# **Abstract**-

In many machine learning scenarios involving structured data, researchers frequently work with several models that demonstrate similar predictive performance but differ significantly in their structure and interpretability. Selecting one model from this set can be non-trivial, especially in situations where domain constraints or transparency requirements play a critical role in decision-making. The TreeFARMS framework addresses this issue by generating a Rashomon set, a diverse collection of sparse decision trees that are all near-optimal in terms of performance but structurally varied.

However, unpacking and comparing such a large set of models introduces its own challenges. While the models are theoretically useful, manually analyzing each tree is often not feasible. To make this process more accessible, we developed an interactive system that enables researchers to upload structured datasets, generate Rashomon sets using TreeFARMS, and explore the resulting trees through a visual interface.

# **Introduction-**

In modern machine learning applications, especially those involving structured tabular data, researchers often aim to balance predictive accuracy with Explainable structure. Complex models such as neural networks and ensemble methods offer high performance but they are typically difficult to interpret,usually called the black box model This makes them unsuitable for scenarios where transparency and explanation of inner workings are essential. In contrast, simpler models like decision trees are usually preferred in domains like healthcare, education, and judiciary.

A common challenge in these situations is that multiple models may achieve nearly identical accuracy on the same dataset, yet differ significantly in structure. Each of these models can offer a valid, but distinct, explanation of the data. This leads to the idea that the space of possible models may contain many near-optimal solutions. As a result, selecting just one model becomes difficult and may not provide the complete picture needed for confident decision-making.

To address the challenges of model selection,using treefarms[1] recent research has introduced the concept of the Rashomon set [2]. This concept is based on the Rashomon Effect [3], a principle in statistical learning theory that describes the presence of a large number of models that achieve nearly identical predictive accuracy on a given dataset, despite differing significantly in their internal structure, or decision logic. The Rashomon Effect [3] challenges the conventional notion of a single best model by demonstrating that multiple, structurally diverse models can explain the data equally well. This inherent multiplicity introduces ambiguity into model selection and raises concerns about transparency, fairness and interpretability, particularly in high-stakes domains where understanding the rationale behind model predictions is essential.

The Rashomon set [2] is formally defined as the collection of models whose predictive performance falls within a narrow margin of the empirical optimum. These models are typically sparse and interpretable, yet represent a broad spectrum of decision-making strategies. Each model within the set offers a distinct perspective on the data, reflecting different trade-offs and structural patterns that may align with various domain-specific constraints or values. While the existence of such a set provides researchers with greater flexibility and deeper insight into model behavior, it simultaneously introduces a new challenge: how to effectively navigate, compare, and interpret a large number of near-optimal yet heterogeneous models.

This project introduces a system that helps solve this problem. It allows researchers to upload structured and preprocessed datasets, generate Rashomon sets [2] of decision trees using the TreeFARMS [1] framework, and explore the resulting models through an interactive and web-based visualization tool. The system automatically constructs a wide variety of decision trees and collects them into a compact form that highlights both the similarities and differences in their decision paths.

The visualization tool makes it easier to explore these decision paths, observe patterns, and focus on trees that align with specific domain requirements. Instead of displaying individual models one at a time, the system provides an interactive view that allows researchers to understand the full range of decision strategies. Rather than creating a new learning algorithm, the focus of this work is to simplify the process of working with many interpretable models. The system offers a complete workflow that goes from uploading a dataset to visually analyzing the structure of the generated models. This makes it easier for researchers to engage with model interpretability, even if they do not have advanced technical expertise.

All this concludes, the project contributes a practical tool that connects interpretable model generation with exploration and analysis. It supports more thoughtful and transparent model selection, especially in domains where understanding the decision process is critical for trust, fairness, and accountability.

# **Background-**

Decision trees are widely used machine learning algorithms due to their accuracy and flexibility,most tools train and display only one decision tree,one that performs best on certain metrics.This approach hides the fact that many other trees might perform just as well, each using slightly different paths. TimberTrek [4] addresses this by working with what’s known as the Rashomonset [2], which includes a large collection of such models. Instead of choosing one, TimberTrek [4] helps users explore and compare all the possible paths.

Decision trees are gaining popularity in health care and criminal justice,because it is simple enough to memorize by human.Still generating Rashomon [2] and visualizing them into Timbertrek from any dataset is a difficult and complicated task,thus we build a pipeline for automating this process.

# **Related work-**

## Visual analytics for model selection-Machine learning requires the process of selection of suitable models. The visual analytics system has provided success in this process where researchers have the opportunity to apply domain knowledge in creation of models. An example is the interactive system BEAMES [5] allowing a domain expert to specify constraints and to search linear models satisfying these requirements.Moreover, there are a number of tools that are developed for visualization and interpretation of tree-based models. These systems are however, mostly set up to assist the comprehension of individual forests of trees, e.g., random forests and gradient-boosted trees, but not correlation of numerous independent decision conjectures.

## Some of the tools which have the closest connection with the objectives of TimberTrek are TreePOD [6] and a system proposed by Padua et al [7]. The two systems help the users to choose the appropriate decision trees by modifying algorithmic parameters. draws a contrast to TimberTrek [4] that is meant to visualize all of Rashomon's set [2] of sparse decision trees and which offers several levels of abstraction. Every tree in the system will be both fulfilling performance and interpretability requirements and it can be an organized method to investigate different but valid models.

**TimberTrek [4]** is a tool that lets users explore multiple sparse decision trees using an interactive sunburst visualization. It helps compare tree structures and feature usage across models, but requires users to manually generate and upload tree data TreeFARMS [1] .also generating the rashomon set from scratch is a difficult task.

Our system simplifies this by integrating TreeFARMS [1] on the server. Users can upload processed CSV files, and the system automatically trains the models and visualizes them,making the process faster and easier without any manual preprocessing.

# **Methodology-**

## **Overview**

The system is built to support the generation, summarization, and visualization of a Rashomon set [2] of sparse decision trees derived from a structured dataset. It is composed of a modular architecture with distinct backend and frontend components that interact seamlessly to deliver an end-to-end model exploration experience.

The **backend** is implemented using **Python** and uses the **Flask web framework** to handle data ingestion, API endpoints, and communication with the TreeFARMS modeling engine. Upon receiving a dataset uploaded from the client, the backend invokes **TreeFARMS**, which performs model training and produces a Rashomon set containing multiple decision trees. The resulting model data is serialized and prepared for visualization in the frontend.

The **frontend** is developed using **React.js**. Decision tree data is passed from the backend to the frontend via JSON over RESTful APIs. For the visualization, a separate **Svelte-based component**, embedded via an iframe, renders the sunburst-style view of the tree ensemble, using TimberTrek [4]. Communication between the React and Svelte layers is handled using the **postMessage** API to ensure cross-context messaging in a secure manner.

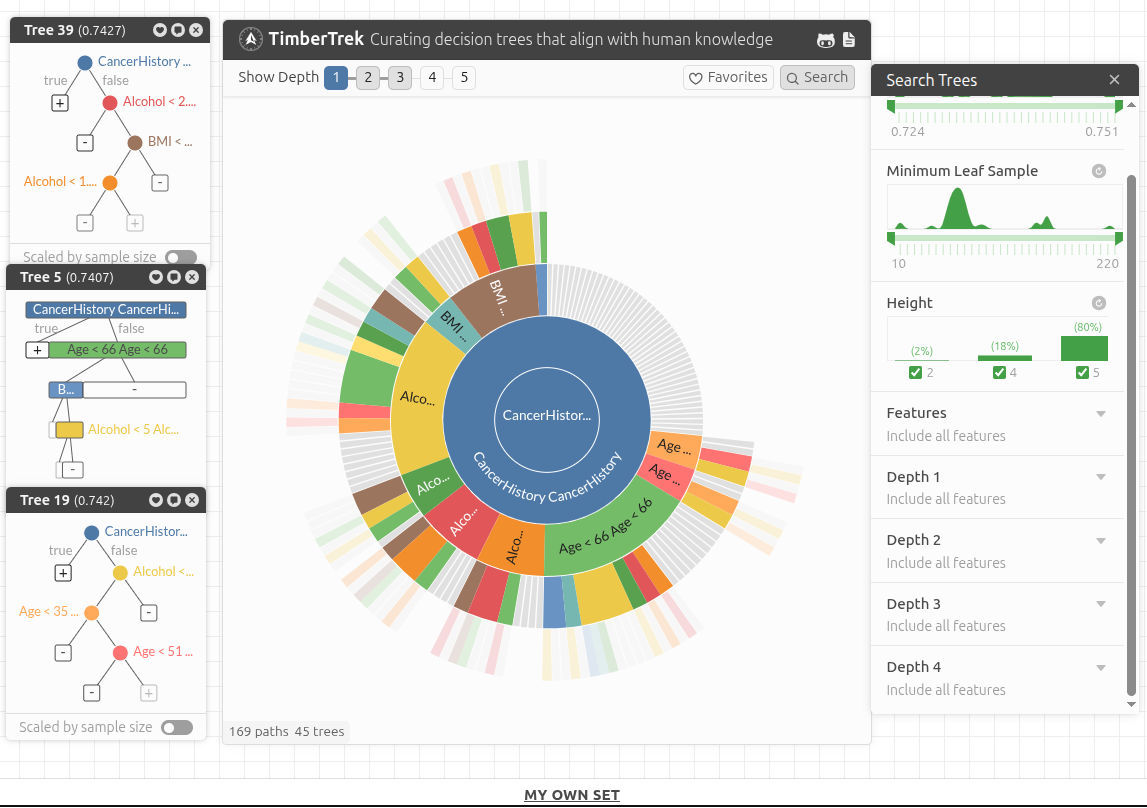


Fig 1 : complete overview

## **System Architecture**

The system (fig-1) is designed using a modular client-server architecture that facilitates a seamless flow from data input to model visualization. Each component in the pipeline plays a specific role in allowing users to interactively explore a set of decision tree models derived from their dataset.

### **1. Data Upload**

The user uploads a structured CSV file through the React-based frontend interface. This component provides a clean and user-friendly form for selecting and submitting datasets. Before the data is transmitted to the server, client-side validation checks are performed to ensure that the file is in the correct format and free from obvious structural errors. This step reduces the chance of processing failures and ensures a smoother experience when interfacing with the backend.

### **2. Model Training**

Once the data is validated, it is sent to the Flask-based backend, which is responsible for handling model generation. The core of this process is powered by TreeFARMS [1], a tool designed to construct sparse decision trees that belong to the Rashomon set [2]. TreeFARMS [1] processes the uploaded dataset and generates a set of decision trees. The resulting rashomon set is exported in a JSON, to support visualization.

### **3. Model Serialization and Summary**

After TreeFARMS [1] completes model generation, the backend proceeds to post-process and serialize the resulting trees. Since the Rashomon set [2] can contain a large number of models, it becomes essential to summarize and organize the data effectively. The system converts the tree structures into a summarizable format such as a decision-path trie or a frequency table of paths. This summarized representation captures the core predictive patterns across the set while reducing redundancy, making the data suitable for efficient and scalable visualization. The backend then packages this summary and prepares it for transfer to the frontend visualization component.

### **4. Interactive Visualization**

The frontend, built with React, embeds the TimberTrek [4] visualization module using an iframe. Once the Rashomon set [2] is generated and summarized by the backend, the data is sent to TimberTrek [4] via the postMessage() API. This enables dynamic and seamless interaction without manual file uploads. TimberTrek [4] uses a Sunburst layout to help users explore decision paths across thousands of sparse decision trees at various levels of abstraction, supporting effective model comparison and selection.

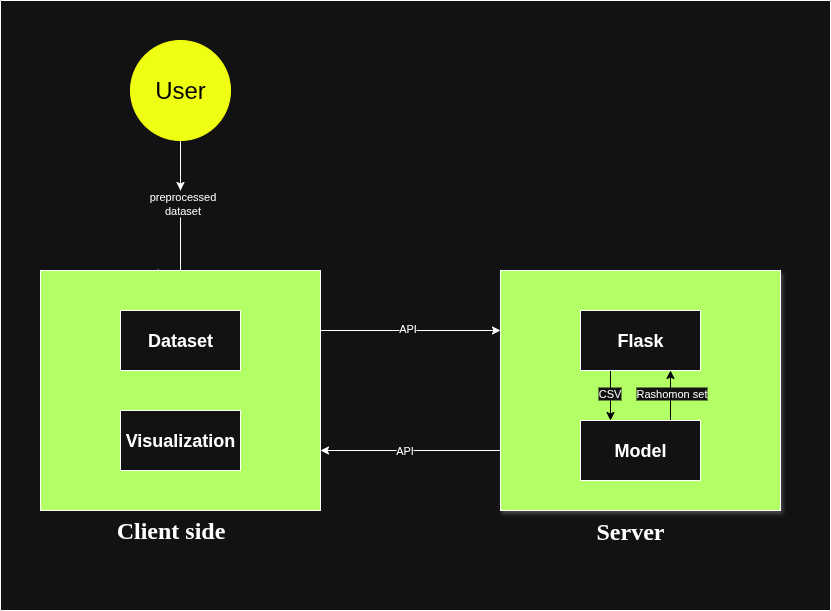


Fig 2 : System Architecture

## **Use Case Demonstration: Cancer Risk Dataset**

To demonstrate the practical use of the system, we used a structured dataset that contained data related to cancer risk assessment. The dataset included both continuous variables such as **Alcohol consumption** (< 1.2, < 2.3, etc.), **BMI** (< 21, < 27, etc.), and **Age** (< 35, < 51, etc.), as well as binary attributes such as **CancerHistory** and the target classification label (**Target**).

First we did preprocessing the dataset. First handled the null and missing value and removed the irrelevant features such as their id’s and location.

Then we processed the feature selection using domain knowledge and removed the irrelevant features. The goal was to retain only those information that highly contribute to the classifying the cancer diagnosis.binning applied to the bmi ,alcohol consumption and age feature to convert them from the numeric value to the categorical values and applied the one hot encoding..

Using TreeFARMS [1], we generated a Rashomon set[2] consisting of multiple sparse decision trees, each of which captured a different plausible decision-making strategy. The resulting trees were visualized using our interactive sunburst-style interface. The system highlighted frequent decision paths involving **Age < 51** and **BMI < 27**, which were commonly used among several models to reach a high-confidence prediction. In contrast, some less common trees utilized **Alcohol < 1.2** and **CancerHistory** earlier in the decision path.

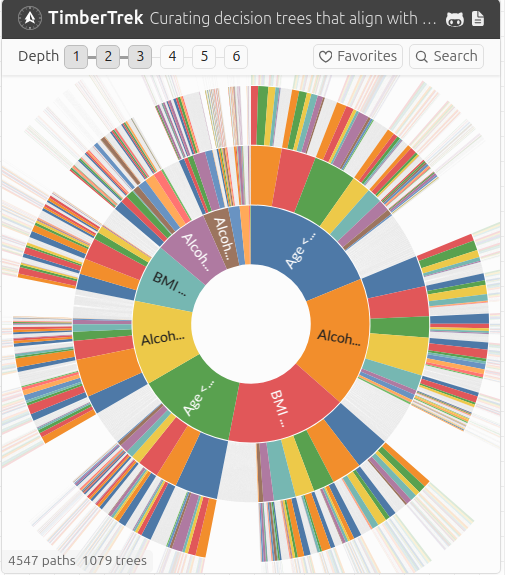


Fig 3 : Timbertrek visualization of Cancer dataset

The visualization helped identify structural similarities and differences across the model set. For instance, in some models, younger age groups and low BMI thresholds contributed significantly to a lower risk prediction, while others prioritized historical cancer presence. These insights were made accessible without requiring manual inspection of each individual tree, emphasizing the system’s capability to support comparative reasoning across diverse models.

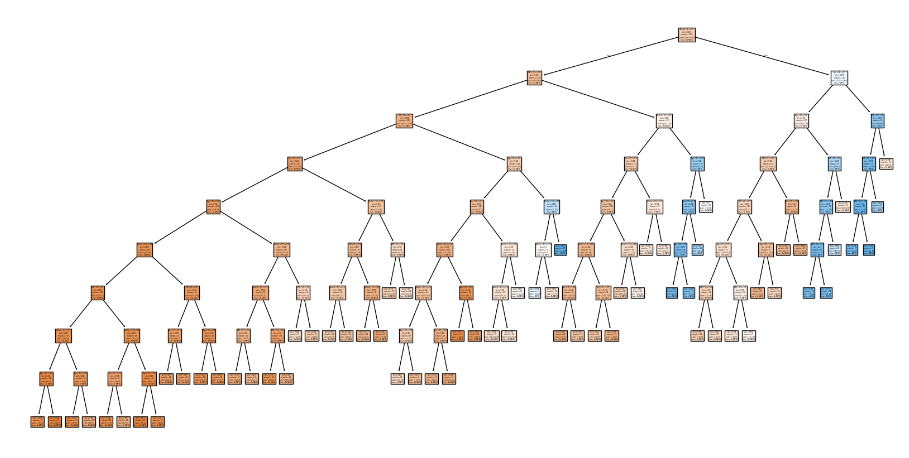


Fig 4: Full decision path

(Fig 3) is extracted from the the rashomon set, this tree provides the information about a how a single root node effects the decision path of decision tree

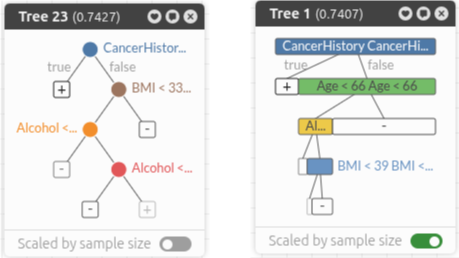


Fig 5: Different decision trees

# **Conclusion and Future work-**

This project presents a complete pipeline for generating, visualizing, and analyzing Rashomon sets[2] of decision trees. The system integrates dataset uploading, sparse model generation using TreeFARMS [1], and interactive visualization through a user-friendly interface. Tested on a cancer dataset, the tool demonstrated its ability to reveal frequent patterns, model diversity, and structural similarities in an accessible format.

While the system effectively enables comparative model exploration, there are opportunities for further improvement. Future work may involve:

* Adding support for larger datasets with better performance
* Extending the backend to allow preprocessing and feature engineering directly within the interface.
* Integrating evaluation metrics (e.g., accuracy, sensitivity, feature importance) alongside structural views to support more informed model selection.

Ultimately, this work contributes toward making interpretable machine learning more usable and transparent by connecting model diversity with meaningful, human-centered exploration.

# **References-**

[1] TreeFARMS, “A Python toolkit for generating Rashomon sets of sparse decision trees,” GitHub. [Online]. Available: <https://github.com/arbml/treefarms>

[2] C. J. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova and C. Zhong, “Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges,” \*Statistical Surveys\*, vol. 16, pp. 1–85, 2022.

[3] L. Breiman, “Statistical Modeling: The Two Cultures,” \*Statistical Science\*, vol. 16, no. 3, pp. 199–231, 2001, doi:10.1214/ss/1009213726.

[4] Z. J. Wang et al., "TimberTrek: Exploring and Curating Sparse Decision Trees with Interactive Visualization," 2022 IEEE Visualization and Visual Analytics (VIS), Oklahoma City, OK, USA, 2022, pp. 60-64, doi: 10.1109/VIS54862.2022.00021.

[5] S. Das, D. Cashman, R. Chang and A. Endert, "BEAMES: Interactive Multimodel Steering, Selection, and Inspection for Regression Tasks," in IEEE Computer Graphics and Applications, vol. 39, no. 5, pp. 20-32, 1 Sept.-Oct. 2019, doi: 10.1109/MCG.2019.2922592.

[6] T. Mühlbacher, L. Linhardt, T. Möller and H. Piringer, "TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 174-183, Jan. 2018, doi: 10.1109/TVCG.2017.2745158.

[7] L. Padua, H. Schulze, K. Matković, and C. Delrieux, "Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections," \*Computers & Graphics\*, vol. 41, pp. 64–75, June 2014, doi: 10.1016/j.cag.2014.02.004.