Evaluation Metrics for Supervised ML Models

Hands-on: Regression Metrics

Hands-on: Regression Metrics

Learning Objective

By the end of this lesson, students will be able to:

- Compute key regression metrics such as **Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2 (R-squared).**
- Interpret these metrics to assess the performance of a regression model.
- Apply Python's `sklearn.metrics` module to evaluate a machine learning model.

Understanding Regression Metrics

Regression models predict **continuous values** (e.g., house prices, sales forecasts). Different metrics assess different aspects of prediction quality:

Metric	Formula	Best Use Case
Mean Absolute Error (MAE)	$\frac{1}{n}\sum$	Measures average absolute error. Good for easy interpretability.
Mean Squared Error (MSE)	$\frac{1}{n}\sum(y_{\text{true}} - y_{\text{pred}})^2$	Penalizes large errors more than MAE. Useful when large errors must be avoided.
Root Mean Squared Error (RMSE)	$\sqrt{\text{MSE}}$	Similar to MSE, but easier to interpret in the same unit as the target variable.
R² (R-squared, Coefficient of Determination)	$1 - \frac{\sum(y_{\text{true}} - y_{\text{pred}})^2}{\sum(y_{\text{true}} - \bar{y})^2}$	Measures how well the model explains variance in the data. 1 = perfect fit, 0 = poor fit.

Hands-On: Computing Regression Metrics in Python

We will use **scikit-learn** to evaluate a simple regression model.

Step 1: Import Libraries

Copy

```
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

Step 2: Create Sample Predictions

Copy

```
# True values (actual outcomes)
y_true = np.array([3.2, 4.1, 5.6, 6.8, 8.0])
```

```
# Model's predicted values
y_pred = np.array([2.9, 4.5, 5.4, 6.6, 7.9])
```

Step 3: Calculate Key Metrics

Copy

```
mae = mean_absolute_error(y_true, y_pred)
mse = mean_squared_error(y_true, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_true, y_pred)

print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")
```

Try It Yourself!

Modify `y_true` and `y_pred` to see how different predictions affect the metrics.

- What happens when **predictions are closer to actual values**?
- How does **R² change** when predictions deviate more?

Key Takeaways

- **MAE** is more interpretable but **MSE/RMSE** penalize larger errors more.
- **RMSE** is in the same units as the target variable, making it useful for reporting.
- **R²** helps compare models but doesn't always indicate good predictions.

< Previous

© 2025 General Assembly
Attributions

Next >