Modules 2: Training, Testing, and Cross Validation

Training, Testing, and Cross-Validation — Full Summary

Purpose

In machine learning, the goal is not just to fit the training data but to build models that generalize well to unseen data. We split data and use cross-validation to estimate performance reliably and select appropriate model complexity.

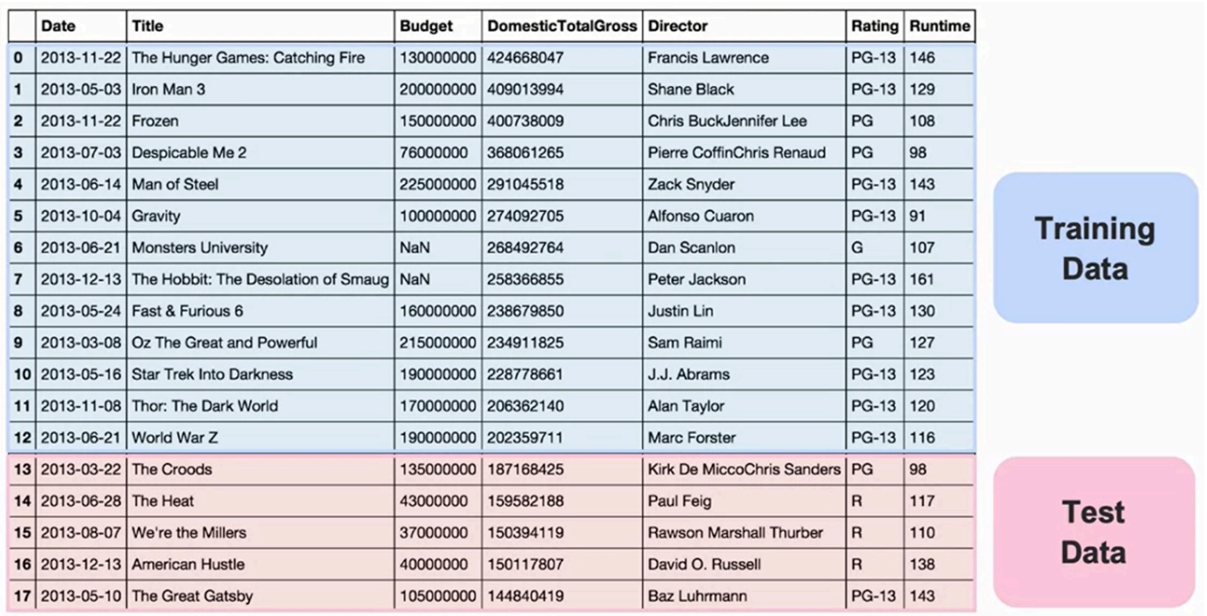
---

1. Dataset Splits: Concepts

- Training set: Fit model parameters.
- Validation set: Tune hyperparameters and select models.
- Test set: Final, unbiased evaluation after all decisions are fixed.

> Golden rule: **Keep the test set untouched until the very end.**

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143



2. Recommended Split Ratios

Guideline table for split ratios by dataset size. Balance training data with reliability of estimates.

Summary of recommended split ratios by dataset size

Dataset size	Suggested split	Notes
Very small (< 1,000)	80/20 or 70/30	Favor more training data to reduce variance.
Medium (1,000–50,000)	75/25	Balanced learning and evaluation.
Large (> 50,000)	90/10	Less test data is sufficient at scale.
With a separate validation set	60/20/20 (train/val/test)	Prevents test leakage during tuning.

- Rule of thumb: 80/20 is a safe default.

3. Error Measurement and the Bias–Variance Trade-off

- Empirical risk (training error)

$$R_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n} L\big(y_i, f(x_i)\big)$$

For regression with MSE: $L(y, hat y) = (y - hat y)^2$.

- Total prediction error

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

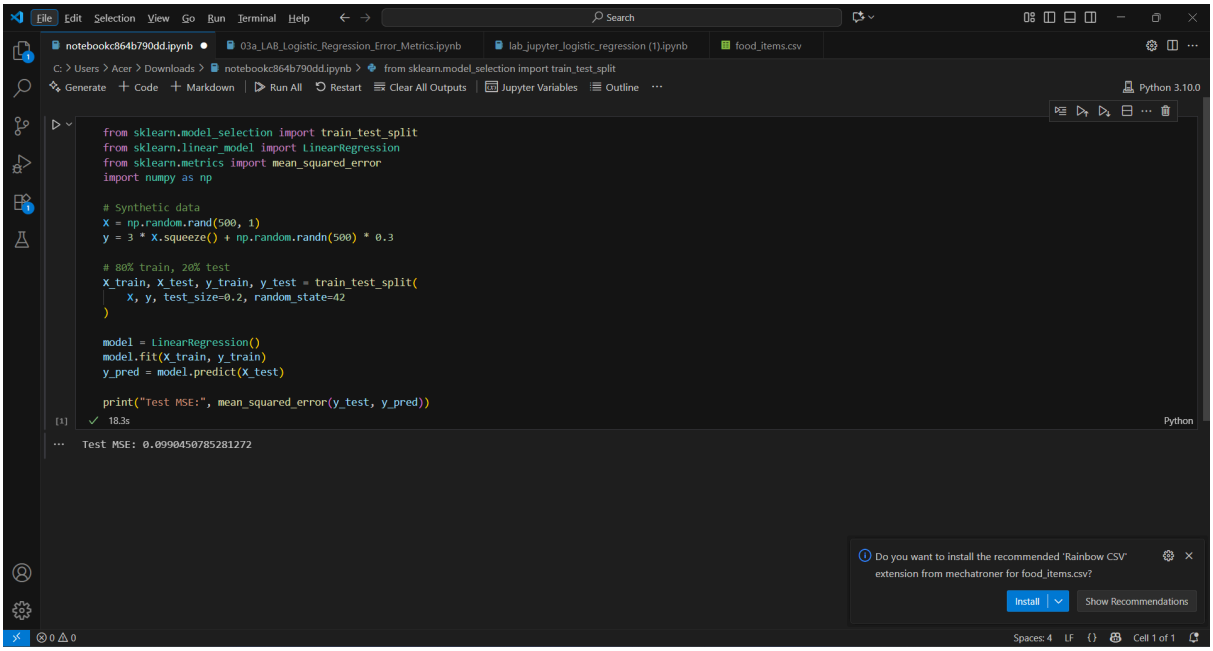
- Intuition:
  - Bias: Error from overly simple assumptions → underfitting.
  - Variance: Error from sensitivity to the training data → overfitting.
  - Irreducible error: Noise we cannot remove.

> Aim for the sweet spot: low bias and low variance.

---

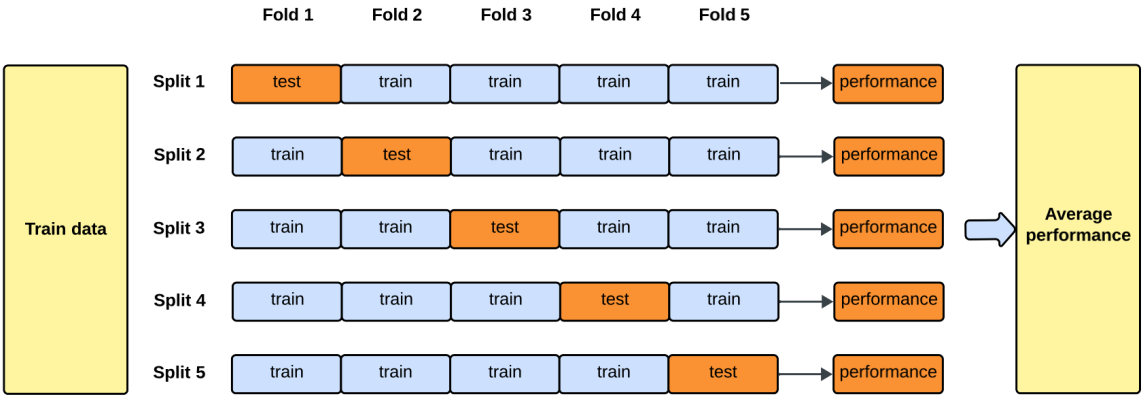
4. Example: Simple Train/Test Split

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

# Synthetic data
X = np.random.rand(500, 1)
y = 3 * X.squeeze() + np.random.randn(500) * 0.3

# 80% train, 20% test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("Test MSE:", mean_squared_error(y_test, y_pred))
```



5. Cross-Validation (CV)

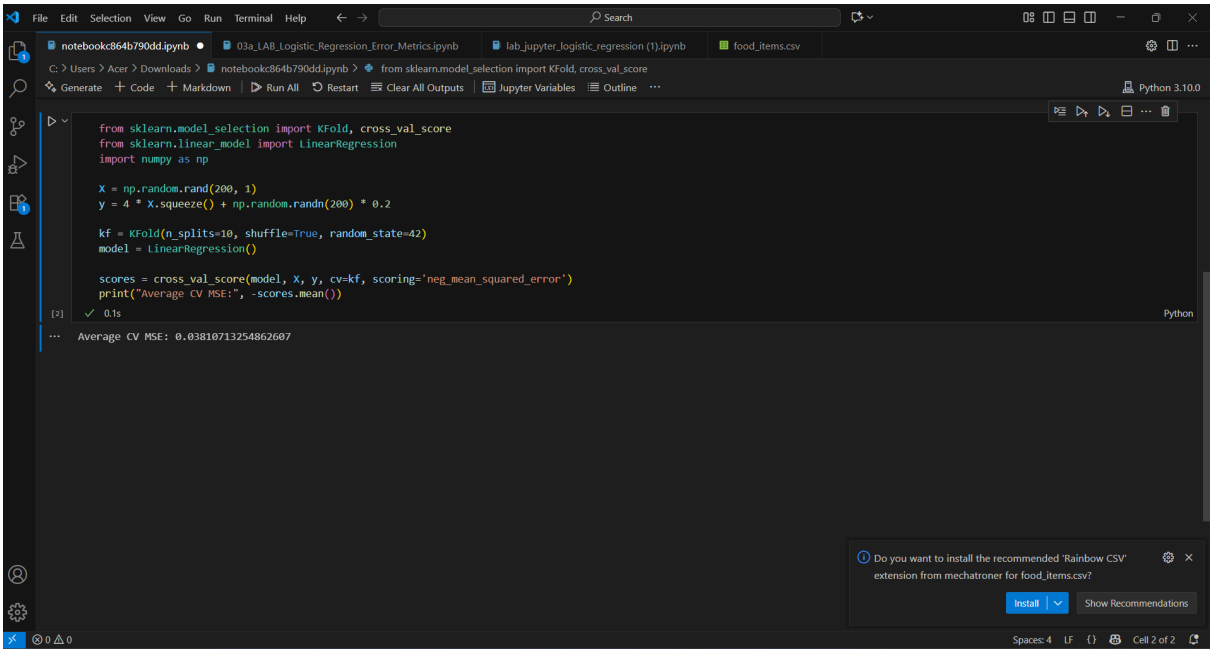
- Purpose: More accurate and stable performance estimates by repeating training and evaluation across multiple splits.
-



- Common methods
  - K-Fold CV: Default choice. k = 5 or 10 → each fold uses about 80–90% for training and 10–20% for validation.
  - Stratified K-Fold: For classification, especially with class imbalance.

    - LOOCV: Very small datasets (< 100). k = n. Maximizes training data, but computationally expensive.
    - Repeated K-Fold: Repeat K-Fold several times to reduce estimate variance.
    - Time Series Split: For sequential data. Preserves temporal order, no shuffling.
- Approximate train/test per fold
    - k = 3: train ~67%, test ~33%
    - k = 5: train ~80%, test ~20%
    - k = 10: train ~90%, test ~10%
    - LOOCV: train ~99%, test 1 sample
- Example: K-Fold CV

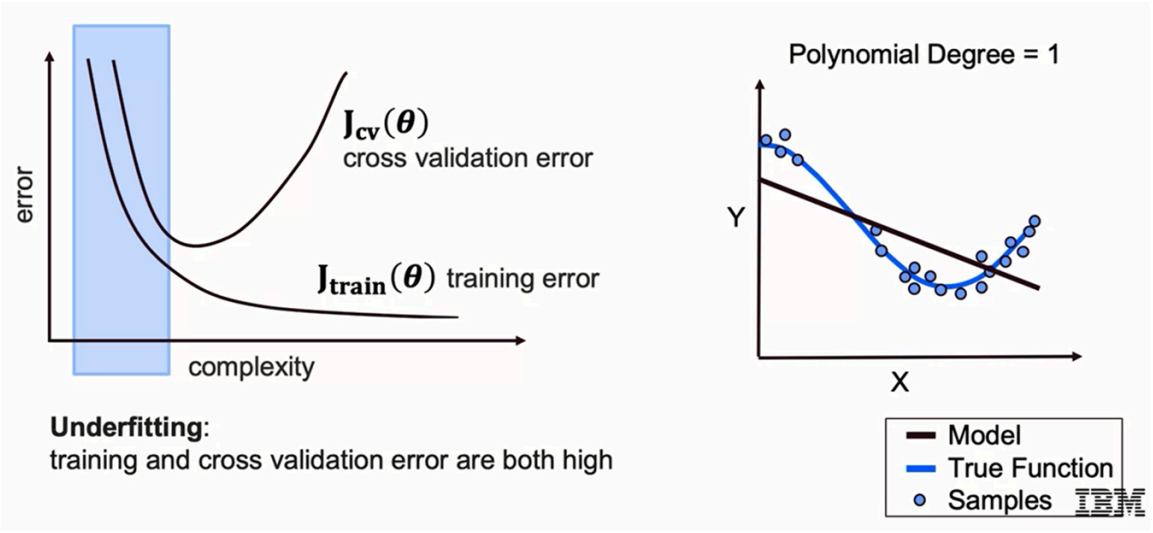
```
from sklearn.model_selection import KFold, cross_val_score
from sklearn.linear_model import LinearRegression
import numpy as np

X = np.random.rand(200, 1)
y = 4 * X.squeeze() + np.random.randn(200) * 0.2

kf = KFold(n_splits=10, shuffle=True, random_state=42)
model = LinearRegression()

scores = cross_val_score(model, X, y, cv=kf, scoring='neg_mean_squared_
error')
print("Average CV MSE:", -scores.mean())
```



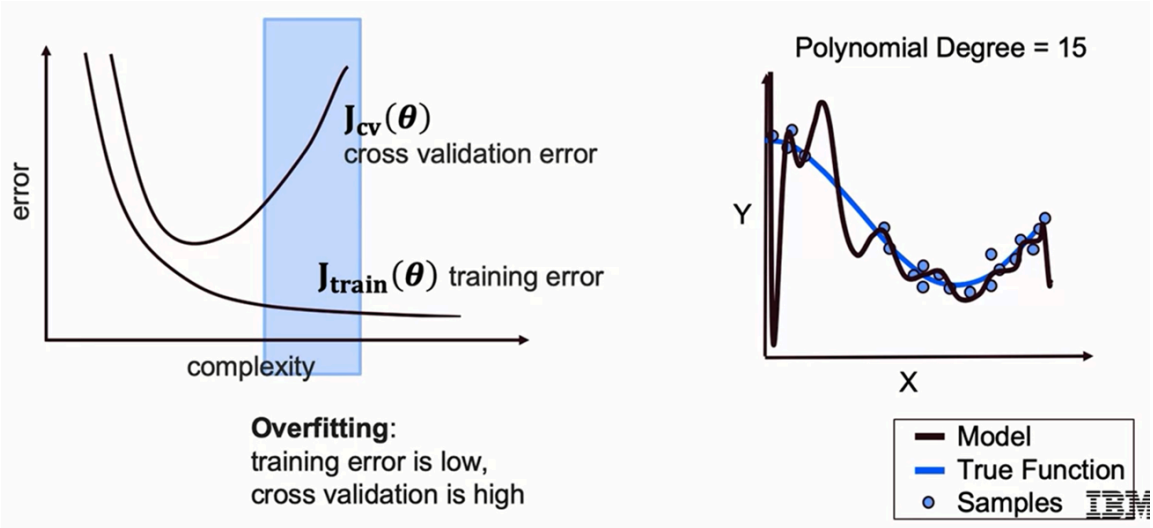
6. Model Complexity vs. Bias–Variance

6.1 Clarifying Underfitting, Overfitting, and “Just Right”

- Underfitting
  - What it is: Model is too simple to capture the true pattern.
  - Symptoms: High training error and high validation/test error. Learning curves plateau early at high error.
  - Common causes: Too few features, overly strong regularization, model family too simple.
  - How to fix: Add features or interactions, reduce regularization, choose a more expressive model, train longer.

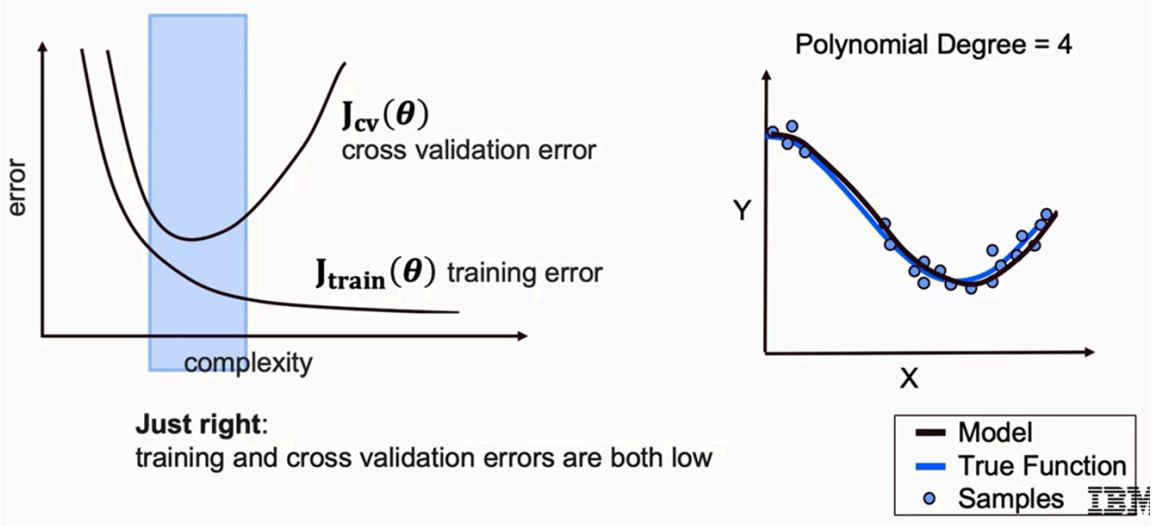


- Overfitting
  - What it is: Model memorizes noise and idiosyncrasies of the training data.
  - Symptoms: Very low training error but much higher validation/test error. Large train–validation gap.
  - Common causes: Model too complex, too many parameters, data leakage, insufficient data or weak regularization.
  - How to fix: Increase regularization, simplify the model, collect more data or use augmentation, apply early stopping, use cross-validation, eliminate leakage.



- "Just right" (good generalization)
  - What it is: Model complexity matches the signal in the data.

- Symptoms: Low training error and similarly low validation/test error. Small train–validation gap.
- How to achieve: Tune hyperparameters with cross-validation, monitor learning curves, pick the model at the CV error minimum.



- Quick diagnostics
  - Train high + Validation high → Underfitting.
  - Train low + Validation high → Overfitting.
  - Train low ≈ Validation low → Just right.
- As model complexity increases:
  - Training error monotonically decreases.
  - CV error typically decreases then increases. The minimum marks the optimal complexity.

Complexity	Bias	Variance	Test error
Too simple	High	Low	High
Balanced	Medium	Medium	**Lowest**
Too complex	Low	High	High

7. Choosing the Right Strategy

- Very small (< 1k): LOOCV or 5-fold CV → maximize training data.
- Medium (1k–10k): 80/20 split or 5–10 fold CV → balanced estimates.

- Large (> 10k): 90/10 split or 5-fold CV → faster training, stable estimates.
- Imbalanced classification: Stratified K-Fold → preserves class ratios.
- Time series: TimeSeriesSplit → maintain chronological order.

---

8. Quick Summary

- Train/Test Split: simple baseline → often 80/20.
- Train/Val/Test: for hyperparameter tuning → 60/20/20.
- K-Fold CV: robust performance estimate → k = 5 or 10.
- Stratified K-Fold: preserves class balance.
- LOOCV: maximizes data for tiny datasets.
- Time Series CV: no shuffling, respect time order.

Practical Recommendations

- Keep the test set completely unseen until final evaluation.
- Prefer K-Fold CV (k = 5 or 10) when compute budget allows.
- For imbalance, use Stratified splits and consider metrics like ROC-AUC and F1.
- For time series, use TimeSeriesSplit and avoid leakage from the future.
- Choose split ratios based on dataset scale to balance training size and estimate reliability.
- Track learning curves and the CV error minimum to select model complexity.