Lesson 23: Regression Metrics

Data Science with Python – BSc Course

45 Minutes

Learning Objectives

The Problem: We've built a regression model, but how good is it? How do we compare different models objectively?

After this lesson, you will be able to:

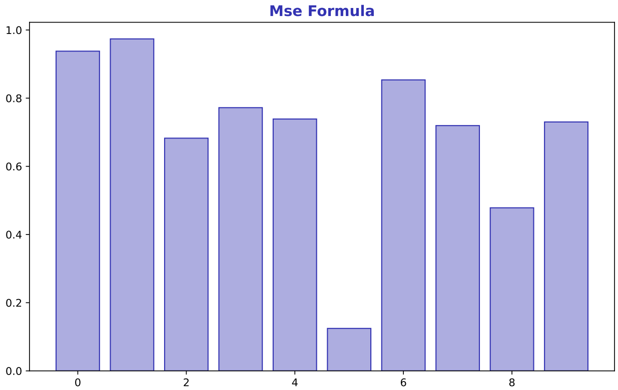
- Calculate MSE, RMSE, and MAE for prediction quality
- Interpret R^2 as variance explained
- Use adjusted R^2 to penalize model complexity
- Apply time series cross-validation for financial data

Finance Application: Evaluating return prediction models before deployment

Mean Squared Error (MSE)

The Foundation Metric

- $\text{MSE} = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$
- Units are squared (e.g., dollars squared) – hard to interpret directly

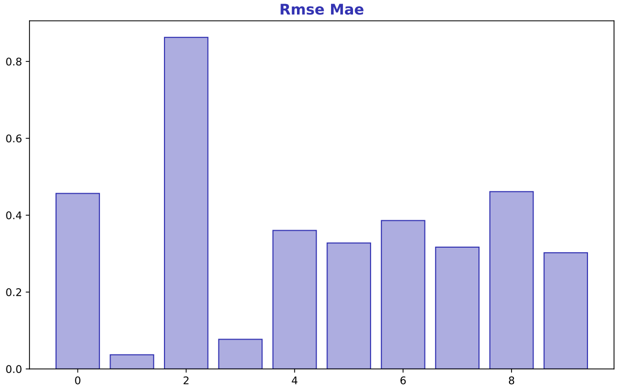


MSE penalizes large errors heavily due to squaring – sensitive to outliers

RMSE and MAE

Interpretable Error Measures

- RMSE $= \sqrt{\text{MSE}}$ – same units as y, penalizes large errors
- MAE $= \frac{1}{n}\sum |y_i - \hat{y}_i|$ – average absolute error, robust to outliers

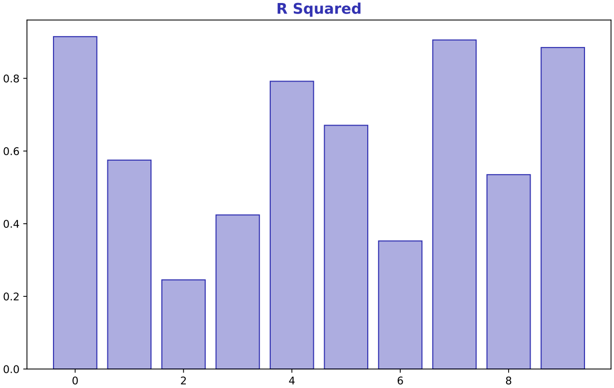


Rule: RMSE $\geq$ MAE always. If RMSE $\gg$ MAE, you have large outliers.

R-Squared (R^2)

Proportion of Variance Explained

- $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$ – ranges from 0 to 1
- $R^2 = 0.7$ means model explains 70% of variance in y

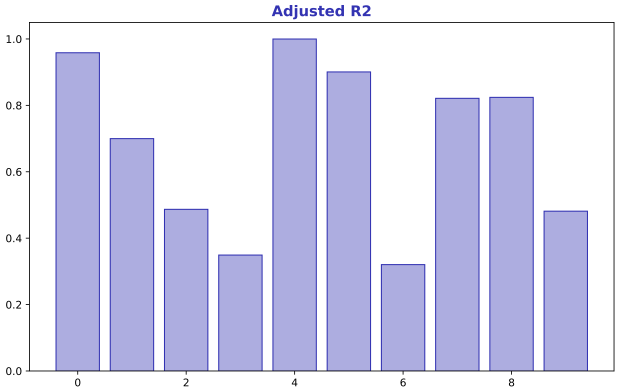


Warning: R^2 **always increases with more features – can be misleading**

Adjusted R^2

Penalizing Model Complexity

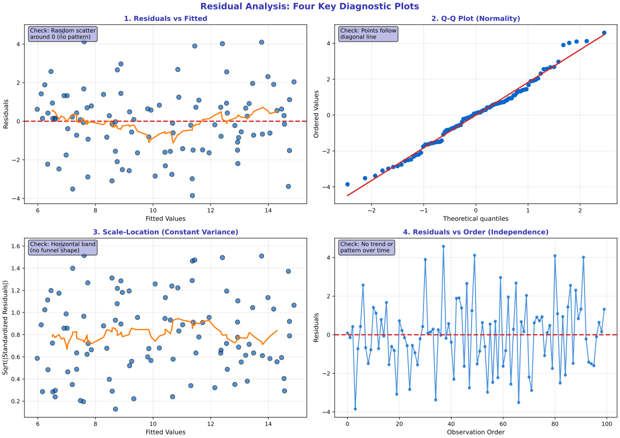
- Adjusted $R^2 = 1 - \frac{(1-R^2)(n-1)}{n-p-1}$ where p = number of features
- Only increases if new feature improves fit more than expected by chance



Use adjusted R^2 when comparing models with different numbers of features

Checking Model Assumptions

- Plot residuals vs fitted values – should show no pattern
- Patterns indicate missing nonlinearity or heteroscedasticity

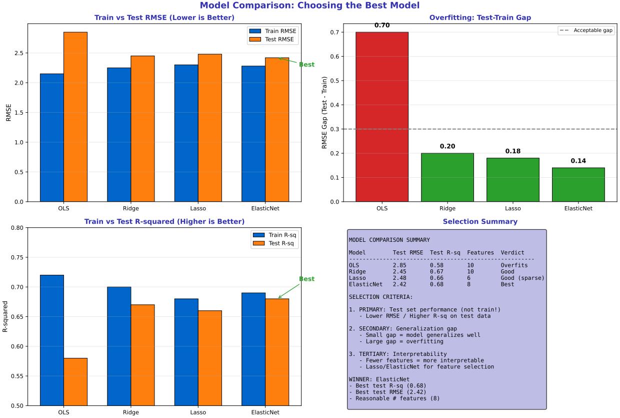


Good practice: Always plot residuals before trusting your model

Model Comparison

Fair Comparison Requires Same Data

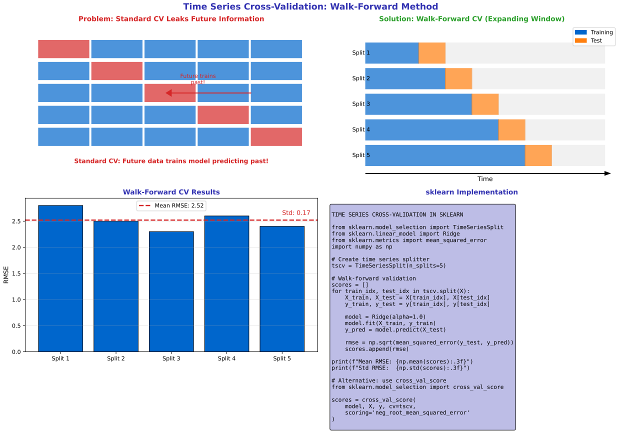
- Compare on held-out test set, not training set
- Use cross-validation for robust comparison



Never compare training R^2 values – always use test set performance

Respecting Temporal Order

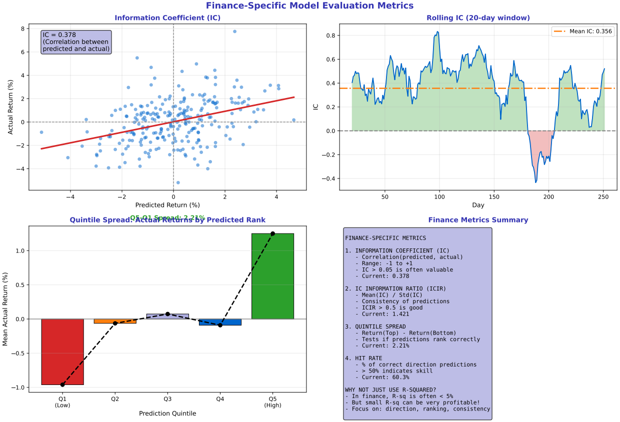
- Standard CV shuffles data – invalid for time series (future leaks into past)
- Rolling window: train on $[t_1, t_n]$, test on $[t_{n+1}, t_{n+k}]$



Finance rule: Never train on data from the future – use TimeSeriesSplit

Beyond Standard Regression Metrics

- Information Coefficient (IC): correlation between predicted and actual returns
- Hit Rate: percentage of correct direction predictions



In trading, predicting direction matters more than magnitude

Hands-On Exercise (25 min)

Task: Evaluate a Stock Return Prediction Model

1. Fit a linear regression predicting next-day returns from lagged features
2. Calculate MSE, RMSE, MAE, and R^2 on a test set
3. Plot residuals vs predicted values – any patterns?
4. Use `TimeSeriesSplit` for proper cross-validation

Deliverable: Summary table of metrics + residual plot.

Extension: Calculate Information Coefficient and compare to R^2

Lesson Summary

Problem Solved: We can now objectively measure and compare regression model quality.

Key Takeaways:

- RMSE/MAE: interpretable error in original units
- R^2: proportion of variance explained (0 to 1)
- Adjusted R^2: penalizes unnecessary complexity
- Time series requires special validation (no data leakage)

Next Lesson: Factor Models (L24) – multi-factor return prediction

Memory: RMSE punishes outliers, MAE treats all errors equally, R^2 is % explained