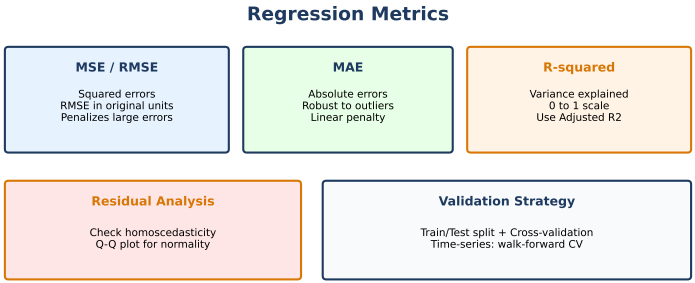
Lesson 23 Summary: Regression Metrics

Data Science with Python – Key Concepts

Data Science Program

Regression Metrics Overview



mean_squared_error(y_true, y_pred) | *r2_score(y_true, y_pred)*

Choose metrics based on business requirements and error tolerance

Mean Squared Error (MSE)

Average of squared residuals:

- **Formula:** $\frac{1}{n}\sum(y_i - \hat{y}_i)^2$
- **Property:** Penalizes large errors heavily
- **Units:** Squared units of target

Use when:
Large errors are much worse than small errors.

MSE is the default optimization target for regression

Root Mean Squared Error (RMSE)

Square root of MSE:

- **Formula:** $\sqrt{MSE}$
- **Advantage:** Same units as target variable
- **Interpretation:** Typical prediction error size

RMSE is the most commonly reported regression metric

Mean Absolute Error (MAE)

Average of absolute residuals:

- **Formula:** $\frac{1}{n} \sum |y_i - \hat{y}_i|$
- **Property:** Robust to outliers
- **Interpretation:** Average deviation

Use when:
All errors matter equally regardless of size.

MAE gives less weight to extreme errors than RMSE

R-squared

Proportion of variance explained:

- **Formula:** $1 - \frac{SS_{res}}{SS_{tot}}$
- **Range:** 0 to 1 (can be negative on test data)
- **Interpretation:** How much better than mean baseline

R-squared of 0.8 means 80% of variance is explained

Adjusted R-squared

Penalizes adding features:

- **Problem:** R-squared always increases with more features
- **Solution:** Adjusted R-squared penalizes complexity
- **Use for:** Comparing models with different feature counts

Prefer Adjusted R-squared for model comparison

Residual Analysis

Visual diagnostics:

- **Residuals vs Fitted:** Check for patterns (should be random)
- **Q-Q Plot:** Check normality of residuals
- **Scale-Location:** Check homoscedasticity

Patterns in residual plots indicate model problems

Cross-Validation

Robust performance estimation:

- **K-Fold:** Split data into K parts, rotate test set
- **Benefit:** Uses all data for training and testing
- **Output:** Mean and std of metric across folds

5-fold or 10-fold CV is standard practice

Time Series Considerations

Walk-forward validation:

- **Problem:** Random splits leak future information
- **Solution:** Always train on past, test on future
- **Method:** TimeSeriesSplit or manual rolling

Never use random train-test split for time series

Quick Reference

Essential Commands:

Metric	Code
MSE	`mean_squared_error(y, pred)`
RMSE	`np.sqrt(mean_squared_error(...))`
MAE	`mean_absolute_error(y, pred)`
R-squared	`r2_score(y, pred)`
Cross-val	`cross_val_score(model, X, y, cv=5)`

Always report test set metrics, not training metrics