Statistics and Data Analysis
Unit 04 – Lecture 08 Notes
Cross-validation and Hyper-parameter Tuning

Tofik Ali

February 17, 2026

Topic

Train/test split, k-fold cross-validation, and hyper-parameter tuning (grid/random search).

How to Use These Notes

These notes are written for students who are seeing the topic for the first time. They follow the slide order, but add the missing 'why', interpretation, and common mistakes. If you get stuck, look at the worked exercises and then run the Python demo.

Course repository (slides, demos, datasets): `https://github.com/tali7c/Statistics-and-Data-Analysis`

Time Plan (55 minutes)

- 0–10 min: Attendance + recap of previous lecture
- 10–35 min: Core concepts (this lecture's sections)
- 35–45 min: Exercises (solve 1–2 in class, rest as practice)
- 45–50 min: Mini demo + interpretation of output
- 50–55 min: Buffer / wrap-up (leave 5 minutes early)

Slide-by-slide Notes

Title Slide

State the lecture title clearly and connect it to what students already know. Tell students what they will be able to do by the end (not just what you will cover).

Quick Links / Agenda

Explain the structure of the lecture and where the exercises and demo appear.

- Overview

- Cross-validation
- Hyper-parameter Tuning
- Exercises
- Demo
- Summary

Learning Outcomes

- Explain train/validation/test split roles
- Describe k-fold cross-validation
- Explain grid search vs random search
- Avoid data leakage using pipelines

Why these outcomes matter. **Cross-validation (CV)** repeatedly splits data into train/validation parts to estimate generalization more reliably. It reduces the dependence on a single lucky/unlucky split and is essential for selecting hyperparameters without touching the test set. **Grid search** tries every combination of hyperparameter values in a predefined grid. It is simple but can be expensive. **Random search** can be more efficient when only a few hyperparameters really matter.

Cross-validation: Key Points

- CV estimates generalization more stably than one split
- k-fold repeats train/validate across folds
- Average score guides selection

Explanation. **Cross-validation (CV)** repeatedly splits data into train/validation parts to estimate generalization more reliably. It reduces the dependence on a single lucky/unlucky split and is essential for selecting hyperparameters without touching the test set.

Hyper-parameter Tuning: Key Points

- Grid search tries all combos
- Random search samples combos efficiently
- Never tune on the test set

Explanation. A **parameter** is a fixed (but usually unknown) number that describes the population (e.g., μ, σ). A **statistic** is a number computed from the sample (e.g., $\bar{x}$, s). Statistics vary from sample to sample, which is why we talk about uncertainty. **Grid search** tries every combination of hyperparameter values in a predefined grid. It is simple but can be expensive. **Random search** can be more efficient when only a few hyperparameters really matter.

Exercises (with Solutions)

Attempt the exercise first, then compare with the solution. Focus on interpretation, not only arithmetic.

Exercise 1: Grid size

3 parameters with 4 values each: how many combinations?

Solution

- $4^3 = 64$

Walkthrough. A **parameter** is a fixed (but usually unknown) number that describes the population (e.g., μ, σ). A **statistic** is a number computed from the sample (e.g., $\bar{x}$, s). Statistics vary from sample to sample, which is why we talk about uncertainty.

Exercise 2: Leakage

Is scaling on full dataset before split leakage?

Solution

- Yes; fit preprocessing on training only.

Walkthrough. An **ROC curve** plots true positive rate vs false positive rate across all thresholds. It helps compare classifiers without committing to a single threshold. **AUC** summarizes the ROC curve: higher AUC generally indicates better ranking of positives above negatives. **Data leakage** happens when information from the future or from the test set influences training. Typical examples: scaling before splitting, using target-related features, or using random splits for time series. Leakage can produce very good-looking accuracy that disappears in real deployment.

Exercise 3: Why CV

Why is a single train-test split misleading sometimes?

Solution

- Performance depends on split; CV reduces variance.

Mini Demo (Python)

Run from the lecture folder:

```
python demo/demo.py
```

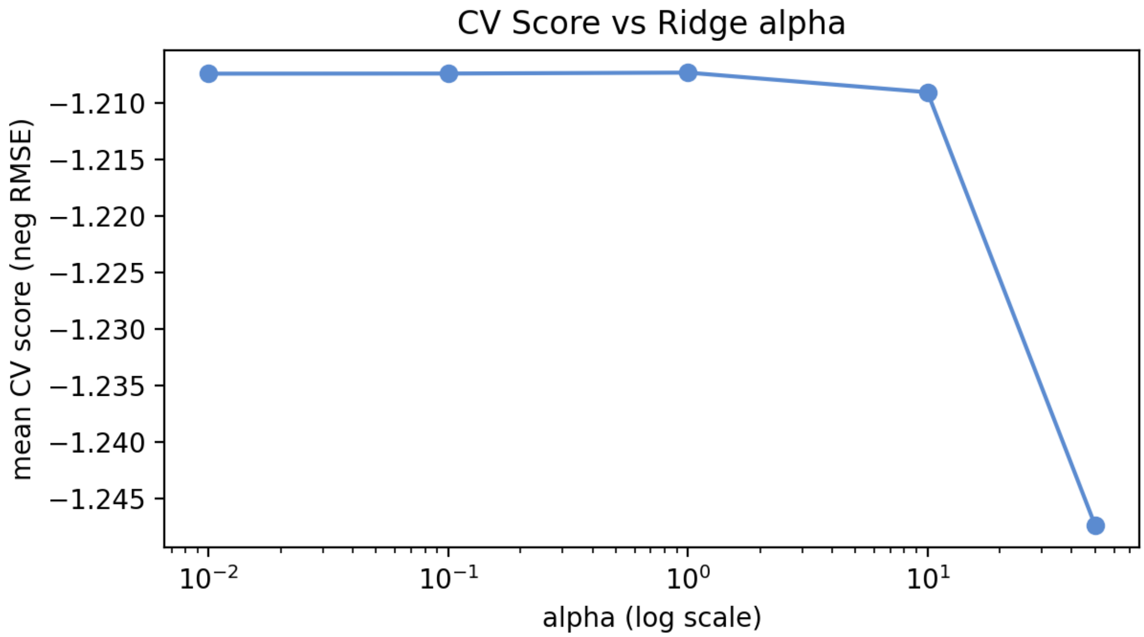
Output files:

- `images/demo.png`
- `data/results.txt`

What to show and say.

- Runs a small CV-like loop and compares model settings across folds.
- Shows why tuning on the test set is leakage and gives optimistic results.
- Use it to explain grid size and why we use pipelines.

Demo Output (Example)



Summary

- Key definitions and the main formula.
- How to interpret results in context.
- How the demo connects to the theory.

Exit Question

Why must you never use the test set to choose hyperparameters?

Suggested answer (for revision). Using the test set for tuning leaks information and makes performance look better than it will be on truly new data; keep test set untouched.

References

- Montgomery, D. C., & Runger, G. C. *Applied Statistics and Probability for Engineers*, Wiley.
- Devore, J. L. *Probability and Statistics for Engineering and the Sciences*, Cengage.

- McKinney, W. *Python for Data Analysis*, O'Reilly.

Appendix: Slide Deck Content (Reference)

The material below is a reference copy of the slide deck content. Exercise solutions are explained in the main notes where applicable.

Title Slide

Quick Links

Overview Cross-validation Hyper-parameter Tuning Exercises Demo Summary

Agenda

- Overview
- Cross-validation
- Hyper-parameter Tuning
- Exercises
- Demo
- Summary

Learning Outcomes

- Explain train/validation/test split roles
- Describe k-fold cross-validation
- Explain grid search vs random search
- Avoid data leakage using pipelines

Cross-validation: Key Points

- CV estimates generalization more stably than one split
- k-fold repeats train/validate across folds
- Average score guides selection

Hyper-parameter Tuning: Key Points

- Grid search tries all combos
- Random search samples combos efficiently
- Never tune on the test set

Exercise 1: Grid size

3 parameters with 4 values each: how many combinations?

Solution 1

- $4^3 = 64$

Exercise 2: Leakage

Is scaling on full dataset before split leakage?

Solution 2

- Yes; fit preprocessing on training only.

Exercise 3: Why CV

Why is a single train-test split misleading sometimes?

Solution 3

- Performance depends on split; CV reduces variance.

Mini Demo (Python)

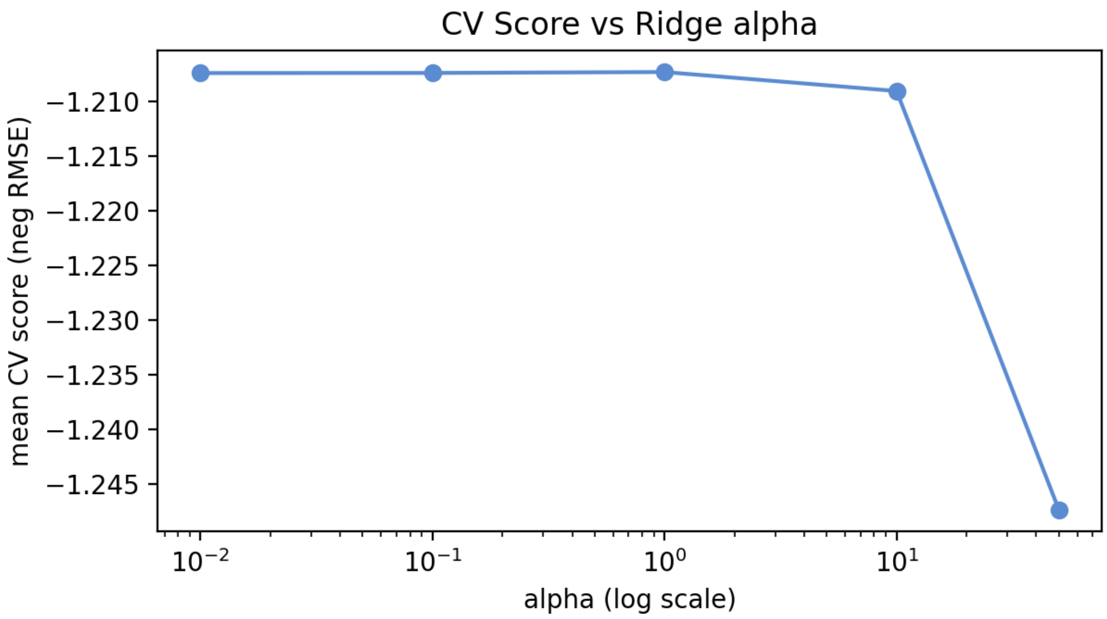
Run from the lecture folder:

```
python demo/demo.py
```

Outputs:

- `images/demo.png`
- `data/results.txt`

Demo Output (Example)



Summary

- Key definitions and the main formula.
- How to interpret results in context.
- How the demo connects to the theory.

Exit Question

Why must you never use the test set to choose hyperparameters?