Statistics and Data Analysis
Unit 05 – Lecture 06 Notes
Advanced Feature Engineering for Multivariate Data

Tofik Ali

February 17, 2026

Topic

Interactions, aggregations, time features, and leakage avoidance.

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

- Interactions
- Aggregations
- Exercises
- Demo
- Summary

Learning Outcomes

- Create interaction features when meaningful
- Create aggregation features from transactional data
- Engineer time-based features (lags/rolling)
- Avoid leakage and look-ahead bias

Why these outcomes matter. **Bias** is a systematic error: your method tends to be wrong in the same direction again and again (too high or too low). It does not disappear by taking more samples if the sampling process is flawed. Fixing bias usually requires changing the data collection procedure (sampling frame, selection method, non-response handling). **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.

Interactions: Key Points

- Products and ratios capture combined effects
- Use domain knowledge
- Validate with CV

Aggregations: Key Points

- Per-user totals/means/counts
- Rolling windows (last 7/30 days)
- Avoid using future data

Exercises (with Solutions)

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

Exercise 1: Interaction

Give one interaction feature for house price.

Solution

- size_m2 * location_score (example).

Exercise 2: Aggregation

Name one per-user aggregation for churn prediction.

Solution

- days_since_last_purchase (example).

Exercise 3: Leakage

Is using next-30-days spend to predict churn today leakage?

Solution

- Yes; it uses future info.

Walkthrough. 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.

Mini Demo (Python)

Run from the lecture folder:

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

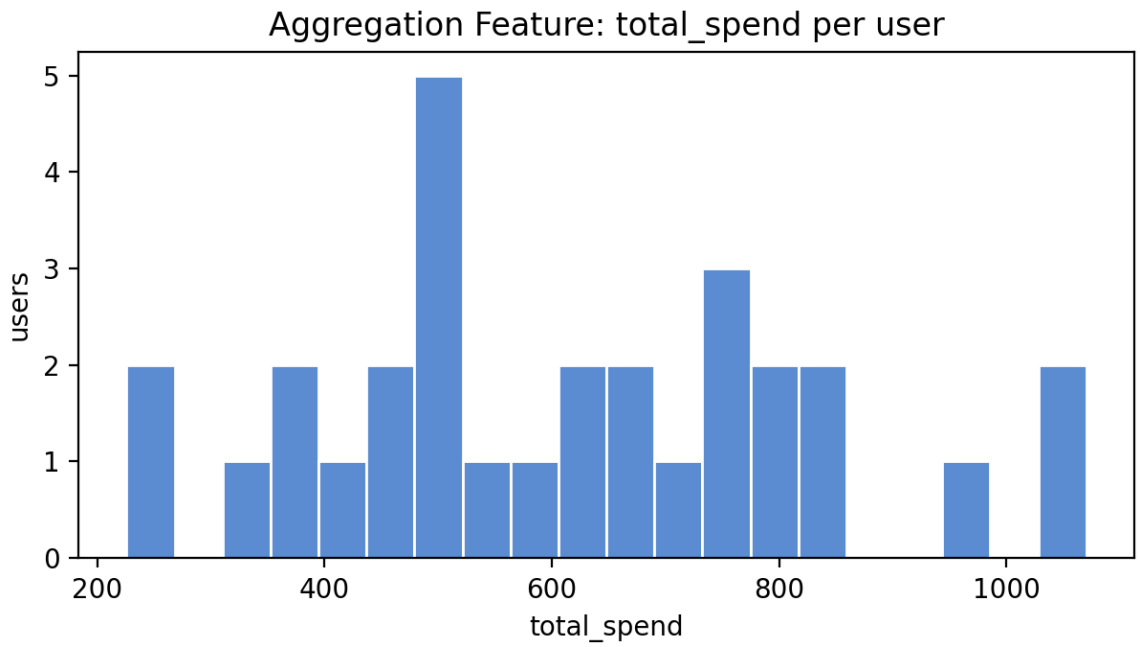
Output files:

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

What to show and say.

- Creates interaction and aggregation features from a toy transactional dataset.
- Shows how leakage can happen if you use future information in features.
- Use it to emphasize validation and time-aware splits when needed.

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

How does cross-validation help detect whether engineered features overfit?

Suggested answer (for revision). Cross-validation (and proper time splits) reveal whether engineered features generalize; if score drops on validation, features may overfit/leak.

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

https://github.com/tali7c/Statistics-and-Data-Analysis

Quick Links

Overview Interactions Aggregations Exercises Demo Summary

Agenda

- Overview
- Interactions
- Aggregations
- Exercises
- Demo
- Summary

Learning Outcomes

- Create interaction features when meaningful
- Create aggregation features from transactional data
- Engineer time-based features (lags/rolling)
- Avoid leakage and look-ahead bias

Interactions: Key Points

- Products and ratios capture combined effects
- Use domain knowledge
- Validate with CV

Aggregations: Key Points

- Per-user totals/means/counts
- Rolling windows (last 7/30 days)
- Avoid using future data

Exercise 1: Interaction

Give one interaction feature for house price.

Solution 1

- size_m2 * location_score (example).

Exercise 2: Aggregation

Name one per-user aggregation for churn prediction.

Solution 2

- days_since_last_purchase (example).

Exercise 3: Leakage

Is using next-30-days spend to predict churn today leakage?

Solution 3

- Yes; it uses future info.

Mini Demo (Python)

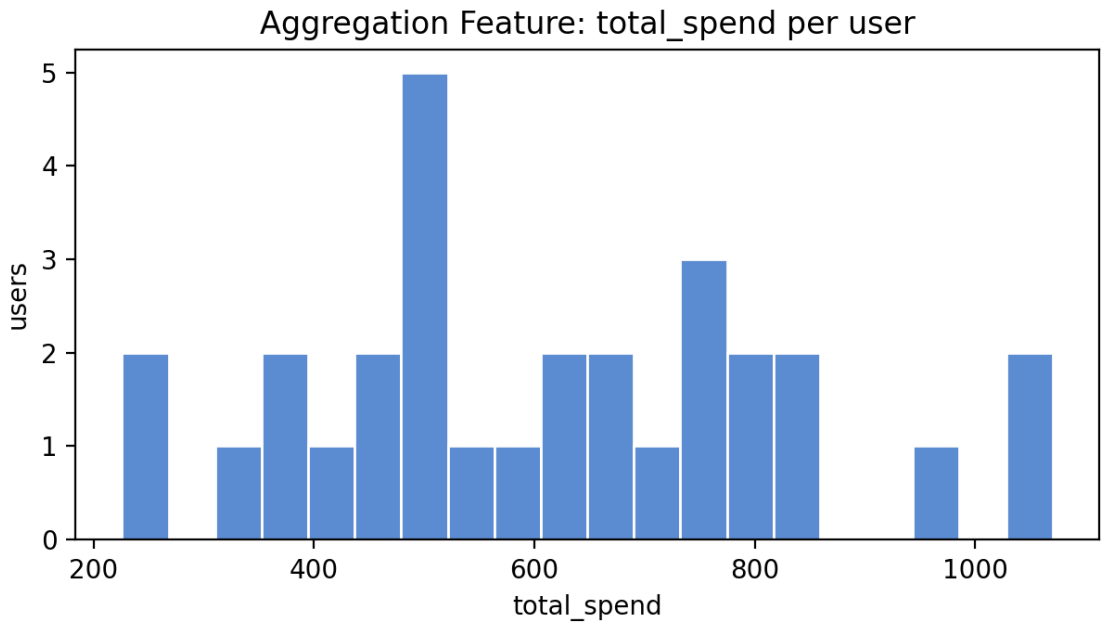
Run from the lecture folder:

`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

How does cross-validation help detect whether engineered features overfit?