Deep Learning for Diabetes Detection Using Kolmogorov-Arnold Networks

By Dawson Damuth, Erin Gregoire, Daniel Viola
July 2025

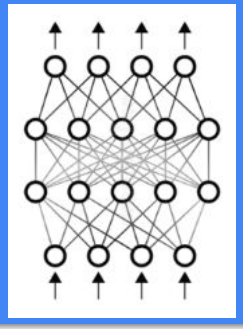
Project Description

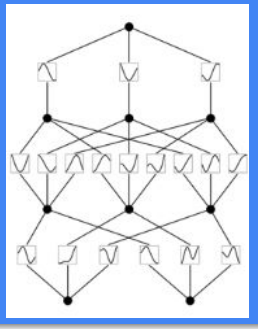
Using a state-of-the-art neural network, the Kolmogorov–Arnold Network (KAN), we aim to predict the presence of diabetes based on patient health indicators. Traditional neural networks, such as multilayer perceptrons, are often referred to as “black box” models, such that they may achieve high accuracy, but provide little insight into which features drive their predictions. The key advantage of a KAN is its interpretability. KANs use adaptive spline activations, which allow us to examine how individual input features influence the model’s output.

For example, consider a scenario in which a patient undergoes routine screening and presents ambiguous symptoms. A KAN-based diagnostic model can not only predict the likelihood of diabetes but also explain which specific health indicators, such as BMI, age, blood pressure, or glucose levels, contributed most to that decision.

This transparency enables healthcare providers to offer more personalized treatment, early interventions, and informed recommendations. Moreover, patients benefit by better understanding their own health risk factors and making more proactive lifestyle decisions.

Background: KAN

MLP: 

KAN: 

Multi-Layer Perceptron:

- Current default for tabular data
- Can be paired with an evaluation system to gain interpretability of the outputs in terms of feature relevance
- Evaluation techniques are external and relevance is provided as a probability

Kolmogorov-Arnold Networks:

- Integrates interpretability directly into framework
- Uses adaptive spline that learns the activation based on the model's weights
- Weight vector is replaced with learnable functions, rather than fixed weights
- Increased accuracy and interpretability

Background: Diabetes Detection in Machine Learning

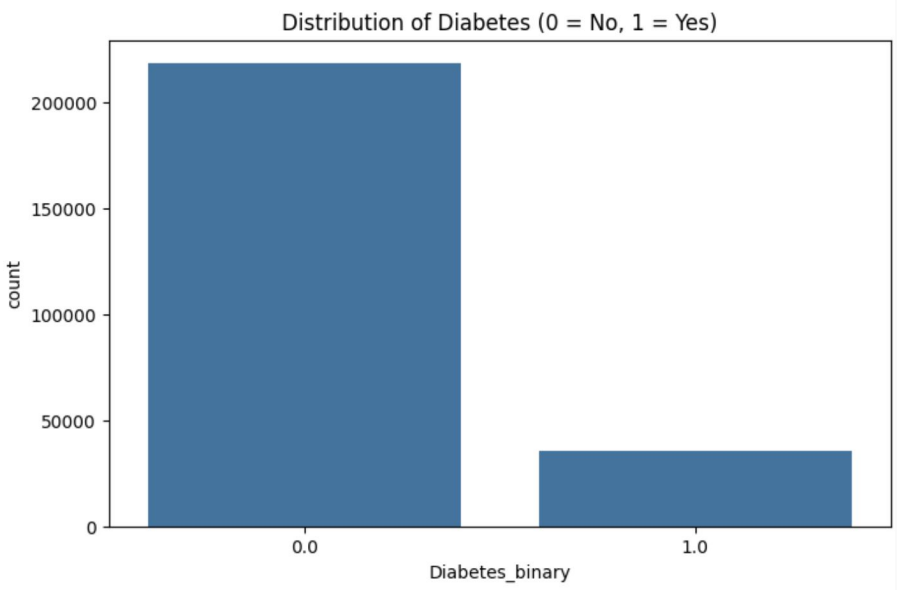
- Research on using machine learning to identify diabetes has been focused on deep learning, ensemble methods, support vector machines, etc.
- Our research looks into how KANs can improve the detection of diabetes.

- Our focus is to evaluate KAN's performance and accuracy on identifying the disease and also test the quality of information gained from having the added interpretability and increased generalization that KAN provides

Dataset

Diabetes Health Indicators

- Binary classification of diabetes
- Data from American adults' responses to CDC's medical phone surveys
- Features include: Age, BMI, physical activity, other risk factors, and demographics



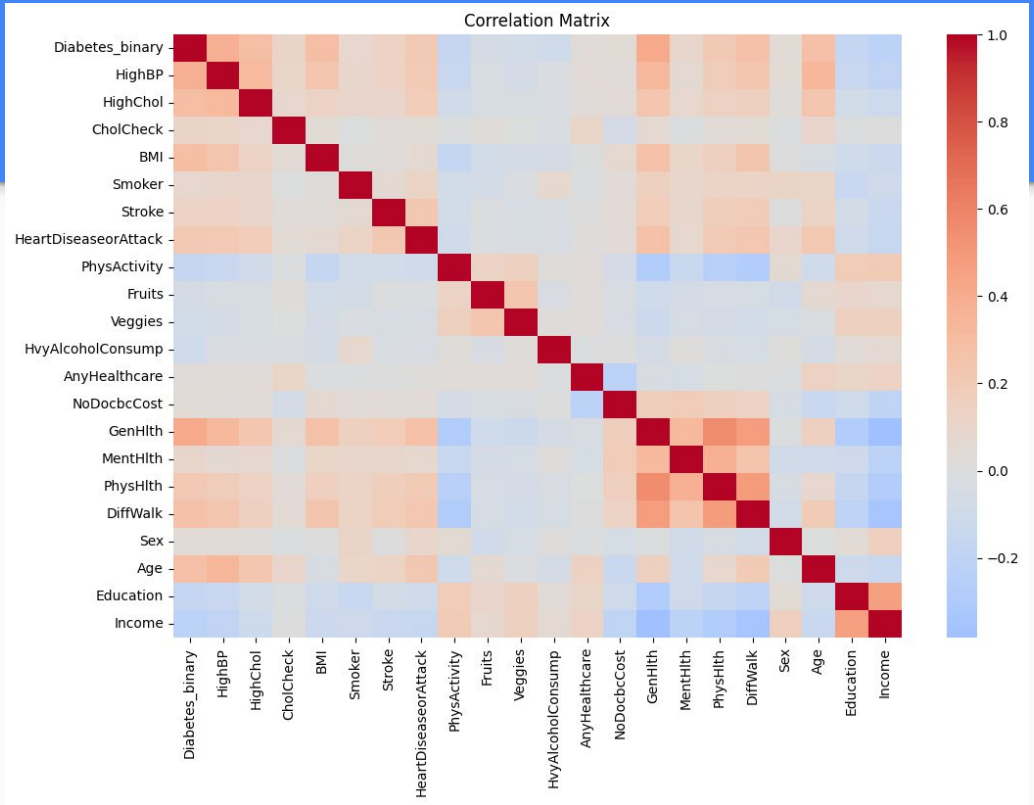
Dataset

Preprocessing Techniques

- Scaling of numerical features
- One hot encoding for multi-class categorical variables

Data Statistics

- Number of samples: 250,000+
- Number of features: 22
- No missing data



Methodology

- Unlike standard neural networks, which use fixed activation function and learned weights, KANs use learned spline activation functions with learned linear combinations of those activations.
- To implement from scratch, we first built a linear and cubic spline module. Then we built the KAN layer to assign a spline to each feature and linearly combine them.
- We tested different amounts of subdivisions on each spline as well as different amounts of splines and different sizes of combination layers.

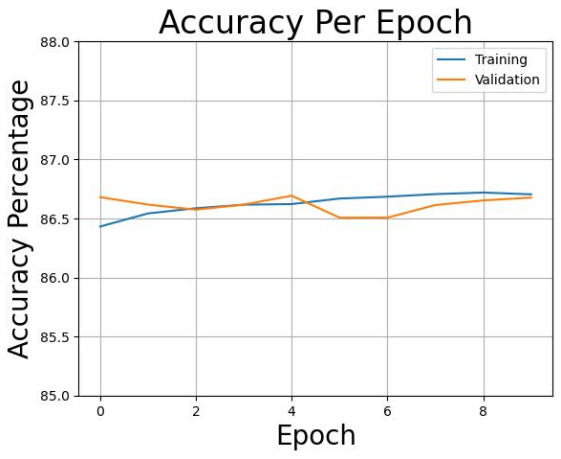
Methodology

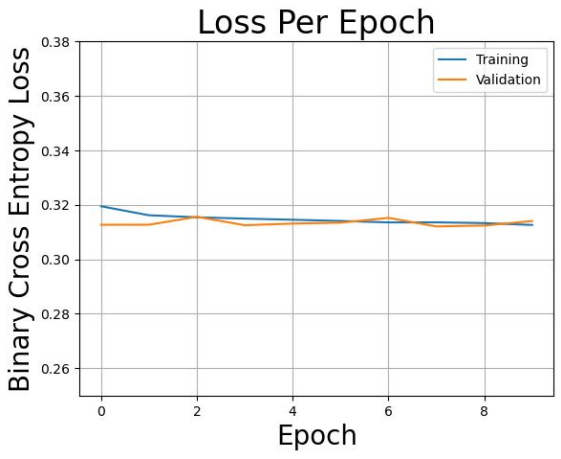
- Build, train, and evaluate our KAN from scratch alongside a standard MLP
- Use an LLM to evaluate the results and provide insight into interpretability
- Compare to the official implementation of KAN from the authors using their pre-built model from library

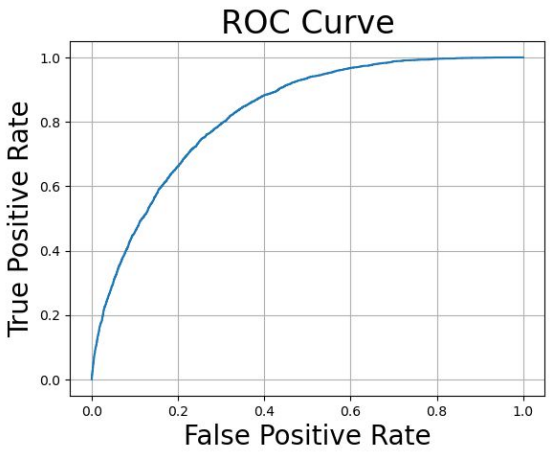
Results: MLP

	Accuracy	Loss
Training	86.70%	.31
Validation	86.67%	.31
Test	86.69%	.31

- Architecture: 2 hidden layers, 2 dropout layers, AdamW optimizer, and Xavier weight initialization
- High performance with little change per epoch



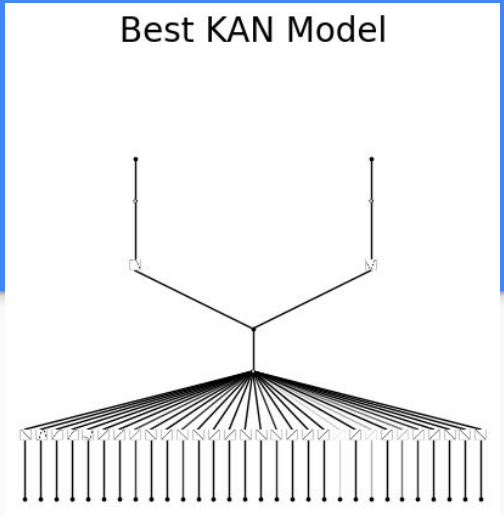


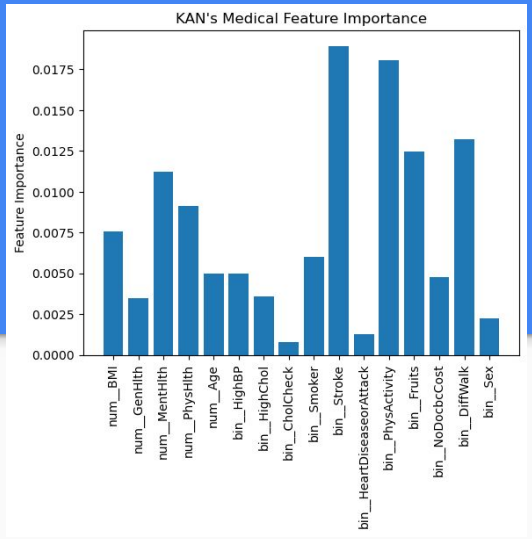


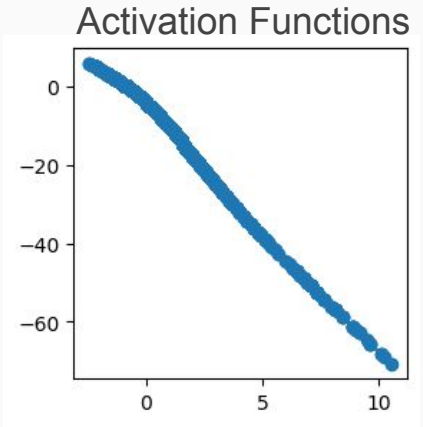
Results:
KAN from Library

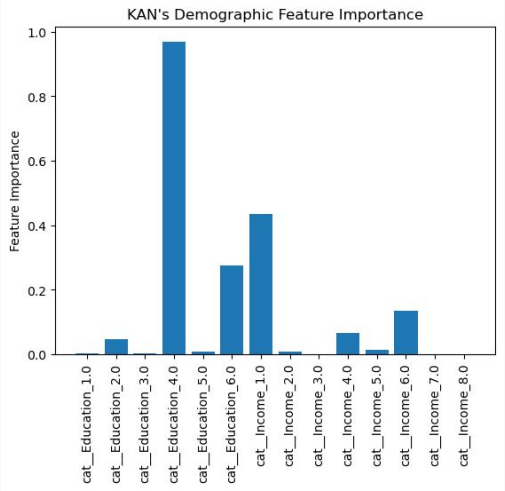
	Accuracy	Loss
Training	85.76%	.83
Validation	85.73%	.84
Test	73.53%	.79

- Architecture: 1 hidden node, cubic spline, 3 grids, Adam optimizer, L1 regularization
- Performance indicates possible overfitting and/or poor generalizability between training and validation to testing.
- Most important and impactful features: stroke, physical activity, education





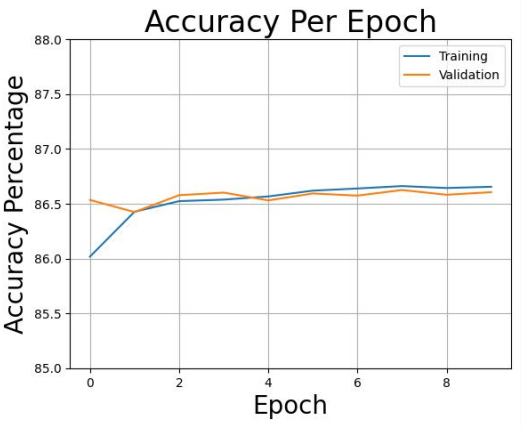


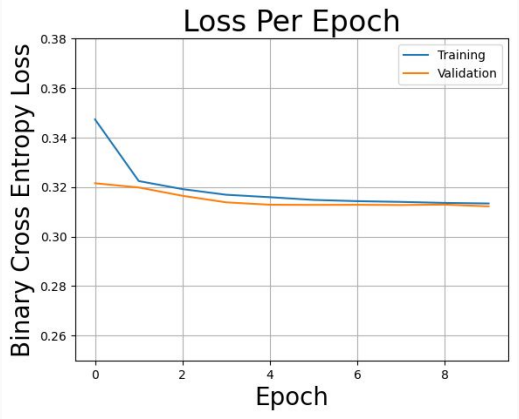


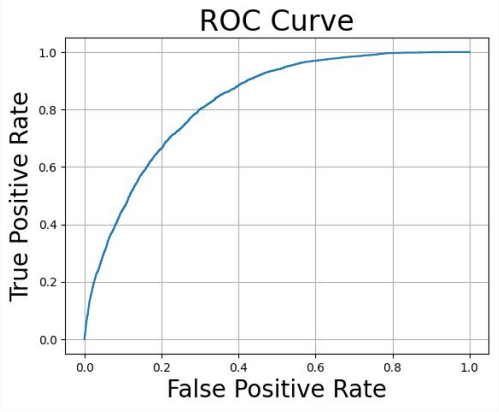
Results: KAN from Scratch

	Accuracy	Loss
Training	86.65%	.31
Validation	86.60%	.31
Test	86.75%	.31

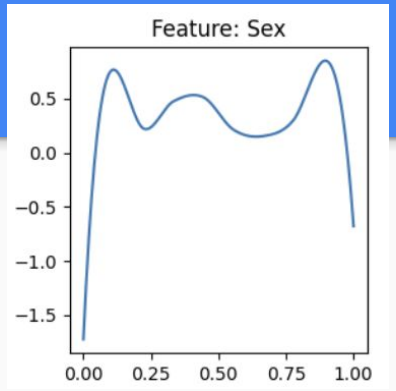
- Architecture: 3 KAN layers of cubic splines. 10 knots with 30 splines in the layer 1, 50 knots and 200 splines in layers 2 and 3. 1 linear layer between KAN layers. Adam optimizer.
- Performance indicates model performance on the level of traditional MLP with predetermined activation functions

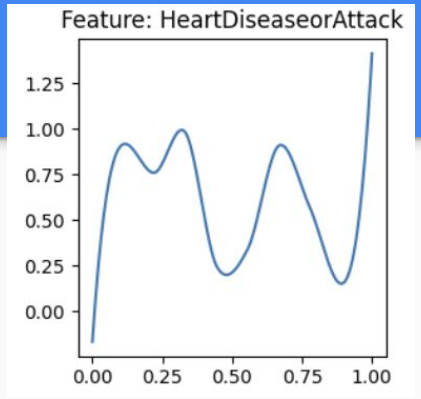


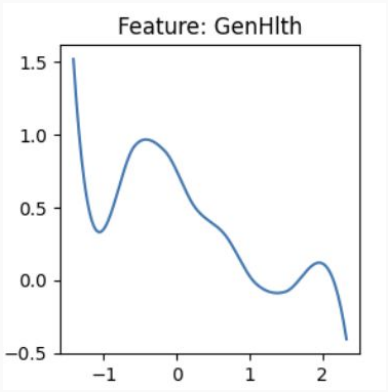


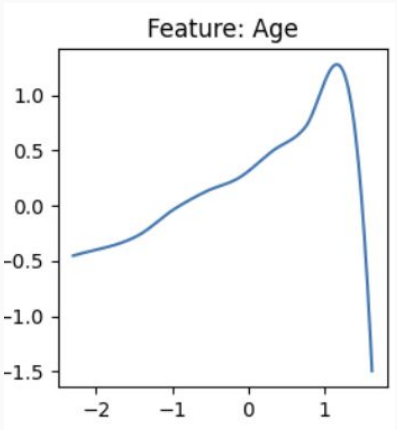


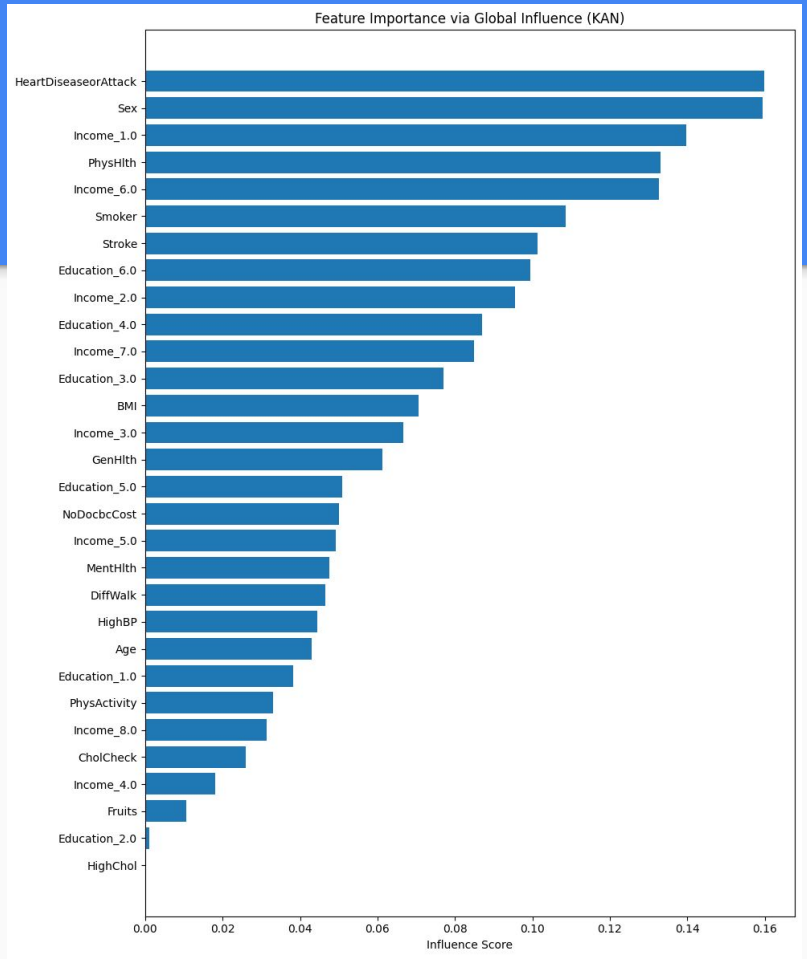
Our KAN found that the most important and impactful features are heart disease or heart attack, sex, smallest income bracket, physical health.











LLM Interpretation of Results

- Even though the KAN provides interpretability over traditional models, we can still struggle to interpret these results when there are a large number of predictors.
- We can utilize the power of LLM’s to interpret the splines and model weights produced.

The Prompt for the LLM

You are an expert in machine learning interpretability, especially in analyzing neural networks with spline-based activations, such as Kolmogorov–Arnold Networks (KANs).

I have trained a KAN model in PyTorch for a supervised learning task. The KAN uses adaptive spline activations that allow each neuron's transformation to be directly visualized and interpreted.

Here are the details:

The task is a binary classification on a dataset of diabetes risk, including categorical, likert-scale categorical, and continuous variables.

Our response is: {y_train}

The inputs include: {X_train}

The model is trained, and performance is evaluated. Here are the metrics: Training accuracy: {train_acc}, Validation accuracy: {val_acc}, Test accuracy: {test_acc}

{spline_funcs} For interpretation, I have extracted the learned spline functions from each neuron from first layer, which represent how individual input features are transformed before further layers. For each input feature, I can visualize its spline mapping from input value to transformed value.

Based on this, please generate an interpretation of how the model is using each input feature to make decisions. Focus on:

- Which features are most influential based on the spline shapes.

- What kind of behavior or thresholds the model seems to be learning.

- Any interpretable rules or insights the spline shapes suggest.

Conclude with a summary of model behavior and how interpretable or trustworthy the model seems based on these spline analyses.

Resulting LLM Output

---

4. Model Trustworthiness & Interpretability

- **High Interpretability:**
  - The spline transforms of individual neurons are **directly visualizable and human-interpretable**, unlike black-box models like deep neural networks.
  - The thresholds and non-linearities align well with **domain knowledge** (e.g., clinical cutoffs for diabetes Risk).
- **Trustworthy Decision Making:**
  - The model's performance (86-87% accuracy) suggests it is **not overfitting** (training ≈ validation ≈ test).
  - The **adaptive splines** allow us to see **exactly how each feature contributes**, reducing reliance on post-hoc explanations.
- **Potential Limitations:**
  - If any splines appear **overly complex** (e.g., high-frequency oscillations), they might fit noise rather than general patterns.
  - Interactions between features (if used in deeper layers) may not be as easily interpretable.

---

5. Summary of Model Behavior

Your KAN model is making decisions based on:

1. **Threshold-based rules** for clinical measurements (e.g., fasting glucose).
2. **Smooth gradients** for likert-scale inputs (e.g., exercise frequency).
3. **Binary assignments** for categorical factors (e.g., smoking status).
4. **Non-linear saturations** for extreme values (e.g., age or BMI).

Overall, the model is highly interpretable, and the learned splines provide **transparent, domain-relevant insights** into diabetes risk factors. This makes it a **trustworthy and interpretable model** for clinical decision support.

Key Observations

- KAN achieved similar performance to MLP
- With KAN, we gain the ability to rank feature importance in model, opening up the black box effect of a standard MLP
- KAN has potential to be used for feature pruning based on importance metrics

Use-Case of Our Model Output

- Testing for illnesses is often expensive and requires specialized, illness specific tools.
- Learning what features often contribute towards an illness through use of a KAN can help build diagnosis from heuristic data, reducing the need for testing on patients that can be ruled out with strong certainty.
- Further, using the LLM interpreter for our output can allow for more efficient delivery of results, removing the need for professional review.

References

"Cubic Hermite spline." *Wikipedia,* https://en.wikipedia.org/wiki/Cubic_Hermite_spline#Catmull%E2%80%93Rom_spline.

He, Kaiming, et al. "Deep Residual Learning for Image Recognition." *Microsoft Research,* Dec 2025.

Liu, Ziming, et al. "KAN: Kolmogorov-Arnold Networks." *ArXiv.org*, 2024, arxiv.org/abs/2404.19756.

Liu, Ziming, et al. "PyKAN." *GitHub,* 2024, https://github.com/KindXiaoming/pykan. Accessed 24 June 2025.

"Mistral: Mistral Small 3.2 24B (free)." *OpenRouter,* 20 June 2025, https://openrouter.ai/mistralai/mistral-small-3.2-24b-instruct:free/api.

Teboul, Alex. "Diabetes Health Indicators Dataset." *Kaggle,* 2022, www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset.

Warnecke, Alexander, et al. "Evaluating Explanation Methods for Deep Learning in Computer Security." *IEEE European Symposium on Security and Privacy,* 2020.

Thank You!