APPENDICES OF

VAR: Visual Analysis for Rashomon Set of Machine Learning Models' Performance

Anonymous

7 Parameter Settings in the Experiment

In the FICO dataset, the depth budget (depth_budget) is set to 4, which constrains the maximum depth of generated trees and ensures interpretability while maintaining reasonable model complexity. The Rashomon bound adder (rashomon_bound_adder) is set to 0.03, determining the acceptable performance gap from the optimal solution - this means we include trees whose performance is within 3% of the best-performing tree. The regularization parameter (regularization) is set to 0.02, which helps balance model complexity with performance by penalizing the number of leaves in the trees. Additional configuration parameters include "rashomon_bound_multiplier" set to 0 and "trivial_extensions" set to True, which help control the diversity of the generated tree set while avoiding redundant model structures. Through these parameter settings, we collected a Rashomon set consisting of 152 models from the FICO dataset.

As the COMPAS dataset is relatively simpler compared to the FICO data, the "depth_budget" is set to 5, allowing for greater tree depth. The remaining parameters include the "rashomon_bound_adder" set to 0.03, regularization set to 0.02, "rashomon_bound_multiplier" set to 0, and "trivial_extensions" set to True. Through these parameter settings, we collected a Rashomon set consisting of 32 models from the COMPAS dataset.

8 Comparison of the different kernel functions for the RBF heatmap visualization

We present 16 different types of kernel functions using the same data points. Four are in each cluster with the mathematic format above the Figure. In all the kernel functions below, r represents radial distance, σ and $c = 1.0$ are kernel function parameters.

By comparing the visualizations of these 16 kernel functions, we can gain a better understanding of their characteristics in RBF heatmaps, providing valuable insights for selecting the appropriate kernel function for specific application scenarios. Overall, the classical kernel functions in the first group perform the most consistently, while custom kernel functions may deliver superior results in specialized contexts.

8.1 Group 1: Classical Kernel Functions

The first group shown in Fig. 5 includes the most commonly used classical kernel functions: Gaussian kernel, multiquadric kernel, inverse multiquadric kernel, and thin plate spline kernel. These kernel functions are widely applied in the field of machine learning and possess excellent mathematical properties. From the visualization results, it is evident that this group of kernel functions generally exhibits good smoothness and continuity, making them particularly suitable for capturing the overall distribution trends in data. The mathematical formats of the kernel functions in Group 1 are shown below:

Gaussian Kernel: $k(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$

Multiquadric Kernel: $k(r) = \sqrt{r^2 + c^2}$

Inverse Multiquadric Kernel: $k(r) = \frac{1}{\sqrt{r^2+c^2}}$

Thin Plate Kernel: $k(r) = r^2 \ln(r)$

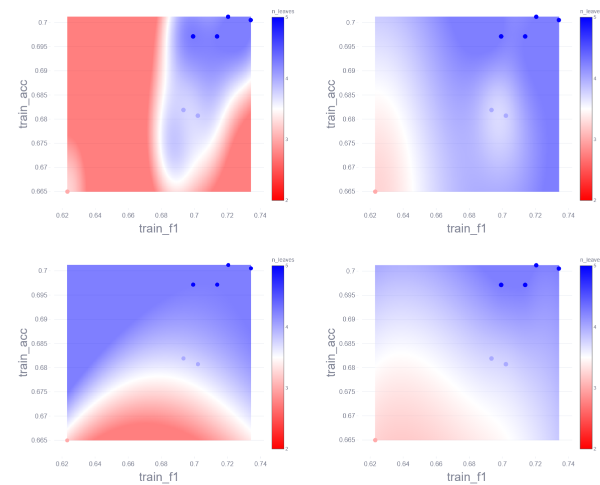

Figure 5: Comparison of different kernel functions used in the RBF-heatmap mode: Gaussian (top left), Multiquadric (top right), Inverse Multiquadric (bottom left), and Thin Plate (bottom right).

8.2 Group 2: Basic Polynomial Kernel Functions

The second group shown in Fig. 6 is mainly composed of basic polynomial functions, including cubic kernels, linear kernels, quadratic kernels, and inverse quadratic kernels. These kernel functions are simple in form and computationally efficient, but they may not be as flexible as the first group when dealing with complex patterns. Observations from the heatmap show that this group of kernel functions provides a relatively direct interpolation effect, and the boundary features are more obvious. However, some kernel functions may not be able to handle some boundaries, resulting in noise points. The mathematical format of the kernel functions in the second group is as follows:

Cubic Kernel: $k(r) = r^3$

Linear Kernel: $k(r) = r$

Quadratic Kernel: $k(r) = r^2$

Inverse Quadratic Kernel: $k(r) = \frac{1}{r^2+c^2}$

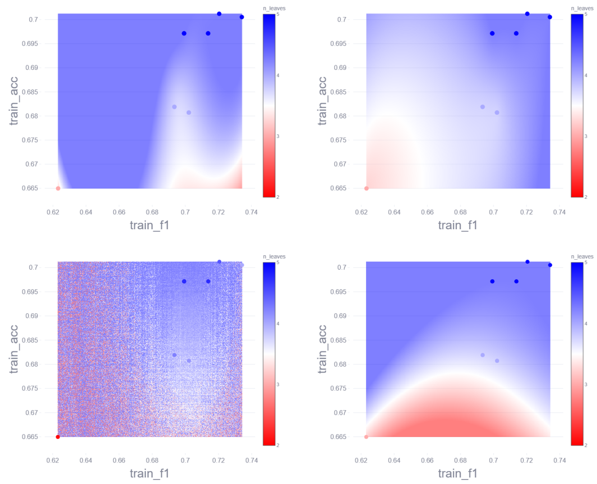

Figure 6: Comparison of different kernel functions used in the RBF-heatmap mode: Cubic (top left), Linear (top right), Quadric (bottom left), and Inverse Quadric (bottom right).

8.3 Group 3: Mixed-Type Kernel Functions

The third group shown in Fig. 7 contains spline kernel, Beckmann kernel, wave kernel, and logarithmic kernel, which combine different mathematical properties. Visualizations show that this group excels in expressing local details, each with its unique ability to capture different patterns in the data. The mathematical formats of the kernel functions in Group 3 are shown below:

Spline Kernel: $k(r) = r\ln(r)$

Beckmann Kernel: $k(r) = \exp\left(-\frac{r^2}{2c^2}\right)$

Wave Kernel: $k(r) = \frac{\sin(r)}{r}$

Logarithmic Kernel: $k(r) = \ln(r+1)$

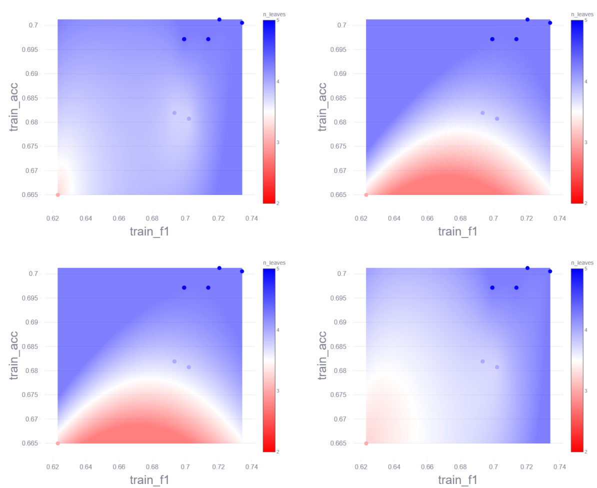

Figure 7: Comparison of different kernel functions used in the RBF-heatmap mode: Spine (top left), Beckmann (top right), Wave (bottom left), and Logarithmic (bottom right).

8.4 Group 4: Custom Kernel Functions

The final group shown in Fig. 8 comprises specially designed kernel functions, including those used in specific research papers and several composite kernel functions. By combining multiple mathematical properties, these kernel functions aim to achieve better performance in specific application scenarios. Heatmaps reveal that these functions can maintain overall smoothness while highlighting local features. The mathematical formats of the kernel functions in Group 4 are shown below:

The paper used kernel: $k(r) = \frac{r \cdot \log(1+r^{0.5})}{1+r^{0.1}}$

Exponential Root Kernel: $k(r) = \frac{\exp(-r)\cdot\sqrt{r+1}}{1+r}$

Sine Logarithmic Kernel: $k(r) = \frac{\sin(r)+\log(1+r)}{1+r^2}$

Hyperbolic Polynomial Kernel: $k(r) = \frac{arctanh(\tanh(r))+r^{1.5}}{1+r^{0.5}+r^3}$

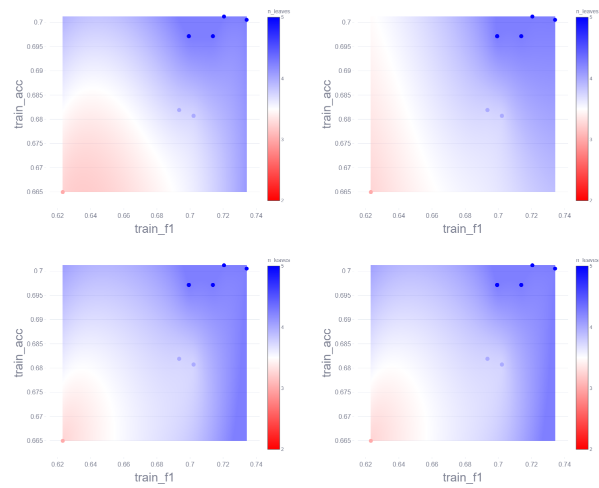

Figure 8: Comparison of different kernel functions used in the RBF-heatmap mode: Paper used (top left), Exponential Root Kernel (top right), Sine Logarithmic Kernel (bottom left), and Hyperbolic Polynomial Kernel (bottom right).

9 High Resolution Images

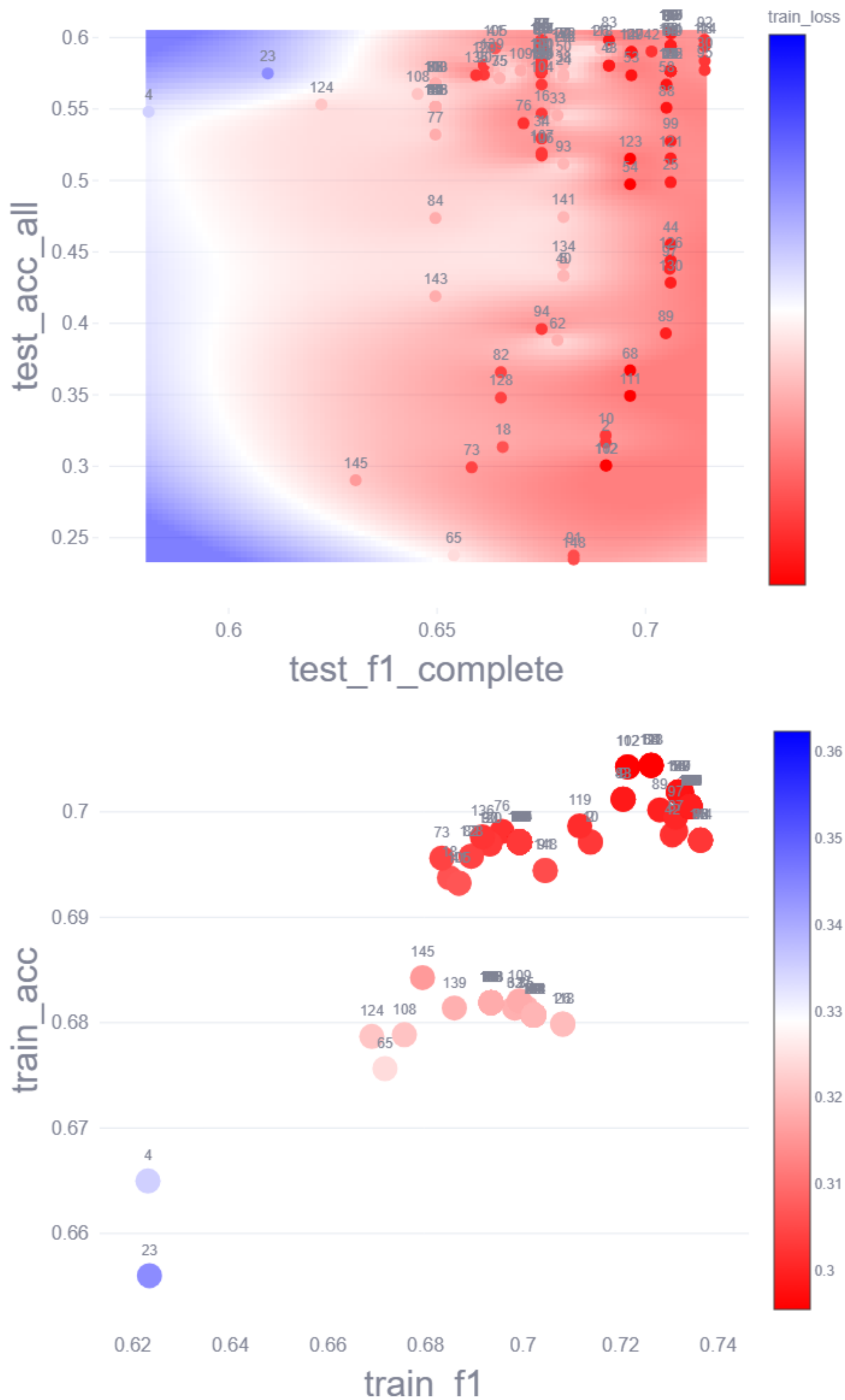

Figure 9: (a) RBF-heatmap mode (Left). Comparison of the performance of 152 models in the Rashomon set on the test set. To ensure that the color of the dots does not completely blend with the background, the color of the dots has been darkened. The color represents the train loss. (b) RBF-dot mode (Right). Comparison of the performance of the same 152 models on the training set. The color represents the train loss.

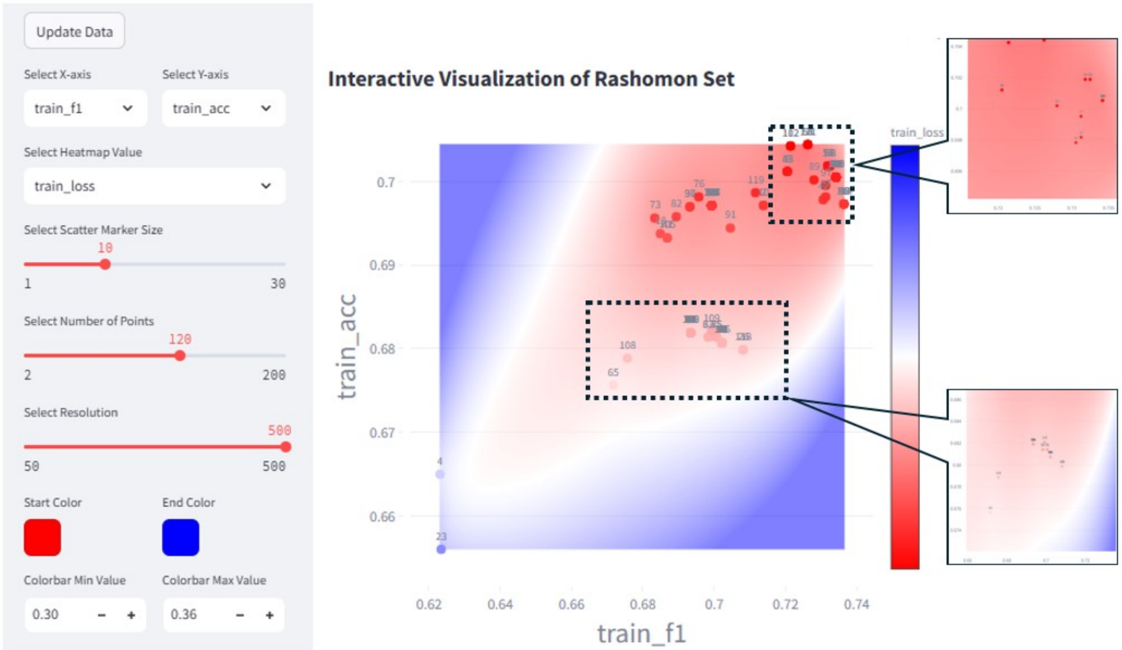

Figure 10: A screenshot of VAR showing functions in the control panel on the left and an RBF visualization plot on the right.

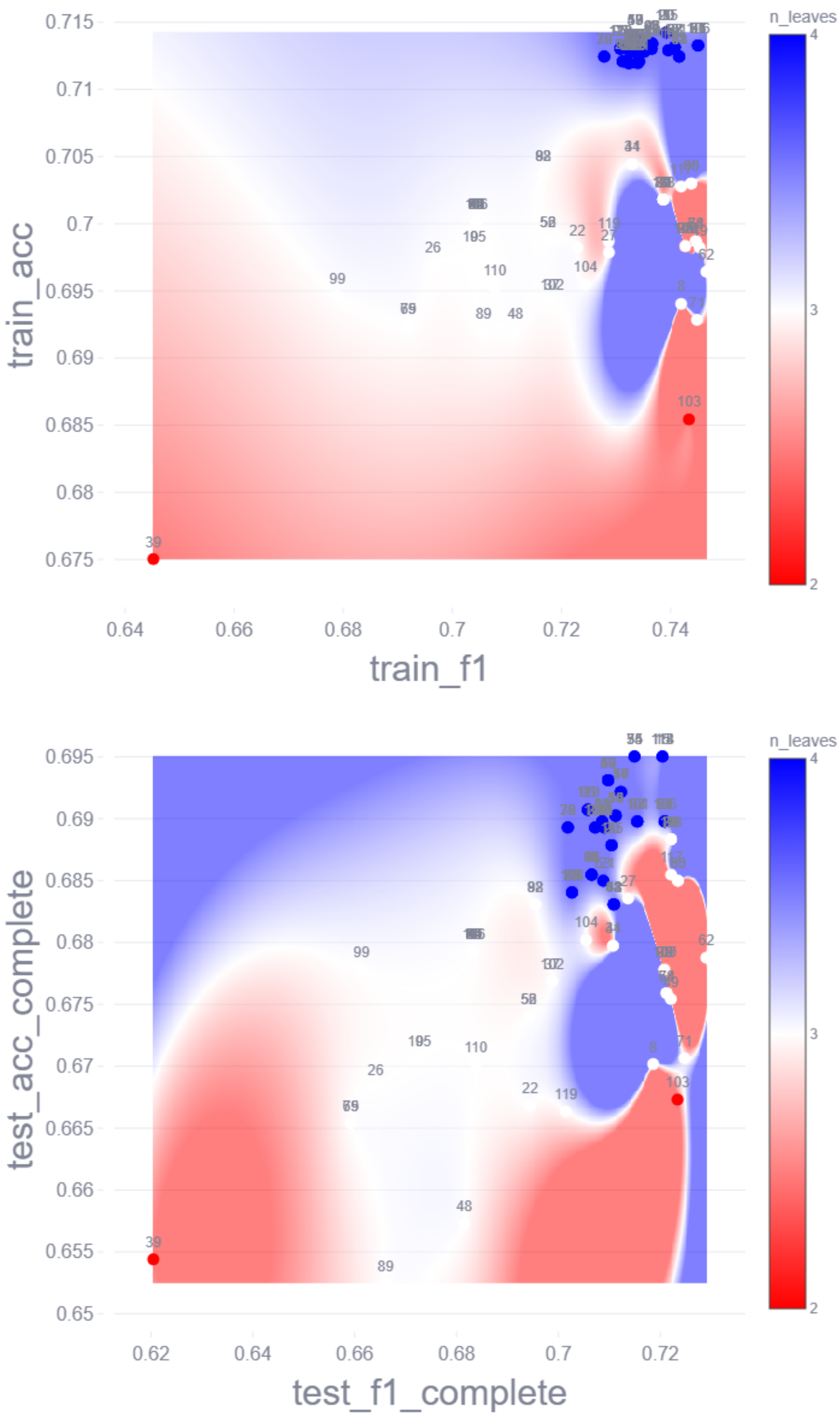

Figure 11: Comparison of the FICO dataset train and test performance. The color represents the number of leaf nodes in a decision tree model.

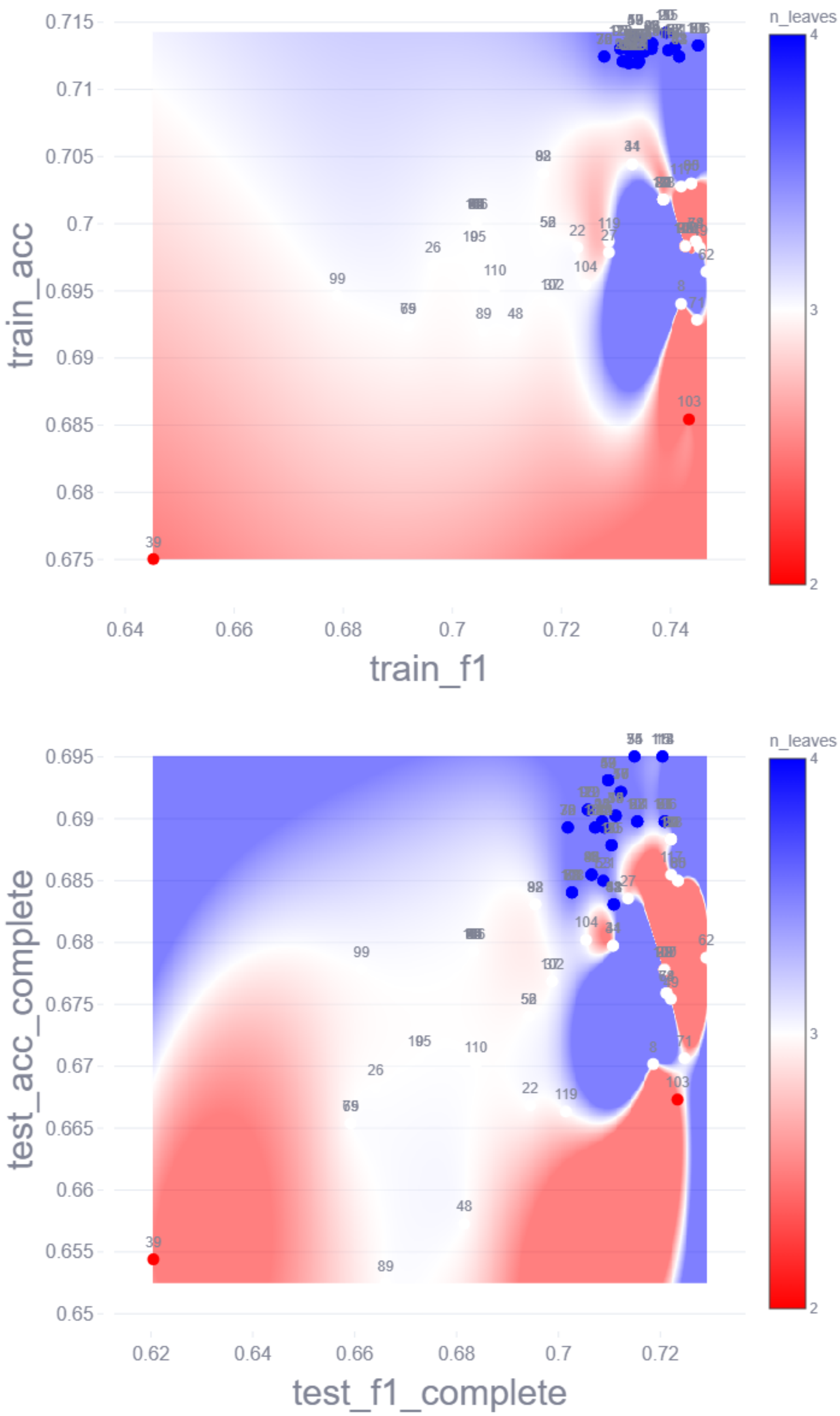

Figure 12: Comparison of the FICO dataset train and test performance. The color represents the number of leaf nodes in a decision tree model. Some models like the label-89 model and the label-48 model are the special models that the ML model developers trying to find.

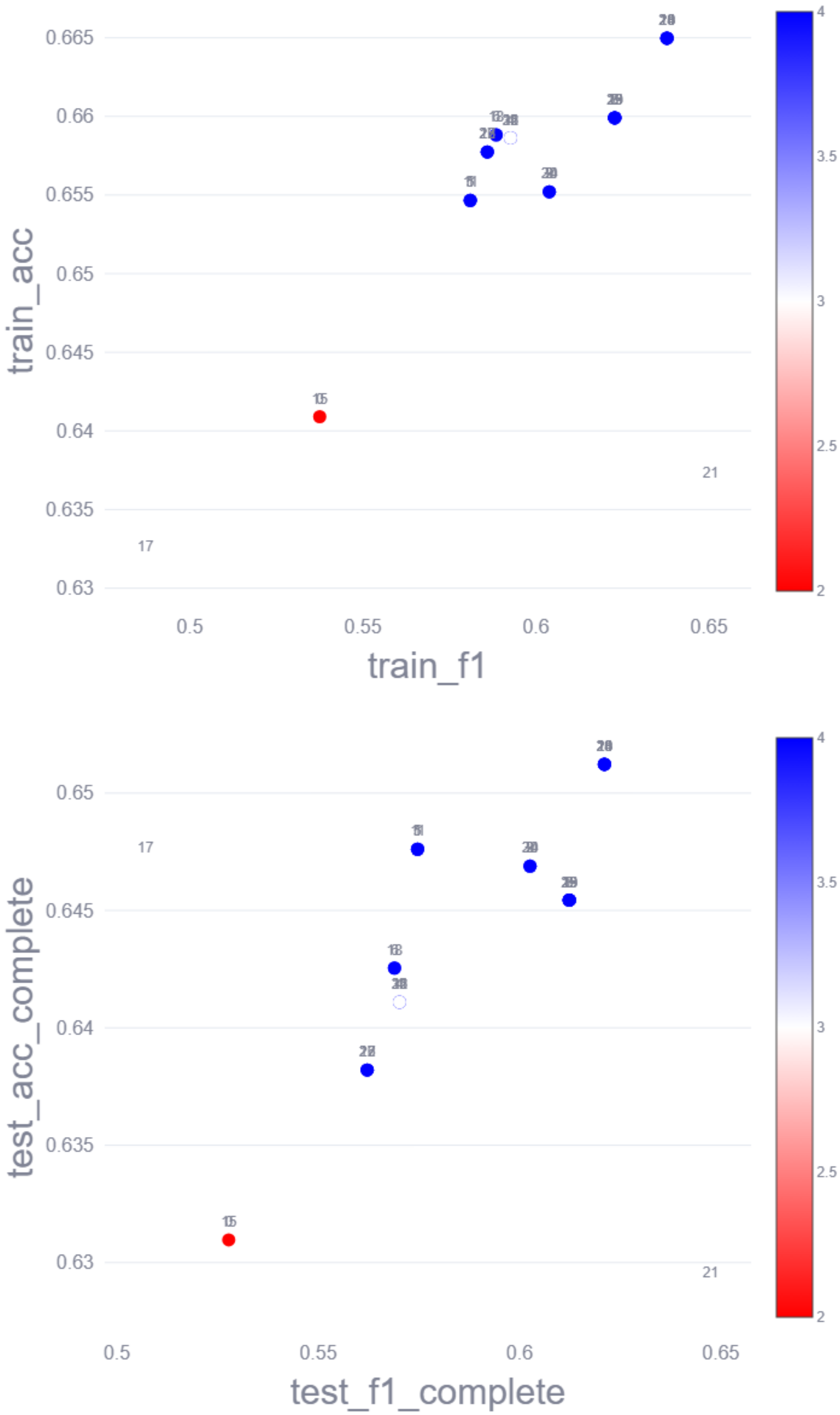

Figure 13: Comparison of the COMPAS dataset train and test performance. The color represents the number of leaf nodes in a decision tree model. The model labeled in 17 is the special model that the ML model developers trying to find.