathering information from (a sample of) entities for the purpose of constructing quantitative descriptors of the larger population of which the entities are members'. The word 'systematic' is used in this definition to differentiate surveys from any other way of collecting information. The expression 'a sample of' indicates that sometimes surveys target everyone in a population and sometimes just a sample.

For a long time, building a questionnaire was considered an art, but over the past years, considerable research has proved that it is a science. There is a lot of science behind designing a successful questionnaire, and we discuss the pitfalls and best practices below.

Designing a questionnaire is a multistage process that entails bearing in mind lots of details. It is somehow a complicated process, as the survey can question a certain topic in many degrees of detail. Questions can be written in various ways and in different orders, which may affect how people interpret later questions (Cox & Cox, 2008). Here are the guidelines for building and conducting a perception questionnaire:

Step 1. Define the objectives and target population (or sample of)

- Describe the goals.
- Outline the ultimate use of the questionnaire results.
- Define target addressees.

Step 2. Draft the questions

- Draft simple and clear questions (this will be detailed later in this section).

- Make sure that respondents can report any problems.

Step 3. Conduct and accommodate the survey

Step 4. Data collection

- Make sure that the sample size will result in indicative conclusions.
- Pick the data collection approach: interviews, internet surveys, email surveys, etc.

Step 5. Run the survey

- Send a letter of invitation to participate in the survey.
- A good letter helps maximise the response rate.

Step 6. Analyse the results

The key point that should be taken into consideration is how to write a good question. A survey is said to be beneficial when it gathers accurate data. And to say that the information is accurate, questions must be written precisely and not be open to many interpretations. Ensuring as much specificity as possible when crafting questions will ensure the actual intent is reflected in the question. This evades any misunderstanding from the respondent side, which reduces answering time as well as potential respondent frustration. If the qustions are not specific enough, the answers will not be comparable and the results will lead to misleading conclusions (Cox & Cox, 2008; Fowler & Cosenza, 2008). Here is a checklist for drafting good questions:

1. The expected answers to the question should help meet the main objectives of the questionnaire.
2. The language should be as simple as possible.
3. Ask one question at a time.

4. Questions should be clear and precise.
5. The answer choices or scales must be clearly understood by the respondents.
6. The survey should be brief enough to ensure that respondents focus on the whole survey.

2.5. Analysis

In user modelling, we can gain the benefit of the hybrid approach (implicit and explicit modelling) in lessening the burden on the user as well as offering some means of user control. This approach in implementing models enables the user to efficiently change the system behaviour whenever there is new input. Classic Machine Learning can be beneficial in building the implicit part of the model, while we can employ interactive ML techniques to build explicit user control.

The major issue with interactive ML techniques is that they require humans to understand the domain and interact with these algorithms. It is a challenge to improve the ability of non-experts to train Machine Learning algorithms (Harrison & Riedl, 2016). So we are utilising interpretable ML in order to develop a glass-box explainable model. This is to enable the user to understand the model behaviour and how important features affect the output decision.

We can say that employing interactive ML in building a user model enhances the users' sense of control over the model and improves the users' trust. Hence we can implement scrutability in a Machine Learning-driven user model. Moreover, interpretable ML could be employed to implement another aspect of the model scrutability, which is model understandability.

Since we are concerned with Decision Tree-based models, the appropriate interpretation method here is the white-box interpretation, so the approach that will be followed in this research is the global and model-specific explanation.

By employing both interactive and interpretable ML techniques, we can implement the scrutability factors mentioned in section 2.2.

For evaluation, each part of this study (interactive ML and interpretable ML) should be handled with the appropriate technique. The Interactive Machine Learning (Decision Tree-based model) part will be assessed with quantitative metrics (accuracy, precision, recall). The other part of this study is the model interpretation section. Since the model here will be delivered to layman users, the appropriate evaluation technique for this interpretation is human-grounded evaluation. This will be handled through conducting a subjective assessment.

The subjective evaluation will be delivered through an assessment questionnaire. Although there is more than one type of survey activity, this research focuses on surveys that have the following characteristics: 1) Information is gathered primarily by asking people questions, 2) information is collected from only a subset of the population to be described – a sample, 3) data is gathered from layman users, and 4) the questions will be delivered to the respondents via email.

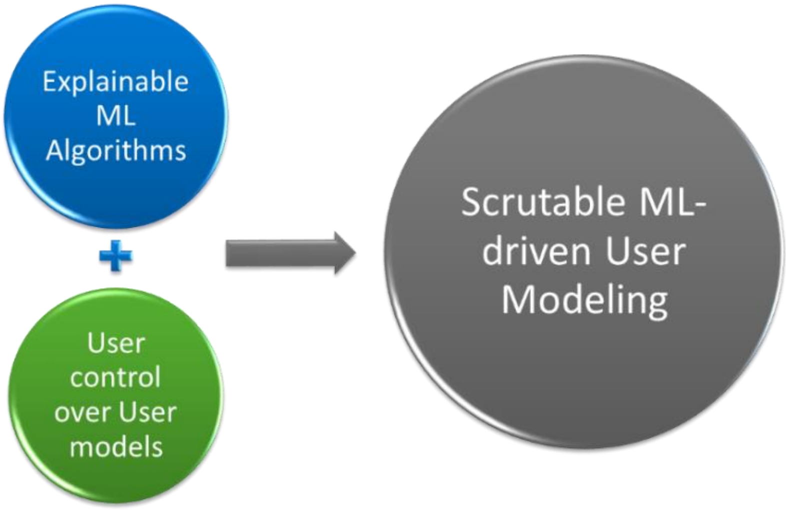
3. Experimental Approach

3.1. Problem Definition

As mentioned in the state-of-the-art chapter, user modelling is, fundamentally, saying the right thing at the right time in the right way, and scrutability is an important requirement of the modelling. This allows users to maintain control of what is modelled about them and improve their trust in the model.

Machine Learning has been employed for user modelling, which typically attempts to mimic the user's behaviour. Generally, users can get good predictions and suggestions with acceptable accuracy by utilising ML to develop such a smart system. Moreover, it can be trained on data from a user's history, but the data usually has limited coverage of the user's activities and interests.

Figure 3-1
UM-IML


Explainable ML Algorithms
+
User control over User models
Scrutable ML-driven User Modeling

Since ML-driven user models are considered black-box, the major concern is user understanding. Users tend not to be interested in an unexplainable model. Thus, we observed that users would gain added value (regarding scrutability) by interpreting the model and engaging in the ML modelling process.

In this research, we are studying the approach of employing Machine Learning related techniques – specifically interactive Machine Learning and interpretable Machine Learning techniques – to implement a scrutable user model. This approach is called SUM-IML (see Figure 3-1). The challenge here is maintaining the scrutability of the model in an ML environment as well as considering the understandability of the model. The development carried out for this work can be sketched in the following goals:

- To explore the possibilities and techniques of interactive Machine Learning in order to develop an approach to support the research question (supplementing ML intelligence with user control).
- To explore the possibilities and techniques of interpretable Machine Learning in order to implement human-understandable explanations for model behaviour. This is to support the research question (demonstrating model behaviour).
- To develop and implement a case study that mimics the proposed approach.
- To explore the suitable technologies and datasets to help build the required case study.

3.2. PROPOSED APPROACH

The approach considered in this study will start by building a case study that validates the idea of SUM-IML. The case study is simply a

Decision Tree-based user model. The model here is a Machine Learning model that is constructed to reproduce the concept of SUM-IML. It is trained on the user's personal data and hence it can predict their reaction. The system basically reads user's email messages and learns user preferences through reading these emails. On an incoming new message, the model can predict the recipient's behaviour and whether the user will be interested in this message or not.

We are working on this target by exploring user data. Here, the main data source to be used in building users' profiles is their email messages. This is used to construct the initial user model. These emails include inbox messages, sent messages, messages in folders, and deleted messages. This large set of user data contains important and unimportant information for the participant. We need to explore this huge number of facts and capture the vital chunks. This selected data should indicate the user's likes and dislikes. This information then goes through the learning phase in which we can build a personalised user model.

The model presents the prediction to the user and asks the user to evaluate this prediction. The model would include this user feedback in the next training iterations. This involvement would benefit from user feedback on the model. Then the model behaviour and results will be explained to the user. The model explanation will be evaluated by users. This is in order to assess how far this explanation is human-readable and understandable.

The important aspect here is investigating the effect of user involvement on the ML process. This involvement is simulated by considering user feedback. The next step is discussing the model results and comparing them with and without the user engagement and analysing how far the (simulated) user feedback affects the model results.

Since the constructed model here is an ML-driven model, the user will not understand the details of this black-box. This is the main challenge in implementing scrutability. To attain this, the model will provide a human-understandable explanation for the output prediction.

When thinking about evaluating the SUM-IML approach, the other point that should be taken into consideration is assessing user understanding and satisfaction with the model interpretability. This would go through field study experimentation by providing the users with a qualitative explanation for the model behaviour and the causality relationship between the modelling components, followed by an assessment of their understanding.

Evaluating the Machine Learning model is an essential part of any work. The evaluation here includes a number of aspects: the first is concerned with assessing the model performance; different ML algorithms have different performance metrics. Most of the time, classification accuracy is used to measure the performance of the model; however, it is not enough to truly judge it. The literature review (section 2.4) discussed all the evaluation techniques that will be used.

3.3. METHODOLOGY

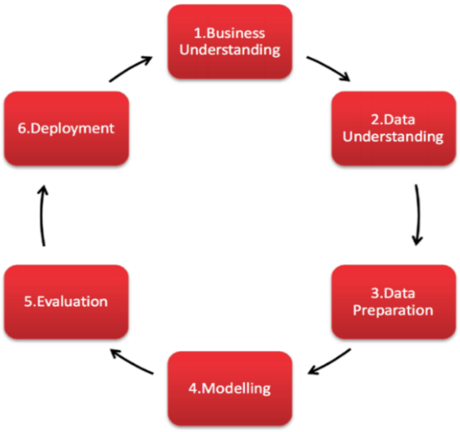
Building predictive data analytics solutions for this kind of problem involves a lot more than choosing the right Machine Learning algorithm. To maximise the chances of success, a structured project management methodology needs to be applied to the development plan. One of the most commonly used methodologies is CRISP-DM.

CRISP-DM stands for Cross-Industry Standard Process for Data Mining (Kelleher et al., 2015). It is a methodology that helps in structured development planning. It is flexible and useful when implementing an analytics model. CRISP-DM is basically the sequence of tasks shown in

. These instructions can be carried out in a different order or repeated, depending on the nature of the model at hand.

Choosing an ML technique can be a difficult process, and in this research, the CRISP-DM framework is employed to help make this task simpler. This is because CRISP-DM gives a clearer structure and set of norms that can be used.

Figure 3-2
CRISP-DM flow (Wirth, 2000)



As shown in Figure 3-2, the six key steps of the predictive analytics project lifecycle as defined by CRISP-DM are: business understanding, data understanding, data preparation, modelling, evaluation, and deployment. In the following, each phase is outlined briefly:

- Business Understanding

The project starts by understanding the problem at hand and the main objectives. This knowledge is then translated to a Machine Learning problem definition and an initial project plan.

- Data Understanding

This phase includes data collection and activities. This includes understanding the data to realise the data quality and the expected difficulties. In Machine Learning modelling, there is a critical dependence between business understanding and data understanding.

- Data Preparation

The data preparation phase is the logical step after understanding the available data. It covers all actions that can provide the final dataset that will be used in the modelling process.

- Modelling

Typically, many techniques can be applied to the same problem type. In the modelling phase, the target is to select the appropriate methodology with the optimum parameters. These selections depend directly on the data details and the problem definition.

- Evaluation

Before proceeding to deploy the model for user usage, it is crucial to thoroughly evaluate the model and its results and whether the main objective is achieved.

- Deployment

Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable ML modelling process. Generally, it is important to realise which activities will be needed to make use of the constructed model.

Mostly, CRISP-DM instructions can be carried out in different orders or repeated, depending on the nature of the model at hand. These steps in the context of this work will be illustrated in more detail in section 3.3.

3.4. **ENRON EMAIL DATASET**

Since the context in this research is about user modelling of personal data, the target was to find the appropriate personal data to build the model on. When searching for personal data to work on, privacy becomes a big issue. Privacy is a crucial point that should be taken into consideration when trying to find personal data. To avoid privacy violations, publicly published data is needed.

The other aspect that should be studied here is whether the data is sufficient: it is important to evaluate how large the training set must be to achieve a sufficient estimate of model performance. Such data is usually limited due to the disclosure of the data included.

The searching task for suitable data for the problem at hand ended with the discovery of the Enron email dataset (*The PAL Framework*, 2017). The Enron dataset is one of the few available datasets that suits the problem of this study. The dataset was primarily gathered and prepared by the CALO Project (Cognitive Assistant that Learns and Organises). The Enron dataset was first publicly published by the Federal Energy Regulatory Commission (Cohen, 2015).

To address privacy and confidentiality concerns, the CALO project has made some modifications to the original data. For example, all email attachments were removed, some messages have been deleted as requested by certain employees, and invalid email addresses were converted to something different, for example, no_address@enron.com (Cohen, 2015). More detail about the data will be presented when implementing the baseline architecture in the Step 2: Data Understanding section.

This step is concerned with selecting the subset of all available data that we will be working with. There is always a strong desire for including all available data, that the maxim ‘more is better’ will hold. We need to consider what data we need in order to address the problem at hand.

The Enron corpus contains data from a number of users, mostly senior management of Enron. Since this study is concerned with personalising user modelling, and in order to investigate the data from the user perspective, this research worked on one user only. How this one user was selected will be discussed in section 3.1.

3.5. **DEVELOPMENT ENVIRONMENT**

The proposed solution (SUM-IML) is a scrutable Machine Learning model. Machine Learning modelling includes several phases of development that start with data analysis and processing and end with model building and evaluation. To develop this scrutable model, interactive ML and interpretable ML techniques were the targets applied. They are ML methods that are implemented in several Machine Learning programming libraries.

We tended to find integrated Machine Learning techniques and programming languages such as R and Python in the search space. After investigating different development options, Python was selected as the core programming language, based on the availability of multiple open source ML libraries. As an open-source programming language, there is the option to choose from a wide range of open-source Python frameworks and development tools. It has several libraries that help in data preparation (this includes cleaning, transformation, and normalisation). Examples of these libraries are NumPy[1], and SciPy[2].

Scikit-learn[3] API (Pedregosa et al., 2011) is an open-source library that provides an implementation of a wide range of Machine Learning algorithms and data preparation methods. It is a simple and efficient tool that is mainly built on NumPy, SciPy, and matplotlib. Scikit-learn is commercially usable (BSD licence).

Skater [4] is a unified framework to enable Machine Learning interpretation for all forms of predictive model. It is an open-source Python library designed to demystify the structures of black-box models (Choudhary et al., 2018). If a developer can obtain inputs and use a function to obtain outputs, Skater can reveal the internal decision policies in a human-interpretable way.

[1] http://www.numpy.org/
[2] https://www.scipy.org/
[3] http://scikit-learn.org/stable/
[4] https://github.com/datascienceinc/Skater

4. EXPERIMENTS AND RESULTS

The ultimate goal here is to answer the research question, or in other words, to assess if it is feasible to implement a scrutable Decision Tree-based user model. As mentioned earlier, the target is developing a case study that demonstrates the proposed contribution (SUM-IML methodology).

The approach followed here starts by building a user model – using the Decision Tree algorithm – that is trained on the user's mailbox. These emails include inbox messages, messages in folders, and deleted messages. The model learns the user preferences from this data. Then, on an incoming new email message, the model will predict whether the user will be interested in this email. All the experiments work on the same user model but target different points in the research.

The work in this model will go through three phases: 1) Constructing the base architecture; this is the baseline of the research development in this study. Details will be presented in section 4.1. 2) Initial user control: this experiment simulates user control in the form of user feedback. The setup and results of this part are shown in section 4.2. 3) Explaining the model behaviour: this experiment interprets the model globally to the user in a simple, readable, and understandable manner. This part is discussed in section 4.3.

The second and third stages are concerned with implementing model scrutability. By this design we are maintaining and investigating the model scrutability as follows:

1. User control: The experiment simulates the user control over the model by giving user feedback as inputs.
2. User understandability: It explains the model behaviour to the user in a simple, readable, and understandable manner.

4.1. **Baseline Architecture**

The main objective of this phase is to build a baseline framework. The next two sections (4.2 and 4.3) show two experiments that depend on the model that is built here. This user model is mainly a Machine Learning model that is learning user preferences so it can predict user actions.

As we mentioned in the previous chapter, the user model in this research is trained on the user's data. This data is in the form of their email messages. Through this training, the model can learn the user preferences and the cases that the user becomes more interested in. On an incoming message, the model will be able to predict whether the user is interested in this message.

The approach followed here is CRISP-DM (Kelleher et al., 2015). Implementing this methodology helps in planning to develop the solution effectively. It goes through a six-stage lifecycle: business understanding, data understanding, data preparation, modelling, evaluation, and deployment. These steps are expanded in the framework of this work, below.

Step 1: Problem Understanding

This research is mainly concerned with studying the implementation of scrutable user models when using Machine Learning. This study focuses on two main factors in scrutability: user control and model understandability. The target here is to build a personalised scrutable predictive model that provides two things: 1) predictions to a user and 2) an explanation for this decision.

In this case study, user preferences are learnt from their history of interactions collected from their email messages. The model can then predict the importance of any incoming email messages.

Step 2: Data Understanding

It is critical to find the right data to build the right model – the model that will answer the research question. In this research, we are working on the Enron email dataset (Cohen, 2015), described in section 3.4. The Enron dataset is one of the few available datasets that suits the problem of this study. Such data is limited due to the confidentiality of the data included.

This step is concerned with selecting the subset of all available data that we will be working with. There is always a strong desire for including all data that is available, that the maxim 'more is better' will hold. We need to consider what data we need in order to address the problem at hand.

The Enron corpus contains data from many users, mostly senior management of Enron. The number of email messages differs from one user to another, and since this study is concerned with personalised user modelling, this research worked on one user only.

The selected user was chosen according to four criteria:

- There should be a sufficient number of email messages to carry out Machine Learning modelling.
- The user data should include different types of email messages (for example, inbox, sent, deleted, and emails in folders).
- These messages are personal and non-personal data.
- There is not much missing data.

These conditions should be covered in the selected mailbox to achieve the research aim. The selected mailbox consists of 1,512 instances (email messages) with about 18,000 features. This data is labelled 'zero' and 'one'. One means the user is interested in this instance, and otherwise is zero. This labelling was done manually according to several criteria and took into consideration that the model is not converted to a rule-based model.

Step 3: Data Preparation

After selecting the data, we needed to study how we are going to use the data. This preprocessing phase was mainly about getting the selected data into a form that was ready for the modelling phase. The data preprocessing was divided into four stages; formatting, cleaning, sampling, and transformation:

- Formatting:

We formatted the data we had selected in a form that was suitable to work with. In the beginning, the email messages were in separate text files, which was not an easy format to work on. Through a programming phase for formatting the data, all the email messages are now in one comma-separated file (.csv).

- Cleaning:

As quality data is a crucial prerequisite for predictive models, we need to avoid 'garbage in, garbage out'. It is a mandatory task to pre-process the text to be ready for Machine Learning modelling, and we had to work on this important step carefully. In the case of text data (email bodies), it is called text cleaning.

Sometimes there are defective data instances: incomplete, noisy, or inconsistent. Text cleaning includes removing the unwanted data that may mislead the model training. This includes stripping whitespace, stop words, numbers, and punctuation. URLs and links were also deleted. If there was a message like:

deal! Let's go shopping and then have dinner
Check this restaurant: www.pizzahut.com
Bye

After the cleaning phase, it would look as follows:

deal Let’s go shopping and then have dinner Check this restaurant Bye

- Data Transformation:

This step is also referred to as feature engineering. As per the problem we are targeting, we are working on analytics for the text data in the user’s email messages. This is the process of transforming text into a single canonical form. It is primarily text normalisation, which is an important step, as it guarantees data consistency before operations are performed on the data.

Most of what we are going to do with the language relies on first separating or tokenising words from running tokenisation text. For example, if the input is: ‘Friend, lend me your car’, the output will be: ‘Friend’, ‘lend’, ‘me’, ‘your’, ‘car’. Next comes text stemming, which is a simpler version of lemmatisation in which we mainly strip suffixes from the end of words to arrive at the common root forms. For example ‘running’ becomes ‘run’.

Step 4: Modelling

The Modelling phase of the CRISP-DM process comes when the Machine Learning work occurs. We developed an ML model that is trained on the user’s mailbox, with the objective of constructing a user model that represents the user’s interests (i.e., it learns a set of signals that help identify whether a new incoming email will be of interest to the user or not).

The focus in this study is on the Decision Tree as a modelling algorithm. There are many reasons to choose this model, and our main ones are:

1. Decision Trees usually simulate human-level thinking, making it easy to understand and make sense of the data.

2. Decision Trees enable the user to figure out the logic of the data to interpret, as they are cumulative and hierarchical, which makes more sense to the user.
3. A Decision Tree is a relatively white-box model that can be easily interpreted.

Step 5: Evaluation

In the context of an ML-driven model, the evaluation usually assesses the model accuracy. However, in addition to model performance, this research is concerned with model scrutability. Each of the experimentation phases works on a certain part of this problem, so the assessment of each phase depends completely on its main objective. This step will be discussed in detail in each experiment.

Step 6: Deployment

This is the last phase, covering the work carried out to successfully integrate a Machine Learning model into the process within an organisation. This phase is not applicable in our research.

Finally, by going through the five stages, it can be said that in this part, the baseline framework is implemented as starting point for the coming experiments.

4.2. **EXPERIMENT 1: INITIAL USER CONTROL – CONSIDERING USER FEEDBACK**

4.2.1. EXPERIMENT HYPOTHESIS

The projected hypothesis of this experiment is that user feedback can enhance model performance. As we mentioned earlier, user feedback to a user model provides the user with control over the system and its performance in the next iterations. So this experiment is assessing the importance of user feedback in user modelling.

4.2.2. Experiment Objective

It would be effective to involve the user input in the modelling process. In this part, the model is taking into consideration the human input at the end of the modelling process. The main objective is employing Machine Learning for building a user model and simulating user feedback over the model.

4.2.3. Experiment Setup

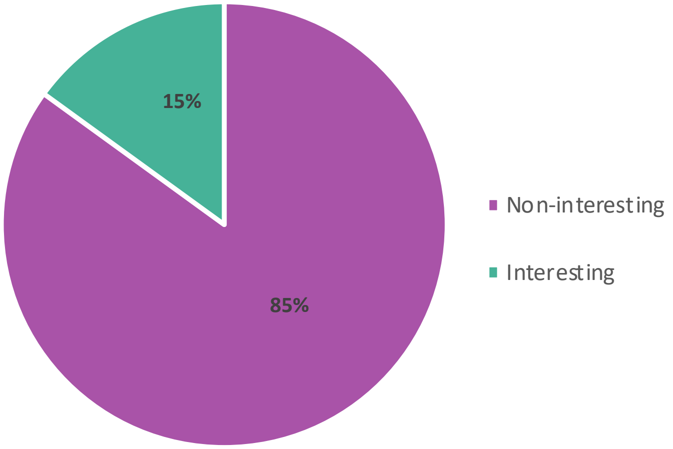
The model here is built over the baseline architecture, where the Decision Tree classification model was trained on the user's mailbox. It consists of 1,512 instances with about 18,000 features. This data is labelled 'zero' and 'one', where one means the user is interested in this message, and zero means otherwise.

Through supervised learning, the Decision Tree-based model is trained twice. The first time is before the user feedback, and the second time is after involving the user feedback. First, the model is trained on 70% of the data (randomly selected). It is then executed to predict the values of a further 15% (randomly selected); then the labels are used to evaluate the model performance. These labels are used for simulating user feedback. The selection of the two parts (70% and 15%) was randomised. This step was repeated 100 times, in order to avoid auto-correlation and to ensure that the data was not biased.

The second training phase uses 85% of the data (70% plus 15% user feedback). After combining these two parts, the model is executed to predict the last 15%. This part is used for validating user performance after including user feedback, in order to benefit from the user input. As in the first iteration, the selection of the two parts (85% and 15%) was randomised and rerun 100 times to make sure that the data was not biased.

The data in this experiment is represented in unbalanced classes, meaning that the individual classes do not contain the same number of elements. Here, there are only two classes: The first is non-interesting messages (emails that the user won't be interested in checking), and this includes 1,279 instances (85% of the whole data). The second class is for interesting messages (those that the user would be interested to know about), containing 233 instances (15% of the whole data), as shown in the following chart (Figure 4-1).

Figure 4-1
Dataset distribution



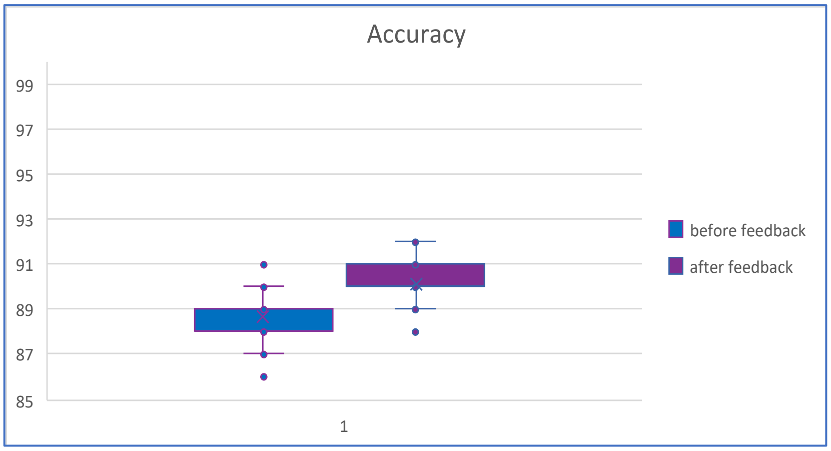
4.2.4. Experiment Evaluation and Results

By the discussed setup, the model prediction would be evaluated before and after the user feedback. As mentioned, the model will be run 100 times to get the distribution of the model results. These predictions would be assessed in more than one way, which we will discuss here.

Accuracy

In this experiment, we ran the model a hundred times and represented all the results of the model in the following chart. The chart shown in Figure 4-2-2 is a boxplot that illustrates the data distribution of these hundred runs of the model. The mean before user feedback was 88.5%, while after the feedback it became 89.7%. This highlights the improvement of the model prediction results after taking the user input in the feedback stage.

Figure 4-2
Accuracy improvement before and after feedback



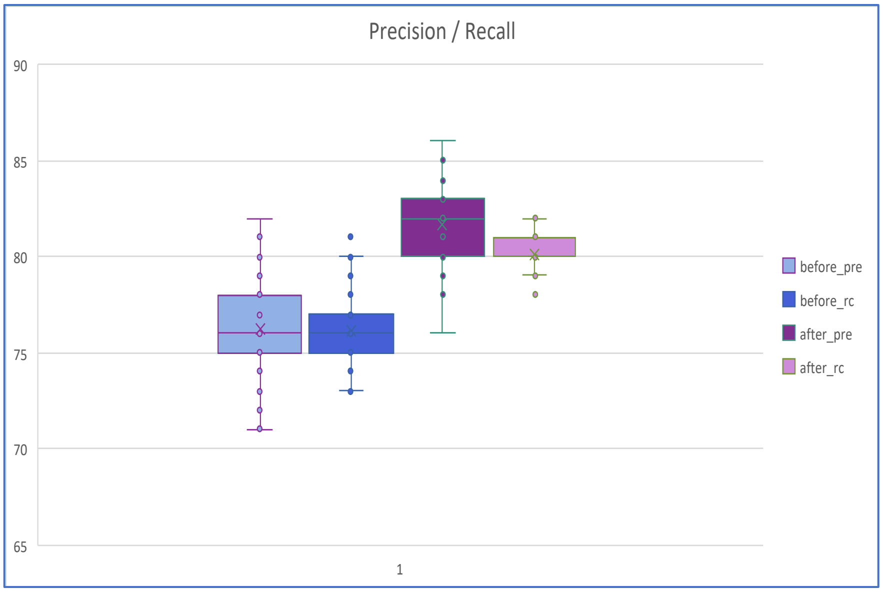
The imbalanced distribution of data in this experiment results in an evaluation challenge. The accuracy here cannot be considered the only good measure of the model performance. A model that just predicts ‘0’ every time will yield an 85% accuracy even though it is a bad model that does not yield any insight or scientific advancement, even though ‘85% accuracy’ sounds like something good. So the baseline of accuracy in this problem is 85%. Hence the model will be evaluated with more approaches to figure out the sense of improvement.

Precision and Recall

Since the data in this experiment is imbalanced, the recall metric would be more indicative than the accuracy. The improvement in recall values shows that the prediction in the minor class specifically has improved.

In Figure 4-3, The light blue and the dark blue each represent the precision and recall before the user feedback, while the light purple and dark purple represent them after the feedback. Both precision and recall have been improved; however, recall is more important in this context.

Figure 4-3
Precision & recall improvement



After calculating the precision and recall of the model, we found that both are significantly improved after considering the user feedback, and this is shown in Table 4-1. It shows the average precision and recall of the model results (for the hundred runs). In these results, the recall was 76% then became 80%, showing a gain of 5.26%.

Table 4-1
Precision & recall before and after feedback

	Before feedback	After feedback
Precision	76%	82%
Recall	76%	80%

Confusion Matrix

By running the model 100 times, we calculated the average of each of the values: TP, FN, FP, and TN. Table 4-2 represents the numbers before taking the user feedback, while table 4-3 shows the numbers after the user feedback was taken as an input. The two tables show that after user input, the model's ability to identify interesting messages (minority class) rose by 11.8%, and this is considered a significant improvement in this context.

Table 4-2
Confusion matrix result before user feedback

	Classified Positive	Classified Negative
Actual positive	94%	41%
Actual negative	6%	59%

Table 4-3
Confusion matrix result after user feedback

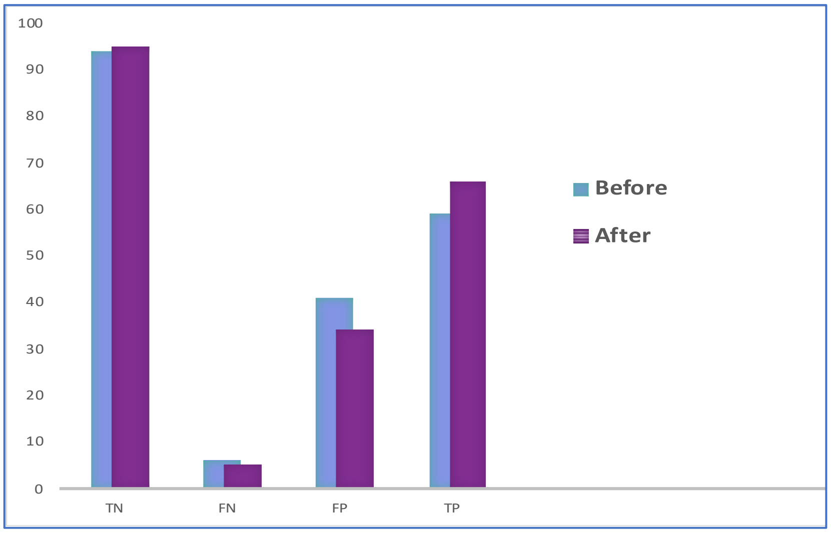
	Classified Positive	Classified Negative
Actual positive	95%	34%
Actual negative	5%	66%

Table 4-4
Confusion matrix result after user feedback

	Classified Positive	Classified Negative
Actual positive	95%	34%
Actual negative	5%	66%

The following chart (Figure 4-4) represents the difference between the model performance with versus without user feedback. The blue bars represent the values before considering user feedback, while the purple bars show the results after taking the user input. The values are the average of the 100 runs. The chart shows a general improvement in the model results.

Figure 4-4
Confusion matrix improvement



In conclusion, this experiment simulated a simple user involvement in the modelling process. Hence we can see an initial part of user control over a user model and the data used in this process. Despite only showing a slight improvement in the model accuracy, the results showed significant improvement in other metrics.

4.3. EXPERIMENT 2: EXPLAINING THE MODEL BEHAVIOUR

4.3.1. EXPERIMENT HYPOTHESIS

The anticipated hypothesis of this experiment is that we can deliver a global human-understandable explanation for the behaviour of a Decision Tree-based model to enable model scrutability.

4.3.2. EXPERIMENT OBJECTIVE

Further to the step of developing the model, another common problem in the field of ML is that ML models are not understandable to their users. It is becoming important to not just understand the ML model itself but to also understand why it produces certain outputs in certain cases or why it made a certain decision. This study aims to implement the model explanation in a human-understandable form and evaluate this interpretation.

The objective here is to interpret the model globally and explain its behaviour to the user. This presentation provides a qualitative understanding of the relationship between the model's components (e.g., words in the text) and the output prediction. This experiment is considered an extension of the previous one. It aims to construct a system that takes the model built earlier as an input and return a human-understandable explanation for it. This explanation will be a global interpretation for a Decision Tree-based model.

4.3.3. EXPERIMENT SETUP

This study aims to implement and evaluate techniques for representing Machine Learning models in a user-understandable form and to produce an easy-to-understand explanation. The experiment here explains the model behaviour in an understandable representation and highlights the most important features that affect the model prediction (i.e., the data attributes on which the model bases its decision to classify an email as interesting or not interesting).

The interpretation can be considered a model on its own. It is built using Skater (mentioned in section 3.5), a Python package for interpreting predictive models (either via post-hoc evaluation or rule extraction). Skater can unpack the internal mechanics of arbitrary models and use model inputs and a function to obtain outputs. It relies internally on NumPy, Pandas, Scikit-learn, and the DataScience.com fork of the LIME package.

The interpretation model concludes the most important features that affect the model decision. It then constructs a simple Decision Tree with a small number of features in order to present the model behaviour in a human-understandable presentation. Each node in this tree contains a set of information that illustrates the model factors and how they work. The simplified tree is then represented to the user with a brief illustration.

This experiment will be carried out in a field study. The examiners will evaluate the system explanation using a qualitative assessment task. The questionnaire was crafted as follows:

Step1. Define the objectives and target population

The objective of the questionnaire is to assess whether the model interpretation was human-understandable and to gauge the user opinion of this type of explanation.

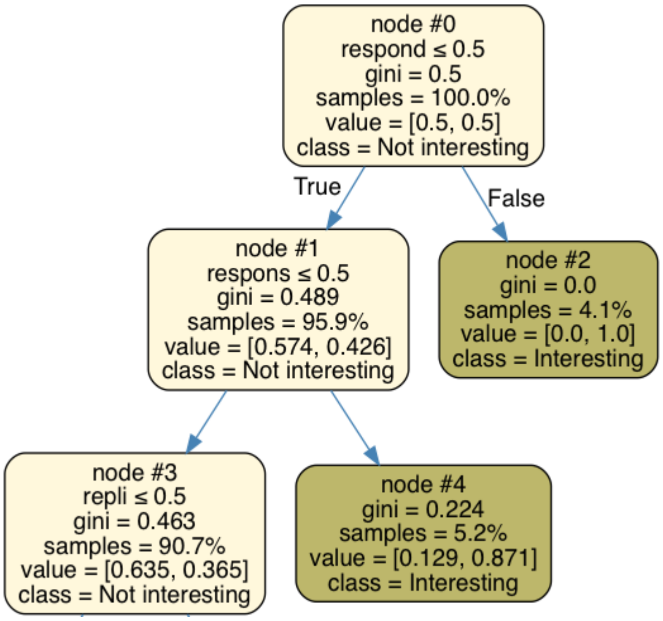
According to the nature of the problem, the addressees are layman users. The main characteristics of these users are that they don’t need to have any technical background and they use their emails frequently. The age ranged from 25 to 45 years old, and they have all attended tertiary education.

Step 2. Design the survey and draft the questions

The questionnaire (seen in Appendix A) is composed of three parts. The first part is an introduction to the survey, which highlighted the problem and the purpose of such a model and its explanation as well as the drive of the assessment.

The second part of the survey is an example demonstrating the concept of the Decision Tree. This section includes a very simple Decision Tree (a three-level tree) and some attributes to illustrate how the Decision Tree works and how decisions are being taken. The following figure is an example of what was represented to the respondents. The data in each node represents the feature or the factor used in this part of the classification method (sender email address, word from the message subject, or word from the message body) and some descriptive information about this feature. The importance of the features descends from the upper to the lower nodes.

Figure 4-5
Part of the model explanation



In the tree shown in Figure 4-5, the first line in each box (node) is its indentifier (ID). Let's look at the first node in more detail:

- node: is the node number (ID).
- respond: is a Y/N question. The answer to this question affects the next step, either to the right or the left direction. 'respond' is a word (feature affecting the decision making) found in the email body. Its occurrence (whether 0 or 1) is the question.

  'respond' is a truncated word that stands for respond, responding, and responds. If one of these words was found in the body of an email (*occurrence* = 1), then this email would be classified as an interesting one.

  When the occurrence of the word 'respond' is (1) ≥ 0.5, then this means that this is an 'interesting' email to the user. Otherwise, the tree will go in the other direction to find other important features that may help in classifying the email (decision making). So, it will go through the next question found in node#1.

- gini: is the importance of this feature (the word 'respond' in this case).
- samples: number of data entries that contain this word (feature).
- value: is the ratio of the two classes when divided by this feature.
- class: is the class name, either 'interesting' or 'not interesting'

Generally, the attributes illustrated here explain how the Decision Tree works and how the decision is being taken. The actual tree used in the implemented model is the same concept but a little deeper. By going through it, the user will be able to interpret the behaviour of the model and how it takes the final decision. The third and final part of the survey is the questions section. This will be discussed in the next point.

Drafting the questions

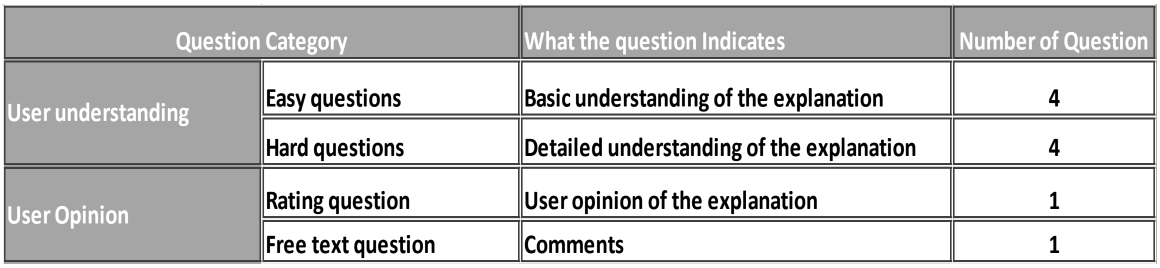
The assessment task (attached in Appendix A) was implemented to evaluate two things: 1) user understanding, and 2) user opinion. As we mentioned earlier, the survey needed to be brief enough to keep the respondents' focus throughout the whole survey, so the questionnaire here entailed 10 questions. They were divided into two parts, as shown in Table 4-4.

The first part was composed of eight multiple choice questions to assess whether the examining user had understood the model explanation. To assess different levels of user understanding, the questions varied from easy to hard. The four easy questions showed the general understanding of the model, then the four hard ones came to show whether the user had understood the fine details of the model. Answering at least four questions out of eight showed a basic understanding of the provided explanation. If the user couldn't answer the four easy questions, they did not qualify to pass the assessment. So a 50% success rate was required for the user to pass.

The second part was concerned with the user's general satisfaction with the provided explanation. This part was composed of two questions. The first one obtained user opinion through explicit rating questions. The user was asked how sufficient the explanation was for them. The answers ranged gradually from *strongly agree* to *strongly disagree*. The second question asked the user to write a free text commenting on the whole experience.

We took into consideration that the language used in writing the questions would be as simple as possible. We asked one question at a time in a clear and precise way, and ordered the questions to ensure user engagement in the survey.

Table 4-5
Questionnaire design



Question Category		What the question Indicates	Number of Question
User understanding	Easy questions	Basic understanding of the explanation	4
	Hard questions	Detailed understanding of the explanation	4
User Opinion	Rating question	User opinion of the explanation	1
	Free text question	Comments	1

Step 3. Conduct and accommodate the survey

We sent the survey to few users as a draft and collected their feedback. After addressing their comments, we came up with the final version of the questionnaire.

Step 4. Data collection

The data collection approach followed here is email surveys.

Step 5. Run the survey

The final version of the survey was sent to fifty participants. We sent them an invitation letter and the details of the survey. Thirty people out of fifty responded to the questionnaire.

Step 6. Analyse the results

The results of this experiment are of two types: assessment of the user's understanding and user opinion. The next section will detail the results of the experiment and what they reflect.

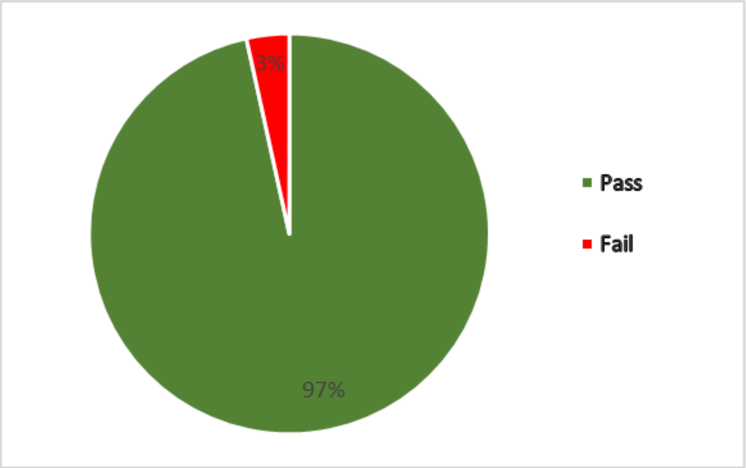
4.3.4. Experiment Evaluation and Results

The model explanation was represented to a number of users. Thirty users examined it and evaluated its understandability. The analysis of the results of this assessment is divided into two parts: user understanding and user opinion.

User understanding

User understanding was assessed through eight multiple-choice questions. These questions varied from easy to hard (see Appendix A for more detail) to assess different levels of user understanding. Easy questions show the general understanding of the model, while the hard ones show whether the user understood the fine details of the model. The users answer the questions, and the marks show how far the user had understood the model and answered the questions correctly. Answering at least four questions out of eight showed basic understanding of the provided explanation. If the user couldn't answer the four easy questions, then they did not pass the assessment. Therefore, a 50% success rate showed that the user had passed.

Figure 4-6
General marking

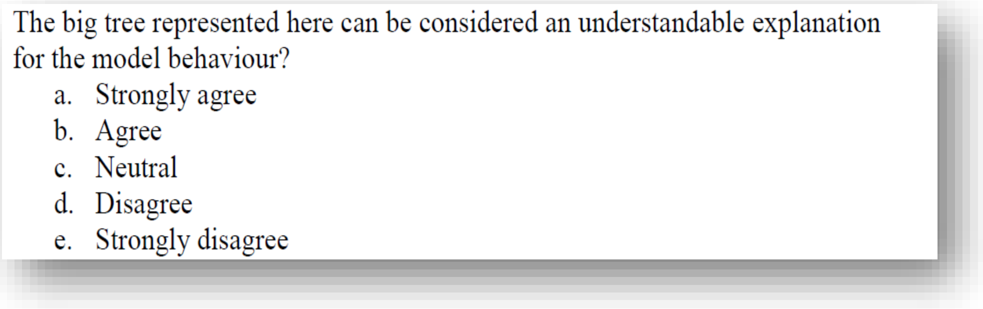


The pie chart shown in Figure 4-6 shows that 97% of the users (29 users) succeeded in answering the assessment task, while only 3% (1 user) failed. The scores vary from A to F. A represents the range from 80% success to 100%. B is for 70% to 80%, while C is from 60% to 70%. F shows the failures (a score of under 50%). The chart here shows that a large portion of users got a score of A (between 80% and 100%). The high scores reflect the acceptable readability and understandability of the model explanation.

User Opinion

At the end of the assessment task, participants were asked about their opinion of the provided explanation. There were two forms for questioning this review. The first one was in the form of a rating, and Figure 4-7 states the question that was asked to the users. The user rated the model interpretation and if it was enough to understand the model behaviour.

Figure 4-7
User satisfaction question

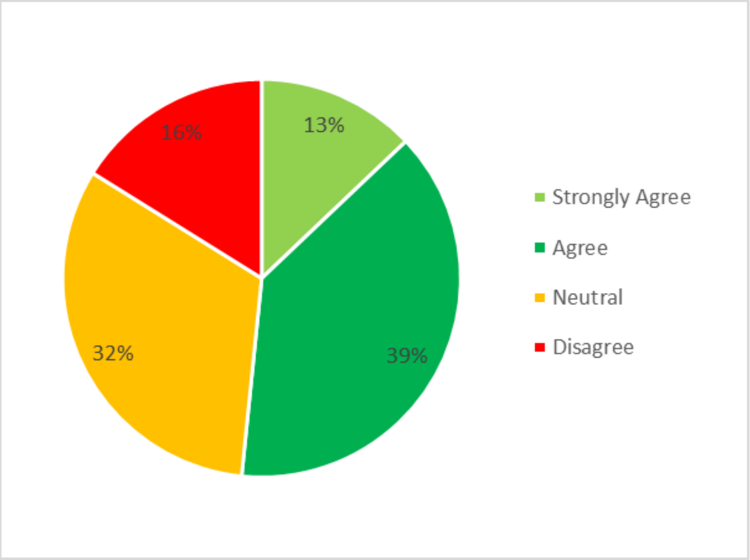


The big tree represented here can be considered an understandable explanation for the model behaviour?

a. Strongly agree
b. Agree
c. Neutral
d. Disagree
e. Strongly disagree

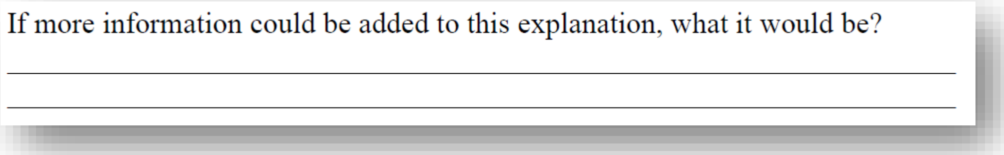
Figure 4-7 presents the actual results of this question. The answers varied from *strongly agree* to *strongly disagree*. Figure 4-8 shows that the largest portion rated it as *Agree*. We can see that the *Strongly disagree* is 0%. The total of *Strongly disagree* and *Disagree* is only 16% of the respondents. This demonstrates a high level of satisfaction on the user side.

Figure 4-8
User satisfaction



The second method for questioning user satisfaction was in the form of a free-text comment. It is highlighted in Figure 4-9.

Figure 4-9
User satisfaction – Comments



If more information could be added to this explanation, what it would be?

____

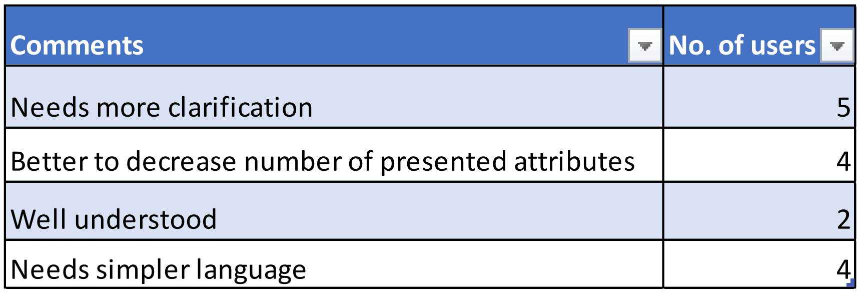
____

As the number of participants involved is not too large, analysing this data was done manually. We searched for some basic factors that could be used in understanding and aggregating the users' comments. These factors are concerned with a number of questions that can be summarised as:

1) Does the explanation need more clarification?
2) Is the number of presented attributes too large?
3) Does the explanation need to be presented in simpler language that can be understood by a layman user?

Table 4-6 shows the analysis of the user comments and opinions of the provided explanation.

Table 4-6
Comments analysis



Comments	No. of users
Needs more clarification	5
Better to decrease number of presented attributes	4
Well understood	2
Needs simpler language	4

4.4. DISCUSSION

The research question in this study centres on the feasibility of implementing a scrutable user model that is a Decision Tree-based ML model. Here we mention the steps of the research in brief, concerning the research questions and demonstrating the objectives and how they were addressed.

The first objective was analysing the state of the art of three main topics: personalisation, user modelling, and scrutability. These are the three main topics that the research lies within. Analysing these fields leads to knowing what the gaps and limitations are. The detail of this objective is referred to in the state-of-the-art section.

After studying the research gap, the technologies and techniques that could address the research problem were analysed. The potential fields are interactive and interpretable Machine Learning – especially the Decision Tree case. This is the second objective. Analysing the state of the art of these two topics was mandatory to examine the problem and find the appropriate solution. To do this, the technologies and techniques of interactive and interpretable Machine Learning needed to be studied.

This thorough studying resulted in a full understanding of the related fields. This understanding enabled us to propose a solution to support the research question: SUM-IML. The idea is simply that applying interactive ML and interpretable ML can maintain model scrutability.

After developing a better vision for the solution, we continued to the next step: designing and implementing the model. The implementation was intended to prove the concept proposed and whether it is feasible.

The design included the three steps of experimentation, which were discussed precisely. The idea behind each step is to implement a part of the SUM-IML, so by the end, we could conclude that the proposal was feasible.

We divided the scrutability into two main parts: user control, and user understanding. Each of the two scrutability aspects was implemented and examined in a separate experiment.

The first experiment was concerned with the user input in the modelling process. This experiment hypothesised that user feedback can enhance model prediction. User feedback at the end of the modelling process was simulated in this experiment. The test showed better results when user input was taken into consideration to retrain the model. Hence we can see an initial part of user control over a user model.

In the second experiment, the hypothesis was that we could deliver to the user a global understanding of the ML model's behaviour, so we could enable scrutability (user understanding aspect) for user modelling. The goal was to interpret the model globally and explain its behaviour to the user, so we can provide the (non-expert) user with a qualitative understanding of the relationship between the instance's components (e.g., words in the text) and the model's prediction. The experiment result showed acceptable readability and understandability of the model explanation.

By the results of these two experiments, we can say that the two aspects of scrutability were enabled in a Machine Learning environment, and it is feasible to have a scrutable ML-based user model.

5. Conclusion

In an information-rich environment managing human attention in user modelling poses a real challenge. Personalised user modelling generally attempts to offer information and services that are customised to meet a user's individual preferences. It can be said that the crucial and limited commodity in such cases is not only to avail the information at any time, at any place and in any form but to lessen the information overload and facilitate access to relevant information in a system. To manage that, we can employ Machine Learning techniques. A vital aspect here is maintaining the model scrutability.

In Machine Learning-driven systems, providing scrutability is a real challenge. This research concerns two main scrutability aspects. The first is user control over the model and providing user feedback to be taken into consideration in the modelling loop. The other is user understanding of the model which is achieved by providing a qualitative explanation of the model behaviour as well as the relationship between model components and model output.

The major contribution of this work is introducing an approach of combining interactive ML and interpretable ML to implement model scrutability. Employing these two ML approaches helps in maintaining the important factors of a scrutable model where assessing the results of each of these phases is an important input.

The approach followed for this research is building a model that demonstrates the proposed approach. The first experiment aimed at building an ML-driven user model that includes the user input by giving their feedback. The model, after taking the user input, was able to identify interesting messages (minority class) significantly by 12% gain, which is considered a significant improvement in this context.

To achieve the user involvement effectively, the user needs to have a qualitative understanding of the model behaviour. This was the target of the second experiment. It concerned building an interpretation model that explains the behaviour of the user model and its results in a human-understandable manner. The experiment involved providing users with an explanation intended to be understood by layman users and assessing their understanding through assessment questionnaire. The results of the survey showed that 97% users understood the provided explanation, and 52% of users indicating that the interpretation was understandable.

In conclusion, the work of this research has shown that it is feasible to maintain model-scrutability in the case of employing ML. Experimentation showed that both user control and understandability could be implemented even in the case of having an ML model.

This study was concerned only with Decision Tree based models. The next step would be working on a wider range of Modelling algorithms. Another aspect is that model scrutability includes many other ingredients. Through this research, we can say that this is a motivating step towards implementing other considerations as well.

To summarise, this thesis featured a research carried out for a Masters degree. It provided detailed discussions regarding the work and experiments that have been carried out so far for this research study, and discussed the experimentation carried out during the research.

REFERENCES

1. Abras, Chadia, Diane Maloney-krichmar, and Jenny Preece. 2004. “User Centered Design.” Design 37(4): 1–14.
2. Amershi, Saleema, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. “Power to the People: The Role of Humans in Interactive Machine Learning.” AI Magazine 35(4): 105.
3. Assad, Mark, David Carmichael, Judy Kay, and Bob Kummerfeld. 2007. “PersonisAD: Distributed, Active, Scrutable Model Framework for Context-Aware Services.” In Pervasive Computing, Springer, 55–72.
4. Baykara, B. 2015. “No Title.” Impact of evaluation methods on decision tree accuracy (Master’s thesis).
5. Van Belle, V., S. Van Huffel, J.A.K. Suykens, and S. Boyd. 2012. “Interval Coded Scoring Systems for Survival Analysis.” ESANN 2012 proceedings, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 07(April): 2007–11.
6. Billsus, Daniel, and Michael J Pazzani. 2000. “User Modeling for Adaptive News Access.” User modeling and user-adapted interaction 10(2–3): 147–80.
7. Blom, Jan. 2000. “Personalization - A Taxonomy.” Conference on Human Factors in Computing Systems (April): 1–2.
8. Bra, Paul De. “Design Issues in Adaptive Web-Site Development.” : 29–39.
9. Buhalis, Dimitrios, and Aditya Amaranggana. 2015. “Smart Tourism Destinations Enhancing Tourism Experience through Personalisation of Services.” In Information and Communication Technologies in Tourism 2015, Springer, 377–89.
10. Carvalho, D V, E M Pereira, and J S Cardoso. 2019. “Machine Learning Interpretability: A Survey on Methods and Metrics.” Electronics 8(8): 832.

11. Choudhary, Pramit, Aaron Kramer, and Datascience.com contributors Team. 2018. “Skater: Model Interpretation Library.”
12. Cohen, William W. 2015. “Enron Dataset.” https://www.cs.cmu.edu/%7B~%7D./enron/ (June 12, 2017).
13. Cox, J, and K B Cox. 2008. “Your Opinion, Please!: How to Build the Best Questionnaires in the Field of Education.” Corw.
14. Daniela Petrelli, Antonella De Angeli, and Gregorio Convertino. 2014. “A User-Centered Approach to User Modeling.” In UM99 User Modeling: Proceedings of the Seventh International Conference, , 255.
15. Dietterich, Thomas G. 1997. “Machine-Learning Research.” AI Magazine 18(4): 97.
16. Doshi-Velez, F, and B Kim. 2018. “Considerations for Evaluation and Generalization in Interpretable Machine Learning.” In Explainable and Interpretable Models in Computer Vision and Machine Learning . , Cham, , 3–17.
17. Doshi-Velez, Finale, and Been Kim. 2017. “Towards A Rigorous Science of Interpretable Machine Learning.” (Ml): 1–13.
18. Du, Mengnan, Ninghao Liu, and Xia Hu. 2019. “Techniques for Interpretable Machine Learning.” Communications of the ACM 63(1): 68–77.
19. Fails, Jerry Alan, and Dan R. Olsen. 2003. “Interactive Machine Learning.” Proceedings of the 8th international conference on Intelligent user interfaces - IUI ’03: 39.
20. Fan, Haiyan, and Marshall Scott Poole. 2006. “What Is Personalization? Perspectives on the Design and Implementation of Personalization in Information Systems.” Journal of Organizational Computing and Electronic Commerce 16(3–4): 179–202.

http://www.tandfonline.com/doi/abs/10.1080/10919392.2006.9681199.

21. Fischer, Gerhard. 2001. “User Modeling in Human--Computer Interaction.” User modeling and user-adapted interaction 11(1): 65–86.
22. Flach, P A. 2003. “The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics.” In Proceedings of the 20th International Conference on Machine Learning (ICML-03, , 194–201.
23. Fowler, F J, and C Cosenza. 2008. “Writing Effective Questions.” International handbook of survey methodology 8: 136–59.
24. Freitas, A A. 2014. “Comprehensible Classification Models: A Position Paper.” ACM SIGKDD explorations newsletter 15(1): 1–10.
25. García Barrios, Victor M, Felix Mödritscher, and Christian Gütl. 2005. “Personalisation versus Adaptation? A User-Centred Model Approach and Its Application.” In Proceedings of IKNOW05 Graz Austria June 29 July 1 (Iicm): 120–27.
26. Gilpin, Leilani H. et al. 2018. “Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning.”
27. Girolami, Mark A, and Ata Kabán. 2003. “Simplicial Mixtures of Markov Chains: Distributed Modelling of Dynamic User Profiles.” In NIPS, , 9–16.
28. Godoy, Daniela, and Analia Amandi. 2005. “User Profiling in Personal Information Agents: A Survey.” The Knowledge Engineering Review 20(04): 329–61.
29. Groves, R M et al. 2011. Survey Methodology (Vol. 561). John & Sons: Wiley.
30. Gutzwiller, Robert S., and John Reeder. 2017. “Human Interactive Machine Learning for Trust in Teams of Autonomous Robots.” 2017 IEEE Conference on Cognitive and

Computational Aspects of Situation Management, CogSIMA 2017.

31. Hampson, Cormac et al. "Dynamic Personalisation for Digital Cultural Heritage Collections."
32. Han, J, M Kamber, and J Pei. 2011. "Data Mining Concepts and Techniques Third Edition." The Morgan Kaufmann Series in Data Management Systems 5(4): 83–124.
33. Harrison, Brent, and Mark O Riedl. 2016. "Towards Learning From Stories : An Approach to Interactive Machine Learning." Aaai.
34. Holden, Sam, and Judy Kay. 1999. "The Scrutable User Model and Beyond." Ninth International Conference on Artificial Intelligence in Education Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling.
35. Holzinger, Andreas. 2016. "Interactive Machine Learning for Health Informatics: When Do We Need the Human-in-the-Loop?" Brain Informatics 3(2): 119–31.
36. ———. 2017. "A Glass-Box Interactive Machine Learning Approach for Solving NP-Hard Problems with the Human-in-the-Loop." : 1–26.
37. Hothi, Jatinder, and Wendy Hall. 1998. "An Evaluation of Adapted Hypermedia Techniques Using Static User Modelling." In Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia, , 45–50.
38. Joerding, Tanja. 1999. "A Temporary User Modeling Approach for Adaptive Shopping on the Web." In Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Toronto and Banff, Canada. Computer Science Report, , 7–99.
39. Jones, Thouis R. et al. 2009. "Scoring Diverse Cellular Morphologies in Image-Based Screens with Iterative Feedback

and Machine Learning." Proceedings of the National Academy of Sciences 106(6): 1826–31.

40. Kay, Judy. 2006. "Scrutable Adaptation: Because We Can and Must." In International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, , 11–19.
41. Kay, Judy, Bob Kummerfeld, and Piers Lauder. 2002. "Personis: A Server for User Models." Adaptive Hypermedia and Adaptive Web-Based Systems 2347: 203–12.
42. Kelleher, John D, Brian Mac Namee, and Aoife D'Arcy. 2015. "Fundamentals of Machine Learning for Predictive Data Analytics."
43. Kim, Suin et al. 2018. "Consistent and Reproducible Direct Ink Writing of Eutectic Gallium–Indium for High-Quality Soft Sensors." Soft Robotics: soro.2017.0103.
44. Kobsa, Alfred. 2001. "Generic User Modeling Systems." User modeling and user-adapted interaction 11(1): 49–63.
45. Koch, Nora Parcus De. 2001. "Software Engineering for Adaptive Hypermedia Systems and Development Process." Development: 371.
46. Lage, I et al. 2018. No Title. Human-in-the-loop interpretability prior.
47. Letham, Benjamin, Cynthia Rudin, Tyler H. McCormick, and David Madigan. 2015. "Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model." Annals of Applied Statistics 9(3): 1350–71.
48. Lipton, Zachary C. 2016. "The Mythos of Model Interpretability." (Whi).
49. Lum, Andrew Wai Kwong. 2007. "Light-Weight Ontologies for Scrutable User Modelling."
50. Maimon, O Z, and L Rokach. 2014. Data Mining with Decision Trees: Theory and Applications (Vol. 81). World scientific.

51. McBurney, Sarah, Nick Taylor, Howard Williams, and Eliza Papadopoulou. 2009. “Giving the User Explicit Control over Implicit Personalisation.” In Procs. of Workshop on Intelligent Pervasive Environments (under AISB’09), Edinburgh, Scotland,.
52. Miller, Tim. 2017. “Explanation in Artificial Intelligence: Insights from the Social Sciences.”
53. Molnar, C. Interpretable Machine Learning. a Guide for Making Black Box Models Explainable.
54. Monk, A. F., and J. O. Blom. 2007. “A Theory of Personalisation of Appearance: Quantitative Evaluation of Qualitatively Derived Data.” Behaviour and Information Technology 26(3): 237–46.
55. Müller, Martin E. 2004. “Can User Models Be Learned at All? Inherent Problems in Machine Learning for User Modelling.” Knowledge Engineering Review 19(1): 61–88.
56. Muller, Martin E. 2004. “Can User Models Be Learned at All? Inherent Problems in Machine Learning for User Modelling.” Knowledge engineering review 19(1): 61–88.
57. Murdoch, W J et al. 2019. Interpretable Machine Learning: Definitions. and applications: methods.
58. Niederée, Claudia, Avaré Stewart, Bhaskar Mehta, and Matthias Hemmje. 2004. “A Multi-Dimensional, Unified User Model for Cross-System Personalization.” In Proceedings of the AVI 2004 Workshop on Environments for Personalized Information Access, , 34–54.
59. O’Keeffe, Ian et al. 2012. “Personalized Activity Based ELearning.” : 1.
60. Pedregosa, Fabian, and G Varoquaux. 2011. “Scikit-Learn: Machine Learning in Python.” …of Machine Learning … 12: 2825–30. http://dl.acm.org/citation.cfm?id=2078195.

61. Plumb, Gregory, Denali Molitor, and Ameet Talwalkar. 2018. “Model Agnostic Supervised Local Explanations.” In Proceedings of the 32nd International Conference on Neural Information Processing Systems, , 2520–29.
62. Potey, Madhuri A. 2014. “International Conference on Information Communication & Embedded Systems (ICICES 2014).” (978): 2–6.
63. Rahman, Md Mahmudur, Prabir Bhattacharya, and Bipin C Desai. 2007. “A Framework for Medical Image Retrieval Using Machine Learning and Statistical Similarity Matching Techniques With Relevance Feedback.” IEEE Transactions on Information Technology in Biomedicine 11(1): 58–69.
64. Ralph, Daniel, and Stephen Searby. 2004. 8 Location and Personalisation: Delivering Online and Mobility Services. IET.
65. Ribeiro, M T, S Singh, and C Guestrin. 2016. No Title. Model-agnostic interpretability of machine learning.
66. Ribeiro, Marco Tulio, U W EDU, Sameer Singh, and Carlos Guestrin. “Model-Agnostic Interpretability of Machine Learning.”
67. Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. “Why Should i Trust You?: Explaining the Predictions of Any Classifier.” In ACM, 1135–44.
68. Rich, Elaine. 1983. “Users Are Individuals: Individualizing User Models.” International journal of man-machine studies 18(3): 199–214.
69. Roll, I., Baker, R. S., Aleven, V., McLaren, B. M., & Koedinger, K. R. 2005. “How Can Users Edit and Control Their Models in Ubicomp Environments ?” User modeling- Springer Berlin Heidelberg (January): 367–76.
70. Ruping, Stephan. 2006. “Learning Interpretable Models.” : 209.

71. Sahli, H. 2020. "An Introduction to Machine Learning." TORUS 1: 61–74.

72. Schirner, Gunar, Deniz Erdogmus, Kaushik Chowdhury, and Taskin Padir. 2013. "The Future of Human-in-the-Loop Cyber-Physical Systems." Computer 46(1): 36–45.

73. Staikopoulos, Athanasios et al. 2012. "AMASE: A Framework for Composing Adaptive and Personalised Learning Activities on the Web." In International Conference on Web-Based Learning, , 190–99.

74. Tamagnini, Paolo, Josua Krause, Aritra Dasgupta, and Enrico Bertini. 2017. "Interpreting Black-Box Classifiers Using Instance-Level Visual Explanations." Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics - HILDA'17: 1–6.

75. "The PAL Framework." https://pal.sri.com/ (June 12, 2017).

76. Van-Belle, Vanya, and Paulo Lisboa. 2013. "Research Directions in Interpretable Machine Learning Models." Esann (April): 24–26.

77. Vellido, Alfredo, José D. Martin-Guerroro, and Paulo J.G. Lisboa. 2012. "Making Machine Learning Models Interpretable." 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) (April): 163–72.

78. Ware, Malcolm et al. 2001. "Interactive Machine Learning: Letting Users Build Classifiers." International Journal of Human-Computer Studies 55(3): 281–92.

79. Wirth, Rüdiger. "CRISP-DM : Towards a Standard Process Model for Data Mining." (24959).

80. Wohlin, C, and A A Andrews. 2001. "Assessing Project Success Using Subjective Evaluation Factors." Software Quality Journal 9(1): 43–70.

81. Wohlin, C, Von Mayrhauser, and H"ost A. “M., & Regnell, B. (2000). Subjective Evaluation as a Tool for Learning from Software Project Success.” Information and Software Technology 42(14): 983–92.

82. Yang, F, M Du, and X Hu. 2019. No Title. Evaluating explanation without ground truth in interpretable machine learning.

83. Yang, Yuping, M Howard Williams, Lachlan M MacKinnon, and Rob Pooley. 2005. “A Service-Oriented Personalization Mechanism in Pervasive Environments.” In Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference On, , 132–35.

84. Zemmouri, Ahlam, and Mohamed Benslimane. 2015. “Hybrid Approach into the Design of a User Model for Adaptive Recommendation System in MOOCs.” 4(1): 282–84.

85. Zycinski, G. et al. 2012. “Discriminant Functional Gene Groups Identification with Machine Learning and Prior Knowledge.” ESANN 2012 proceedings, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (April): 221–26.

APPENDIX A

Machine Learning-based Model Explanation

Introduction

A common problem that faces a lot of email users these days is “Too many daily incoming mails”. The increasing number of emails that arrive every day is making it increasingly harder for users to process them in their entirety. One of the modern ways to combat this problem is to use Machine Learning (ML) to classify whether an incoming email is of interest to the user (classify as *interesting* vs. *non-interesting*). As part of this research, we developed an ML model that that is trained by processing the emails in the user’s mailbox, with the objective of constructing a user model that represents the user’s interests (i.e. learn a set of signals that would help identify whether a new incoming email would be of interest to that user or not)..

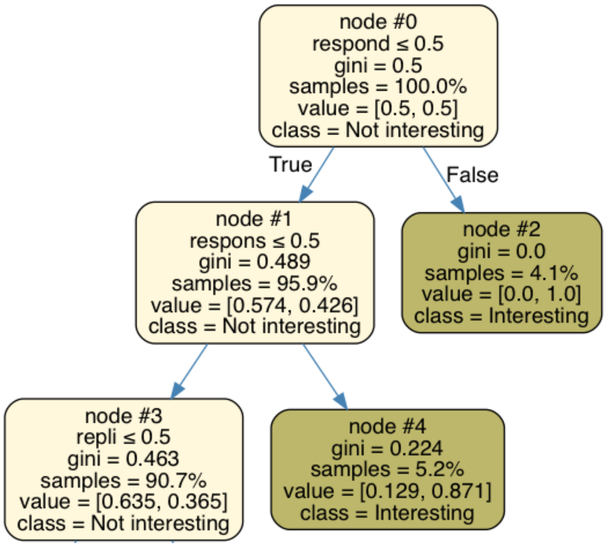
Furthering on the step of developing the model, another common problem in the field of ML (and AI in general) is that ML models are not understandable to the users. It is becoming important nowadays to not just understand the ML model itself, but to also understand why it produces certain output in certain cases or why it made a certain decision. Therefore, one of the main objectives of our research is to develop approaches for explainable ML models. Therefore, this study aims at implementing and evaluating techniques for representing ML models in a user-understandable form and for producing easy-to-understand explanations of the behaviour of ML models.

Based on the above, this experiment presents an explained/explainable ML algorithm and asks questions to verify how far the given explanation is meaningful. In this experiment, the model is constructed using the famous ML algorithm: Decision Tree Classification. The presentation here aims at explaining the model behaviour in an understandable representation and highlighting the most important features that affect the model prediction (i.e. the data attributes on which the model bases its decision to classify an email as interesting or not interesting). Decision Tree is used to identify the next decision with respect to the user's history. It typically starts with a question (or possibility) and branch off into other possibilities. The final result is a tree with decisions (or classes).

The model learns user interests from their history. In this task we are using the user mailbox to reflect these interests. The mailbox consists of 1511 email messages. The data showed that the user is interested in 232 messages of them (15%), while the 85% are not considered interesting to that user. This shows how the data is classified; 'interesting' and 'not interesting' classes. The ML model found that the most important factors that affect this classification are *subject*, *body*, and *sender*.

Example

Here, we can see a simple example that represents how the model works. This is part of a Decision Tree of our model. The data in each node represents the feature or the factor used in this part of the classification method (sender email address, word from the message subject, or word from the message body). This is as well as some descriptive information about this feature. The importance of the features descends from the upper to the lower nodes.



In the tree shown, the first line in each box (node) is its ID. Let's take the first node in more detail.

- node: is the node number (ID)
- respond: is a Y/N question. The answer of this question affects the next step; whether the right or the left direction. 'respond' is a word (feature affecting the decision making) found in the email body. Its occurrence (whether 0 or 1) is the question.

  'respond' is a truncated word that stands for respond, responding, and responds. If one of these words were found in the body of an email (occurrence = 1), then this email would be classified as interesting one.

  when the occurrence of the word 'respond' is (1) ≥0.5, then this means that this is an 'interesting' email to the user.

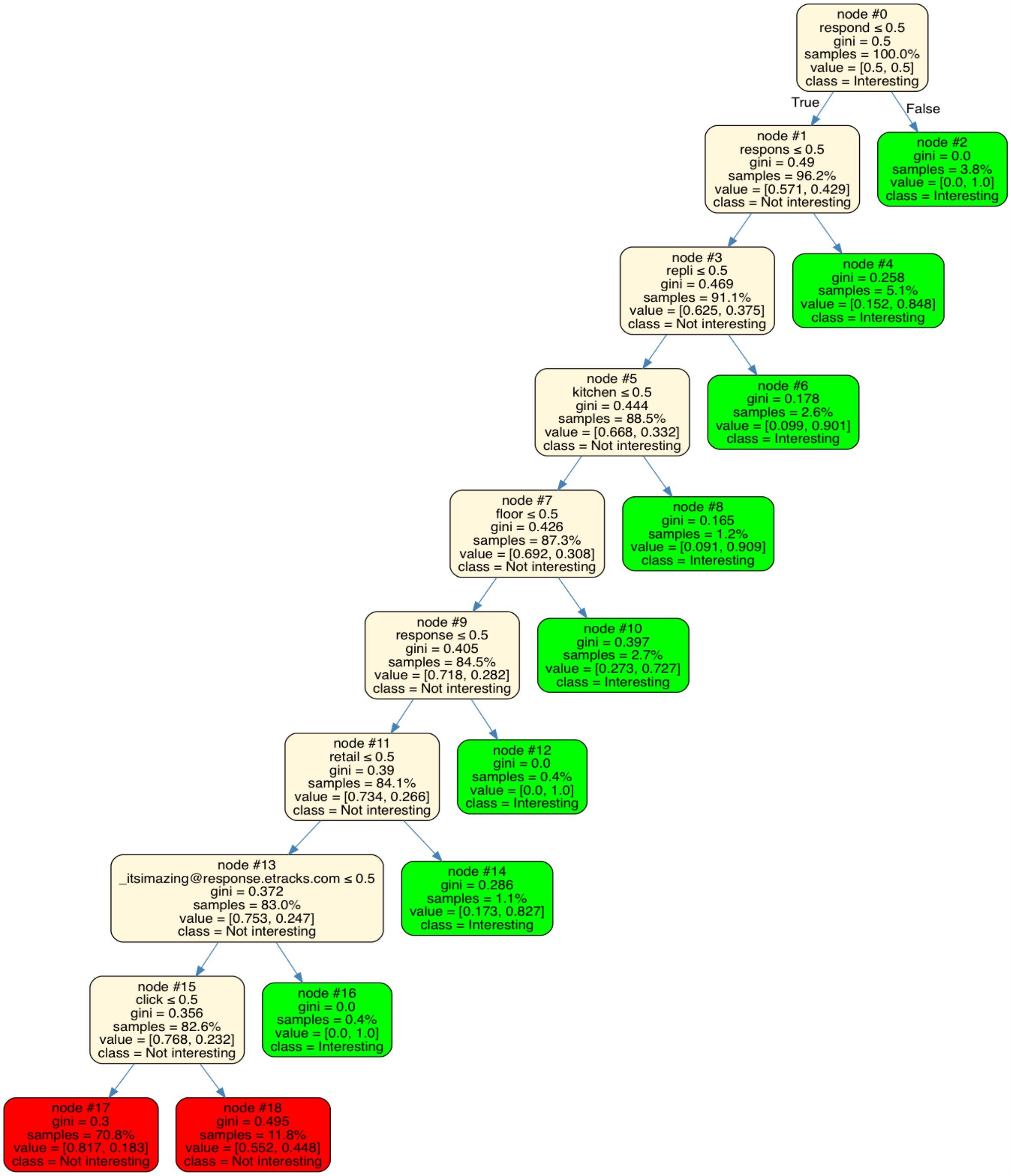
  Otherwise, the tree will go through the other direction to find other important feature that may help in classifying the email (decision making). So, it will go through next question found in node#1.
- gini: is the importance of this feature (the word 'respond' in this case)

- samples: number of data entries that contains this word (feature)
- value: is the ratio of the two classes when divided by this feature.
- class: is the class name; whether ‘interesting’ or ‘not interesting’

Generally, attributes illustrated here explains how the Decision Tree works and how the decision is being taken. Here, we are going to show the actual tree used in the implemented model. It is the same concept, but a little deeper. By going through it, you’ll be able to interpret the behaviour of the model and how it takes the final decision.

Questions

Please take two minutes (maximum) to understand it, then try to answer the following few questions that would not take more than 20 minutes.


node #0
respond ≤ 0.5
gini = 0.5
samples = 100.0%
value = [0.5, 0.5]
class = Interesting
True
False
node #1
respons ≤ 0.5
gini = 0.49
samples = 96.2%
value = [0.571, 0.429]
class = Not interesting
node #2
gini = 0.0
samples = 3.8%
value = [0.0, 1.0]
class = Interesting
node #3
repli ≤ 0.5
gini = 0.469
samples = 91.1%
value = [0.625, 0.375]
class = Not interesting
node #4
gini = 0.258
samples = 5.1%
value = [0.152, 0.848]
class = Interesting
node #5
kitchen ≤ 0.5
gini = 0.444
samples = 88.5%
value = [0.668, 0.332]
class = Not interesting
node #6
gini = 0.178
samples = 2.6%
value = [0.099, 0.901]
class = Interesting
node #7
floor ≤ 0.5
gini = 0.426
samples = 87.3%
value = [0.692, 0.308]
class = Not interesting
node #8
gini = 0.165
samples = 1.2%
value = [0.091, 0.909]
class = Interesting
node #9
response ≤ 0.5
gini = 0.405
samples = 84.5%
value = [0.718, 0.282]
class = Not interesting
node #10
gini = 0.397
samples = 2.7%
value = [0.273, 0.727]
class = Interesting
node #11
retail ≤ 0.5
gini = 0.39
samples = 84.1%
value = [0.734, 0.266]
class = Not interesting
node #12
gini = 0.0
samples = 0.4%
value = [0.0, 1.0]
class = Interesting
node #13
_itsimazing@response.etracks.com ≤ 0.5
gini = 0.372
samples = 83.0%
value = [0.753, 0.247]
class = Not interesting
node #14
gini = 0.286
samples = 1.1%
value = [0.173, 0.827]
class = Interesting
node #15
click ≤ 0.5
gini = 0.356
samples = 82.6%
value = [0.768, 0.232]
class = Not interesting
node #16
gini = 0.0
samples = 0.4%
value = [0.0, 1.0]
class = Interesting
node #17
gini = 0.3
samples = 70.8%
value = [0.817, 0.183]
class = Not interesting
node #18
gini = 0.495
samples = 11.8%
value = [0.552, 0.448]
class = Not interesting

1. In node #13, what does '_itsimazing@response.etrack.com' mean?
    a. '_itsimazing@response.etrack.com' is the sender email address
    b. '_itsimazing@response.etrack.com' is the receiver email address
    c. I don't know
2. What is the colour that denotes the 'interesting' class?
    a. Yellow
    b. Green
    c. Red
    d. I don't know
3. What is the importance of the feature 'kitchen'?
    a. 0.444
    b. 0.275
    c. 0.246
    d. I don't know
4. In node #3, what does 'repli' mean?
    a. A truncated form of replies
    b. A truncated form of replied
    c. a & b
    d. I don't know
5. Assume the following message is sent to the user. Trace the tree to find whether the user will be interested or not.

    Sender: no.address@enron.com
    Subject: GMAT Review available at Enron
    Body:

    Hi, Please review attached requirement document and **respond** to this message with any comments. Once you approve it, I am going to build the application. Thanks, Fangming

    a. User will be interested in this message. It will be located on node #2
    b. User will not be interested in this message. It will be located on node #18
    c. I don't know

6. Assume the following message is sent to the user. Trace the tree to find whether the user will be interested or not.

Sender: phillip.allen@enron.com

Subject: sagewood town homes

Body:

Larry, Just a note to touch base on the sagewood town homes and other development opportunities. It is mentioned that some of the units are the 1308 floor plan. As far as being an investor in a new project, I am still very interested. Call or email with your thoughts. Phillip

   a. User will be interested in this message. It will be located on node #10
   b. User will not be interested in this message. It will be located on node #12
   c. I don’t know

7. Assume the following message is sent to the user. Trace the tree to find whether the user will be interested or not.

Sender: phillip.allen@enron.com

Subject: west desk members

Body:

Dave, Here are the names of the west desk members by category. The origination side is very sparse. Phillip

   a. User will be interested in this message.
   b. User won’t be interested in this message.
   c. I don’t know

8. Assume the following message is sent to the user. Trace the tree to find whether the user will be interested or not.

Sender: _itsimazing@response.etrack.com

Subject: Re: The Stage

Body:

I just spoke to the insurance company. They are going to cancel and prorate my policy and work with the Kuo's to issue a new policy.

   a. User will be interested in this message.
   b. User will not be interested in this message.
   c. I don’t know

9. The big tree represented here can be considered an understandable explanation for the model behaviour?
   a. Strongly agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly disagree

Comment:

____

10. If more information could be added to this explanation, what it would be?

____

____