ML SUMMER SCHOOL 2025

MASTERING MODEL TUNING: FROM OVERFITTING TO CROSS-VALIDATION”

LECTURE BY- SIMRAT KAUR

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

AGENDA

- Overfitting vs Underfitting
- Bias-Variance Tradeoff
- Hyperparameter Tuning
- Grid Search & Random Search
- Upsampling & Data Imbalance Techniques
- K-Fold Cross Validation

OVERFITTING VS UNDERFITTING

✅ **Overfitting**

- The model memorizes the training data, including noise and outliers.
- Performs very well on training data, but poorly on unseen/test data.
- High variance, low bias.

❌ **Underfitting**

- The model is too simple to capture the underlying patterns in data.
- Performs poorly on both training and testing data.
- High bias, low variance.

STEPS TO AVOID OVERFITTING

- hold back of validation dataset
- using resampling techniques like k fold cross validation
- remove the data points which have little or no predictive power(feature selection)

BIAS AND VARIANCE TRADEOFF

🎯 Bias

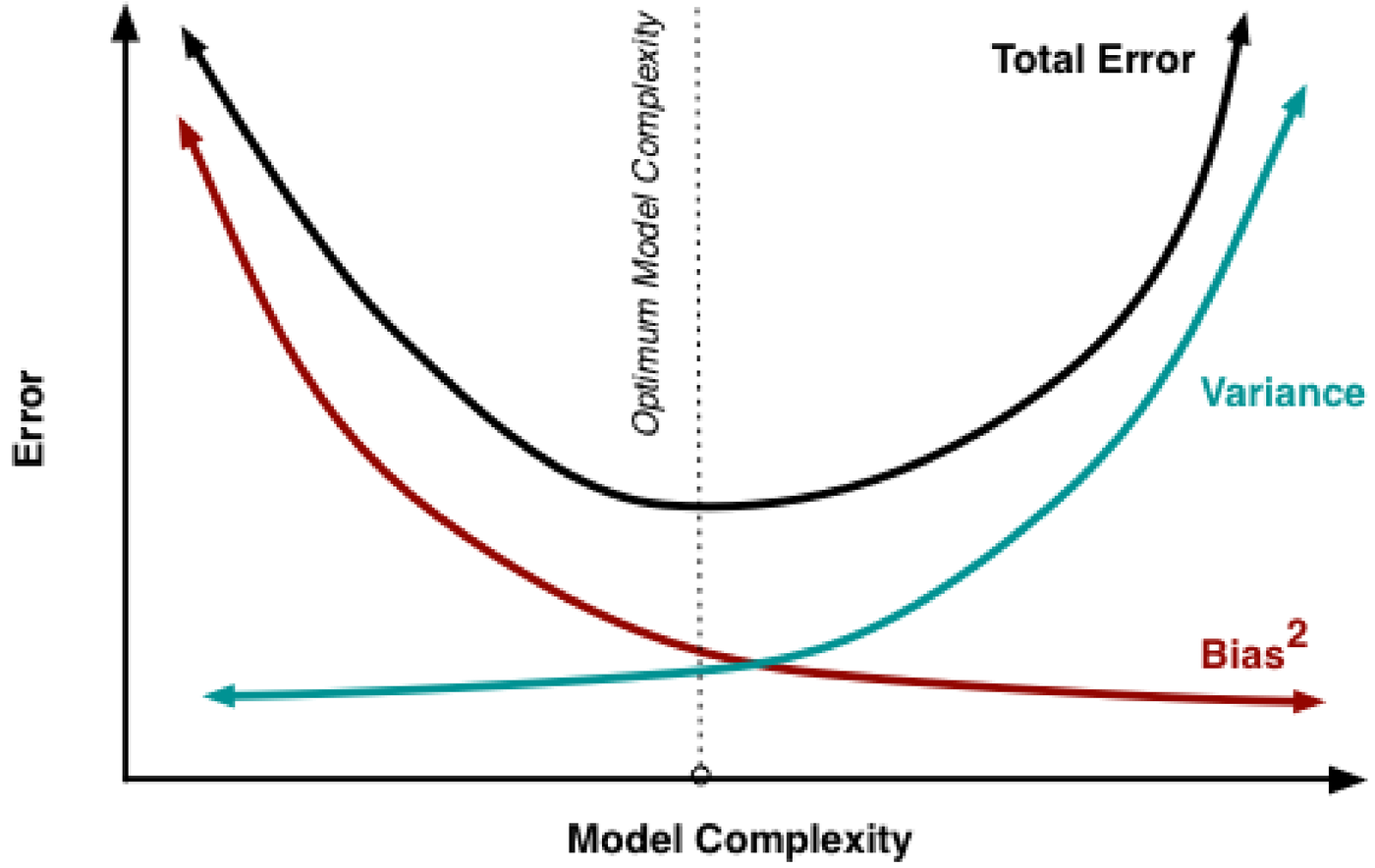
- Bias is the error due to overly simplistic assumptions in the model.
- A high bias model misses important relationships between features and target (i.e., underfitting).
- Example: Using a linear model for curved data.

🔁 Variance

- Variance is the error due to the model's sensitivity to small changes in training data.
- A high variance model fits the training data too closely and fails to generalize (i.e., overfitting).
- Example: Using a high-degree polynomial on small noisy data.

TRADE OFF

Trade off can be controlled by adjusting complexity, a simple model may have high bias but low variance but a more complex model may have low bias but high variance.



SOLUTIONS TO OVERFITTING

- Regularization (L1, L2)
- Early stopping
- Simplifying model
- Cross-validation

REGULARIZATION (L1 AND L2)

- Regularization adds a penalty term to the model's loss function to discourage complexity (like large weights).
- Helps prevent the model from relying too heavily on specific features (especially noisy ones).

Types:

- L1 Regularization (Lasso):
  - Adds $|w|$ (absolute value of weights) to the loss.
  - Encourages sparsity — can shrink some weights to zero → useful for feature selection.
- L2 Regularization (Ridge):
  - Adds w^2 (squared weights) to the loss.
  - Shrinks weights smoothly → discourages large coefficients, but doesn't eliminate them.

EARLY STOPPING

What It Is:

- A technique used in iterative models (especially neural networks).
- Stop training as soon as the validation error starts increasing, even if training error keeps decreasing.

Why It Helps:

- Prevents the model from over-optimizing and fitting noise in the training data.
- Acts like a natural regularizer.

SIMPLIFYING THE MODEL

- Use fewer parameters or a simpler algorithm to reduce the model's capacity.

Examples:

- Reducing the depth of a decision tree.
- Using linear regression instead of polynomial regression.
- Reducing the number of hidden layers or neurons in a neural network.

CROSS-VALIDATION

- A resampling technique used to evaluate models reliably on limited data.
- Common form: K-Fold Cross Validation — Split data into K parts, train on K-1, test on the remaining one, repeat.

Why It Helps:

- Provides a better estimate of test performance.
- Reduces overfitting risk to a single train/test split.
- Helps in choosing more robust models or hyperparameters.

HYPER-PARAMETER TUNING

- Hyperparameters are settings you define before training a model (not learned from data).
- They significantly impact:
- Model accuracy
- Training time
- Overfitting vs Underfitting
- Good tuning → Better generalization and performance.

TECHNIQUES FOR TUNING

Grid Search

- Exhaustively tries all combinations from a specified grid of hyperparameters
- Computationally expensive as dimensions grow

Random Search

- Randomly samples combinations from the hyperparameter space
- Often finds good solutions faster than Grid Search
- More efficient for high-dimensional spaces

DATA IMBALANCE & UPSAMPLING TECHNIQUES

- A dataset is imbalanced when the number of samples in each class is not equally represented.

Example:

- Fraud detection: 98% non-fraud, 2% fraud
- Medical diagnosis: 95% healthy, 5% disease

Random Oversampling

- Duplicates minority class examples randomly until both classes are balanced.
- Simple, fast — but may lead to overfitting on repeated samples.

SMOTE (Synthetic Minority Over-sampling Technique)

- Generates new synthetic samples by interpolating between existing minority class samples.
- Works by:
  - Picking a sample
  - Finding its k nearest minority neighbors
  - Generating points along the lines joining them

```python
# Step 2: Apply Random Oversampling
ros = RandomOverSampler(random_state=42)
X_resampled, y_resampled = ros.fit_resample(X, y)

print("Resampled class distribution:", Counter(y_resampled))
```

```python
# Step 2: Apply SMOTE
smote = SMOTE(random_state=42)
X_smote, y_smote = smote.fit_resample(X, y)

print("Resampled class distribution:", Counter(y_smote))
```

CASE STUDY EXAMPLE

- Model: Logistic Regression
- Dataset: Credit card fraud / Breast cancer
- Walkthrough:
- Train/test split
- Overfitting
- GridSearchCV
- KFold
- Upsampling with SMOTE

THANK YOU