Analysis of Machine Learning Regression Estimators for Richard's Equation Saturation Curves

MELISSA BUTLER, University of Wyoming, USA

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: differential equations, neural networks, initial value problem, boundary value problem, ODE

ACM Reference Format:
Melissa Butler. 2023. Analysis of Machine Learning Regression Estimators for Richard's Equation Saturation Curves. 1, 1 (April 2023), 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM STATEMENT

Richard's Equation models fluid flow through semi-saturated porous media. It is high non-linear and the resulting saturation curve has extreme slopes, making numerical methods involving derivatives unstable. We would like to see if Machine Learning regression algorithms can capture the extreme curvature of the saturation, given a finite sampling of data.

2 SIGNIFICANCE

The study of fluid flow through partially saturated porous media is critical to agriculture, construction, waste disposal, and other significant fields and is an extremely complex process, described by Richards equation. Richards equation is of great interest due to the lack of closed form solutions and the difficulties in numerical approximations.

3 BACKGROUND

Richard's Equation is represented by

$$\begin{cases} \partial_t \theta(u) - \partial_z \left(\kappa(u)\partial_z(u-z)\right) = 0, \text{ in } (0,L)\times(0,T) \\ u(z,0) = u_0(z),\ z\in(0,L) \\ \kappa(u)\partial_z(u-z)\Big|_{(0,t)} = g_0(t),\ u(L,t)=0. \end{cases} \tag{1}$$

The highly nonlinearity nature is seen in the dependence on the pressure head, u, by the hydraulic conductivity, κ, and saturation, θ. This dependence produces rapid changes in the capillary head around the infiltration front, generating possibly non-differentiable zones [**?**]. This, in turn, creates instability in many numerical approximation techniques. We will explore the effect of the non-differentiable zones of the saturation on machine learning regression estimators.

Author's address: Melissa Butler, University of Wyoming, Laramie, WY, USA.

Manuscript submitted to ACM

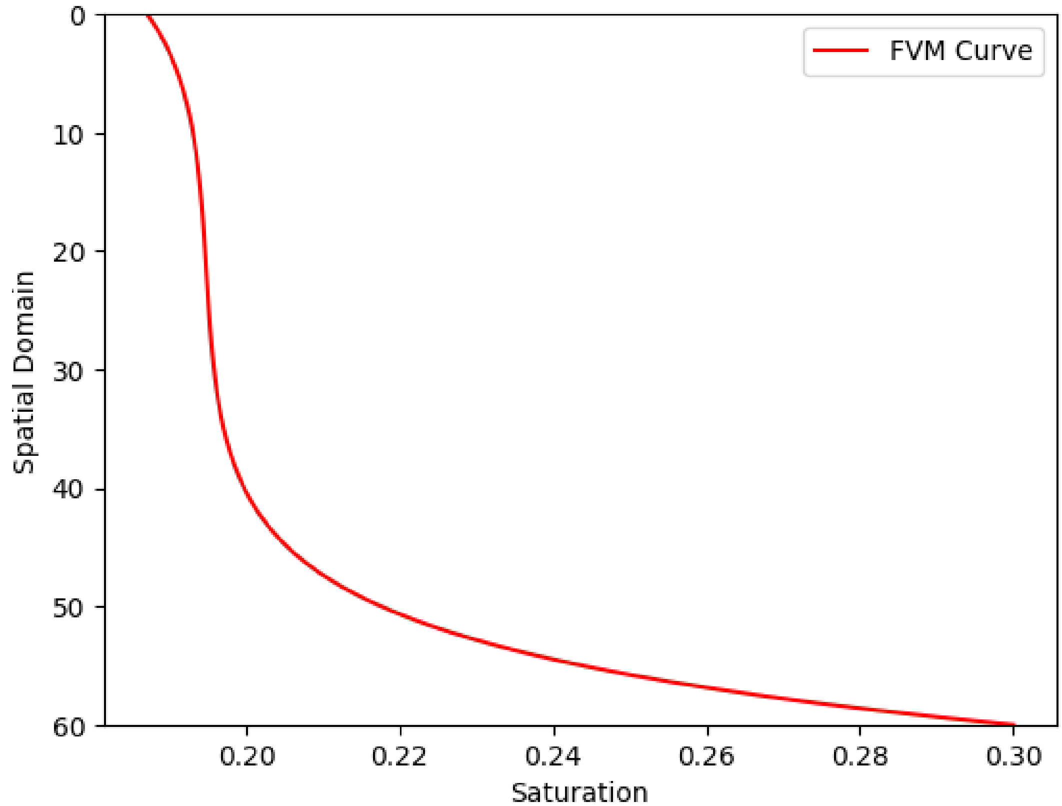

Fig. 1. FVM Curve

By standard convention, we plot the spatial domain on the vertical axis in reverse. This is visual the domain with the surface, $z = 0$, and the top and the water table $z = 60$ at the bottom.

4 DATASET DESCRIPTION

The Finite Volume Method (FVM) is a common numerical approximation tool for PDE's and was used to generate our data points. Our inputs consist of a discretized spatial domain, $z \in [0, 60]$, with $M = 100$ equispaced nodes,

$$\{z_i\}_{i=1}^{M}, \; 0 = z_1 < z_2 < \cdots < z_{M-1} < z_M = 60.$$

The outputs are the approximate values of saturation at each node,

$$\{\theta(u_h(z_i))\}_{i=1}^{M},$$

generated by the FVM. Note that Richard's Equation is also time dependent. Our FVM employed a full-discretization (spatial and temporal) and time marching was used. The extreme curvature occurs in early timesteps, with later times reaching a smoother steady state, so early timesteps were used.

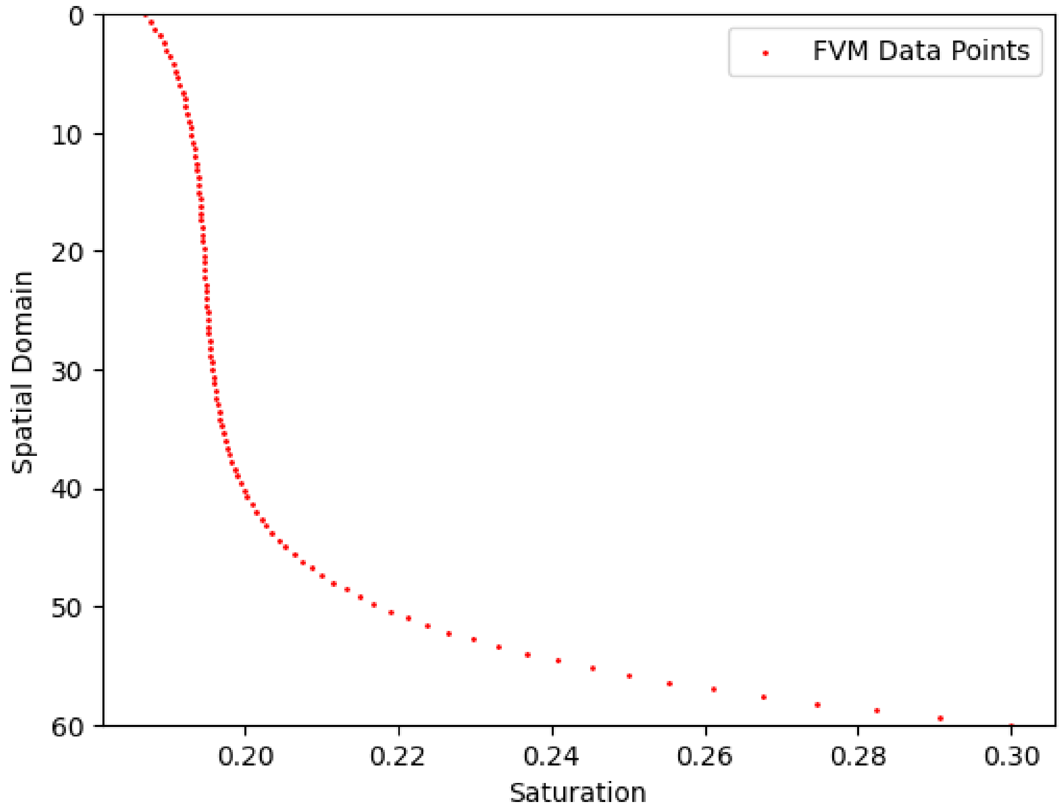

Fig. 2. FVM Generated Data Points

5 METHODOLOGY

To analyze the capabilities of machine learning regression algorithms on the saturation curve, the data was split into various size training and testing sets. The training data was used to fit Polynomial Regression, KNN Regression, Decision Tree Regression, and Random Forest Regression models. Various hyperparameters where also tested, using a grid search. To analyze the accuracy, the Mean Squared Error and Maximum Error were calculated. We first started with two random training splits with 70% training and 30% testing, to analyze a dense set of data. Next, to simulate potential in-field measurements of saturation, we split the data using equispaced grid points with 6 training points and 11 training points.

6 RESULTS

6.1 Random Split 1 - 70% Training Data

In this random split, there were no training data points close to the water table, which resulted in larger errors for the methods normally used for classification.

6.2 Random Split 2 - 70% Training Data

In this random split, we happened to capture data points for the entire domain. This cannot be guaranteed, but also doesn't reflect real world situations, since collecting the saturation level for 70 data points is infeasible. However, it does illustrate the capabilities of the machine learning algorithms to capture high curvature.

6.3 Equispaced - 11 Training Points

For this split we used 11 equally spaced training inputs,

$$\{0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60\}$$

and their corresponding saturation values. The rest of the data was used for testing. This is under the assumption, that a core sample can be taken and the saturation can be measured at various depths. This could then be extrapolated to estimate nearby water content.

6.4 Equispaced - 6 Training Points

For this split we used 6 equally spaced inputs,

$$\{0, 12, 24, 36, 48, 60\},$$

and their corresponding saturation values. The rest of the data was used for testing. This was to observe the accuracy of the model for even sparser data. Reducing the data further resulting in highly inaccurate models.

7 CONCLUSION

TODO - Write Conclusion

8 PRESENTATION

TODO - Record Presention Video Presentation

REFERENCES

[1] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs.* http://github.com/google/jax

https://www.statology.org/sklearn-polynomial-regression/

9 FIGURES

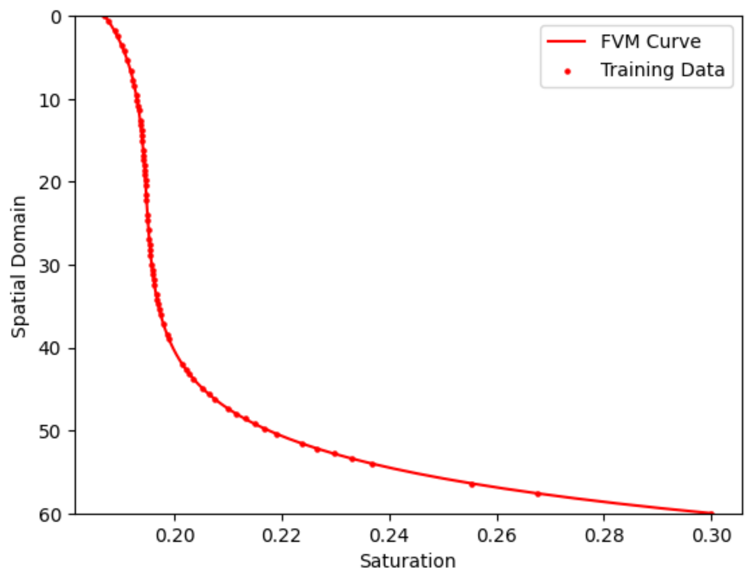

Fig. 3. Random 1 Data

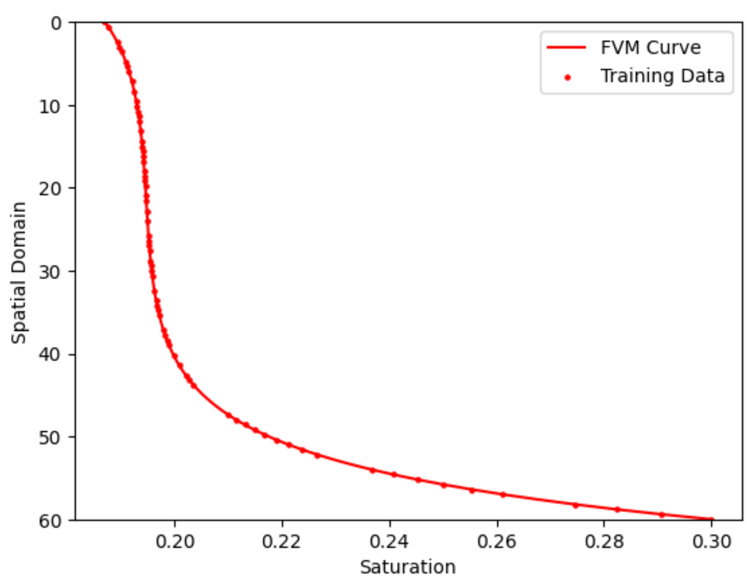

Fig. 4. Random 2 Data

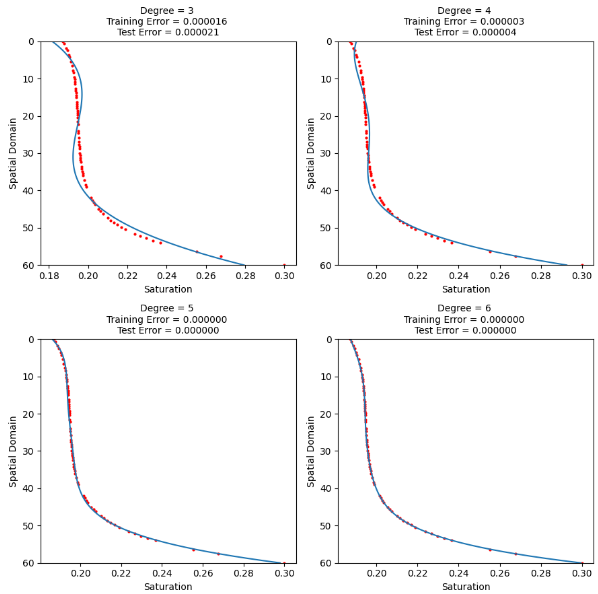

Fig. 5. Random 1 Polynomial Regression

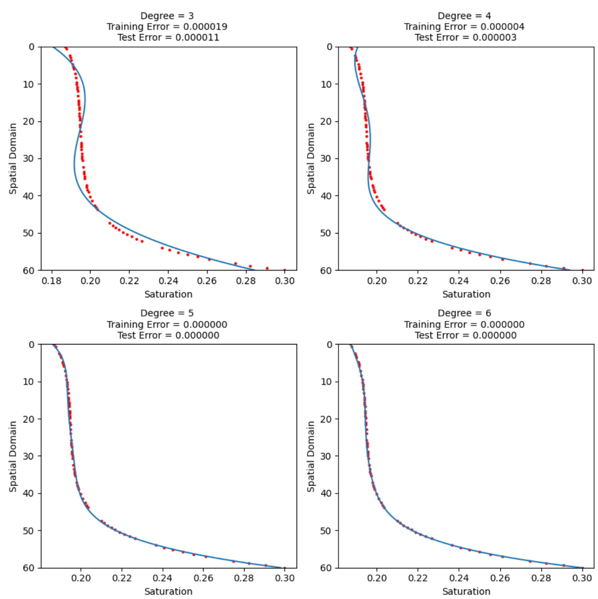

Fig. 6. Random 2 Polynomial Regression

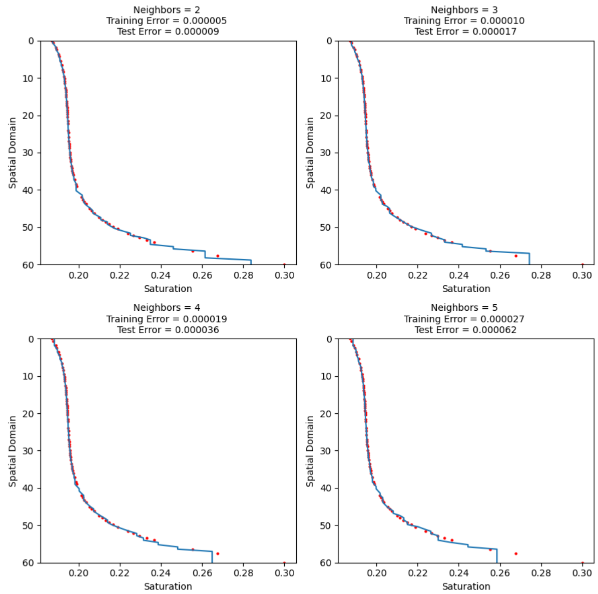

Fig. 7. Random 1 KNN

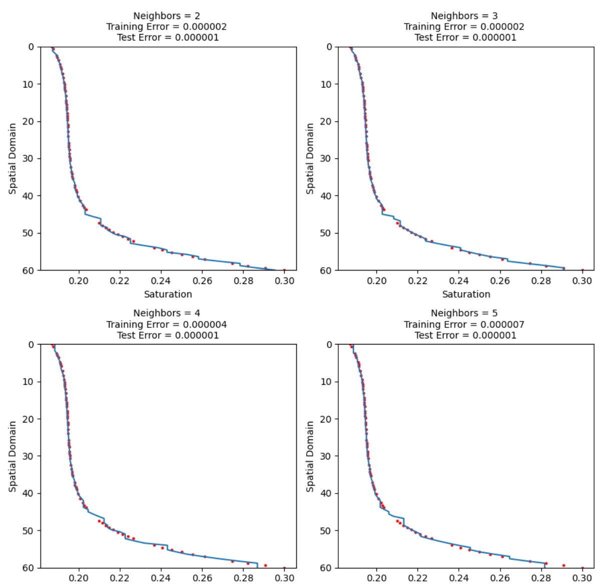

Fig. 8. Random 2 KNN

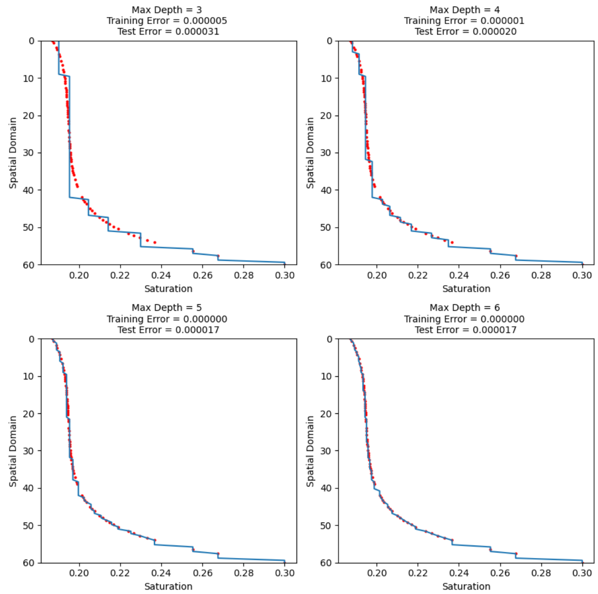

Fig. 9. Random 1 Decision Tree

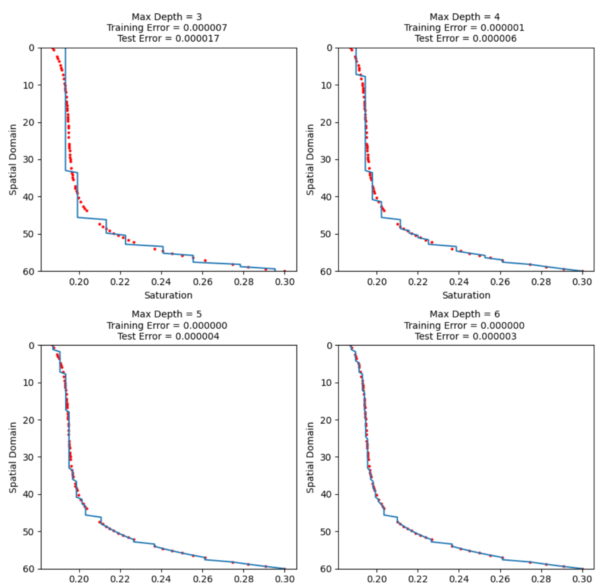

Fig. 10. Random 2 Decision Tree

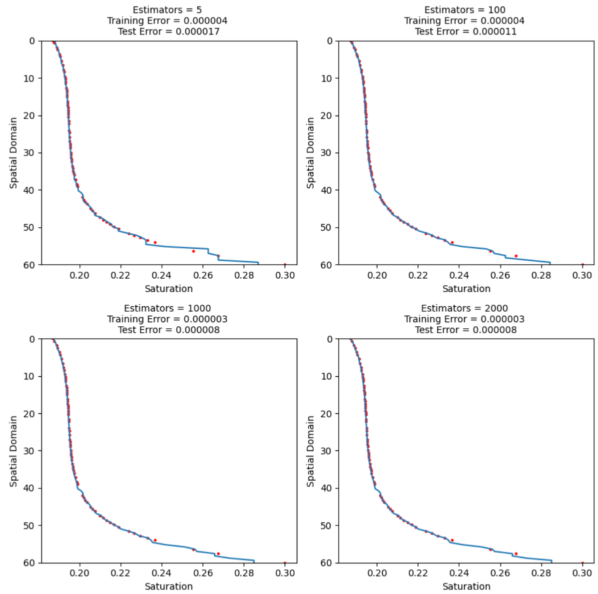

Fig. 11. Random 1 Random Forest

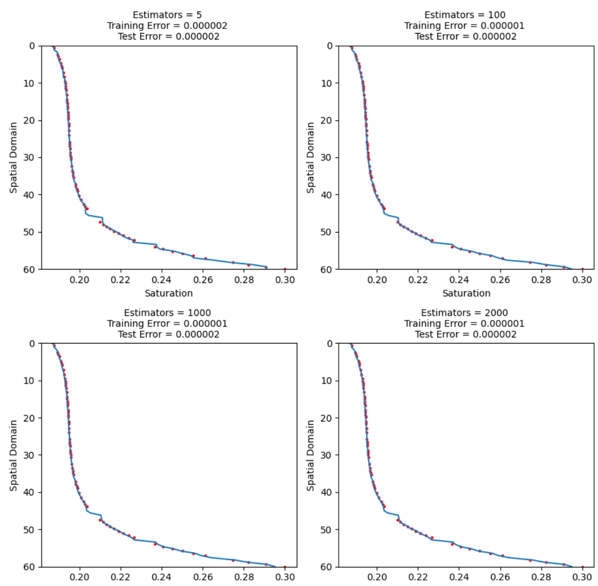

Fig. 12. Random 2 Random Forest

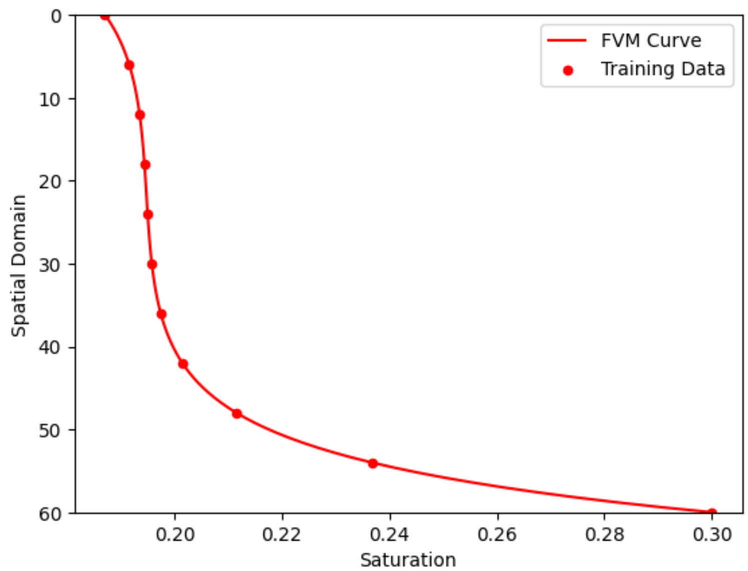

Fig. 13. 11 Equispaced Data

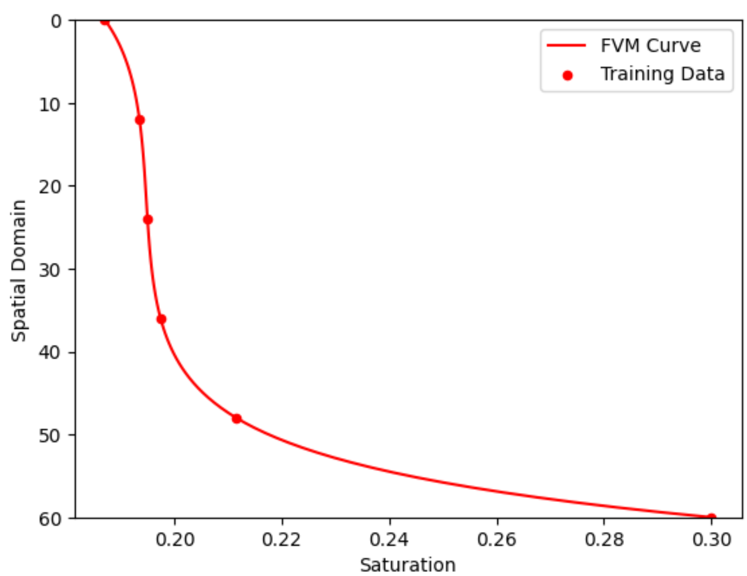

Fig. 14. 6 Equispaced Data

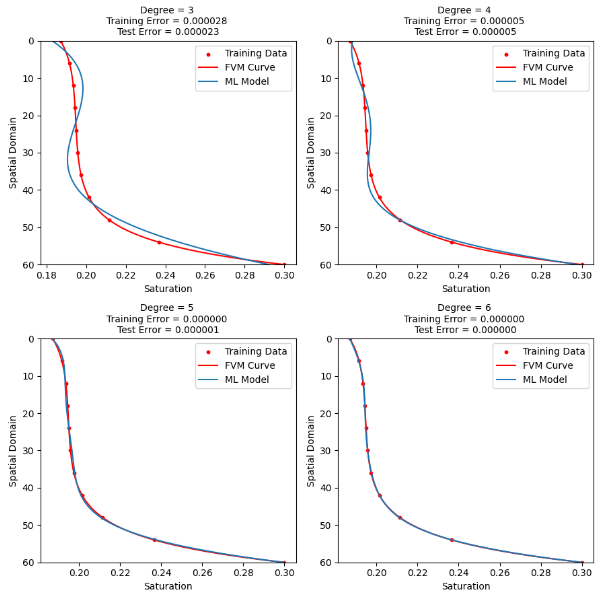

Fig. 15. 11 Equispaced Polynomial Regression

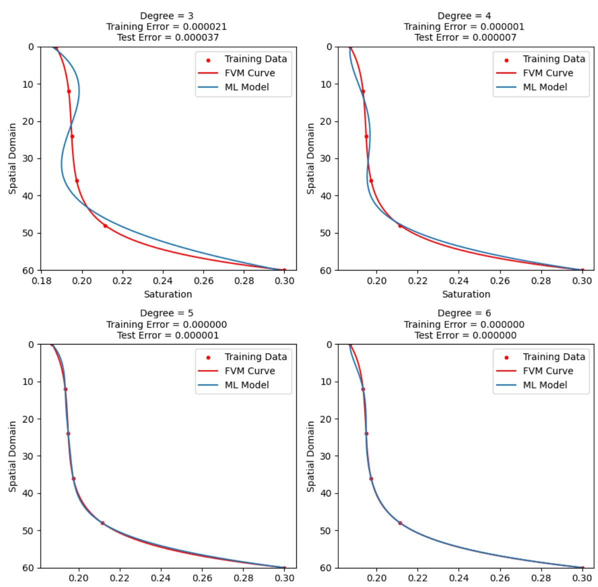

Fig. 16. 6 Equispaced Polynomial Regression

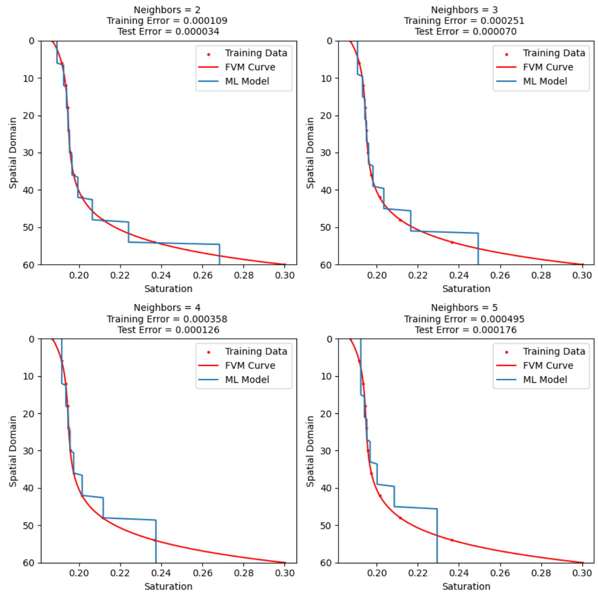

Fig. 17. 11 Equispaced KNN

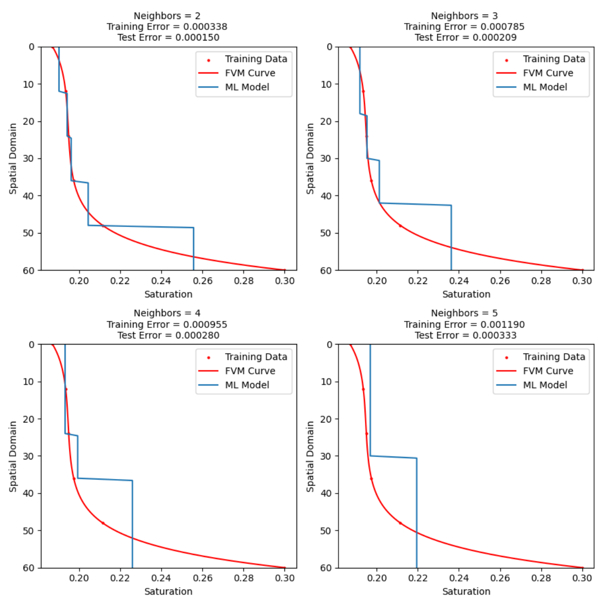

Fig. 18. 6 Equispaced KNN

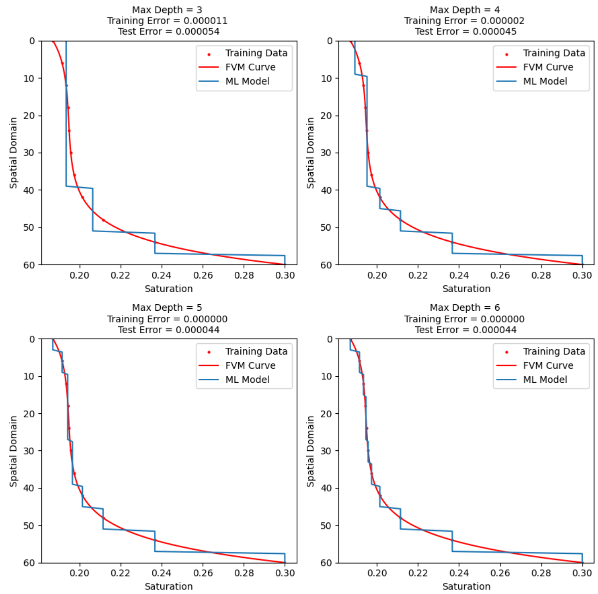

Fig. 19. 11 Equispaced Decision Tree

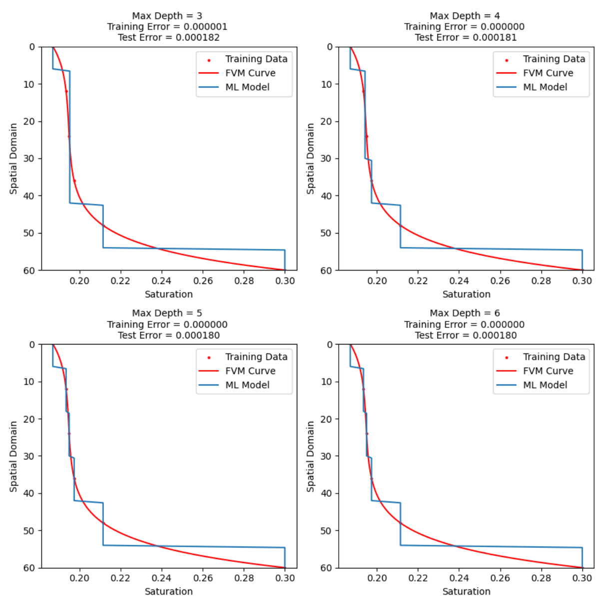

Fig. 20. 6 Equispaced Decision Tree

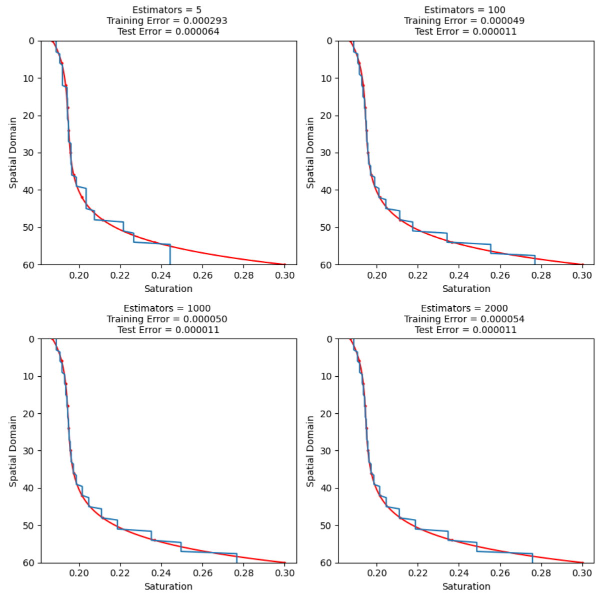

Fig. 21. 11 Equispaced Random Forest

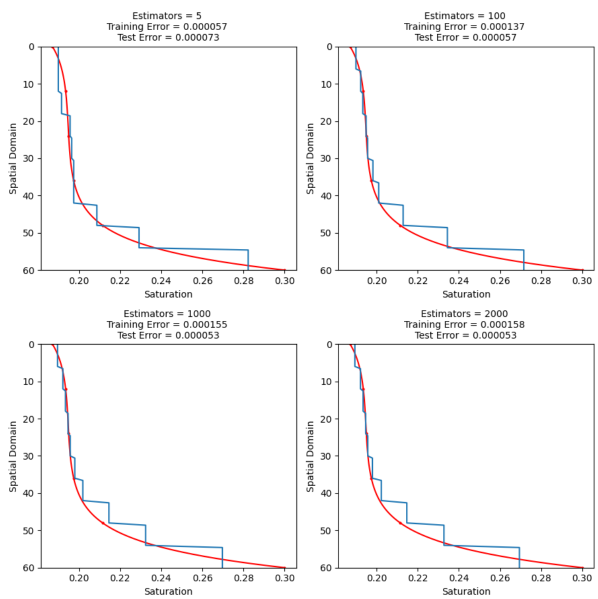

Fig. 22. 6 Equispaced Random Forest

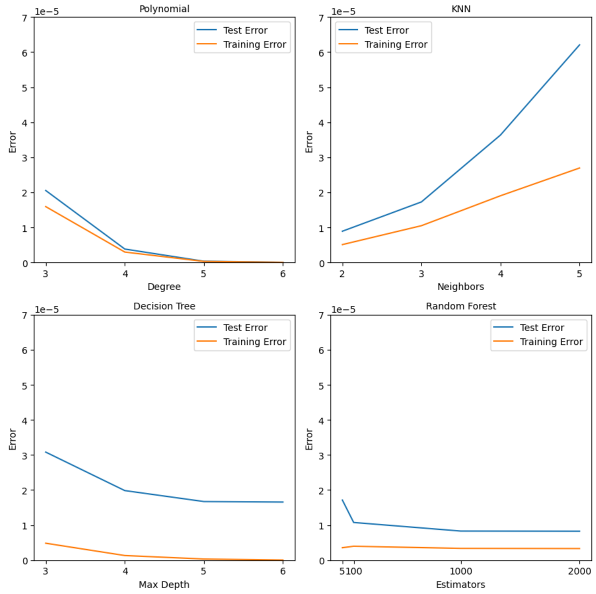

Fig. 23. Random 1 Testing and Training Error

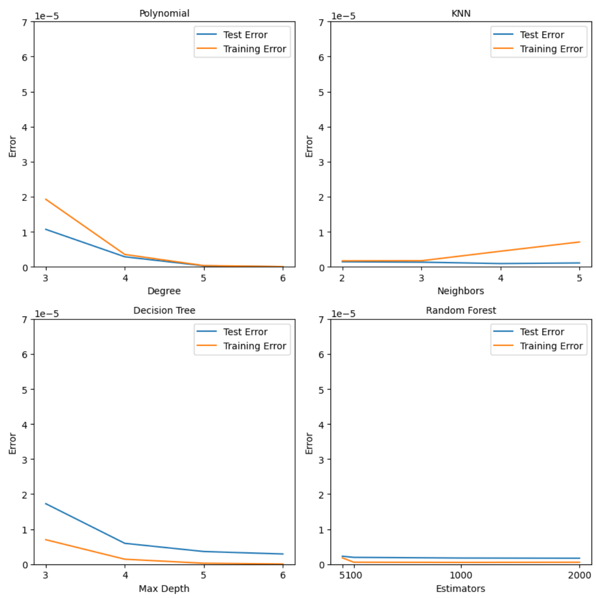

Fig. 24. Random 2 Testing and Training Error

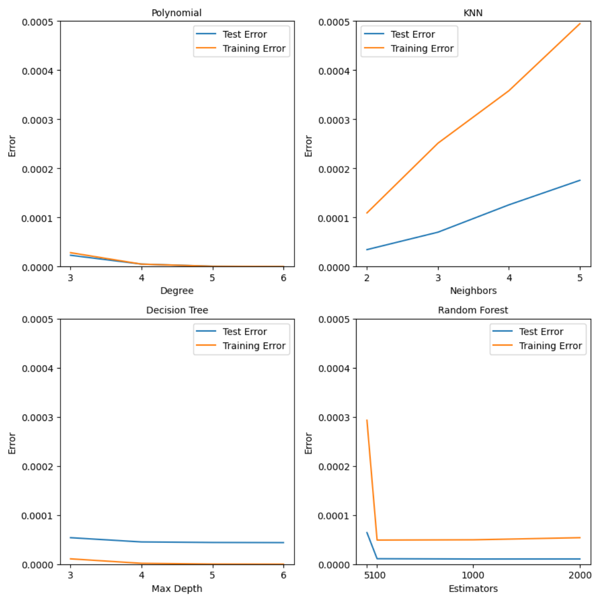

Fig. 25. 11 Equispaced Testing and Training Error

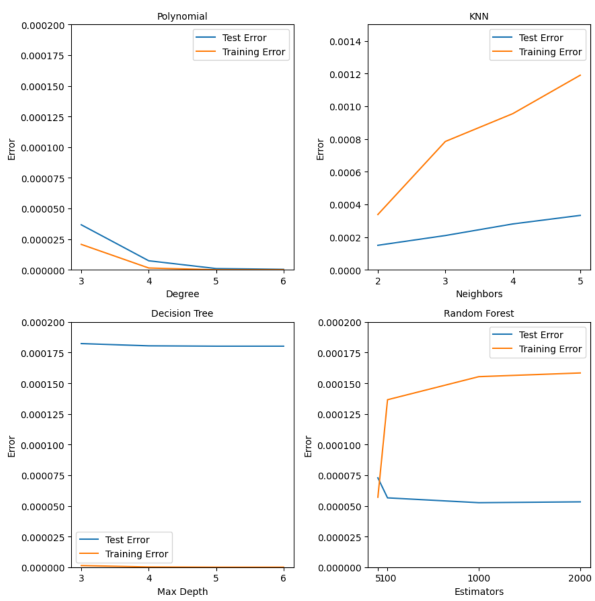

Fig. 26. 6 Equispaced Testing and Training Error

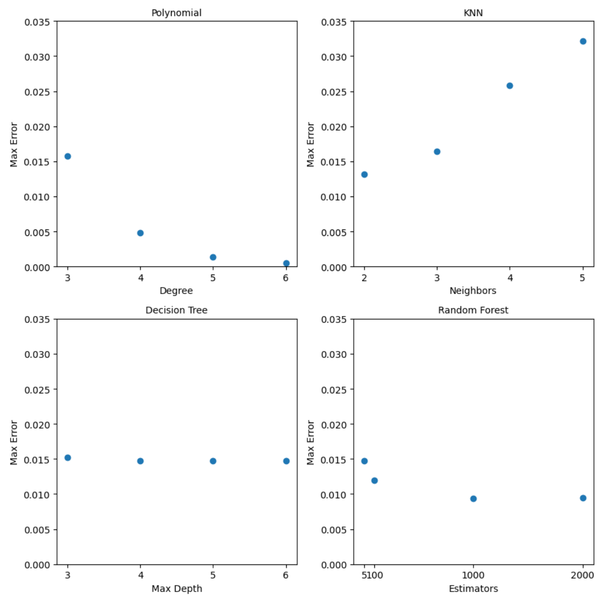

Fig. 27. Random 1 Maximum Error

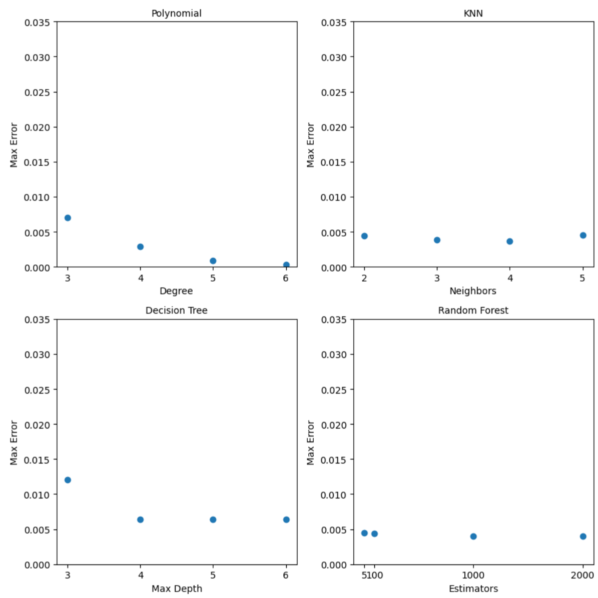

Fig. 28. Random 2 Maximum Error

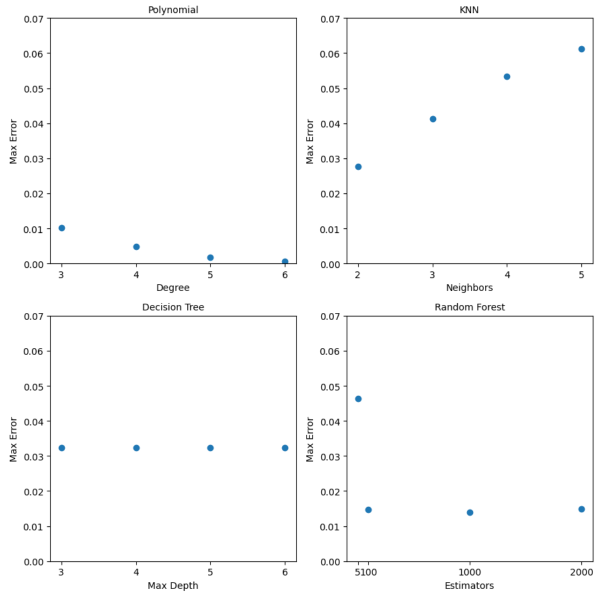

Fig. 29. 11 Equispaced Maximum Error

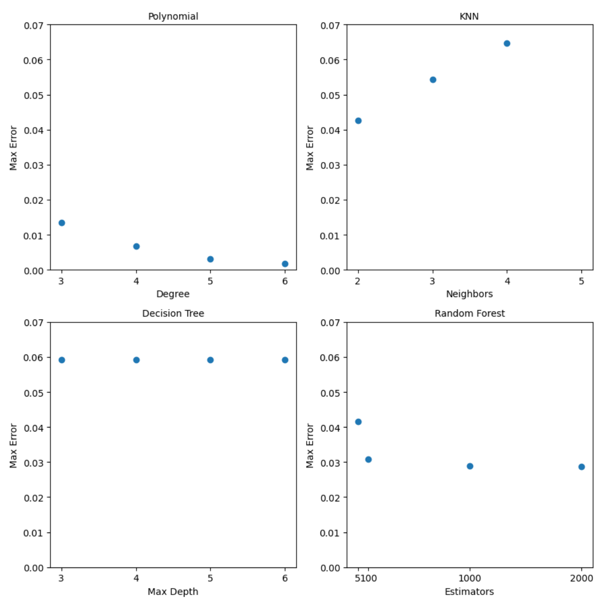

Fig. 30. 6 Equispaced Maximum Error