**PMAI LAB**

Mathematics • Python • Artificial Intelligence

**LINEAR ALGEBRA FOR MACHINE LEARNING**

# Interactive Case Study 01

## Retail Sales Planning with Least Squares and Ridge Regression

**Your role** Act as a retail analytics team. Build a stable sales model, test it on future weeks, and recommend a promotion plan to management.


**Dr. Hira Benish**

Associate Professor of Mathematics


Student Case-Study Handout | Suggested time: 90–120 minutes

# 1 Client Brief and Learning Challenge

A regional retail outlet plans weekly advertising, discount offers, and promotional events. Management wants to estimate weekly sales and decide whether a proposed campaign is financially reasonable. The available features are correlated, so a model may predict well while its individual coefficients remain unstable.

The dataset is a realistic classroom simulation and does not represent the records of a specific company.

**Learning Outcomes**

By completing this case study, students will be able to:

- represent business observations as a feature matrix and target vector;
- identify scale differences and multicollinearity;
- compare least squares with ridge regression;
- use condition number and coefficient norms as diagnostics;
- evaluate models on later, unseen weeks;
- test a proposed business scenario using Python; and
- communicate a recommendation with limitations.

## 1.1 Management Decision

For the next campaign, management proposes:

$$\text{advertising} = 55, quad\text{footfall} = 320, quad\text{discount} = 15, quad\text{event} = 1.$$

Advertising and sales are measured in thousands of PKR, footfall is the weekly customer count, discount is a percentage, and event is a binary indicator.

**Practice Tasks**

Decide whether the campaign is likely to reach a sales target of 1,100 thousand PKR. Your recommendation must include a predicted value, model choice, validation evidence, and at least two cautions.

## 1.2 Team Roles

| **Role** | Responsibility |
|---|---|
| **Matrix analyst** | Define $X$, **y**, shapes, and preprocessing |
| **Numerical analyst** | Examine rank, correlation, and conditioning |
| **Model evaluator** | Compare test errors and model stability |
| **Business lead** | Convert the evidence into a concise recommendation |

## 2 Dataset and Train-Test Design

Use Weeks 1–9 for training and Weeks 10–12 for testing. The time-ordered split imitates the real decision: use past weeks to estimate later sales.

| Week | Ad spend (000 PKR) | Footfall (customers) | Discount (%) | Event (0/1) | Sales (000 PKR) |
|---|---|---|---|---|---|
| 1 | 18 | 160 | 4 | 0 | 578 |
| 2 | 22 | 175 | 5 | 1 | 688 |
| 3 | 25 | 185 | 7 | 0 | 681 |
| 4 | 28 | 205 | 6 | 0 | 705 |
| 5 | 32 | 220 | 8 | 1 | 819 |
| 6 | 35 | 235 | 10 | 0 | 807 |
| 7 | 39 | 255 | 9 | 1 | 903 |
| 8 | 43 | 270 | 12 | 0 | 903 |
| 9 | 47 | 290 | 11 | 1 | 1004 |
| 10 | 52 | 310 | 14 | 0 | 1017 |
| 11 | 57 | 330 | 16 | 1 | 1136 |
| 12 | 62 | 350 | 15 | 0 | 1118 |

### 2.1 Matrix Representation

For all 12 weeks,

$$X \in \mathbb{R}^{12\times 4}, \qquad \mathbf{y} \in \mathbb{R}^{12}.$$

After adding an intercept column,

$$A = \begin{bmatrix} \mathbf{1} & X \end{bmatrix} \in \mathbb{R}^{12\times 5}.$$

For example, Week 5 is represented by

$$\mathbf{a}_5^T = \begin{bmatrix} 1 & 32 & 220 & 8 & 1 \end{bmatrix}, \qquad y_5 = 819.$$

**Why a Time Split?**

A random split mixes early and later business conditions. A time-ordered split gives a more realistic test of forecasting future weeks, although three test observations still provide only limited evidence.

The features do not prove causes. Advertising, footfall, discounts, events, seasonality, and external conditions may influence one another.

# 3 Interactive Investigation: Before Modeling

## 3.1 Individual Prediction - 5 Minutes

Without using Python, answer the following:

1. Which variables should have positive coefficients? Explain your expectation.
2. Which two features appear most strongly related to one another?
3. Why might a one-unit coefficient be difficult to compare across these columns?
4. Is the next campaign an interpolation or an extrapolation relative to the training weeks?

## 3.2 Pair Activity - Shape and Scale Audit

Complete the following before running a model:

| **Check** | Your answer |
|---|---|
| **Shape of $X_{\text{train}}$** | |
| **Shape after adding intercept** | |
| **Largest raw feature scale** | |
| **Likely correlated feature pair** | |
| **Test target values** | |

## 3.3 Standardization

For feature $j$, estimate the training mean $\mu_j$ and standard deviation $s_j$, then compute

$$z_{ij} = \frac{x_{ij} - \mu_j}{s_j}.$$

Use the same training values $\mu_j$ and $s_j$ to transform the test weeks and the proposed campaign.

**Data-Leakage Pause**

Why is it incorrect to calculate the mean and standard deviation using all 12 weeks? Discuss with a partner and write a one-sentence explanation before continuing.

## 3.4 Numerical Questions

Use a correlation matrix and condition number to investigate:

$$\kappa_2(A) = \frac{\sigma_{\text{max}}(A)}{\sigma_{\text{min}}(A)}.$$

- Does standardization reduce conditioning problems without completely removing multicollinearity?
- How may coefficients behave when two predictors carry nearly the same information?

# 4 Interactive Modeling and Decision Table

## 4.1 Model A: Least Squares

Fit

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|A\boldsymbol{\beta} - \mathbf{y}\|_2^2$$

using `np.linalg.lstsq`. Do not compute an explicit inverse.

## 4.2 Model B: Ridge Regression

For $\lambda > 0$, solve

$$(A^T A + \lambda P)\widehat{\boldsymbol{\beta}}_\lambda = A^T \mathbf{y},$$

where $P = \text{diag}(0, 1, 1, 1, 1)$ so the intercept is not penalized.

## 4.3 Team Experiment

Test $\lambda \in \{0, 0.1, 1, 10, 100\}$. Record the test RMSE, slope-vector norm, and next-campaign prediction.

| $\lambda$ | Test RMSE | $\|\boldsymbol{\beta}_{1:}\|_2$ | Campaign prediction |
|---|---|---|---|
| 0 | ______ | ______ | ______ |
| 0.1 | ______ | ______ | ______ |
| 1 | ______ | ______ | ______ |
| 10 | ______ | ______ | ______ |
| 100 | ______ | ______ | ______ |

**Practice Tasks**

Select one model. Your group must defend the choice using all three criteria:

1. unseen-week prediction error;
2. coefficient stability or size; and
3. usefulness for the management decision.

## 4.4 What-If Challenge

Change the proposed footfall from 320 to 280 while holding other inputs fixed. Predict the new sales value and explain why this is a scenario estimate, not proof of what would happen if footfall were changed.

**Machine-Learning Connection**

The model combines feature geometry, projection, conditioning, regularization, and test-set validation in one business workflow.

# 5 Python Lab

```python
import numpy as np

X = np.array([
    [18,160,4,0], [22,175,5,1], [25,185,7,0],
    [28,205,6,0], [32,220,8,1], [35,235,10,0],
    [39,255,9,1], [43,270,12,0], [47,290,11,1],
    [52,310,14,0], [57,330,16,1], [62,350,15,0]
], dtype=float)
y = np.array([
    578,688,681,705,819,807,903,903,1004,1017,1136,1118
], dtype=float)

X_train, X_test = X[:9], X[9:]
y_train, y_test = y[:9], y[9:]
mean, std = X_train.mean(axis=0), X_train.std(axis=0)
Z_train = (X_train - mean) / std
Z_test = (X_test - mean) / std
A = np.column_stack([np.ones(len(Z_train)), Z_train])
A_test = np.column_stack([np.ones(len(Z_test)), Z_test])

penalty = np.eye(A.shape[1])
penalty[0, 0] = 0.0

def fit_model(lam):
    if lam == 0:
        beta = np.linalg.lstsq(A, y_train, rcond=None)[0]
    else:
        beta = np.linalg.solve(
            A.T @ A + lam * penalty, A.T @ y_train
        )
    prediction = A_test @ beta
    rmse = np.sqrt(np.mean((prediction - y_test) ** 2))
    return beta, rmse

print("Raw condition:", np.linalg.cond(
    np.column_stack([np.ones(9), X_train])))
print("Scaled condition:", np.linalg.cond(A))

for lam in [0, 0.1, 1, 10, 100]:
    beta, rmse = fit_model(lam)
    print(lam, rmse, np.linalg.norm(beta[1:]))

campaign = np.array([[55, 320, 15, 1]], dtype=float)
campaign_A = np.column_stack([
    np.ones(1), (campaign - mean) / std
])
beta, _ = fit_model(0.1)
print("Campaign prediction:", campaign_A @ beta)
```

Add a loop that perturbs one training target by $+10$. Compare how much the least-squares and ridge coefficients change.

# 6 Self-Check Results and Management Decision

Complete the analysis independently before using this page to check your results.

## 6.1 Diagnostic Checkpoints

- Training shape: $X_{\text{train}} \in \mathbb{R}^{9\times4}$ and $A \in \mathbb{R}^{9\times5}$.
- The training correlation between advertising and footfall is approximately $0.999$.
- Raw condition number: approximately $12{,}200.94$.
- Standardized design condition number: approximately $79.18$.

Standardization improves scale balance, but the remaining condition number still reflects strong feature dependence.

## 6.2 Model Comparison

| $\lambda$ | Test RMSE | Slope norm | Interpretation |
|---|---|---|---|
| 0 | 17.28 | 153.43 | Low error, unstable opposing slopes |
| 0.1 | 4.41 | 76.50 | Best test RMSE in this grid |
| 1 | 7.89 | 72.69 | More shrinkage, slightly higher error |
| 10 | 75.41 | 53.97 | Underfits the later weeks |
| 100 | 238.70 | 15.56 | Severe underfitting |

For $\lambda = 0.1$, the proposed campaign prediction is approximately

$$1{,}112.62 \text{ thousand PKR.}$$

**Evidence-Based Recommendation**

The point estimate is about 12.62 thousand PKR above the target. A reasonable classroom recommendation is to proceed cautiously with the $\lambda = 0.1$ model as the best candidate in the tested grid, while reporting that the dataset is small, later weeks extend beyond much of the training range, and the model does not establish causal effects.

## 6.3 Discussion Questions

1. Why does unregularized least squares give large, opposing standardized coefficients for advertising and footfall?
2. Why is the smallest coefficient norm not automatically the best model?
3. Which additional variables would make the business model more credible?
4. What validation design would you use after collecting one year of data?

# 7 Student Submission and Assessment

## 7.1 Required Submission

Submit one notebook and a one-page management brief containing:

1. the matrix definition, feature units, and train-test split;
2. a correlation or condition-number diagnosis;
3. least-squares and ridge results for all five $\lambda$ values;
4. a completed comparison table and campaign prediction;
5. one visual showing actual versus predicted test sales;
6. the final recommendation and at least two limitations; and
7. reproducible code with clear comments.

## 7.2 Assessment Rubric

| Criterion | Marks |
|---|---|
| **Matrix representation and preprocessing** | 20 |
| **Least-squares and ridge implementation** | 20 |
| **Conditioning and multicollinearity analysis** | 20 |
| **Validation, scenario analysis, and interpretation** | 25 |
| **Communication and reproducibility** | 15 |
| **Total** | 100 |

## 7.3 Extension Tasks

- Use rolling-origin validation instead of one fixed test block.
- Add seasonal, competitor-price, and inventory features.
- Compare ridge with a PCA-regression pipeline.
- Create confidence intervals using bootstrap resampling.

**Practice Tasks**

Write a 100-word reflection: Which linear-algebra idea most influenced your final business recommendation, and why?

**Core connection:** Feature matrices $\rightarrow$ conditioning $\rightarrow$ least squares $\rightarrow$ ridge stabilization $\rightarrow$ validated decision.