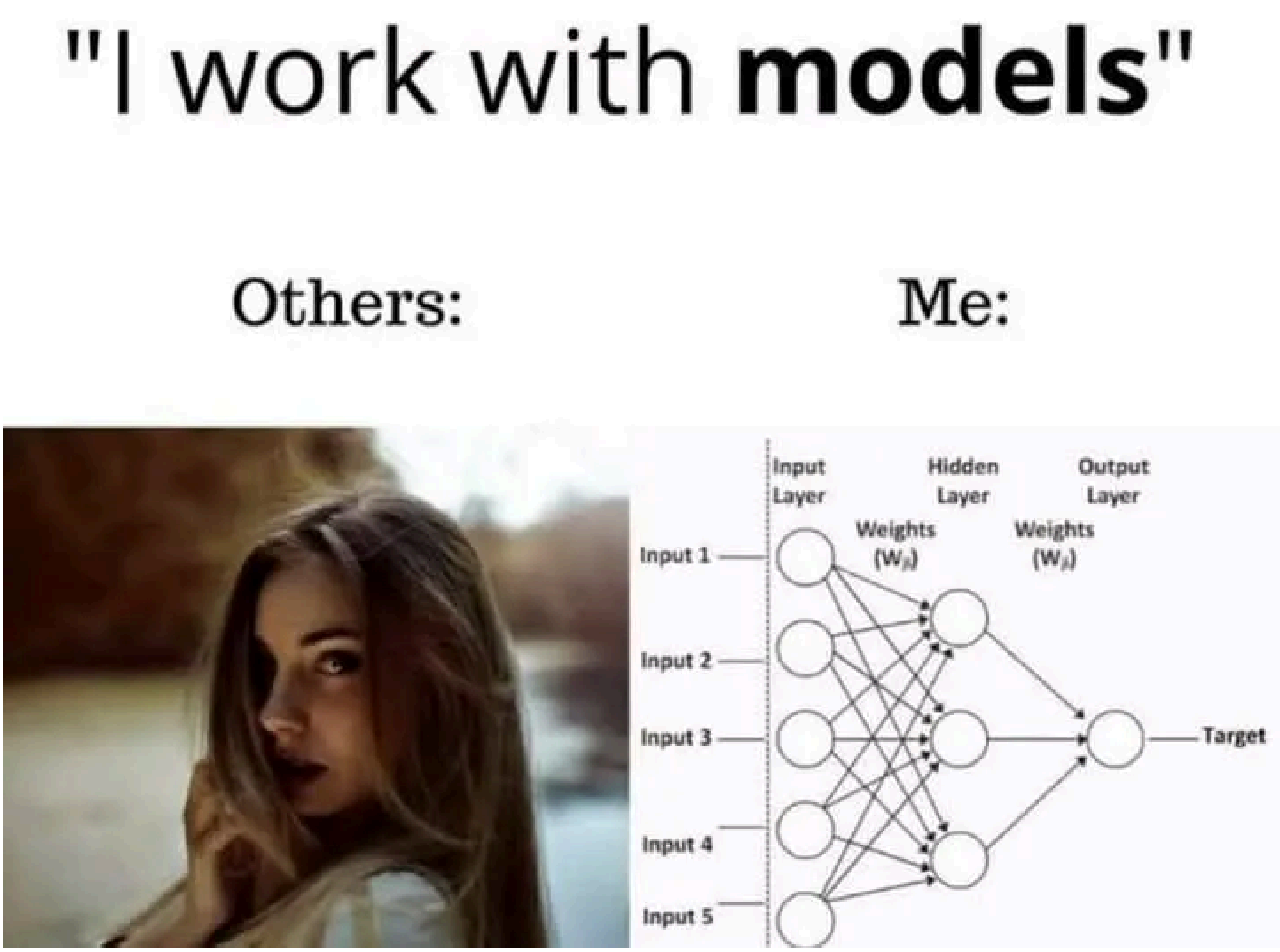
Check Your Model: Is It Getting It Right, or Just Faking It?

Sayemuzzaman Siam

4 min read · Jan 27, 2025

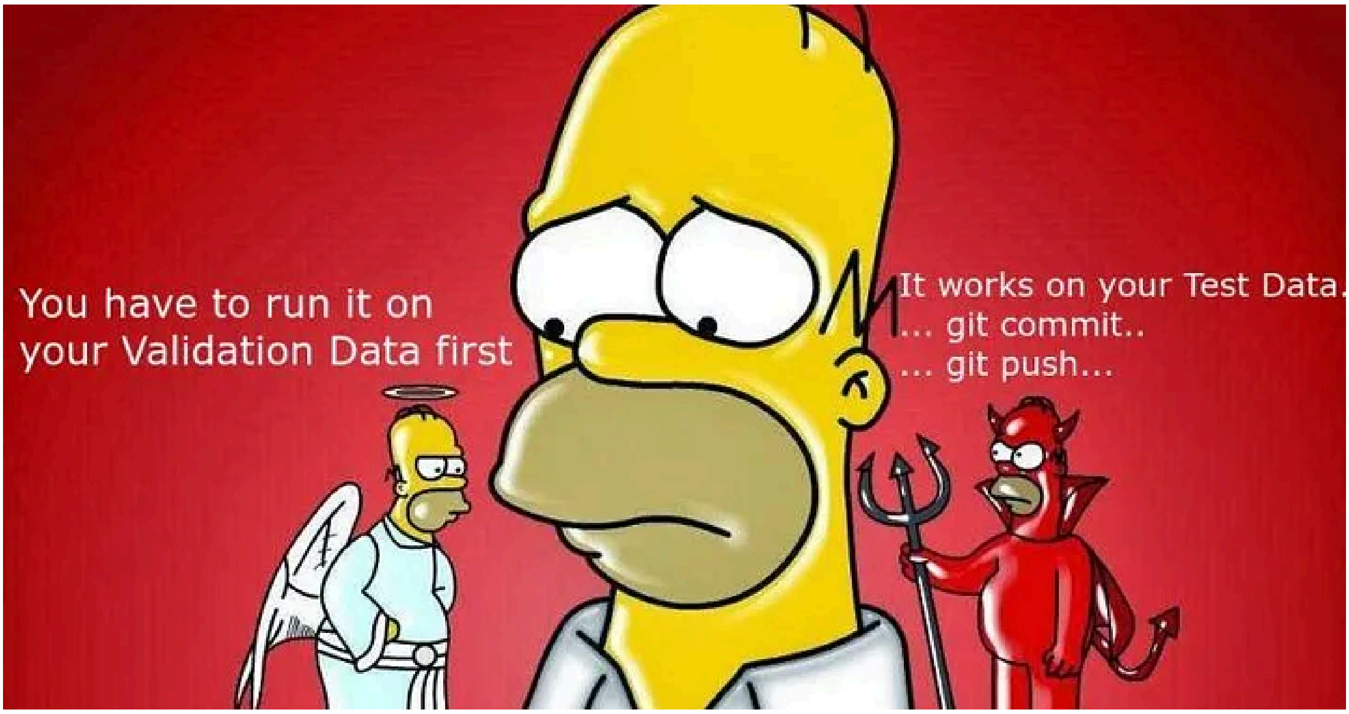
Listen Share More



Your models pose, my models predict. We are not the same😤

Training a machine learning model is like teaching a pet — you want it to learn the right tricks without overdoing it or forgetting the basics! But how do you know if your model is truly learning meaningful patterns or just memorizing noise in the data?

Techniques like **cross-validation, learning curves, and activation visualizations** help you quickly spot whether your model is **underfitting, overfitting, or perfectly balanced.** Let's dive in and find out.


You have to run it on your Validation Data first
It works on your Test Data.
... git commit..
... git push...

Techniques to Identify Underfitting and Overfitting:

1. Cross-Validation

> **How It Works***:*

Split your data into multiple folds, and train and evaluate the model on each fold.

> **What to Look For***:*

Low performance across all folds → **Underfitting.**

High variance between folds → **Overfitting.**

Pro Tip*: Use stratified k-fold for imbalanced datasets to ensure each fold represents the class distribution.*

2. Learning Curves

> **How It Works***:*

Plot the model's performance (**accuracy**↑ or **error**-**loss**↓) on both training and validation sets over time or as the training set size increases.

> **What to Look For***:*

Training and validation performance stabilize at a low level →**Underfitting.**

Large gap between training and validation performance → **Overfitting.**

3. Visualizing Activations (Neural Networks)

> **How It Works***:*

Analyze the activations of layers in a neural network to see if the model is learning useful features or overfitting to noise.

Tools*: TensorBoard, Grad-CAM, or activation heatmaps.*

> **What to Look For***:*

Uniform or uninformative activations → **Underfitting**.

Overly specific activations (memorizing noise) → **Overfitting.**

Overfitting: Symptoms and Solutions

Symptoms:

1.Model performs well on training data but poorly on validation/test data.

2.High variance in cross-validation results.

Overfitted bed: It fits the model perfectly… but good luck getting anyone else in!

Solutions*:*

1.**Reduce Model Complexity**:

Use fewer layers (neural networks) or fewer parameters (reduce tree depth in decision trees).

2.**Regularization Techniques**:

L1 (**Lasso**)**:** Encourages sparsity and feature selection.

L2 (**Ridge**)**:** Smooths weights to prevent over-reliance on specific features.

Dropout: Randomly drop units during training (neural networks).

3.**Early Stopping**: Stop training when validation performance stops improving.

4.**Data Augmentation**: Increase dataset size through transformations

5.**Batch Normalization:** Normalize activations to stabilize training (neural networks).

6.**Use Simpler Models:** Switch to a less complex algorithm.

Underfitting: Symptoms and Solutions

> **Symptoms***:*

Model performs poorly on both training and validation/test data.

Low performance across all cross-validation folds.

You are underfit for this king size bed, get a wife and have children

> **Solutions***:*

1.**Increase Model Complexity:** Add more layers (neural networks) or increase the number of parameters (increase tree depth in decision trees).

2.**Feature Engineering:** Create new features or transform existing ones (polynomial features, interaction terms).

3.**More Data:** Increase the number of data in the dataset.

4.**Hyperparameter Tuning:** Adjust hyperparameters like learning rate, number of layers, or number of estimators.

5.**Ensemble Methods:** Combine multiple models (bagging, boosting) to improve accuracy and handle complex relationships.

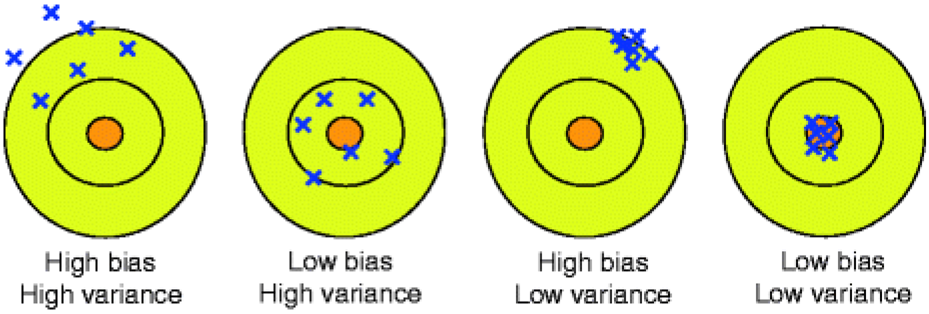
Common Solutions for Both Underfitting and Overfitting:

1.**Clean Data**: Ensure the data is properly preprocessed, free from outliers, and representative of the problem space.

2.**Loss Functions**: Choose or modify loss functions to better suit the problem (focal loss for class imbalance, Huber loss for robust regression).

3.**Change Algorithms:** Experiment with different algorithms that might perform better for your specific problem.

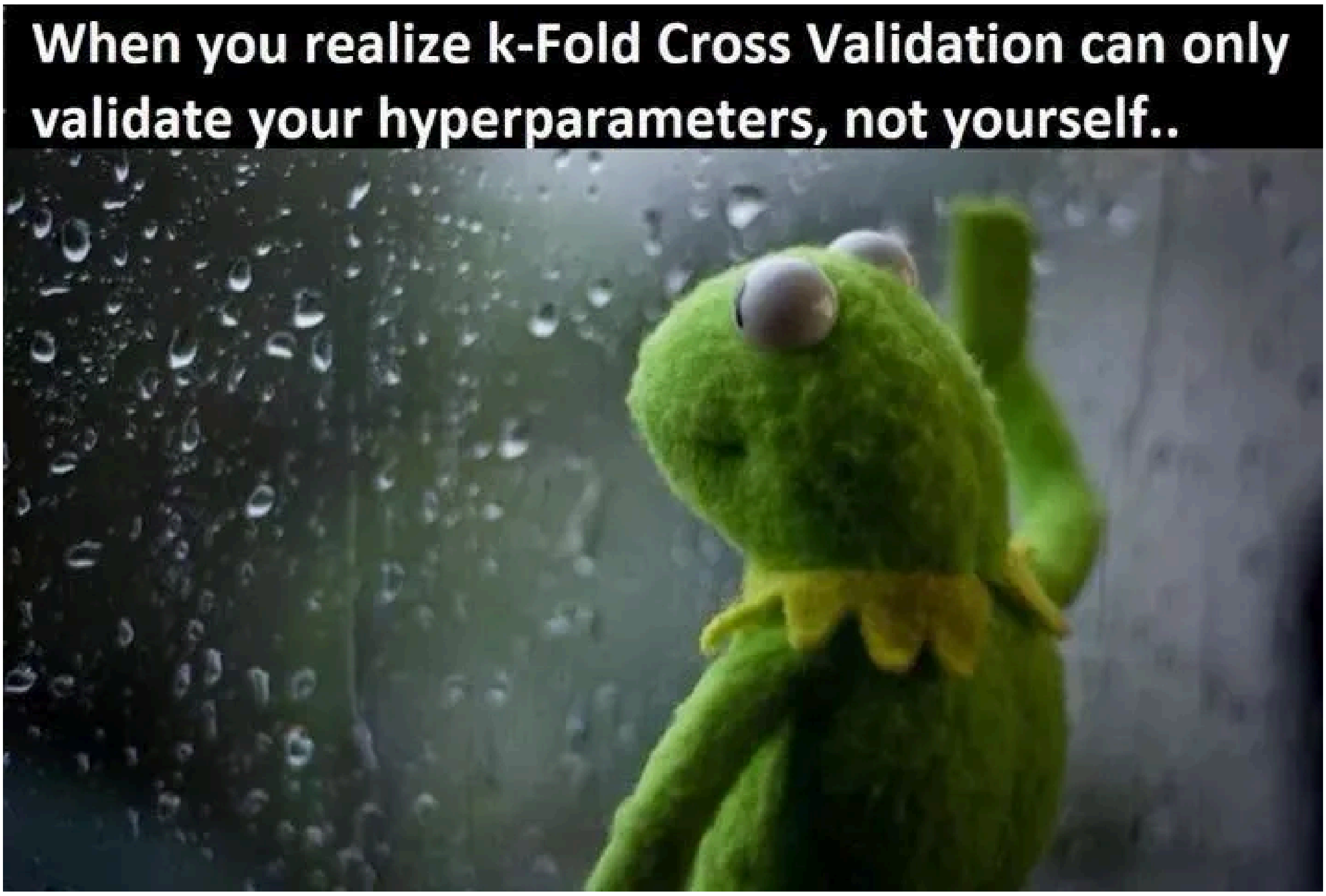
Bias-Variance Tradeoff: The Big Picture



High Bias = You are bias towards someone, not neutral! and High Variance = More scattered!

Underfitting → High Bias: The model is too simple to capture the underlying patterns.

Overfitting → High Variance: The model is too complex and memorizes noise instead of learning generalizable patterns.


When you realize k-Fold Cross Validation can only validate your hyperparameters, not yourself..

SAD Life

Machine Learning

Deep Learning

Neural Networks

Algorithms

Memes

Edit profile

Written by Sayemuzzaman Siam

1 Follower · 16 Following

Deep Learning Enthusiast.