RANDOM FOREST SIMILARITY MAPS: A SCALABLE VISUAL REPRESENTATION FOR GLOBAL AND LOCAL INTERPRETATION

by

Dipankar Mazumdar

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2021

To Dad(Surendra Mazumdar), Mom(Tarulata Das) and Madhusmita.
This wasn't possible without your support.

Table of Contents

List of Tables

List of Figures

Abstract

Machine Learning algorithms have made significant contributions in today's world, leading to an increased usage of these algorithms in fields such as healthcare, finance, judiciary and science. However, as the application of ML algorithm surges, the need for transparent and interpretable models becomes essential. This is particularly applicable to domain experts such as medical practitioners, finance analysts, who rely on these algorithms' results to make decisions that can significantly impact someone's life. Therefore, despite these algorithms' immense predictive power, they tend to be a black box in nature when explaining the results or decisions made. Visual representations have shown to be instrumental in addressing such an issue of increasing model transparency, allowing users to grasp models' inner workings. Visualization touches upon complex algorithms' intrinsic nature and highlights some of the crucial factors needed to interpret them. Despite their popularity, visualization techniques still present limitations in terms of visual scalability, mainly when applied to analyze popular and complex models, such as Random Forests(RF). In this work, we propose *'Random Forest Similarity Maps' (RFMap)*, a scalable interactive visual analytics tool designed to analyze RF models. RFMap focuses on explaining the inner working mechanism to users in a simplistic way through three views, describing individual data instances' predictions, providing an overview of the entire forest of trees, and highlighting instances' input feature values. The interactive nature of RFMap allows users to visually interpret the models' errors and decisions, establishing the necessary confidence and users' trust in RF models to make better use of the algorithm and improve performance. The effectiveness of our technique is demonstrated using two user scenarios and validation of the results are done through a thorough user study, which shows that most of the users were able to interpret and understand RF models using our tool.

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr. Fernando Paulovich, for his constant guidance and support. I am really fortunate to have worked under your supervision and be able to learn and apply my skills. I would also like to take this opportunity to thank my lab colleagues, Mário Popolin Neto and Rakshit Varu for their valuable suggestions and help at every crucial point. Finally, I would like to thank my parents, brother, my partner Madhusmita, and the extended family members for their immense support which has helped me throughout this journey to achieve the best.

Chapter 1

Introduction

Machine Learning (ML) algorithms have seen widespread usage in numerous fields over the past few years. From music recommendations [39] to understanding whether or not a patient has a severe infectious disease [36], ML has become a part of our day-to-day life, and their dependencies have only increased with time. ML algorithms' strong predictive capability has triggered many organizations to rely on these techniques for effective data-driven decision-making. Specifically, in ML, the demand for solving classification problems has been significant, usually involving assigning appropriate class *labels* to new and unseen instances [26].

However, with the increase in the complexity of problems and dataset features, better algorithms are necessary to derive accurate models. The need to have greater predictive performance levels in real-life use cases often leads to a very intrinsic problem, i.e., its interpretation [11]. For instance, COMPAS (Correctional Offender Management Profiling for Alternative Sanctions), a software for judging the likelihood of a criminal defendant becoming a recidivist, has been widely criticized for its biased racial decisions [5]. It was observed from the algorithm results that people of color were at greater risk of recidivism than white defendants. Since ML techniques have become ubiquitous, especially in crucial decision-making involving humans, there is a considerable demand to explain the complex algorithms' work-ability and help decision-makers gain confidence and trust in the algorithm [17].

Visualization has been a natural choice for users to understand the black-box nature of complex ML algorithms [12]. In recent times, visual metaphors such as Node-Link diagrams [43], Decision Tables [25], and Matrix views [41, 42] have been widely used to interpret models. However, these techniques are prone to scalability issues and can often not capture an algorithm's overall working mechanisms since they are usually designed to inform about particular predictions. This hinders the user from getting a holistic picture of the model and can lead to misinterpretations. The issue is particularly seen with ensemble models such as Random Forests (RF) [42, 60], which is a collection of decision trees, and visualizing every tree in a human interpretable form is complex. Another issue happens when auditing results. With the current existing approaches [42, 60], the local interpretability focus on presenting only the decision paths (or logic rules) used by an instance, which is not helpful enough for most of the ML models, specifically for those based on multiple trees such as RF [35]. The problem, in this case, is losing the context, which is deemed one of the essential aspects of visual analytics tasks.

To address the above mentioned issues, we propose a tool to represent the knowledge learned (rules) by an RF model and allow for interpretation.

1.1 Research Questions

In light of the issues referenced previously, the current work will attempt to answer the following research questions:

- Is it possible to understand what knowledge a RF model has learned globally?
- Is it possible to understand the reasons behind a specific decision while preserving the global context?
- Is it possible to have a comparative analysis between two or more RF models?

1.2 Proposal

In this work, we propose *Random Forest Similarity Map (RFMap)*, a scalable and interactive visual analytics tool to address the research questions discussed above. Our visual representations are based on multidimensional projection techniques [54, 40] enabling users to visualize entire models, making them easy to understand. RFMap allows a user to analyze the entire model globally and enables auditing of individual instances or a cluster of instances locally to explain how a specific decision was made without losing the forest's global context. A similar approach has been widely used to understand certain aspects of Deep Neural Networks [24, 21]. However, to the best of our knowledge, this is the first time this is used to interpreting Random Forests.

1.3 Contributions

In summary, the primary contributions of this work are the following:

- An interactive visualization tool that supports explanation of data instances. without losing the global context of the forest.
- Two novel multidimensional projection based visual metaphors to understand the instances and induced logical rules.
- A scalable technique to visualize vast number of trees (decision paths) in a Random Forest model.
- The proposed technique is validated through a user study conducted with 18 participants.

1.4 Thesis outline

The remainder of the thesis is organized as follows: in Chapter. 2, we present the literature review of ML model explainability strategies, general visualization techniques used in model interpretation, and visualizations of RF models. Chapter. 3 presents background information on RF, design goals and analytical tasks supported by our solution. Also, *RFMap* is introduced in detail. In Chapter. 4, we explain the workability of our technique using two different user scenarios and show how our system helps in the interpretation of entire RF models and local instances while maintaining the global context. Finally, we discuss and present our technique's limitations and draw conclusions in Chapter. 6.

Chapter 2

Related Work

As ML algorithms garner more attention with time, the need for Explainable AI (XAI) has also increased substantially. There has been vast progress in the research of explaining the black-box nature of algorithms by using visual representations [42, 41, 60], knowledge extraction methods [41, 42, 37, 45] and influence based techniques [59, 14]. In this work, we focus on visual representations as the primary medium to interpret ML algorithms.

2.1 General Explainability Strategies

In the literature survey of existing explainability strategies, *Adadi et al.* [2] categorize the methods primarily into: (i) complexity-based (ii) scope-based (iii) model-based. The complexity of a model tends to be directly proportional to its interpretation. Hence, the nature of a model in terms of its complexion is non-trivial in explaining it. To reduce complexity in understanding ML models, the overall knowledge learned by the model is typically extracted. These are called knowledge extraction methods. Such knowledge extraction techniques, specifically in black-box algorithms like Artificial Neural Networks (ANN), have proven efficient in simplifying their black-box nature. In this regard, rule-based knowledge extraction methods have leaped in the direction of approximating complex models using simpler, interpretable ones. *Humbird et al.* [22], presents a technique to build deep feed-forward neural networks based on decision trees, thus allowing to build model surrogates that are easy to understand. *Letham et al.* [30] proposes an explainable method (Bayesian Rule Lists) based on decision lists (consisting of if-then statements), making predictive models more interpretable to human. *Mashayekhi et al.* [38] presents another way of extracting rules specifically from Random Forest (RF) models. They employ optimization strategies, specifically 'hill climbing algorithm' to select valuable rules instead of all the rules, thus allowing them to deal with scalability issues. In our work, we make use of rule-based extraction techniques to interpret black-box models, but our main focus lies on being able to represent the rules using visual metaphors and not on the knowledge extraction part.

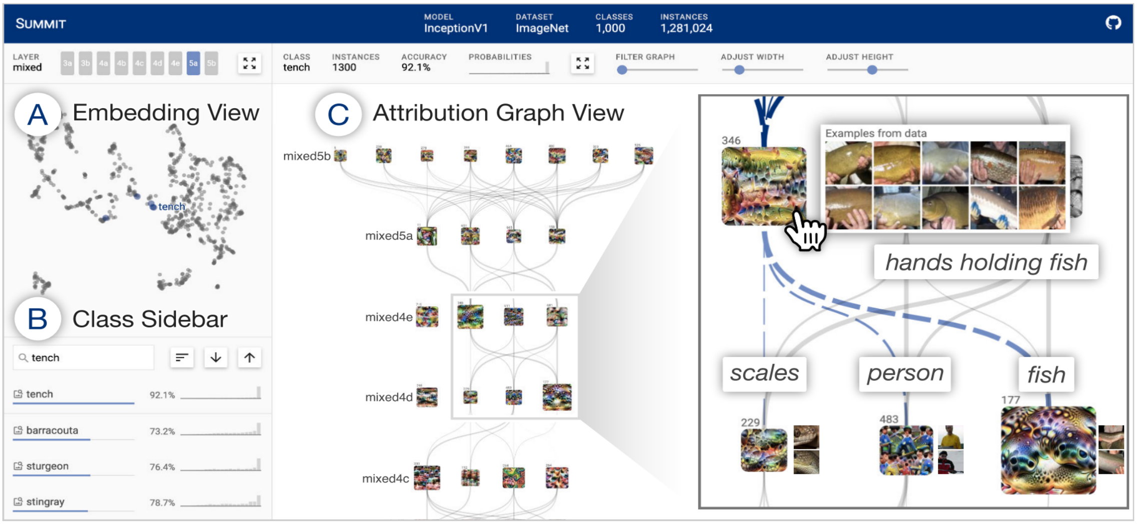
Another strategy takes into consideration the global or local scope of models. In general, a model's global scope targets explaining the entire model logic [55, 30, 57, 33] to help a human understand models' internal behavior. Global explanations touch on the factors such as understanding how a model makes judgments, what features were involved in the decision-making process, and knowledge (decision paths, weights, etc.) learned by the model. On the other hand, local interpretation is about explaining a particular instance and its resulting predictions [50]. Taking inspiration from these techniques, we present a system that explains the local instances and preserves the overall global structure, helping users keep a perspective of how a complex model has learned knowledge, thereby reducing the cognitive load on users.

The third strategy touches upon a model's nature, i.e., it being agnostic or specific. Model-agnostic explanation techniques apply to all models irrespective of their nature [2, 49]. In contrast, model-specific strategies are limited to a particular model and generally apply to simpler models in structure, such as Decision Trees or linear models. However, recent research has allowed for the distillation of complex models so they can be converted into simpler form and hence be transparent [53]. This enables explanations to be made from a particular model perspective, which helps in touching upon that model's specifics. For this research, our target is aiding users in understanding the black-box nature of 'RF models'.

2.2 Visualization for Interpreting Models

Visual metaphors have been seen to address the problem of interpreting ML algorithms for quite some time. In recent years, numerous visual analytics systems have been developed to explain the inner workings of a model [46, 15], extracting information from a model, i.e., post-hoc interpretability [41, 60, 2] or to enable performance diagnosis for building more accurate models [3, 48, 4]. Naturally, visual representations have been an optimal choice when explaining complex ML models.

Fred Hohman et al. [21] propose a visual analytics system (Fig 2.1) for Deep Neural Networks (DNN) that summarizes the activation aggregations from all classes to help to understand the critical neurons contributing to a particular classification using dimensionality reduction techniques. They also provide a graph representation of the 'neuron-influenced' aggregations to show the relationship between different neurons that finally leads to a prediction [21]. In another line of work, *Zahavy et al.* [58] use a 3D t-SNE to visualize the state transitions of the learned policies from a Deep-Q network used in reinforcement learning. ActiVis [24] is another system deployed across Facebook to interpret the black-box nature of deep neural networks. They use a 2-D projection to explain instances' activation using t-SNE and place instances with similar activation values together. *Cantareira et al.* [10] introduce an approach to explore a Neural network's hidden layer activities and simplifying the inner working mechanism of such complex networks. Their novel technique focuses on comparing projections derived from multiple stages in a neural network and visualizing the differences in perception. Taking a similar approach to ours, *Rauber et al.* [47] in their research, focuses on improving classification systems. They present projection-based visualizations that can help developers interpret particular interesting insights in a classification model and improve these systems through feature selection.



MODEL InceptionV1
DATASET ImageNet
CLASSES 1,000
INSTANCES 1,281,024
LAYER mixed
CLASS tench
INSTANCES 1300
ACCURACY 92.1%

Figure 2.1: SUMMIT: summarizes and visualizes what a Deep learning model has learned [21]

In general, these point-based visualization techniques have been focused on explaining deep neural networks. On the other hand, *RFMap* is specific to RF models. It allows users to understand the data instances and the decision paths in a forest

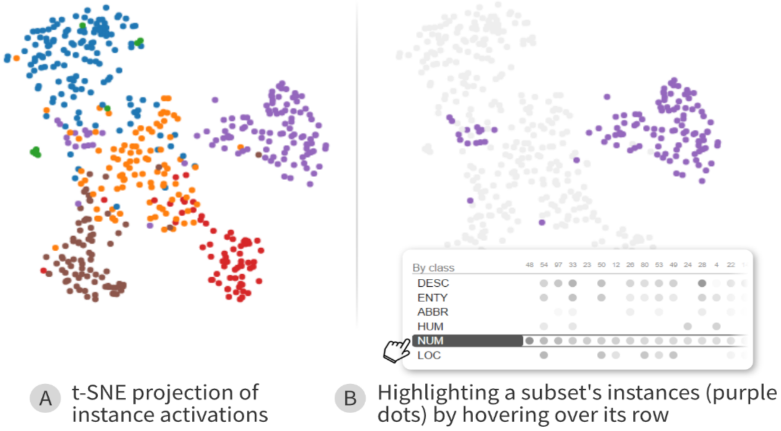


Figure 2.2: ActiVis: Projected instances activations [24]

by using 2-D projections techniques, making our *RFMap* easily interpretable and scalable.

2.3 Random Forest Visualization

A RF model consists of a large number of decision trees that function together as an ensemble. Each tree model makes its predictions (vote), and the class with the most votes is considered to be the final predicted class [8]. Such a large collection of decision trees and their underlying working mechanism makes the model black-box in nature. Hence, visualizing the entire forest of trees to make it interpretable for users is an ongoing research problem. The ability to represent a forest of decision trees to explain each tree's structure and properties is daunting. *Hansch et al.* [18] propose an interactive visualization system based on a botanical approach to interpret and improve RF models. They emphasize that the factors that lead to an increase or decrease in an RF models' performance are not easy to express as scalars. Hence, visualizing the entire RF model with all the decision trees can help in better interpretation. However, their work applies only to RF models based on binary trees. Moreover, the scalability of the forest visualization is a concern when thousands of decision trees are used.

To show the correlation between the decision trees in an RF model and data instances, *Breiman and Wald* [9] presented a multidimensional scaling (MDS) based projection. They use proximity measure, which is inherent to the RF algorithm as similarity measure to display the clusters formed among the trained instances and allow for outlier identification. In our technique, we use their 'proximity measure' as a distance metric to project our instances. However, we extend their method to build a visual representation that allows for the explanation of class separability and enables local inspection of the instances by preserving the context of the forest. *Lau et al.* [29] uses a method of aggregation for the collection of trees based on the number of appearances of a feature at node positions. They focus on feature importance and interaction to derive information on variables applied at each node of a tree. However, these tasks become challenging as they increase the number of feature variables. Additionally, their system cannot handle tree depths greater than 8, which makes it practically non-usable.

Neto et al. [42] in their work, named ExMatrix(Fig 2.3), present a matrix-based visual metaphor to aid in the global and local interpretation of RF models. They propose a novel method to extract each tree's decision path as rules and uses the matrix visualization together with properties such as certainty of rules, coverage, voting committee decision to make interpretation easy for users. While ExMatrix presents a single view of decision paths and their associated properties, they do not allow users to compare and see similarities or differences between the various decision paths. *Ming et al.* [41] introduces a model-agnostic, rule-based visual analytics system, named *RuleMatrix* (Fig 2.4) that targets the explanation of black-box models by using model induction techniques and presents a matrix-based visual metaphor. However, *RuleMatrix* is not scalable when it comes to visualizing thousands of rules.

Closely related to our work is *iForest* [60]. iForest (Fig 2.5) incorporates different visual representations into a platform that aims at explaining an RF model. Although iForest visualizes decision paths for a specific prediction, they fail to preserve the global context among the entire forest of trees when supporting analysis of local predictions. This is very critical in the case of RF models. It will allow a user to gain insights into the knowledge learning process from different parts of the forest, giving a holistic picture of the entire decision-making process. In our work, we specifically focus on this aspect and allow users to understand the decision paths used by an instance while keeping the entire forest of trees in context. Our technique can also

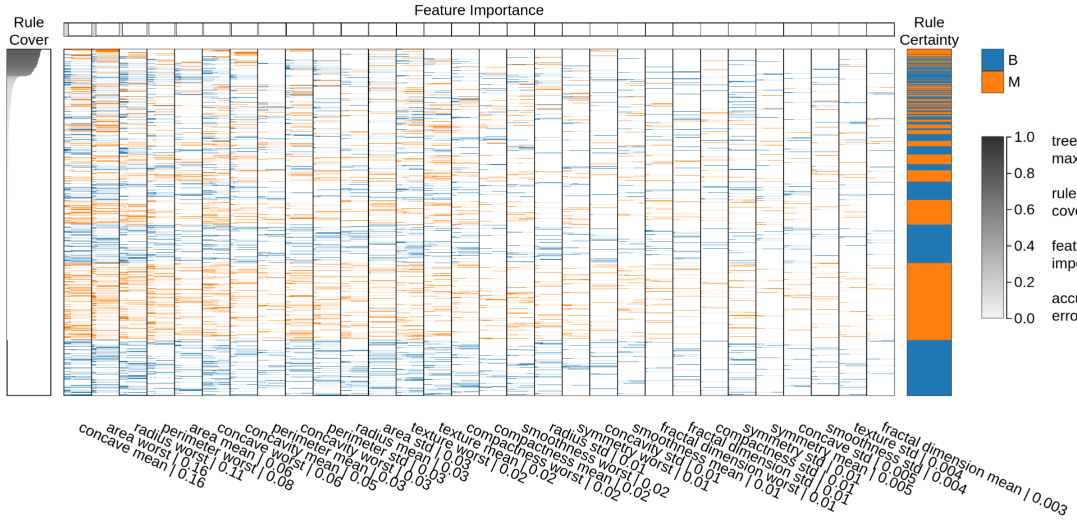


Figure 2.3: ExMatrix: a Matrix based visual representation to interpret RF model [42]

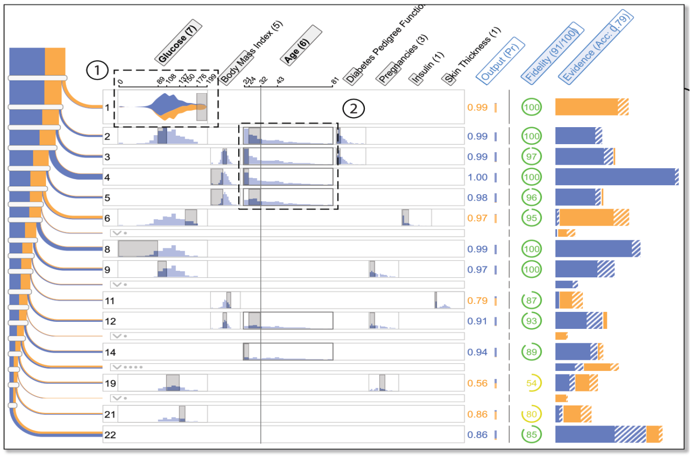


Figure 2.4: RuleMatrix: Another matrix based visualization to understand rules [41]

handle models with thousands of decision trees, thus allowing us to address the visual scalability issue, which is seen with systems like iForest [60], ExMatrix [42] and RuleMatrix [41].

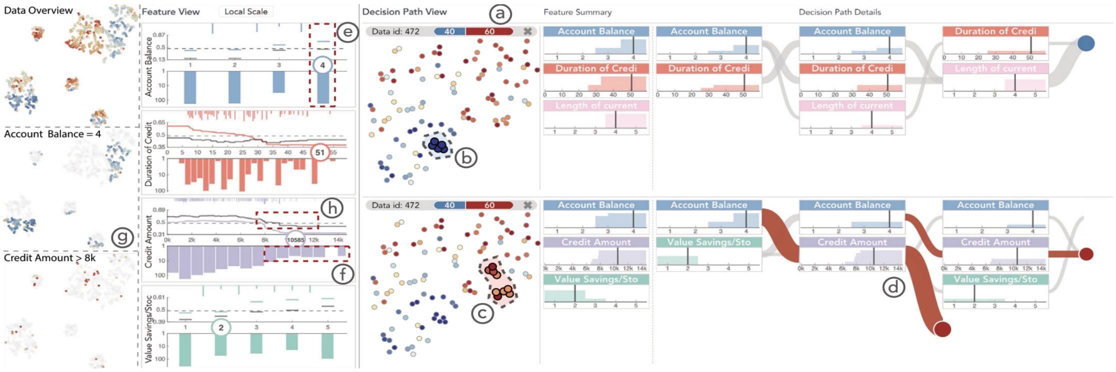


Figure 2.5: iForest: A Visual Analytics system to interpret RF model [60]

Chapter 3

Methodology

3.1 Background

A classification process involves developing a model, i.e. function F by taking a dataset X= {x$_1$,...,x$_N$} of size N and their labeled classes Y = {y$_1$,...,y$_N$} along with their features F={f$_1$,...,f$_M$} to predict the class y_n for new unseen instances x_n. As part of the classification process, the dataset X is divided into two subsets, X_{train} and X_{test}. The function F is trained on X_{train} and the trained function is then applied to X_{test} [42]. A Random Forest (RF) classification model consists of individual decision trees $DT_1,...,DT_k$ as ensembles [8]. Each decision tree is trained on a random subset of sample with replacements derived from the training data along with a randomly selected set of features. The resulting predictions from individual decision trees are then put through a voting process and the class receiving highest votes is considered to be the final predicted class [6, 8]. The construction of the RF algorithm is shown in Algorithm 1[60, 28].

To aid in the interpretation of RF models, our technique uses rule-based knowledge extraction procedures [42] and visualizes each 'decision path' of a tree as logic rules R={r$_1$,..r$_z$}. A decision path is a path starting from the root node to the leaf node in a single decision tree, and the outcome from the leaf node determines the predicted class C after it goes through the voting process. Along with the extracted logical rules R, we also derive two other important properties using *Neto et al.*'s [42] vector extraction technique for supporting interpretation of the rules - rule coverage, r_z^{cov} and rule certainty, r_z^{cert}. Rule coverage, r_z^{cov} is defined as the value obtained after dividing the number of instances in X_{train} that is valid for that rule's class r_z^{class} by the total number of instances of r_z^{class} present in the set X_{train}. Rule certainty, r_z^{cert} is the vector of each class' probabilities that is derived from the decision path, i.e., the leaf. Since a RF model usually comprises many decision trees, using the decision path, as a rule, comes as an effective choice when visualizing the entire forest of trees,

Algorithm 1: *Random Forest(x,J,A,B)*

```
Result: Ensemble of trees, $\{DT_1,..DT_k\}$
  1. for j=1 to J
  2. Select a bootstrap sample of size A from training sample $x$.
  3. Grow a tree using the bootstrap data recursively on each node.
  4. while number of nodes in the tree does not reach a certain criteria do
         Select $b$ features at random from the input feature $B$.
         Pick the best feature for split from $b$.
         Based on a criteria, split the node into child nodes.
     end
  5. Return the ensemble of trees
```

thereby allowing us to design scalable and easy-to-interpret visual metaphors.

3.2 Design Goals

After reviewing the literature in RF models visualization and dimensionality reduction-based classification model analysis, we present our system's design goals. These goals are listed below:

G1: Global interpretation. An RF model is a collection of trees. One of the best ways to interpret the ensemble model's inner working mechanism is to allow users to understand what knowledge the overall model has learned [60]. After the RF model training, the model presents an overview of the relationships between the data instances and the various decision paths, given a target class label. These relationships between decision paths and data instances mirror the RF model's working mechanism at a granular level and help users comprehend the generic knowledge (valid for most of the instances) or specific knowledge (valid for only a few instances) learned by the model. Therefore, it simplifies the model's complex nature and presents the knowledge learned (whether generic or specific), helping to understand how the overall RF models decisions. By enabling the interpretation of the knowledge learned by an RF model, we aim to explain the model globally.

G2: Local interpretation by preserving the global context. Local interpretation describes the reasons behind a specific decision for a particular instance [2]. In most RF models, local interpretation usually involves presenting the decision paths used by the instance [42]. However, to perform local interpretation effectively, preservation of the global context of the forest is essential so users can compare the used logic rules in context with the other rules in the forest and answer questions such as - 'Are the decisions made from the most certain logic rules in the forest?', 'Does these rules have a good amount of data support(coverage) as compared to others in the forest?'. Being able to answer these questions not only helps users develop trust in their local explanation but also allows them to retain the local faithfulness [51] on unseen instances. Local interpretation also means allowing users to find out hidden patterns from the dataset or a specific set of examples and deduce further explanations based on them [60].

G3: Comparative analysis of RF models. Another design goal is the ability to have a comparative analysis between two or more RF models to assist model developers in selecting reliable models [31]. An RF model is built using various parameters such as number of trees, splitting criteria, maximum depth of a tree, maximum number of features to be considered during a split, among others. These factors are very imperative and influence the overall prediction capability of a model [47]. By visualizing and comparing RF models built using different properties, we can interpret their functioning and shed light on some hidden patterns. For example, what happens to a model when we do not limit the trees' depth, allowing all trees to grow indefinitely.

3.3 Analytical Tasks

Based on the related work discussed in Sec. 2 and to fulfill our design goals, we have developed the following analytical tasks.

T1: Analyzing structure and properties of decision paths. The decision path in every RF model tree provides a way to understand the final predicted class. Hence, the analysis of the structural differences between various logic rules is imperative to uncover the black-box nature of an ensemble model (**G1**). For instance, how do we know which group of rules among the forest classifies samples as a particular class? Have they learned anything generic from the training data? We aim to provide users an answer to these questions through our technique. Other than analyzing structural similarities between logic rules, it is also essential to know about the properties such as a rule's class probabilities (certainty) and how much training data support a decision path has concerning its predicted class (rule coverage). High coverage and certain logic rules are the important ones in the model as they are valid for the majority of the instances (generic knowledge) and important to the RF committee [42].

T2: Visualizing the forest. Visualizing an entire forest of trees is a challenging task, and its complexity increases with the number of trees used in an RF model. To support the case of understanding the working mechanism of complex ensemble models with certain number of trees (**G1**), it is essential to provide a way to visualize the entire forest. Visualizing the forest also helps users in understanding where a decision path is located within the forest and how these decision paths are related to one another [18]. Hence, to summarize the structure and understand various decision paths that form an RF model, visualizing the entire forest is non-trivial.

T3: Interpreting class separation among instances. The primary goal of any ML classification problem is to separate the data instances into their respective classes. By establishing a clear decision boundary between the classes, we can validate the model's accuracy, understand its inner working mechanism (**G1**), and allow for the improvement of models through comparative analysis (**G3**) [47]. Thus, providing a visual metaphor to understand the class separation between the instances in a dataset is crucial.

T4: Knowledge utilized by the model to make a prediction. To understand the prediction of a single instance or a group of instances, it is necessary to know what knowledge was utilized by the model [41], i.e., which logic rules were used to classify an instance. While local interpretation methods allow users to know what rules (decision path) were applied to a sample [42, 60], there is a lack in interpreting the knowledge learned from the entire forest of logic rules visually, preserving the global context (**G2**). The ability to inspect local instances while having a visual cue of the used rules from the forest allows users to perform in-depth interpretation, and it provides insights on the voting process for that instance.

T5: Understanding the instances. Analyzing the structure of instances in a dataset helps provide intuition into specific hidden patterns that can, in turn, assist in understanding the way an RF model works on similar types of instances (**G2**). For example, by analyzing the inter class overlaps among samples in a dataset, we can interpret class errors made by the model in a classification problem. Also, understanding similarities or differences between a group of instances can help the user develop reasoning on how the RF model sees every instance and how it differentiates them.

T6: Performing model diagnosis. To develop an understanding of RF models' performance, i.e., how properly the model can separate the classes, ML model experts and developers often need to drill down and analyze certain aspects, such as detecting the mistakes made by a model using confusion matrices [16] and visually comparing multiple confusion matrices [52]. Although confusion matrices are easy to use, they become challenging to interpret when the number of classes increases in a multi-class classification problem [47]. By having the ability to visualize the patterns formed among the logic rules in a model and compare it with other models (**G3**), users can develop their confidence in the models' overall functioning and select the model that produces the desired result.

3.4 Overview

This section discusses our solution, *RFMap* proposed based on the design goals mentioned in Chapter. 3.2. *RFMap* comprised visual representations, and steps involved in interpreting RF model are presented in Fig. 3.1. Primarily, in our technique, we focus on three visualizations: *Instance view, Forest view & Feature View.* To fulfill our design goals, we have amalgamated the three views into a cohesive and scalable system. The integration of all the three views into one interactive system (Fig. 3.1) allows us to target both global **(G1)** and local **(G2)** interpretability perspectives [32]. We emphasize that to clearly understand the knowledge learned from a vast forest of trees, presenting only the decision paths when explaining the results of classification is not very beneficial [35]. Therefore, preserving the forest's global context is imperative for local interpretation so users can understand from which part of the forest an instance is using the logic rules. Our method also enables users to further drill down into analyzing the used rules contextually with the other forest rules to help them answer some of the crucial questions discussed in **(G2)**. The global preservation approach can also help shed some light on one of the essential components of RF models, i.e., the voting committee.

In our approach, the logic rules are first extracted from an RF model Ⓐ and the necessary instance-rule and rule-instance mappings are derived using the ExMatrix package [42]. Using *RFMap*, the user first explores the *Instance View* Ⓑ to understand how the RF model sees all the data instances and analyzes the separation of classes among instances ❶ of the dataset. This view also helps a user to understand the similar and differing instances from RF models' perspective and enables interpretation of misclassified instances ❷. Users can click on any particular instance ❸ to perform local interpretation to understand which logic rules were used to classify the instance while having the global context of the forest Ⓒ preserved. The preserved forest view helps the user perform contextual analysis of the used logic rules with other forest rules. The highlighted rules ❹ in the *Forest View* Ⓒ displays the used logic rules. The *Forest View* also allows users to derive an understanding of the structure of various logic rules and the certainty of their predictions, as shown in ❺. To understand the classification of instances from a global perspective, users can lasso-select a rule or cluster of rules ❻ from the *Forest View* and the instances being classified would be highlighted like in ❼. The lasso-selection of rules also intuitively presents the feature values of instances using those rules in the *Feature View* Ⓓ.

RFMap is a web-based tool developed using D3.js [7], Vanilla JS, Plotly [44] and Bootstrap for the front-end, and Python for the back-end. The motivations and visualizations for understanding instances and logic rules are further discussed in the following sections.

3.5 Visualization design preliminaries

Based on the effective results achieved by using dimensionality reduction techniques in the interpretation of neural networks [46, 24], we adopt a similar strategy to build both of our *Instance* and *Forest* views. We also incorporate a third view to visualize the feature values of instances used by a rule or group of rules, called *Feature View*.

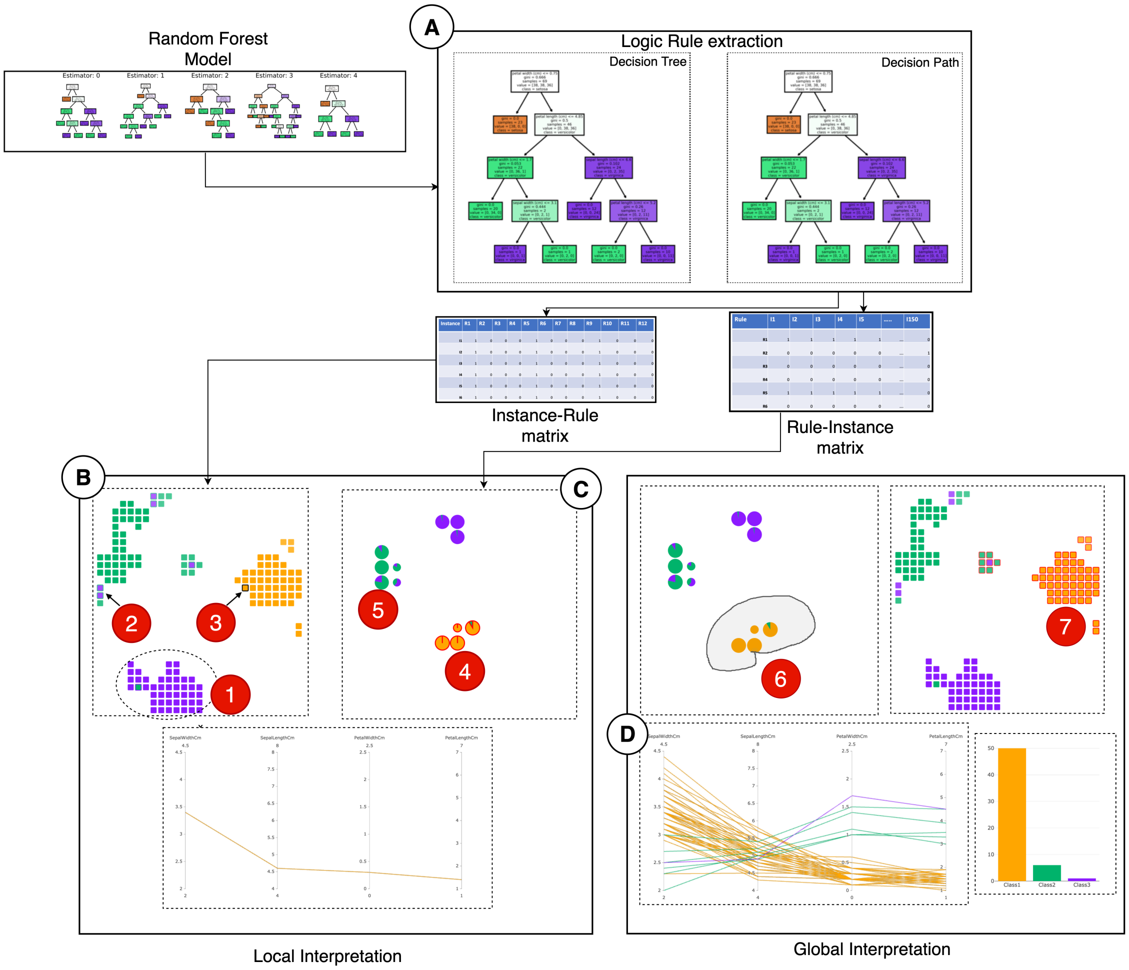


Figure 3.1: RFMap system with Iris dataset. The letters indicate the system's different modules. (A) is the Logic rule extraction part, after which we derive the instance-rule and rule-instance matrix for our projections; (B) is the *Instance View*; (C) is the *Forest View*; (D) is the *Feature View*; the user first explores the *Instance View* and notices the three separated cluster of classes by color. One of the cluster (1) is shown in dotted circles. They take a note of some misclassified instances (2) from the view. They then click on an instance (3) to understand the knowledge learned by the instance from the forest. The highlighted rules in (4) shows the logic rules being used by the particular instance. The user analyzes the structural differences (similar or not similar rules) and certainty, coverage of the various logic rules as shown in (5). To understand a 'knowledge cluster', the user draws a lasso around the certain cluster of rules (6) and the instances covered by the rules are highlighted in (7). For enabling further interpretation and analysis, the knowledge contents are shown in (D).

An inherent problem with using dimensionality reduction methods for visualization is that they can result in cluttered visuals and occlusion issues. This issue particularly applies to our research work since we build our visual representations based on the idea of whether an instance uses a logic rule to be classified or not. Hence it is expected that a lot of instances might end up being in the same position of projection if they use the same rule. To remove the projection overlaps and meet our design goals, we employ a distance-preserving grid layout algorithm [20] that uses a binary space partitioning technique together with the projections obtained for creating orthogonal grid layouts for both *Instance and Forest* views. The method first maps the data instances to 2D points using a projection technique preserving the dissimilarity as much as possible. Then a partition-scheme is utilized to assign each point into a grid cell by maximizing the below cross-correlation function [20]

$$CC = \frac{1}{N^2} \sum_{i}^{N} \sum_{j}^{N} \frac{(\lambda(g_i, g_j) - \overline{\lambda})(\delta(d_i, d_j) - \overline{\delta})}{\sigma_{\lambda}\sigma_{\delta}} \tag{3.1}$$

where $\delta(d_{\mathrm{i}}, d_{\mathrm{j}})$ is the dissimilarity between the samples, $\lambda(g_{\mathrm{i}}, g_{\mathrm{j}})$ is the distance between grid cells, $\overline{\lambda}$ is the mean distance between any two cells, $\overline{\delta}$ is the mean distance between any two pair of samples, and δ_{λ} and δ_{σ} are the standard deviations. Although the grid layout algorithm allows us to tackle the occlusion problem, we still need to maintain the distance of the actual group of instances or logic rules so that our representation is valid. Therefore, we employ the technique of addition of "dummy" points before allocating the projections into the grids as per *Hilasaca et al.* [20]. This method targets the low-density areas of a projection to add points to represent space.

One of the critical goals of our work is to be able to interpret what knowledge an RF model has learned (**G1**) and understand the knowledge learning process for an instance while preserving the global context of the forest (**G2**). However, our focus lies on the principle that interpreting the overall RF model should not be mentally overwhelming for the user. It should be easy to understand from an explainability perspective. By taking motivation from these goals, we decide on visualizing just the decision paths (logic rules) instead of individual decision trees to represent an entire forest of trees.

3.6 Instances view visualization

The first visual representation, *Instance View* (Fig. 3.2), is a 2D projected representation of the instances in a dataset based on an RF model. The key idea behind designing a visualization to interpret data instances is to allow users to understand the structural similarities or dissimilarities (**T5**) between these instances considering what has been learned by the RF model. We intend to give users an idea of how similar the instances are in the RF model's eyes irrespective of whether the data is high-dimensional [19]. Therefore, the instances' similarity will vary based on model to model. Using our technique, we visualize the class separability among the instances (**T3**) and allow model developers to visualize their model's performance, detect outliers, and wrongly classified instances. A near-to-accurate RF model will present a visualization with clear decision boundaries between the classes with very few within-class overlaps [47]. This serves as an initial guide for developers in building effective and robust RF models.

We start with building an RF model based on several parameters, such as the total number of trees, maximum depth, split criteria, and so on. In any projection-based technique, the distance metric choice is critical, and its performance relies on the metric used. Hence, an ideal alternative to aligning with our design goals is the *Proximity measure* suggested in [8]. This measures for two samples x_1 and x_2 the number of times the same leaf node (decision path) classifies both samples t within each decision tree, normalized by the total number of trees in the entire forest. Based on this concept, we derived the following dissimilarity measure

$$Dissimilarity(x_1, x_2) = 1 - \frac{1}{M}\sum_{m=1}^{M}\sum_{t\in\widetilde{\varphi}_{\mathcal{L},\theta_m}} 1(x_1, x_2 \in x_t) \tag{3.2}$$

where $\widetilde{\varphi}_{\mathcal{L},\theta_m}$ represents the set of leaf nodes in a tree $\varphi_{\mathcal{L},\theta_m}$ and M is the total number of trees in the forest [8]. This dissimilarity range in $[0, 1]$ with values closer to 0 indicating samples reaching the same leaf and values closer to 1 samples reaching different leaves, thus representing the instances from a models perspective [34].

To project our dataset using this dissimilarity, we use the Multidimensional Scaling (MDS) technique [54]. We use MDS based on *Breiman et al.*'s adoption [8] of MDS for visualizing training data to understand clusters and outliers from an RF model perspective. Also, MDS is a global technique, and since, in our case, it is vital to enable interpretation of groups of instances (Fig. 3.1A) based on the class labels **(T3)** rather than preserving local neighborhood, it is a reliable alternative.

Projection-based techniques represent data instances with high dimensions into more human interpretable forms, such as 2D. However, it is also critical to understand how well the data is represented in a lower dimension to not present with a wrong representation. In the case of our MDS projection, we measure the effectiveness of projected points using a metric called *Stress* [27]. As per the definition, *Stress* of an entire projection is measured as the difference between the actual distance of all points and their projected values. For this work, we adapt *Kruskal's* formula to compute the stress value per point approximately ($\mathrm{q_i}$) using the below formula.

$$stress(q_i) = \frac{\sum_j^N (d'(p_i, p_j) - d(q_i, q_j))^2}{\sum_i^N \sum_j^N d'(p_i, p_j)} \tag{3.3}$$

where p_i is the original point, *d′ (p_i, p_j)* is the distance between original points and *d(q_i, q_j)* is the distance between projected points. We focus on displaying *stress* per point to bring out the projection method's effectiveness and allow users to build up their trust on the projected points.

Each instance in the *Instance View* is drawn as a rounded-rectangle filled with the color of their original labeled class. In the case of misclassified instances, the outside stroke color surrounding the rectangle represents the model's predicted class, and the filled color inside represents their original class(Fig. 3.2). We also incorporate the computed *stress* value, s for each data point by controlling the opacity, O of the rectangles using

$$O = \begin{cases} 1, & \text{if } 0 \leq s \leq 0.025 \\ 0.8, & \text{if } 0.025 \leq s \leq 0.05 \\ 0.7, & \text{if } 0.05 \leq s \leq 0.1 \\ 0.6, & \text{if } 0.1 \leq s \leq 0.2 \\ 0.5 & \text{if } s > 0.2 \end{cases}$$

Therefore, data instances whose projected representation is almost equal to their original high-dimensional representation will visually be much brighter. Our primary

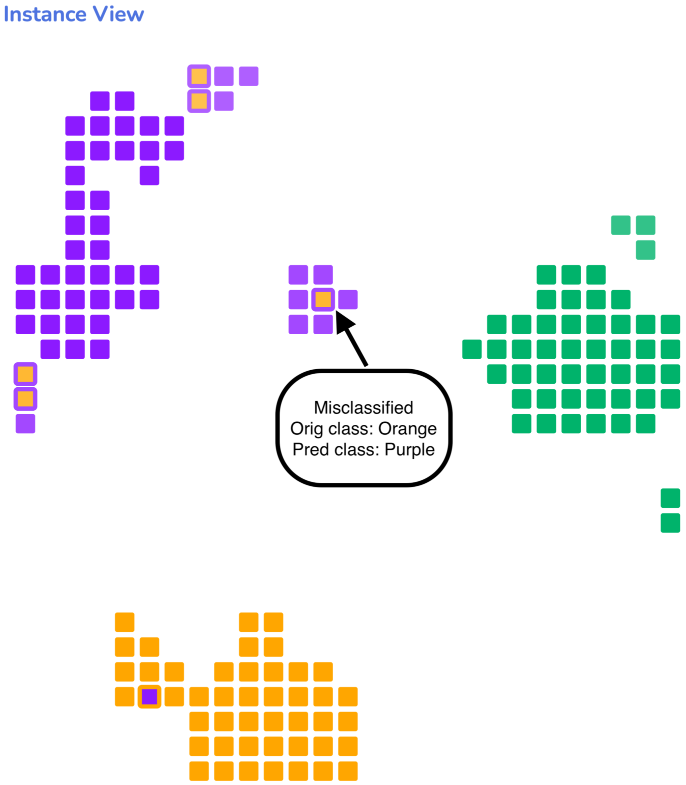


Figure 3.2: Instance View: Represents misclassified instances, decision boundary between the three classes and outliers.

focus is to present a visual metaphor to help the user understand how the RF model sees every instance after being trained. This way, we can establish a hypothesis regarding the instances' structure and derive further explanation on new and unseen data (**T5**). By analyzing the various cluster of instances formed, a user can get an idea of the generic knowledge learned by an RF model and explore how a particular group is different from others in terms of their prediction or feature values. This is very helpful specifically for users who need to derive case-based reasoning based on these instances' interpretation.

Another key element that users can gain insight into using the *Instance View* is detecting the within-class outliers [34]. By computing the average proximity value of an instance with respect to all other instances in a dataset, we can show the instances that have high overlaps with other class instances and affirm that these samples are the ones that the RF fails to classify correctly. Thus, the *Instance view* serves as an effective solution for users to interpret the classification errors, detect outliers and understand the decision boundary.

3.7 Forest view visualization

The second visualization, *Forest View* (Fig. 3.3), was designed to allow users to analyze every decision path (logic rules) in the forest and thereby gain an understanding of what the RF model has learned (**G1**). To derive insights on the structure of various logic rules, it is essential to identify the similarities or differences between various logic rules. In our technique, two or more rules are considered to be exactly identical if they classify the same instances. A near-to-similar set of rules will have an intersection of instances and will be closer to proximity. These similar rules form distinct sets of high certainty clusters, which we refer to as a 'strong knowledge cluster'. A 'strong knowledge cluster' suggests that the model has learned strong (high coverage) relations between training instances given a target class. The 'knowledge clusters' represents a very well-defined piece of knowledge learned by the RF model since these clusters are valid for most of the data instances and, therefore, more generic. To our knowledge, this is the first time an RF model's learning is visualized this way. Other than interpreting generic 'knowledge clusters', analyzing the instances' classification from the point of view of a set of rules can also provide insights into understanding

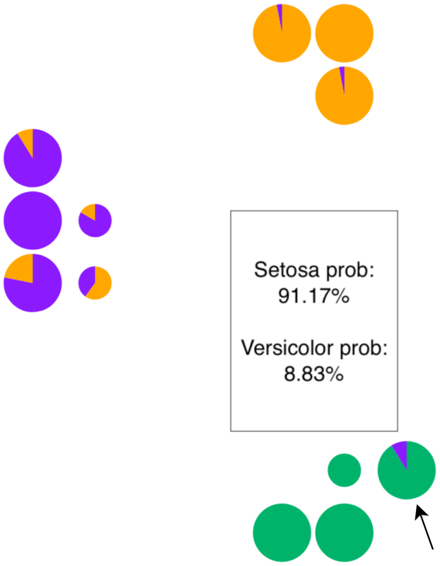


Figure 3.3: Forest View: visualizing the forest of logic rules(decision paths) in the Random Forest model. Shows high(big circles) and low (small circles) coverage paths. Strong knowledge clusters are seen to be formed inside the model. Individual class probabilities can be seen when hovered over a rule, as shown by the arrow.

the patterns defined by these set of logic rules, i.e., what kind of instances a set of rules classify as a particular class? We enable users to answer questions like these by selecting a set of logic rules and visualizing the various hidden patterns.

The view for visualizing the logic rules is also based upon the same underlying principle of projections. However, the choice of distance metric used and the projection technique applied varies here. To do the projection, we utilize an instance-rule matrix derived from the ExMatrix package [42], $\boldsymbol{I_{mn}}$, $i_{mn}\in [0, 1]$ (#instances x #rules) that gives a binary value of 0 or 1 based on whether an instance uses a rule or not for classification, and then apply a transpose on it. The result is a rule-instance matrix(#rules x #instances), $\boldsymbol{R_{ij}}$, $r_{ij}\in [0, 1]$, where 0 represents a rule not used by an instance and 1 means the rule is used. Since the value of the matrix elements is binary, we focus on selecting a distance metric that can help us preserve the closeness between rules specifically for this kind of binary data. After reviewing the literature of distance measures used for a binary dataset, we selected *Jaccard* as the optimal choice for our problem [13]. Then, the non-linear dimensionality reduction algorithm UMAP [40] is used to project the rules into a 2D space. UMAP performs well in preserving the distances between both local and global samples. Since our focus is to maintain the balance between similar logic rules (local) and also the overall rules based on class labels (global) to make explanations on the model workings (**G1**), we choose UMAP as the most suitable projection technique.

We draw each decision path as a Pie chart to allow users to understand the resulting class probabilities for each leaf node. For the decision paths whose certainty of classification is 1, the pie chart would look like a circle filled with the predicted class color. Users can also hover over individual pie charts to gain relevant information on the decision path, such as the rule coverage percentage, rule certainty probability, and final predicted class (**T1**). A forest of trees will generally have a set of high and low coverage decision paths. Since it is critical for users to interpret which rules are important from the RF model perspective, we control the size (radius r) of each pie chart based on the coverage value, c. The outer radius r of the pie charts are defined using the following.

$$r = \begin{cases} 11, & \text{if } c \geq 70 \\ 9, & \text{if } 30 \leq c \leq 70 \\ 6 & \text{if } c < 30 \end{cases}$$

Hence, high-coverage decision paths will be comparatively bigger than the low-coverage ones. We extend the *Forest View* to allow users to compare multiple (**G3**) RF models by placing the views side-by-side in our system.

3.8 Feature view visualization

The *Feature view* (Fig. 3.4) is designed as a combination of Parallel Coordinate plot (PCP) and a Bar chart to help support analysis of the instances' feature values from both global (**G1**) and local (**G2**) perspectives. From a global point of view, we focus on understanding the patterns defined by the logic rules through this visualization displaying the valid instances with their feature value ranges so users can effectively compare them and derive certain insights, such as how a set of rules are different from others in terms of the instances they classify. Locally, we enable users to analyze the input feature values of the selected data instances to understand the differences or similarities between them from not the perspective of the RF model but the actual

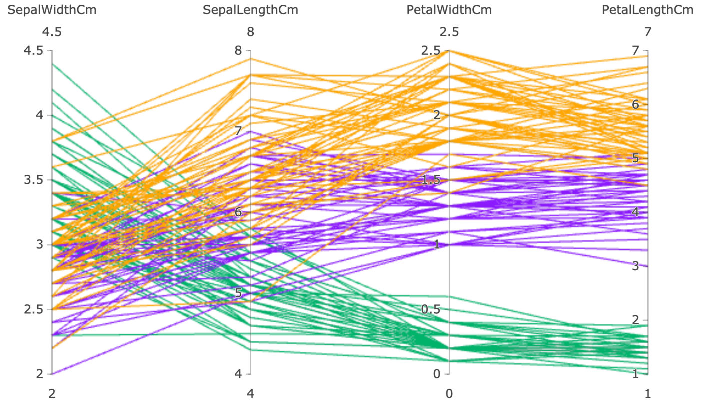


Figure 3.4: Feature View: PCP showing the relationship among various features of instances in the Iris dataset.

dataset.

Each feature in a PCP is represented as a vertical bar, and the advantage of using this plot is that these bars can have different ranges and units. It has also proved to be beneficial in representing high-dimensional dataset [23]. The lines in the PCP are colored as per the original class label for each instance. To locally inspect differences among features, users can select a group of instances in the *Instance View*, and the PCP will present their feature values. For global analysis, we allow users to lasso select a cluster of logic rules from the *Forest View*, and the PCP will display all the instances that leverages this cluster for its classification. To support the analysis of the logic rules using the PCP, a bar chart is incorporated into the *Feature View* that displays a count per class for instances using the set of logic rules.

Chapter 4

Experiments and Results

In this section, we present three usage scenarios to evaluate the effectiveness of *RFMap* in interpreting and visualizing Random Forest (RF) models. We leverage the *ExMatrix* [42] package to perform rule extraction as discussed in Sec. 3.

4.1 Usage Scenario 1: Breast Cancer Diagnostic

In this usage scenario, we describe Karen, a Data Scientist working with a healthcare company who needs to understand the overall working mechanism of an RF classification model to align it to her understanding. Interpreting the classification model is imperative for her company so the decisions can be trusted and put to real-time use for medical experts. To do that, Karen uses our RFMap system to visualize and interpret an RF model she has developed to classify breast cancer diagnosis. The dataset she uses to train the RF model is from the University of Wisconsin (Wisconsin Breast Cancer Diagnostic [56]), and it contains samples of solid breast masses collected from 569 patients, out of which 357 were labeled as Benign (B) and 212 as Malignant (M). The total number of features in the dataset is 30. Karen develops the classification model by randomly selecting 70% of the dataset as training and the remaining 30% as testing. To build the model, she sets the total number of trees as ten and does not limit their depths. For evaluating the quality of a split, she uses 'Gini' as criteria. The resulting model comprises 184 decision paths (rules), and the accuracy of testing data is 98.8%.

Karen loads the RF model in the 'RFMap' system and is presented with the *Instance, Forest, and Feature Views.* She first inspects the *Instance View* and notices two significant clusters of classes, Orange (Malignant) and Blue (Benign) as seen in Fig. 4.1 ❶ & ❷. The clusters have a clear visible separation between them, with only a few instances from both the classes on the top and bottom center parts. Seeing the clear separation between the two classes visually in the system allows her to affirm to

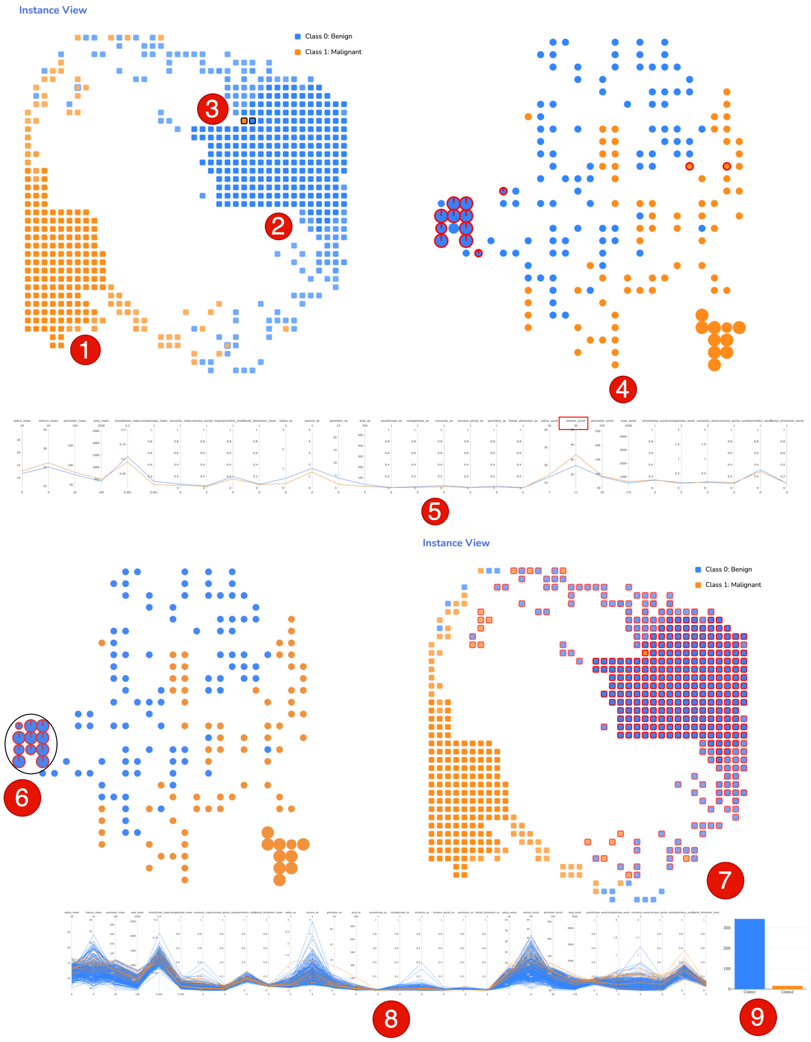


Figure 4.1: RFMap system with Breast Cancer dataset. (1) & (2) RF model clearly separates the majority of the instances based on predicted classes and forms two clusters. (3) Misclassified instance 136 and its neighbour 161 selected. (4) Rules used by the instances highlighted in the *Forest View*. (5) Parallel coordinate shows the difference between the two selected instances. (6) A lasso-drawn to understand a 'knowledge cluster'. (7) Instances using the 'knowledge cluster' highlighted in the *Instance View*. (8) PCP shows the knowledge content defined by the cluster. (9) Bar chart displays the count per class for all instances using the 'knowledge cluster'.

the understanding that the model's performance is good, and it is confused only about a few instances (**T3**). She also observes that the not properly separated instances are less brighter than the clearly separated ones. This means their position cannot be fully trusted, which matches the model's perception in these instances. She quickly notices three instances that have outer stroke color 'Blue' and the inside circles are filled with 'Orange', giving her an indication that the model has wrongly classified these three instances as 'Benign' (**T5**). Identifying false negatives is very important, especially for her organization, which deals with cancer diagnosis. So, she hovers over these three data points to know which specific patient IDs were wrongly classified so they can be investigated in depth.

Karen then focuses her analysis on one specific misclassified instance, i.e., instance 136. She understands from the bright transparency that its position in the view can be relied upon, and it definitely belongs there. The instance lies at a close proximity to the other 'benign' instances, which means the RF model sees these instances as similar. She wants to drill down more to understand if the model is confused and wrongly predicts or if this is because of the instance having similar feature value ranges with the neighboring instances. The *RFMap* system allows her to click on an individual instance or a group of instances to analyze their feature values in the *Feature View* (**T5**). So she clicks on instance 136 and its immediate neighbor 161 (Fig. 4.1 ❸), and goes to the *Feature View*. She realizes that other than 'texture_worst', the rest of the features have almost similar values for both the samples (Fig. 4.1 ❺).

Karen then moves to the *Forest View* to analyze the knowledge (**T4**) utilized from the forest to classify these two instances. From the highlighted red rules, she gets a clear picture of the decision paths used by both of these instances. Karen notes that the rules used to classify instance 136 and neighboring ones have differences. She also does a contextual analysis of the rules used by instance 136 in terms of the other forest rules to understand their certainty and coverage. Karen notices that eight of the rules are from the blue 'knowledge cluster' and two are 'orange' rules from the middle of the forest. She notes that although the two 'orange' rules are certain in their predictions, they have little coverage since they are smaller in size. She hovers over these two 'orange' rules to get the actual coverage value and sees that they are 0.6% and 5%. These values are significantly less than the other 'blue' rules from the strong cluster of knowledge, which suggests that these have learned considerably less from the training data and therefore have no significant impact on the 'voting process'. This contextual analysis level in terms of the entire forest helps Karen develop trust in her local explanations. Karen thinks instance 136 might be a rare case since position wise it belongs to the 'blue' cluster and has similarities with other blue instances, but its ground truth label is 'orange'. She decides to bring it to an oncologist's attention so they can investigate in detail the case. Being able to perform this kind of interpretation is very beneficial for her as this way, she can increase her trust in the model so it can be used to classify unseen instances.

After developing her understanding of the instances locally, Karen decides to gain insights on the predictions from the forest of trees' perspective (**T2**). Her objective is to look into the inner working mechanisms of the RF model to make it transparent. She observes two 'strong knowledge clusters'. One with 'blue' rules and other with 'orange' rules as represented in Fig. 4.1 ❹. Karen understands the significance of these strong clusters of knowledge and wants to gain more insights, specifically on the 'blue benign cluster'. She draws a lasso around this cluster ❻ and the *Instance View* highlights all the instances using this set of rules ❼. The first thing she spots is that this 'knowledge cluster' covers all the perfectly classified benign instances, which means that this cluster has learned a very well-defined piece of knowledge from these instances and is very generic. To further understand the pattern between all the rules in this cluster, she scrolls down to the *Feature View* and sees from the bar chart ❾ that 344 instances are from the 'blue' class and 16 are from the 'orange' class. Karen observes the pattern among the 'blue' cluster rules in terms of the instances' features ❽. This helps her make hypotheses on any unseen instance (patient) as they will probably be classified as 'benign' if they have similar features. She notes the features that are too low or high in values so domain experts can develop a clear understanding of 'benign' class cells.

Karen also grabs the attention of the low coverage decision paths in the middle of the forest (Fig. 4.1 ❹) that have not formed strong clusters. These rules have internally developed some homogeneous groups but have no class-level separation. She assumes that using these rules might lead to instances not being separable in

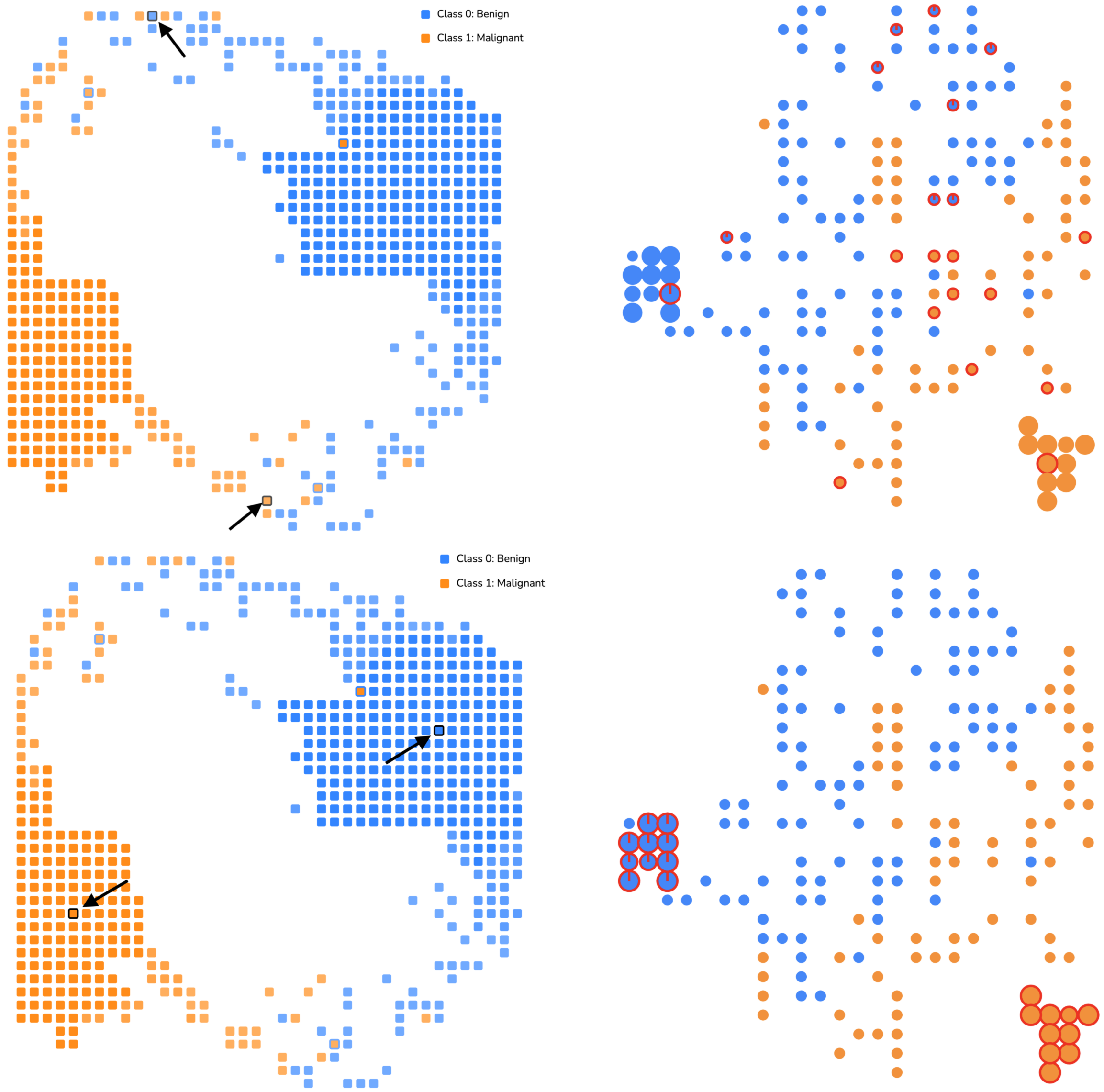


Figure 4.2: (A) Left: Instance View where instances selected are marked by an arrow. These instances are not clearly separated, and hence Karen wants to investigate if they use the rules from the center part in Forest View. Right: Forest View shows the used decision paths highlighted in Red. The majority of the rules belong to the center part, which has all the low-coverage rules. (B) Left: Instance View with two perfectly classified instances selected. Right: These instances use high-coverage decision paths from the two Blue and Orange clusters.

their true classes and decides to analyze if her hypothesis is true. So, she clicks on one Blue and Orange instance (**T4**) from the top and below portion of the *Instance View* which are not clearly separated by the model as shown in Fig. 4.2(A). She notices that these instances primarily use the low-coverage decision paths (Fig. 4.2A (Right)). She also selects two perfectly classified instances (Fig. 4.2B) and spots the high-coverage decision path clusters in the *Forest View*. These explanations help her validate that the initial understanding was correct, and the model is unable to classify the instances that were not clearly separated in the *Instance View* when using these rules. The ability to learn these insights from using the *RFMap* system makes Karen's model transparent and gives her the confidence to put it into real-time practice.

4.2 Usage Scenario 2: Election Votes

The second usage scenario presents John, a Senior Data Scientist, who is working with a Strategy & Analytics team that develops ML models to predict the potential tendency of voting for US presidential elections given users' preferences. He receives two RF models that a team member has developed. As part of his role, John needs to evaluate various ML models so they can be deployed to the production systems. In the past, John has seen models performing very well with high accuracy on training and testing data, but when it comes to using them real-time on new and unseen data, the models have failed to generalize at times. This has led to wrong predictions, and the trust in such type of application is lost. Therefore, John's goal is to select a robust and generic model from among the two, so it works well on unseen samples in the future. The first model (Model 1) is built using 20 decision trees with a max_depth of 7, and the accuracy 94.48%. The second model (Model 2) has 30 decision trees with a depth limit of 2 and accuracy of 94.3%. Although both models' accuracy does not significantly differ, he would like to analyze the models in detail, so he does not just select a model based on 'high accuracy'. John learns about the ability to do comparative analysis among various RF models using the *RFMap* system and starts his analysis.

He first loads Model 1 into the system as shown in Fig. 4.3. This model is large, comprising 1596 rules (decision paths). His previous experience with visualizing bigger models, such as this, has been overwhelming because the systems were either

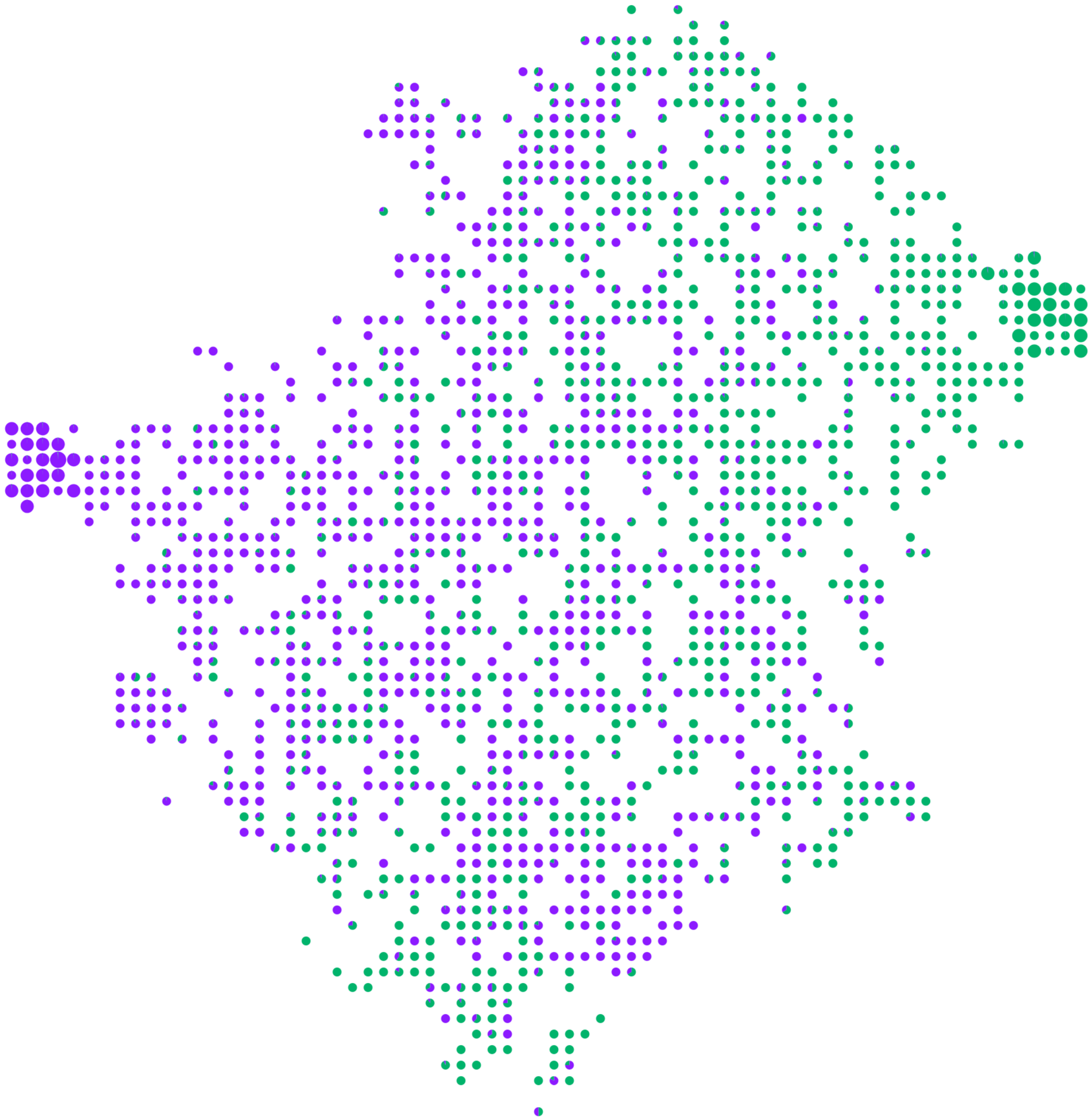


Figure 4.3: A complex model with 1596 rules. The model has 20 trees, max_depth:7, Accuracy: 94.48%.

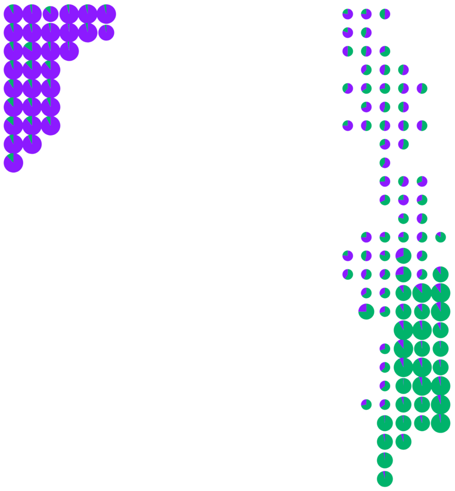


Figure 4.4: A simple model with 120 rules. The model comprises of 30 trees, max_depth:2, Accuracy: 94.3%.

unable to visualize the entire model or the complexity in interpretation was very high. The scalable nature of *RFMap* helps him visualize the whole model with various logic rules. He notices that there are two knowledge clusters in this model, but most of the rules are very specific (low coverage), with some being certain and some not. More specific low-coverage rules imply that they have not learned much from the training data and hence not very useful. John then loads up Model 2 (Fig. 4.4), and from the *Forest View* he sees that there are two strong knowledge clusters in this model with only a few specific (low-coverage) rules. This model's strong knowledge clusters are mostly very high in coverage, meaning they have learned good relationships from the training data samples and, therefore, more generic. He inspects the rules' certainty in the two strong knowledge clusters and is satisfied that they have a high inclination towards their respective classes.

John concludes from his comparative analysis (**T6**) that a balance between coverage and certainty is a very important criteria in case of evaluating RF models, i.e., a model comprising of rules that are high on certainty but low on coverage (like Model 1) will lead to specific knowledge learning. However, a model with higher coverage and lower certainty rules will mean the model has not learned any robust knowledge and results in lower prediction capabilities. Therefore, in this case, Model 2 might be the ideal one for John to select as they have more generic rules which are also highly certain, thus establishing a right balance between certainty and coverage. The ability to derive insights about the knowledge clusters (generic rules), specific rules, and certainty-coverage balance is not available in traditional model evaluation techniques, and being able to compare multiple RF models using the *RFMap* system helps John in assessing the model not just from the perspective of 'accuracy', thus giving him a much effective model. He is also convinced that the marginal loss in accuracy by selecting Model 2 (94.3%) instead of Model 1 (94.48%) is acceptable as long as he has a generic model that can predict well on unseen data.

4.3 Usage Scenario 3: University Admit

This usage scenario presents Sam, a prospective Graduate (Masters) student looking for admissions into universities in the US. He wants to use a Machine Learning algorithm to predict his chances of getting Admit(1) or No Admit(0) based on features - GRE score, GPA, and University Rank. The dataset used for training his model is from UCLA[1] and has a record of 400 students, out of which 273 received no admit, and the remaining 127 were able to secure admits. Sam's primary focus is to interpret his algorithm's workings with the dataset so he can develop trust and confidence in the predictions and finally use it to predict his chances of getting admission or not. To achieve this goal, Sam trains a RF model and employs the *RFMap* system to deduce insights. He adds ten trees in the forest with a maximum depth of 5 assigned to each tree and uses 'Gini' as splitting criteria. The result is a model with an accuracy of 74.1% and 233 decision paths(rules). He understands that although the model's accuracy is not very ideal for individual predictions, it can be used to derive certain conclusions in terms of knowledge learning, misclassifications, etc.

Sam opens up the *RFMap* tool, and from the legend in the View, he understands that purple bubbles represent an 'Admit' and green bubbles represent 'No Admit'. He goes to the 'Instance View' and notices that there is no clear separation(**T3**) between the two class of instances(Figure 4.5A), which is correct since the model is not very accurate and makes many mistakes. However, he does see that the top-left portion consists of instances mostly belonging to 'Admit' class, and the below portion towards the right are instances from the 'Non Admit class. Sam then moves to the *Forest View* (Figure 4.5B) and is interested in exploring the forest of trees (**T2**) to

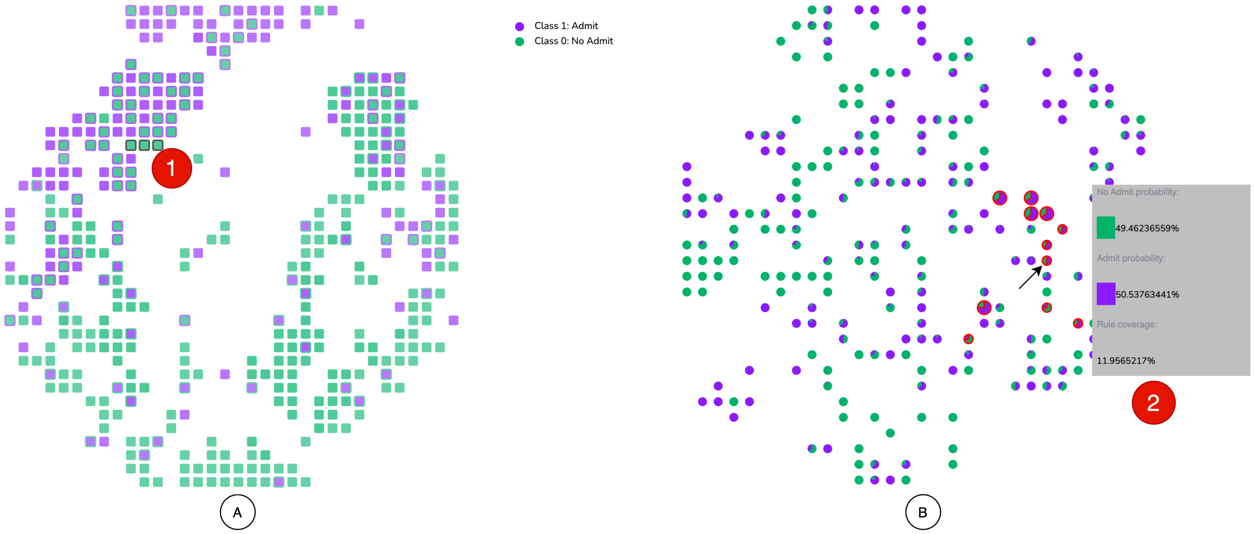


Figure 4.5: RFMap system with University Admit dataset. (A)(1) Sam clicks on a few misclassified instances(highlighted in black) and the corresponding logic rules are highlighted in *Forest View*; (B)(2) hovering over a logic rule shows marginal probabilities.

develop his understanding with the various decision paths. He notes from the pie charts in the view that most of the rules do not have 100% certainty when predicting instances and believes these rules are the ones which results in the model being less accurate, resulting in misclassifications. He also does not see any ‘strong knowledge cluster’ formed inside the forest(**T1**) and every rule is mixed up with others. This gives him the idea that the model has not learned enough from the training data and that he might need to get some more data to build up a concrete model.

Sam notices many misclassified ‘Non Admit’ instances on the top left part of the *Instance View* and decides to perform some local interpretation to understand more. He selects a few instances (highlighted in black) from the *Instance View* as shown in Figure 4.5A(1) and observes the decision paths in the *Forest View* (**T4**). He hovers over the logic rules (Figure 4.5B(2)) and realizes that majority of these paths have individual class probabilities on the margin i.e.,(55-45% or 60-40%). Sam now affirms to his analysis that the decision paths with certainty values in boundary, may lead to misclassifications.

Since Sam’s target is to secure an admit into a university, he is curious to understand which features significantly impact positive and negative predictions. So he

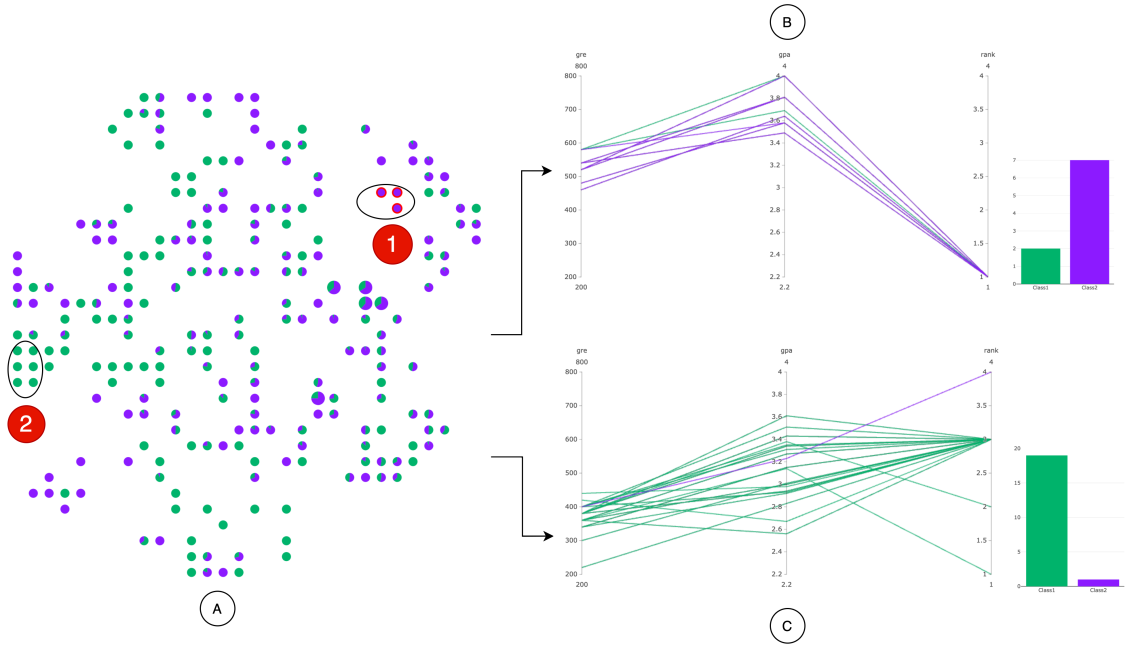


Figure 4.6: RFMap system with University Admit dataset. (A)(1): A lasso is drawn around a cluster of 'purple' logic rules in the *Forest view*; (2): A lasso is drawn around a cluster of 'green' logic rules; (B): knowledge content of the 'purple' group of rules visualized in the *Knowledge-Content view* to understand 'Admit' criteria; (C): knowledge content of the 'green' group of rules visualized in the *Knowledge-Content view* to understand 'Non Admit' criteria;

decides to select a group of certain 'green' cluster and 'purple' cluster to perform analysis (**T1**). He draws a lasso on the small cluster of three purple rules (Figure 4.6A1) and goes to the 'Knowledge-Content View' (Figure 4.6B). He notices that there are 15 purple-class and 5 green-class instances. The purple instances using these rules have GPA>3.5, university rank of 1 and GRE score>470. Now, he lasso selects the green group of rules (Figure 4.6A2) and observes the feature values (Figure 4.6C). This time, for majority of the green instances, GPA is less than 3.4, university rank is 3 and GRE<470. This contrastive analysis globally, helps Sam understands the feature range needed for an 'Admit' into US universities, which allows him to make hypotheses on unseen data(**T5**). E.g., if someone has a GPA of 3.5, their university rank is less than 3 and GRE score is 500, they will probably have higher chances of getting admissions.

Chapter 5

User Study

5.1 Study Design and Tasks

A quantitative and qualitative study was conducted to evaluate the effectiveness and ease-of-use of *RFMap* in interpreting RF models. Our primary goal is to understand whether users can derive general insights about an RF classification model by interacting with the system and explaining their predictions so they can build up trust in the model. By evaluating the ability to accomplish the tasks, a general consensus is computed among all the participants, thus helping us determine the usability of the tool.

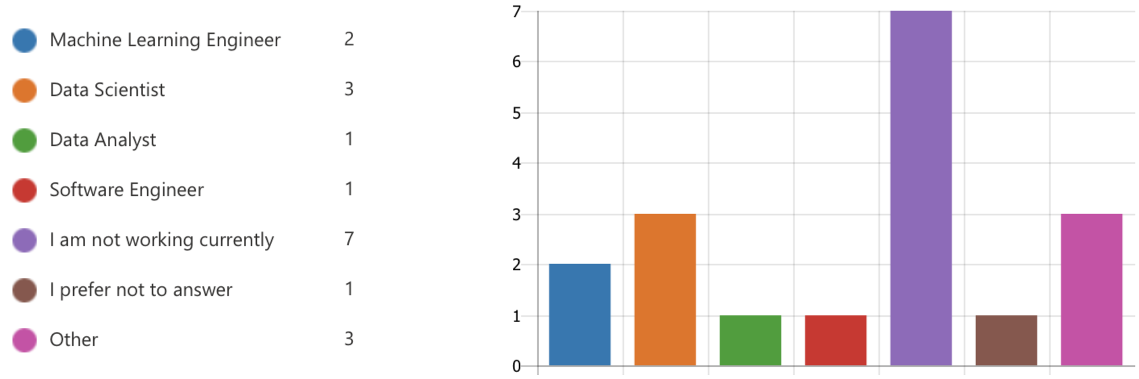


Figure 5.1: Figure shows the current job role for all the participants in the study.

The total number of participants recruited for the study was 18. Out of them, 7 were ‘not working’ and were from a pool of graduate students comprising of Masters and PhD students with very good knowledge of ML and specifically RF models. Three were working as Applied and Research Data scientists, 2 were ML engineers, 1 was a Data Analyst in the software industry. The remaining 4 had other positions (Data engineer, Cloud engineer, Developer) and 1 did not disclose their working role (Fig 5.1). Among these participants, 16 had a ‘Computer Science’ degree and the remaining 2 were from ‘Mathematics’ background (Fig 5.2). Also, from the entire

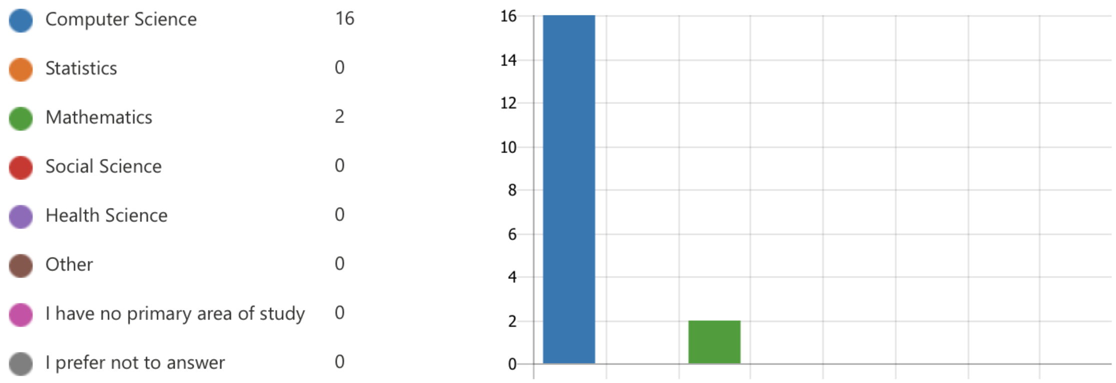


Figure 5.2: Plot shows the field of study of all the participants in the study.

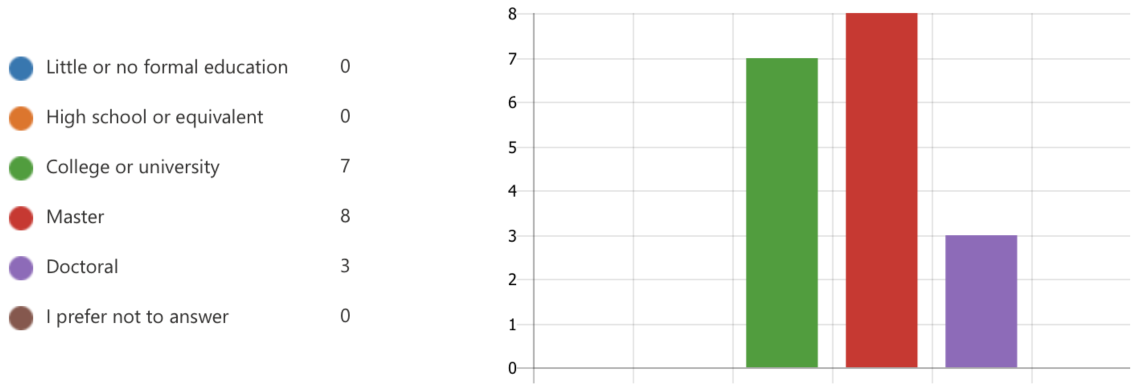


Figure 5.3: Plot shows the level of education of all the participants in the study.

population, 8 of them had completed 'Masters', 7 had University degree and rest 3 were PhD's as seen in Fig 5.3.

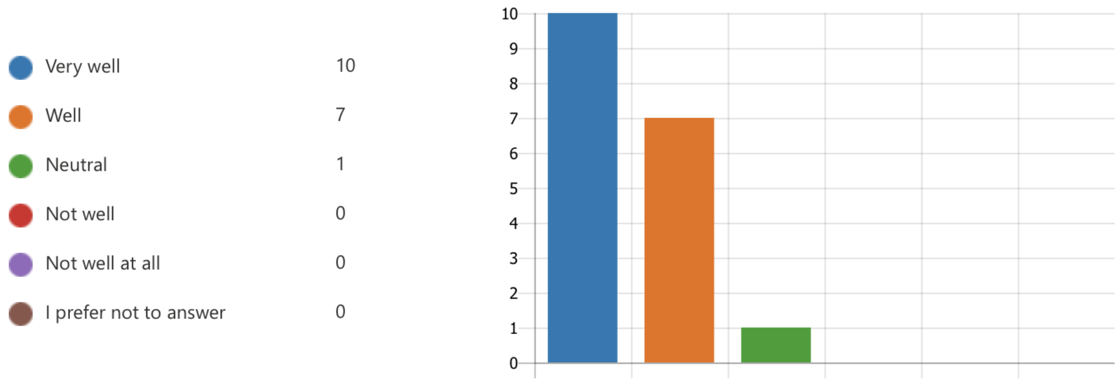


Figure 5.4: Chart shows the familiarity of the participants with visualization plots.

All the participants were interacting with the tool for the first time but had an

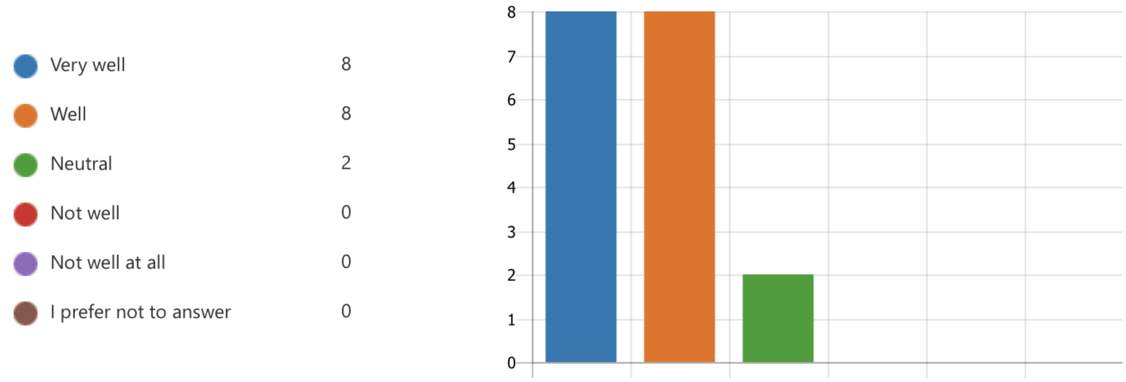


Figure 5.5: Chart shows the familiarity of the participants with Random Forest model.

extensive background with Data visualization plots(17 well to very well familiarity and 1 neutral, Fig 5.4) and Random Forest models (16 well to very well familiarity and 2 neutral, Fig 5.5). Twelve individuals rated their familiarity with Visual Analytics tool high (well to very well), and the other 6 had neutral or not well exposure as shown in Fig 5.6.

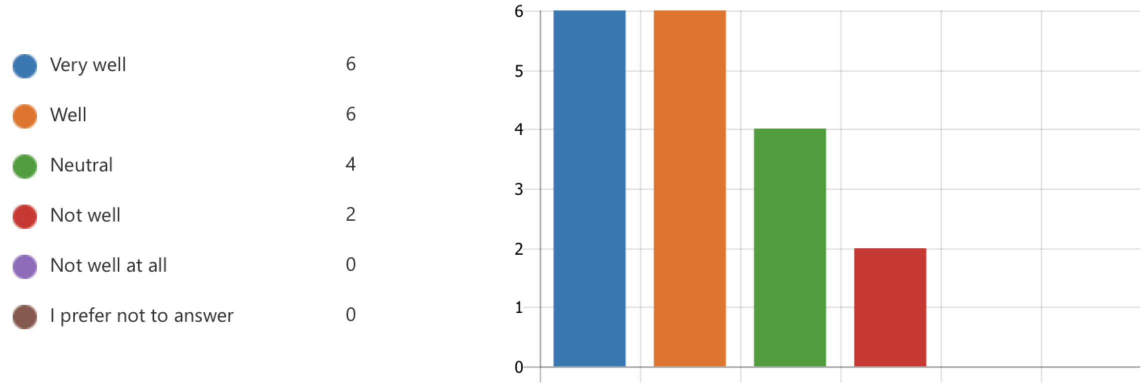


Figure 5.6: Responses to: "How familiar are you with Visual Analytics tools, such as Qlik, Tableau, or Power BI?"

All of our participants were comfortable (13 extremely, 4 very and 1 comfortable) with using interactive user interfaces (Fig 5.7) and 94.4% among them used interactive interfaces frequently (often, very, extremely) as seen in Fig 5.8. The entire group(100%) said that they use tools involving visual representations often, very often or extremely often as compared to text (Fig 5.9).

The user study was conduced remotely. Each participant was initially presented

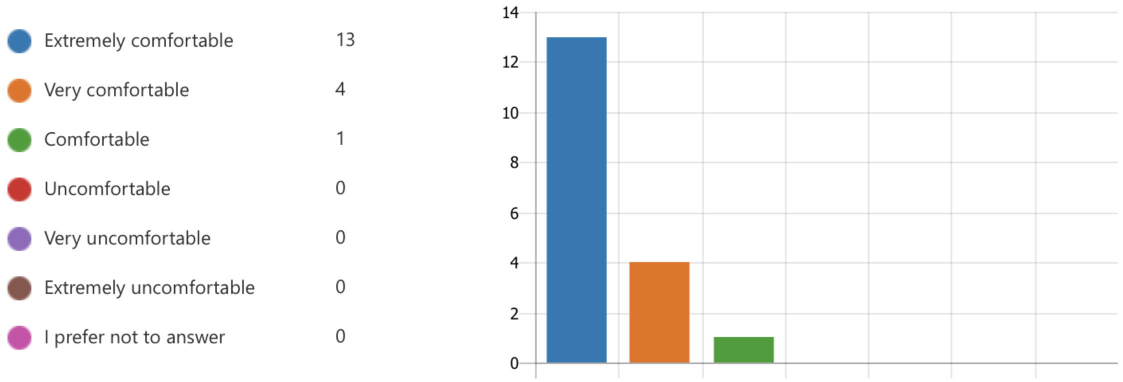


Figure 5.7: Chart shows the comfort level of the participants with interactive interfaces.

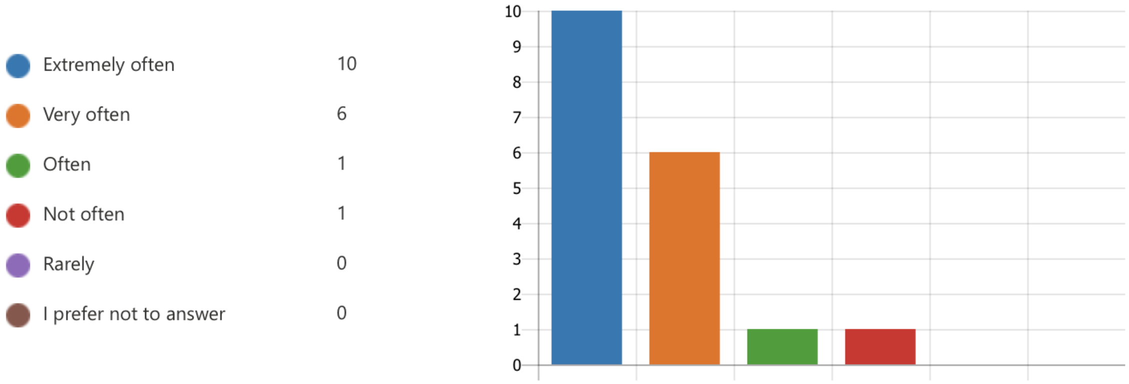


Figure 5.8: Figure shows the frequency of the participants with using interactive interfaces.

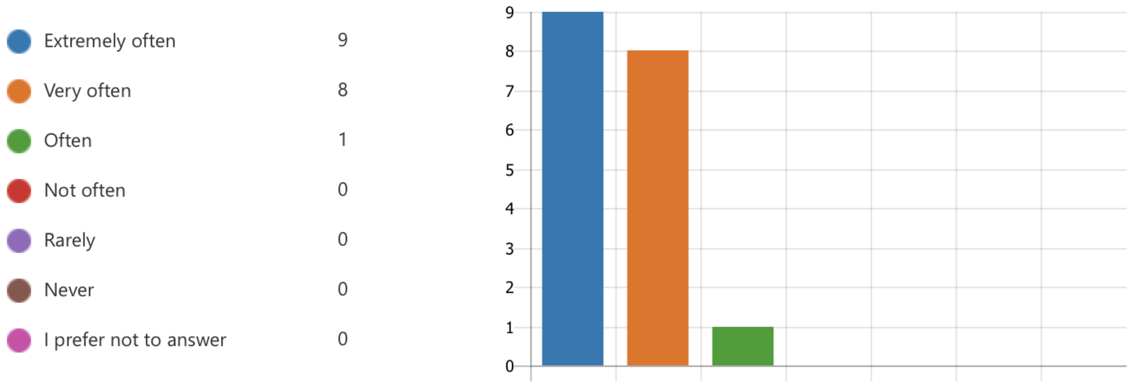


Figure 5.9: Chart shows the frequency of the participants with using visual tools instead of text.

with a tutorial video on RF models and logic rules extraction, followed by another demonstration on using the *RFMap* system. Every individual had an opportunity to explore the tool and ask any questions after developing some familiarity with the tool. Participants were then presented with a set of demographic questions. For the first part of the study, we designed tasks to allow users to use the tool's various visualization components and interactivity functions practically, for example, interpreting misclassified instances, knowledge clusters, making selections on instances to understand the decisions made by the model, analyzing similar or dissimilar instances and understanding the reasons behind misclassifications. After executing every task using the system, the participants were asked to respond to two sets of 5-point Likert scale assessment to judge their experience and complexity level with performing the tasks and get an overall experience of the system. Finally, two open-ended questions were presented to have user's subjective feedback on the overall functionality, quality and any suggestions for future improvement of the tool.

5.2 Results and Feedback

The average time taken by all the participants to complete the tasks and qualitative questions was 38'(including two tutorial videos). The tutorial of the system can be accessed at - https://www.youtube.com/watch?v=ZcUyCBjlVpE. The study comprised of 9 task-based questions that covered both global and local perspectives for RF models interpretation. A summary of the tasks and their relationship to our design goals and tasks is presented in Table 5.1.

The descriptive statistics is computed for the participants response to the questions and is presented as follows. For the first two questions (**Q1,Q2**), participants were asked to interpret misclassified instances from the 'Instance View'. Among all the participants, 94.4% of them responded correctly with 3 as the total number of misclassified instances and 'Malignant' as the original class for misclassified instances. The next two questions involved interpreting 'knowledge clusters' and analyzing the logic rules used to classify the instances from the Forest View. For the first one (**Q3**), 94.4% of the participants gave the correct answer and for the second(**Q4**), 100% were able to understand the decision making process for local instances.

Another question(**Q5**) touched upon interpreting the structural differences among instances using the Feature view, to which, 94.4% of them responded correctly. The

Task	Goal	Question	Percentage of correct responses
T5	G2	Q1: How many misclassified instances do you spot from the 'Instance View'?	94.4%
T5	G2	Q2: The misclassified instances in the 'Instance View' were originally from which class?	94.4%
T1, T2	G1	Q3: How many strong 'knowledge clusters' can you spot in this Random Forest model from the 'Forest View'?	94.4%
T4	G2	Q4: Which set of rules are used to classify Instance 136?	100%
T5	G2	Q5: Instance 136 and its neighbouring instance 161 have a significant difference for which feature value?	94.4%
T3, T5	G1, G2	Q6: What can we infer from the orange 'knowledge cluster' in this Random Forest model? Select all the option that applies.	77.7%, 88.8%, 83.3%
T4, T5	G2	Q7: What is the reason behind the misclassification of Instance 264?	83.3%
T3	G1	Q8: From the 'Instance View', I can validate that the RF model is mostly able to separate the instances into two groups(classes).	94.4%
T1, T2, T4	G1, G2	Q9: The 'outlier' instances like the highlighted ones in the image are not clearly separated because?	88.8%

Table 5.1: Summary of the user study's tasks and their relation with the design goals.

reasons behind misclassification of instances was also asked (**Q7**) to determine using the system and 83.3% of the recruited participants responded correctly to it.

The remaining three questions were targeted towards deriving insights from a knowledge cluster (**Q6**), ability to interpret effectiveness of the model by understanding the decision boundary (**Q8**) and understanding rule properties (coverage, certainty) (**Q9**). The first one was a multiple choice question and no one selected the wrong answer. The second question was rightly answered by 88.8% and the final one had 94.4% right responses.

After the completion of tasks, the participants' responded to the the 5-point Likert-scale questions to assess the user interface, complexity in performing the tasks and the overall system. The results are summarized in the Table 5.2 and 5.3. Most of the participants agreed(66.6% Strongly Agree, 22.2% Somewhat Agree) that the logic rules and their properties were easy to analyze from the system. Regarding interpreting classified or misclassified instances, 17 out of the total 18 participants showed strong agreement. When asked about how complex it was to interpret and compare instances using the various visualization in the system, 83.3% strongly agreed it was easy for them. Finally, 72.2% of the participants strongly agreed that they would be able to interpret more RF models on their own using this tool.

The feedback on the system was also collected through subjective questions. Two of the ML Engineers mentioned - 'they would like to use the tool in their day-to-day job' and 'a very good system to understand RF models'. Another two participants recommended having a feature to understand 'feature importance'. We also received some feedback on incorporating the "values of accuracy and loss to support interpretation". One participant provided feedback for future work saying that the system could help "understand how much weight the model applies to each logic rule / feature / etc. when it classifies a particular point; what is emphasized currently is the coverage, which is also interesting, but different".

Table 5.2: Post Evaluation user feedback for the RFMap's interface.

Question	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
The decision paths (rules) and their properties are easy to interpret from the system.	0	0	2	4	12
The functionalities in the tool are self-explainable and easy to use.	0	0	2	6	10
I understand what class an instance is classified/misclassified as effectively.	0	0	0	1	17
I can differentiate between high and low coverage rules.	0	0	1	3	14
Having a Visual Analytics tool to visualize the entire Random Forest model is overwhelming.	11	5	1	1	0
Having a visual system to explain the workability of Random Forest makes it easy to interpret.	0	0	0	5	13
The 'Parallel coordinate plot' used to display the feature values is very helpful.	0	1	1	5	11
The number of functionalities provided are sufficient to interpret the Random Forest model.	0	1	4	7	6
You can perform the tasks shown in the video with ease.	0	0	1	3	14
The graphs were easy to interpret and compare different instances.	0	0	1	2	15
The labels were clear, and choices of datasets were enough to understand interpretation of Random Forest.	0	0	0	7	11
The choice of color used for Visualization are effective.	0	1	1	3	13

Table 5.3: Post Evaluation user feedback for the Software usability.

Question	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
I think that I would like to use this system frequently	0	0	3	3	12
I found the system unnecessarily complex	17	1	0	0	0
I thought the system was easy to use	0	0	0	4	14
I found the various visualizations in this system were well integrated.	0	0	1	2	15
I thought there was too much inconsistency in this system	16	2	0	0	0
The video was very effective for me to quickly learn to use the system	0	0	1	5	12
I would be able to interpret more Random Forest models on my own now using the system.	0	0	2	3	13

Chapter 6

Conclusions

In this research work, we present *Random Forest Similarity Maps (RFMap)*, an interactive visualization tool that supports the explanation of RF models globally and locally by preserving the forest context. *RFMap* uses multidimensional projection techniques to visualize the forest of decision paths and explain the structural relationships between the data instances from the model's perspective. It also helps users visualize RF models with thousands of logic rules, one of the most scalable visualization solutions for RF analysis. To prove the effectiveness of our technique, we present three user scenarios and the results from a user study that summarizes the useful applications of *RFMap* in various domains. The results show that the tool can help understand RF models from the multiple perspectives discussed in this work.

6.1 Discussions

In this section we discuss some of the limitations we found in our work and how we address them in the future.

Application

We have developed the *RFMap* system by considering interpretability for RF models. However, the system can be extended to understanding any ML model based on rules. With the growing demand for knowledge extraction techniques, i.e., transforming complex models like deep neural networks to surrogate models, our system can be beneficial from a visual interpretation perspective. The system also sheds light on some of the important elements of a classification problem like misclassifications, within-class overlaps, accuracy, and so on, which allows model developers to build better models in the future.

Visualization

The two visual components of our system, *Instance and Forest Views* are based on multidimensional projections techniques. Projection-based techniques have enabled us to visualize thousands of decision paths and data instances in one single frame, thus reducing the load on the human mind and producing visual metaphors that are very scalable. Complex machine learning models like ensembles, deep networks have seen a huge rise in their application in the industry and research works. And as we develop more complex models, the need for scalable and robust visual analytics systems is imperative.

Although projection-based methods have allowed us to visualize an entire forest of trees with thousands of decision paths, these techniques often face the problem of reliable representations since converting high-dimensional data to a lower, human interpretable form is practically very complex. In this research work, we keep this problem in mind, especially when designing the *Instance View* and make use of stress [27] to show users reliable and non-reliable data points. We design the non-reliable data instances as less transparent so users can intuitively differentiate the reliable points from the non-reliable ones in the visualization.

Another problem that was observed during the early development phases of the *RFMap* system is that the icons (circles in the Instance View, pie-charts in the Forest View) suffered from a visual clutter issue. Since our whole rationale behind designing the system was based on whether or not an instance uses a particular rule for classification and how similar or not they were, many instances and decision paths occupied the same visual space when they had common elements. The result was visualization with occlusion issues. To tackle this problem, we leveraged a binary partitioning grid algorithm [20]. This helped us place each element (instance or rule) in a single grid while still maintaining the actual distances between them as much as possible.

Other than the positive feedback that we received from the participants of our user study, we have observed that *RFMap* also has a couple of interesting avenues for future research. Incorporating 'feature importance' and 'splitting criteria' is something we plan to be working on in the future to broaden the perspectives of explaining RF models. Other than that, our goal in the future would be to extend our technique to other ML models based on rules. One area that we also plan to research about is how we make the system valuable for users who do not have an in-depth understanding of the ML models. An interesting feedback that we received from one of the participants is - “I am not an expert in ML, but seeing the video and playing with the tool helps me get the basic idea and understand how the random forest model is making it’s decisions”. The feedback helps us understand that the current system has the potential to benefit users with basic knowledge of ML. In the future, we would like to conduct a more focused study with such specific group of people to understand the requirements better.

Bibliography

[1] UCLA hypothetical data for Graduate admissions. `https://stats.idre.ucla.edu/stat/data/binary.csv`.

[2] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[3] Bilal Alsallakh, Allan Hanbury, Helwig Hauser, Silvia Miksch, and Andreas Rauber. Visual methods for analyzing probabilistic classification data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1703–1712, 2014.

[4] Saleema Amershi, Max Chickering, Steven M. Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. Modeltracker: Redesigning performance analysis tools for machine learning. *Conference on Human Factors in Computing Systems - Proceedings*, 2015-April:337–346, 2015.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine Bias, 2016. `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`.

[6] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

[7] Heer Jeffrey Bostock Michael, Ogievetsky Vadim. D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 7(12):2–3, 1997.

[8] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[9] Leo Breiman. Wald lecture ii. looking inside the black box. page 34, 2002. `https://www.stat.berkeley.edu/users/breiman/wald2002-2.pdf`.

[10] Gabriel D. Cantareira, Elham Etemad, and Fernando V. Paulovich. Exploring neural network hidden layer activity using vector fields. *Information (Switzerland)*, 11(9):1–15, 2020.

[11] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible Models for HealthCare. pages 1721–1730, 2015.

[12] Angelos Chatzimparmpas, Rafael M. Martins, Ilir Jusufi, and Andreas Kerren. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization*, 19(3):207–233, 2020.

[13] Seung Seok Choi, Sung Hyuk Cha, and Charles C. Tappert. A survey of binary similarity and distance measures. *WMSCI 2009 - The 13th World Multi-Conference on Systemics, Cybernetics and Informatics, Jointly with the 15th International Conference on Information Systems Analysis and Synthesis, ISAS 2009 - Proc.*, 3(1):80–85, 2009.

[14] Paulo Cortez and Mark J. Embrechts. Opening black box Data Mining models using Sensitivity Analysis. *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*, pages 341–348, 2011.

[15] Federica Di Castro and Enrico Bertini. Surrogate decision tree visualization interpreting and visualizing black-box classification models with surrogate decision tree. *CEUR Workshop Proceedings*, 2327, 2019.

[16] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[17] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision making and a ”right to explanation”. *AI Magazine*, 38(3):50–57, 2017.

[18] Ronny Hänsch, Philipp Wiesner, Sophie Wendler, and Olaf Hellwich. Colorful trees: Visualizing random forests for analysis and interpretation. *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, pages 294–302, 2019.

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[20] Gladys Hilasaca and Fernando V. Paulovich. Distance Preserving Grid Layouts. 00(0):1–15, 2019.

[21] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng Polo Chau. Summit: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1096–1106, 2020.

[22] Kelli D. Humbird, J. Luc Peterson, and Ryan G. Mcclarren. Deep Neural Network Initialization With Decision Trees. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5):1286–1295, 2019.

[23] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[24] Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng Polo Chau. ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):88–97, 2018.

[25] Ron Kohavi. The power of decision tables. In Nada Lavrac and Stefan Wrobel, editors, *Machine Learning: ECML-95*, pages 174–189, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[26] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

[27] J B Kruskal. Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[28] Natalia Kuznetsova. Random forest visualization Eindhoven University of Technology Master Thesis Random Forest Visualization. 2014.

[29] Ken Lau. Random Forest Ensemble Visualization, 2014. `https://www.semanticscholar.org/paper/Random-Forest-Ensemble-Visualization-Lau/9cb33b2459730e8b2bf2b3364bbf5ffc28337523`.

[30] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[31] Yiran Li, Takanori Fujiwara, Yong K. Choi, Katherine K. Kim, and Kwan-Liu Ma. A visual analytics system for multi-model comparison on clinical data predictions. *Visual Informatics*, 4(2):122–131, 2020. PacificVis 2020 Workshop on Visualization Meets AI.

[32] Q. Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the ai: Informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–15, New York, NY, USA, 2020. Association for Computing Machinery.

[33] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[34] Gilles Louppe. Understanding Random Forests: From Theory to Practice. (July). `http://arxiv.org/abs/1407.7502`.

[35] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020.

[36] C. F. Luz, M. Vollmer, J. Decruyenaere, M. W. Nijsten, C. Glasner, and B. Sinha. Machine learning in infection management using routine electronic health records: tools, techniques, and reporting of future technologies. *Clinical Microbiology and Infection*, 26(10):1291–1299, 2020.

[37] D. Martens, B. B. Baesens, and T. Van Gestel. Decompositional rule extraction from support vector machines by active learning. *IEEE Transactions on Knowledge and Data Engineering*, 21(2):178–191, 2009.

[38] Morteza Mashayekhi and Robin Gras. Rule extraction from random forest: the rf+hc methods. In Denilson Barbosa and Evangelos Milios, editors, *Advances in Artificial Intelligence*, pages 223–237, Cham, 2015. Springer International Publishing.

[39] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: Personalizing explainable recommendations with bandits. *RecSys 2018 - 12th ACM Conference on Recommender Systems*, pages 31–39, 2018.

[40] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[41] Yao Ming, Huamin Qu, and Enrico Bertini. RuleMatrix: Visualizing and Understanding Classifiers with Rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, 2019.

[42] Mário Popolin Neto and Fernando V. Paulovich. Explainable Matrix – Visualization for Global and Local Interpretability of Random Forest Classification Ensembles. 2020.

[43] T. D. Nguyen, T. B. Ho, and H. Shimodaira. A visualization tool for interactive learning of large decision trees. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2000-January:28–35, 2000.

[44] Plotly. Plotly Parallel coordinate. `https://plotly.com/javascript/parallel-coordinates-plot/`.

[45] J. R. Quinlan. Generating production rules from decision trees. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'87, page 304–307, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.

[46] Paulo E. Rauber, Samuel G. Fadel, Alexandre X. Falcão, and Alexandru C. Telea. Visualizing the Hidden Activity of Artificial Neural Networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, 2017.

[47] Paulo E. Rauber, Alexandre X. Falcaõ, and Alexandru C. Telea. Projections as visual aids for classification system design. *Information Visualization*, 17(4):282–305, 2018.

[48] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D. Williams. Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):61–70, 2017.

[49] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-Agnostic Interpretability of Machine Learning. (Whi), 2016. http://arxiv.org/abs/1606.05386.

[50] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu:1135–1144, 2016.

[51] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1527–1535. AAAI Press, 2018.

[52] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1283–1292, New York, NY, USA, 2009. Association for Computing Machinery.

[53] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. *AIES 2018 - Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310, 2018.

[54] Warren S Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[55] Marco A. Valenzuela-Escárcega, Ajay Nagesh, and Mihai Surdeanu. Lightly-supervised Representation Learning with Global Interpretability. *arXiv*, 2018.

[56] W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology*, 1905(870):861–870, 1993.

[57] Chengliang Yang, Anand Rangarajan, and Sanjay Ranka. Global Model Interpretation Via Recursive Partitioning. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, pages 1563–1570, 2019.

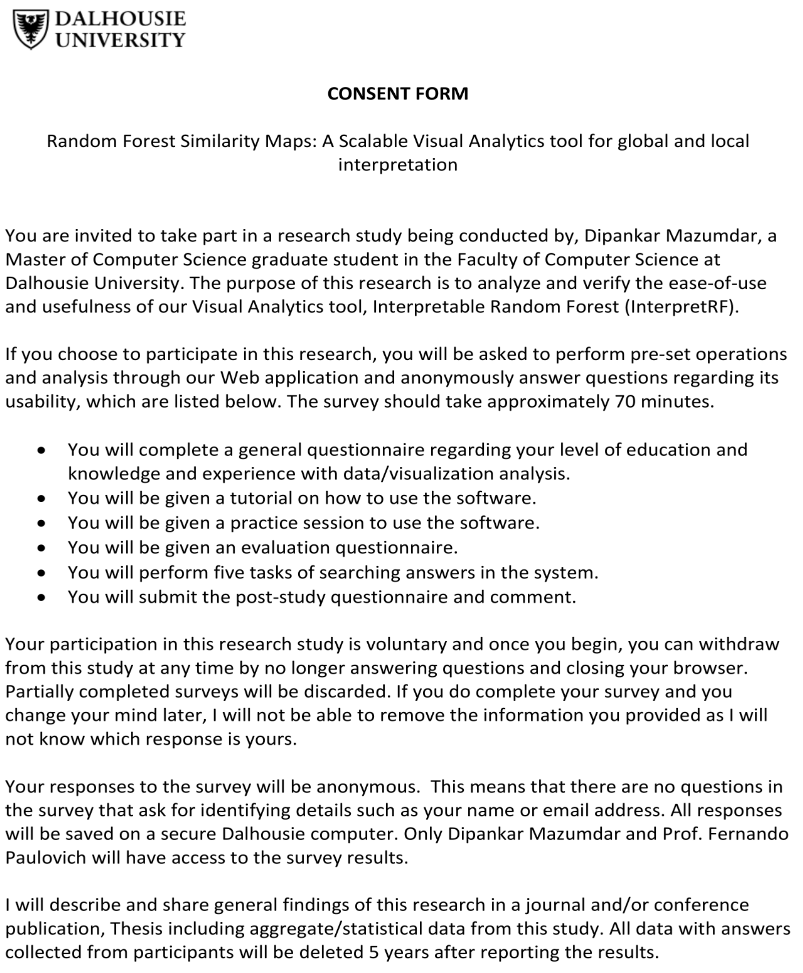
[58] Tom Zahavy, Nir Ben Zrihem, and Shie Mannor. Graying the black box: Understanding DQNs. *33rd International Conference on Machine Learning, ICML 2016*, 4:2809–2822, 2016.

[59] Ye Zhang and Byron Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. 2015.

[60] Xun Zhao, Yanhong Wu, Dik Lun Lee, and Weiwei Cui. IForest: Interpreting Random Forests via Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):407–416, 2019.

Appendix A

Consent Form



DALHOUSIE UNIVERSITY

CONSENT FORM

Random Forest Similarity Maps: A Scalable Visual Analytics tool for global and local interpretation

You are invited to take part in a research study being conducted by, Dipankar Mazumdar, a Master of Computer Science graduate student in the Faculty of Computer Science at Dalhousie University. The purpose of this research is to analyze and verify the ease-of-use and usefulness of our Visual Analytics tool, Interpretable Random Forest (InterpretRF).

If you choose to participate in this research, you will be asked to perform pre-set operations and analysis through our Web application and anonymously answer questions regarding its usability, which are listed below. The survey should take approximately 70 minutes.

- You will complete a general questionnaire regarding your level of education and knowledge and experience with data/visualization analysis.
- You will be given a tutorial on how to use the software.
- You will be given a practice session to use the software.
- You will be given an evaluation questionnaire.
- You will perform five tasks of searching answers in the system.
- You will submit the post-study questionnaire and comment.

Your participation in this research study is voluntary and once you begin, you can withdraw from this study at any time by no longer answering questions and closing your browser. Partially completed surveys will be discarded. If you do complete your survey and you change your mind later, I will not be able to remove the information you provided as I will not know which response is yours.

Your responses to the survey will be anonymous. This means that there are no questions in the survey that ask for identifying details such as your name or email address. All responses will be saved on a secure Dalhousie computer. Only Dipankar Mazumdar and Prof. Fernando Paulovich will have access to the survey results.

I will describe and share general findings of this research in a journal and/or conference publication, Thesis including aggregate/statistical data from this study. All data with answers collected from participants will be deleted 5 years after reporting the results.

The risks associated with this study are no greater than those you encounter in your everyday life.

To thank you for your time you will receive by email a $50 amazon gift card. Your contact information will not be linked in any way to your survey responses.

You should discuss any questions you have about this study with Dipankar Mazumdar or Prof. Fernando Paulovich. Please ask as many questions as you like before or after participating. My contact information is dp976894@dal.ca

If you have any ethical concerns about your participation in this research, you may contact Research Ethics, Dalhousie University at (902) 494-3423, or email ethics@dal.ca (and reference REB file # 20XX-XXXX)."

If you agree to complete the survey, please answer this email with "I accept the consent agreement", and the link to the survey will be sent to you.