Explaining predictions using Neural Additive Models

Fairness in Machine learning –
Final Project Presentation

by

Sai Pradeep Peri – sap187@pitt.edu

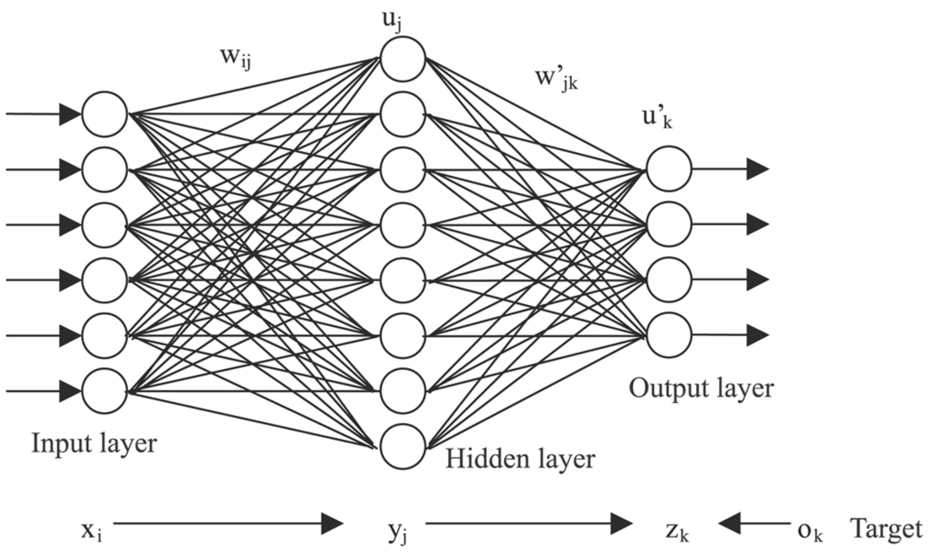
Kinori Rosnow – ksr43@pitt.edu

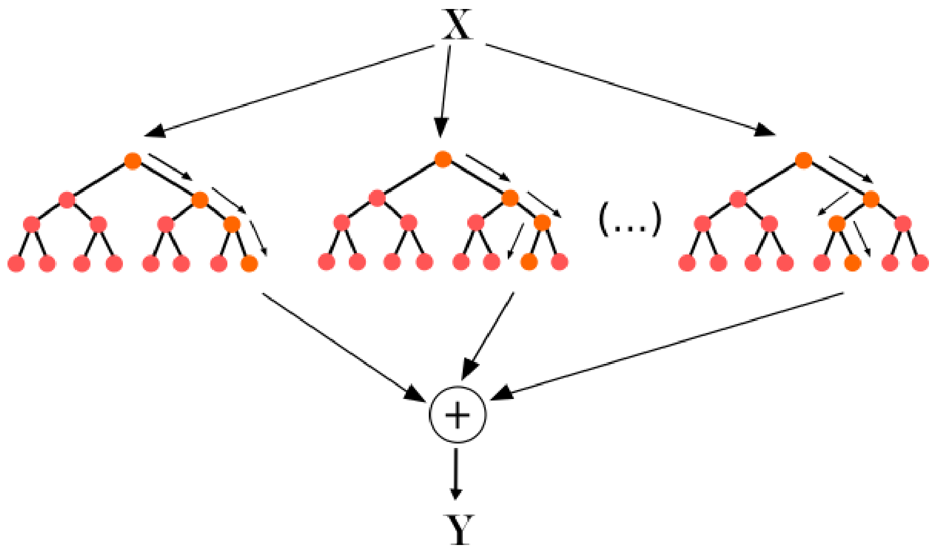
Motivation

- Many models are high performance, but not interpretable
- Others may be low performance, but interpretable
- Goal: Combine high performance models with interpretable design

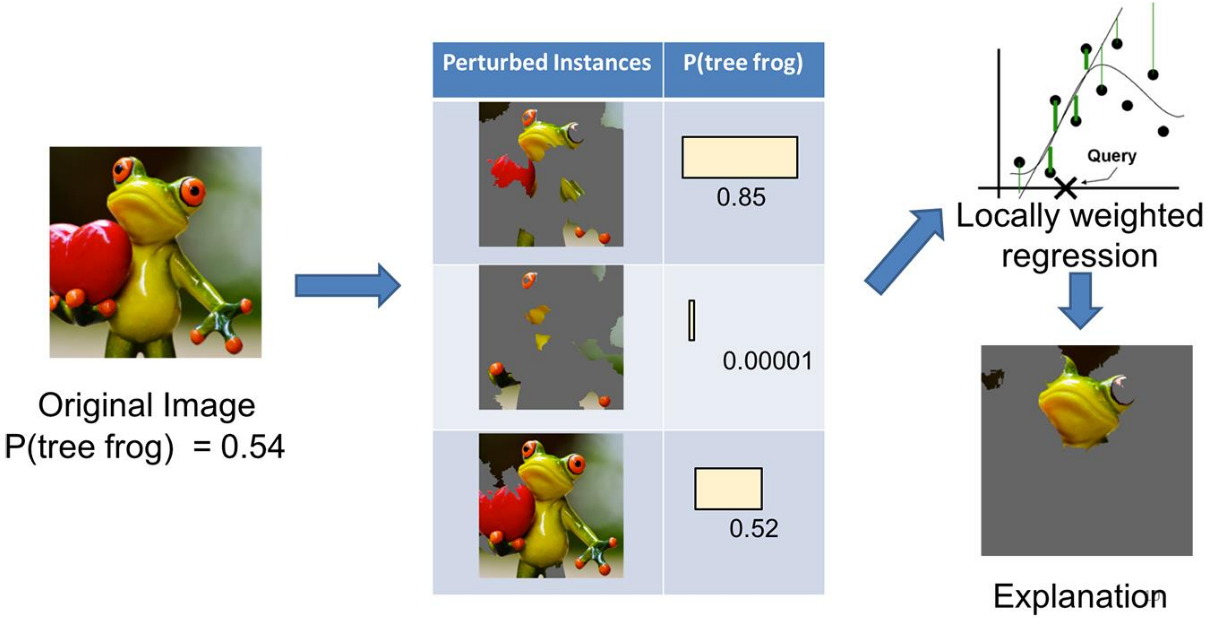
High Performance Models

- Added complexity for more flexibility
- Deep neural networks
- Random Forests





Explaining Black Boxes:



Explaining a prediction with LIME. Sources: Marco Tulio Ribeiro, Pixabay.

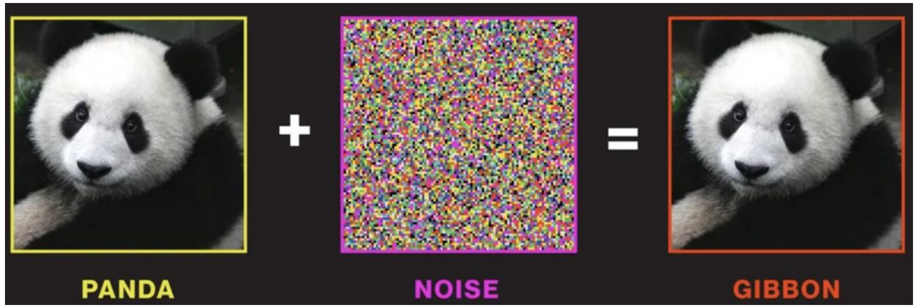
- Past Research based on Surrogate Models.
  - LIME, SHAP

Problems and Limitations:

- Problem – Approximate with linear models, does not actually explain.
- Can be easily fooled – below paper

Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods - https://arxiv.org/pdf/1911.02508.pdf

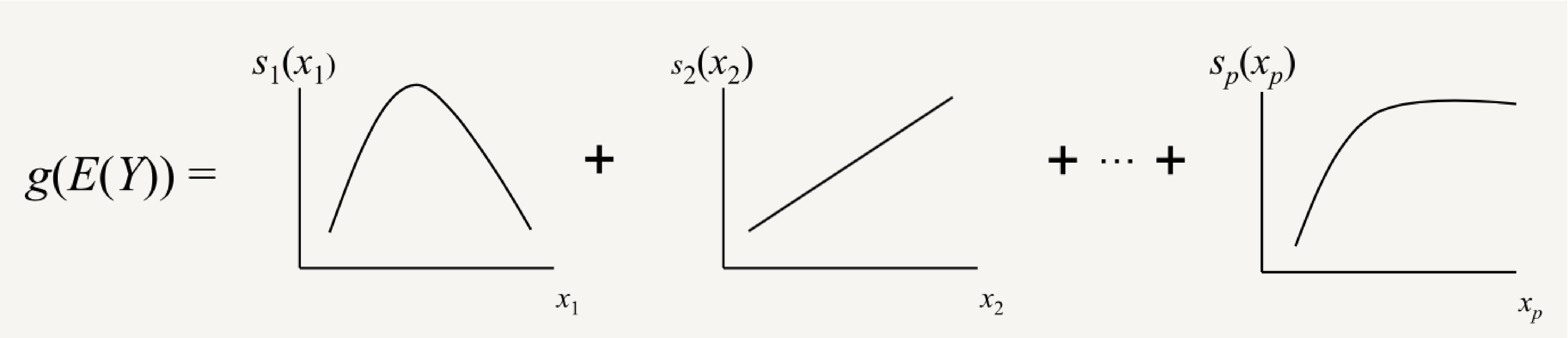
- Can fool the classifier



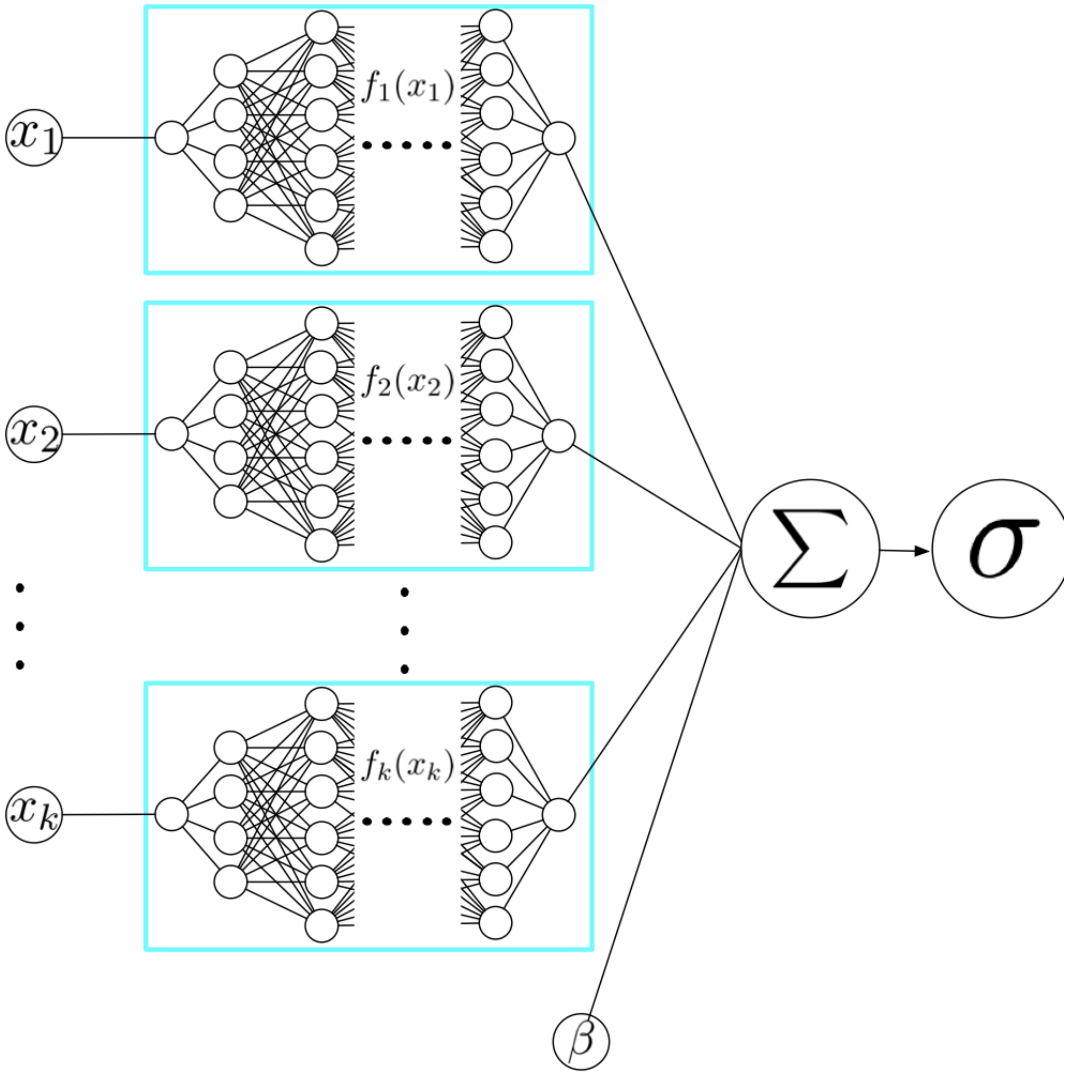
- Proposal – Design the architecture of Neural Networks to make them explainable.

GAMs – Generalized Additive Models :

- The impact of the predictive variables is captured through smooth functions
- g(E[y]) = β + f1(x1) + f2(x2) + · · · + fK (xK)



- Previous State-of-the-Art models used boosting trees to approximate f(x) functions



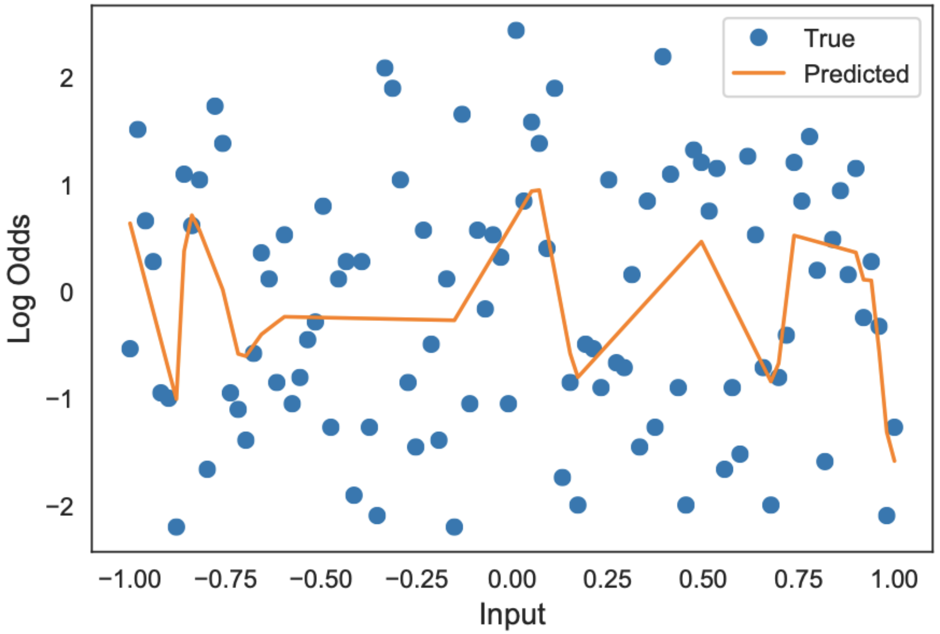
NAMs

- NAMs - linear combination of networks for each input feature:
  - Each fi(xi) is parametrized by a neural network.
- Interpreting NAMs is easy
  - Features are independent of each other
  - Can visualize shape of function (*e.g.,* plotting fi(xi) *vs.* xi).

RELU

- ReLu with mini-batch training provide smooth fits. (different case when using full batch training)

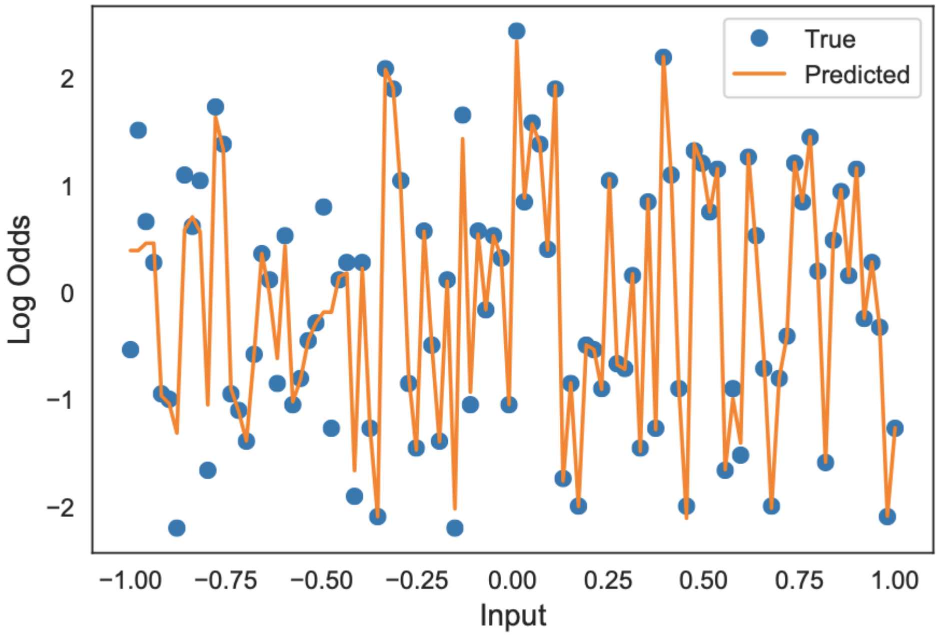
$$h(x) = max(0, wx + b)$$



EXU (exp-centered Units)

- ExU model jagged functions by computing a linear function with steep slope even with small weights.

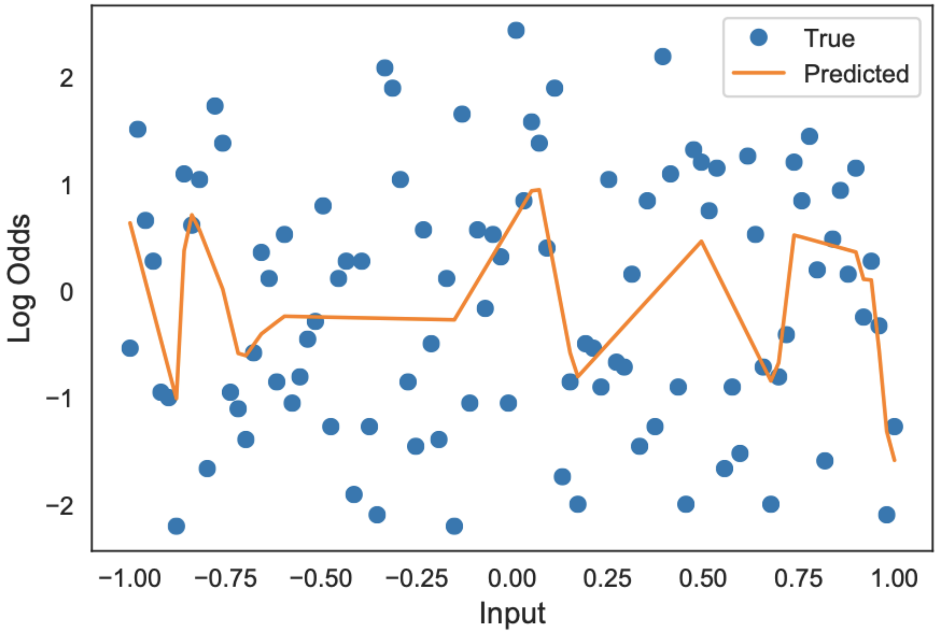
$$h(x) = f\left(e^{w} * (x - b)\right)$$



RELU

- ReLu with mini-batch training provide smooth fits. (different case when using full batch training)

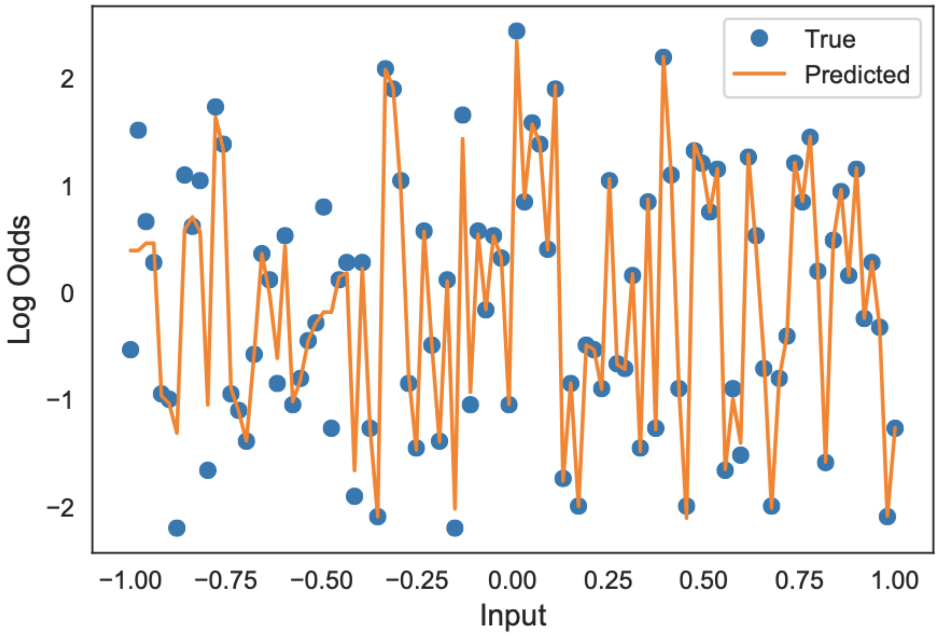
$$h(x) = max(0, wx + b)$$



EXU (exp-centered Units)

- ExU model jagged functions by computing a linear function with steep slope even with small weights.

$$h(x) = f\left(e^{w} * (x - b)\right)$$



Datasets

- COMPAS – Risk Prediction (Classification)
- MIMIC 2 – Mortality prediction in ICU (Classification)
- Credit Fraud detection (Classification)
- California Housing price prediction (Regression)
- FICO score predictor (Regression)

Baseline Models

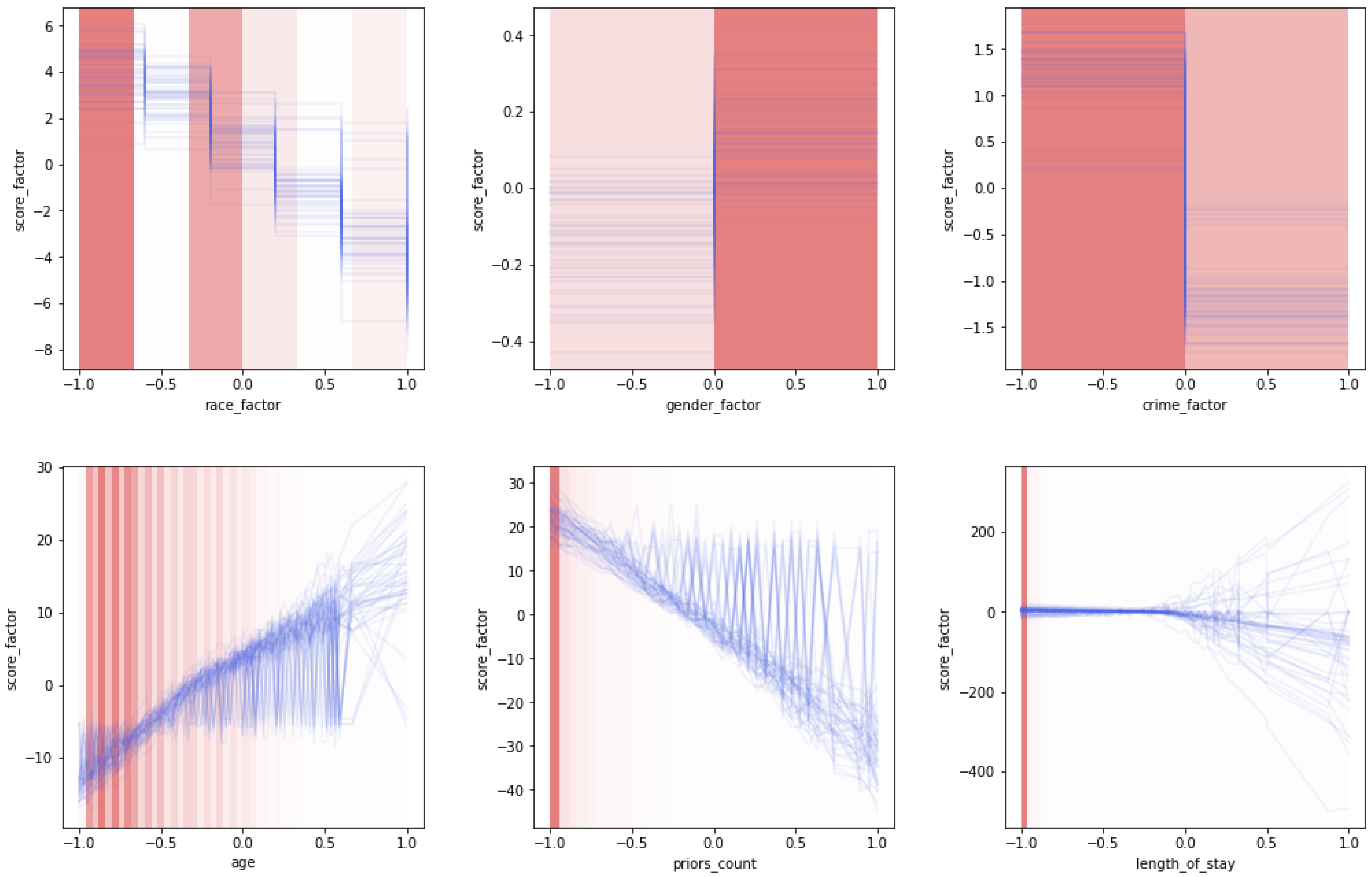
- Logistic/Linear Regression
- Decision Tree
- XGBoost
- Explainable Boosting Machines
- Deep Neural Nets

COMPAS – Risk Prediction

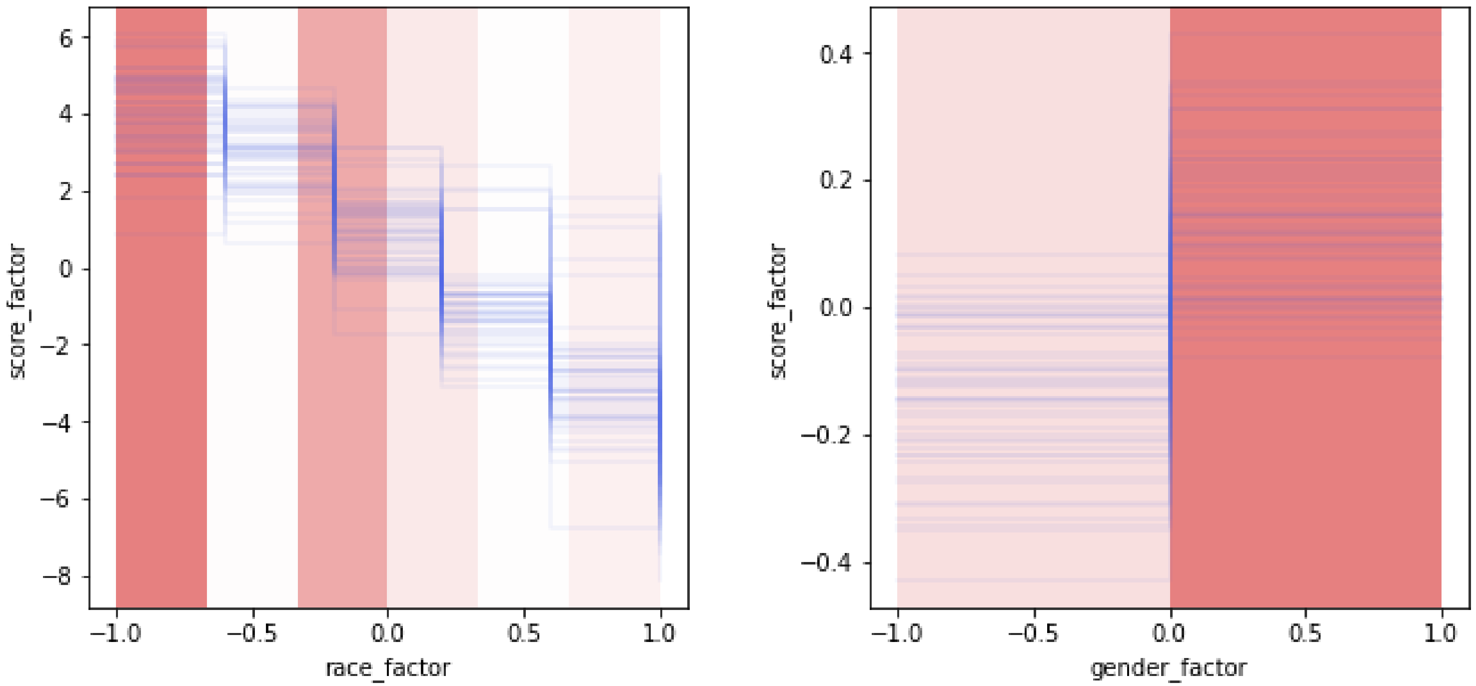
Model	AUC Score
Logistic Regression	0.75
Decision Trees	0.73
XGBoost	0.75
EBMs	0.76
Deep Neural Networks	0.74
NAMs	0.72

Explaining COMPAS NAM model

- Trained 50 ensemble networks.
- The redness indicates the density of the data at that location.



Explaining COMPAS NAM model



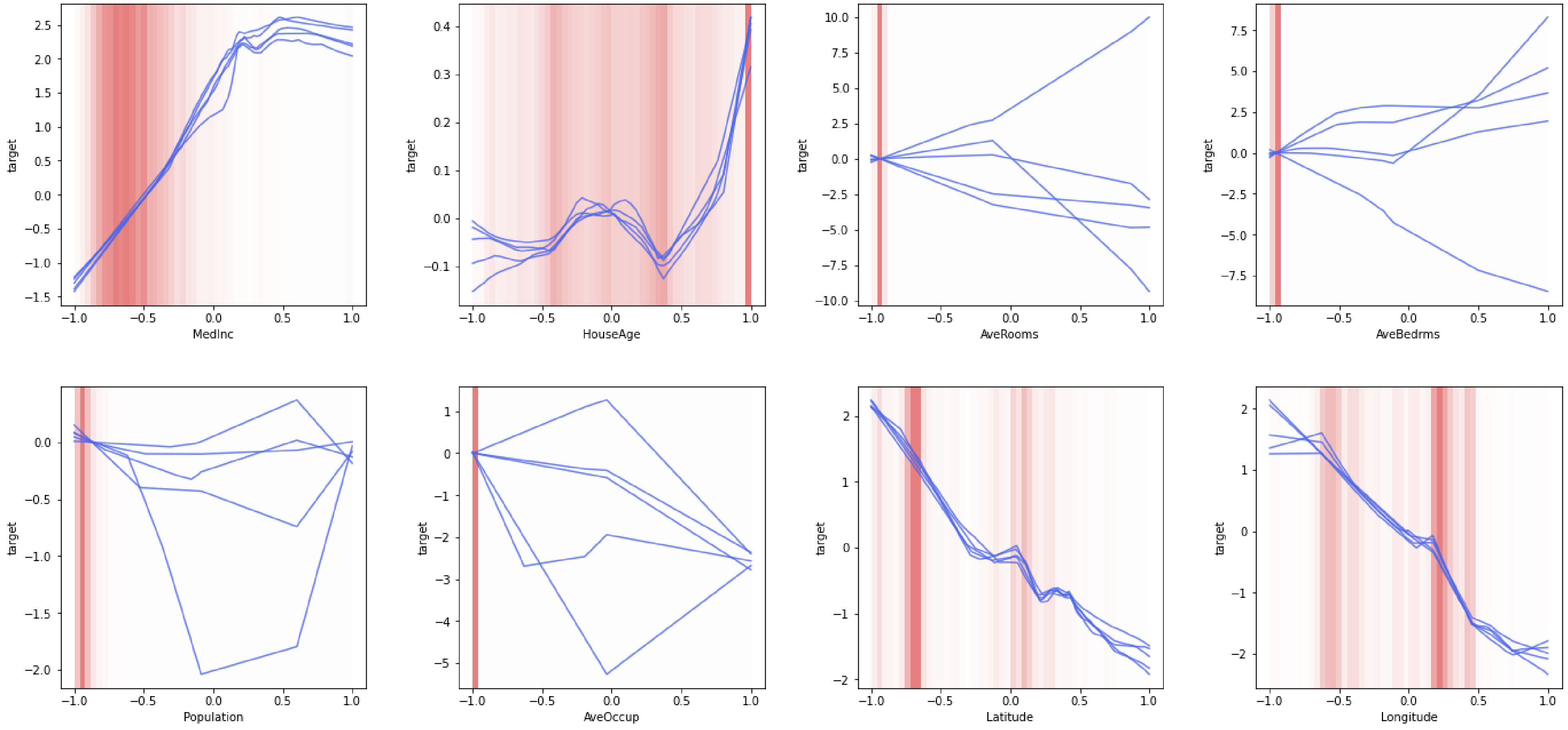
Race: {'African-American': 0, 'Asian': 1, 'Caucasian': 2, 'Hispanic': 3,'Native American': 4, 'Other': 5}

Gender: {'Female': 0, 'Male': 1}

California Housing price prediction

Model	MSE Score
Linear Regression	0.53
Decision Trees	0.52
XGBoost	0.3
EBMs	0.25
Deep Neural Networks	0.46
NAMs	0.43

Explaining Housing price prediction model

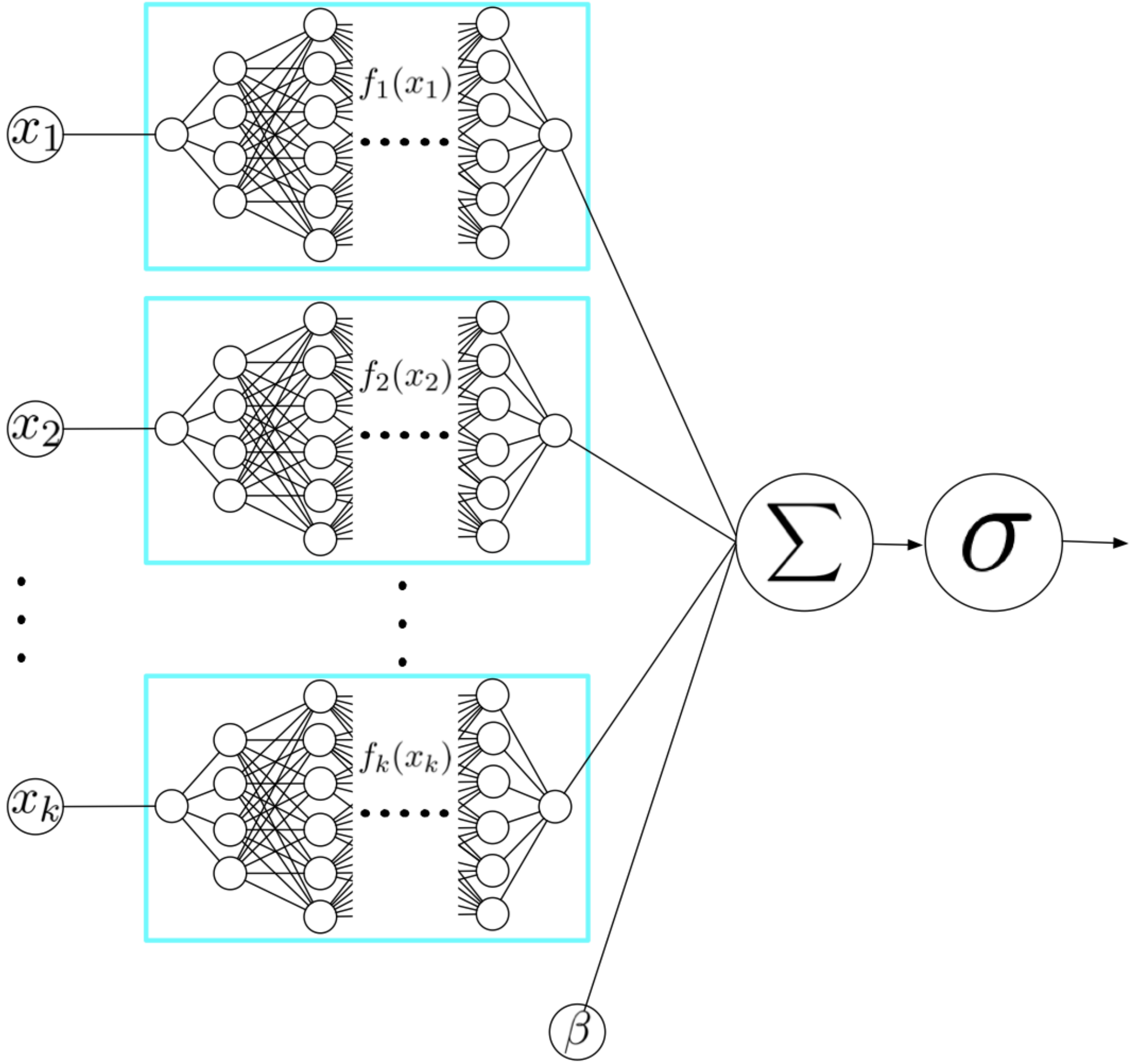


Benefits of NAMs

- Expressive and interpretable
- Feature independence allows for human understandable visualization
- Comparable to high performing models for many problems

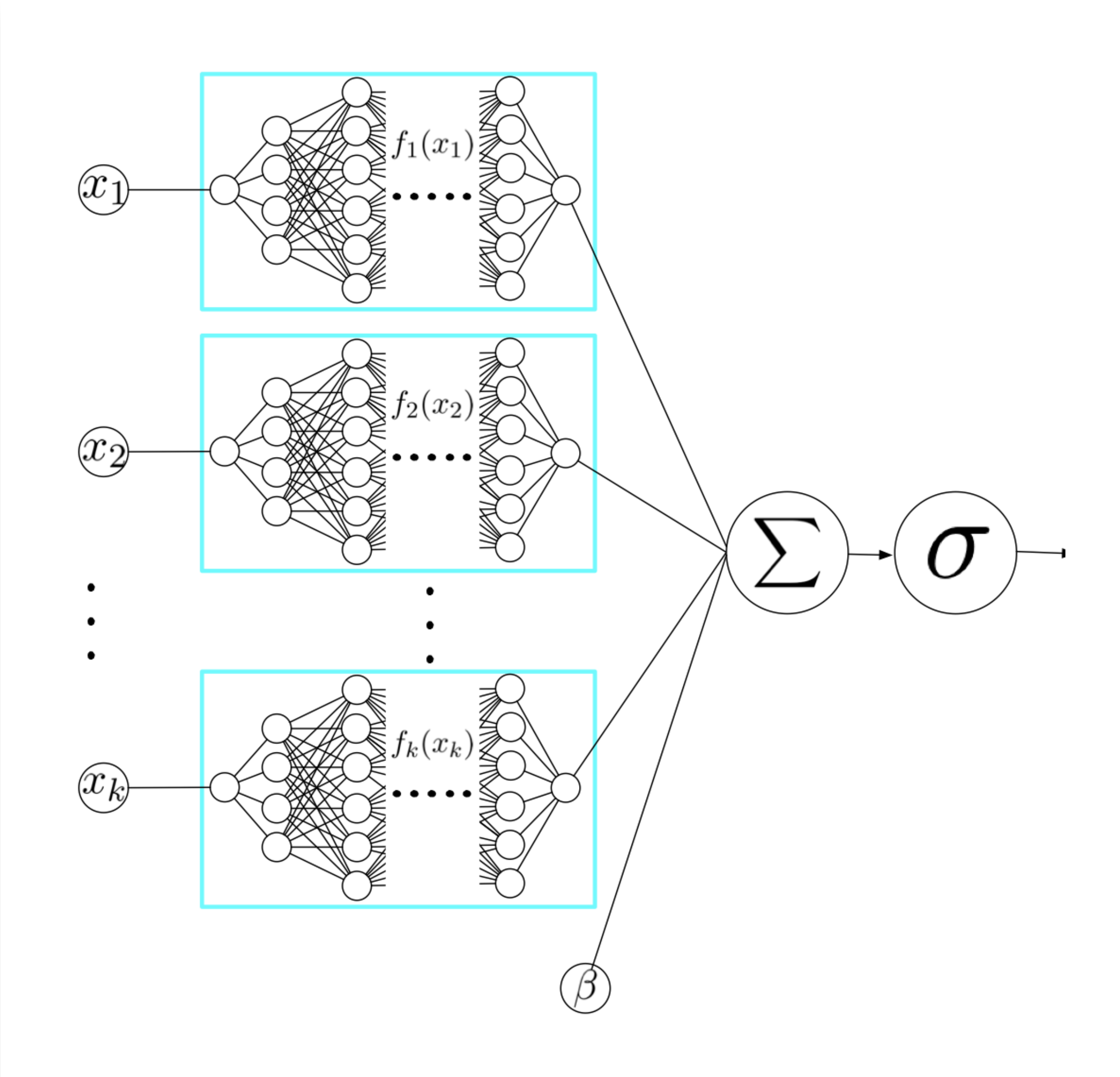
Weakness of current NAM implementation

- Networks for single features
- Lack of feature interactions is a major weakness
- Computationally Expensive



Proposed Solution

- Feature interaction terms
- g(E[y]) = β + f1(x1) + f2(x2) + · · · + fK (xK) + fK+1(x1,x2) + · · ·
- Only slight loss of interpretability
- Many terms O(KCn)
  - K = number of features
  - n = number in combination
  - C = combination/"choose"



Conclusion :

- NAMs are competitive with performant black box models
- NAMs are an explainable alternative to black box models (i.e. DNNs)
- Hyper parameter tuning is tedious
- Computationally expensive
    - DNN for each feature – complexity scales up quickly