To explore decision trees using **TimberTrek** in a Jupyter Notebook, follow these steps. This example assumes that you have a set of decision trees saved in a JSON format, which can be loaded and visualized in TimberTrek.

**Step 1: Install TimberTrek**

Install TimberTrek using pip:

python

Copy code

pip install timbertrek

**Step 2: Load TimberTrek and Import Your Decision Trees**

Here's how to use TimberTrek to visualize and analyze your decision trees:

python

Copy code

import timbertrek as tt

import json

# Load your decision trees (assumes you have them saved in a JSON file)

with open("your\_decision\_trees.json", "r") as f:

decision\_trees = json.load(f)

# Initialize the TimberTrek visualization

tt.show(decision\_trees)

**Step 3: Interact with TimberTrek**

Once you run the code, a TimberTrek interface will open in your Jupyter Notebook. You can:

* **Explore decision trees**: View trees in a Sunburst layout, with each ring representing different levels of decision-making.
* **Filter by properties**: Use sliders and checkboxes to filter trees by accuracy, size, or specific features.
* **Compare Trees**: Click on trees to view them in a side-by-side node-link diagram, where you can assess prediction paths and feature importance.

**Example Dataset**

If you don’t have a JSON file, you can generate a Rashomon set of decision trees from any dataset using tools like TreeFARMS for sparse decision tree generation, which can be exported to JSON format for use in TimberTrek​

[GitHub](https://github.com/poloclub/timbertrek)

​

[ar5iv](https://ar5iv.org/abs/2209.09227)

.

This setup is highly valuable for model interpretability, making it easier to inspect, select, and deploy decision trees that best fit your interpretability or accuracy needs