Evaluation Metrics for Regression in Machine Learning

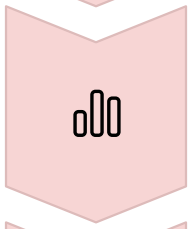
Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.

Muhammad Saeed

Course Overview

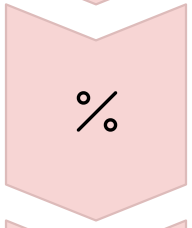
Fundamental Metrics

Mean Absolute Error, Mean Squared Error, Root Mean Squared Error



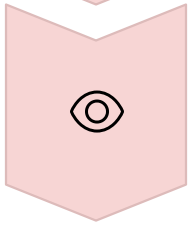
Statistical Metrics

R-squared, Adjusted R-squared, Coefficient of Determination



Percentage-Based Metrics

Mean Absolute Percentage Error, Mean Percentage Error



Visual Evaluation

Residual plots, Q-Q plots, Learning curves

Why Evaluation Metrics Matter

Model Selection

Metrics help us compare different models and select the best one for our specific problem.

Hyperparameter Tuning

They guide the optimization of model parameters to improve performance.

Performance Monitoring

Metrics allow us to track model performance over time and detect degradation.

Business Decision Making

They help translate technical performance into business impact and value.

Mean Absolute Error (MAE)

Formula

$MAE = (1/n) \times \Sigma|y_i - \hat{y}_i|$

Where:

- n = number of observations
- y_i = actual value
- $\hat{y}_i$ = predicted value

Interpretation

MAE measures the average magnitude of errors between predicted and actual values, without considering their direction.

Lower values indicate better model performance.

MAE is in the same units as the target variable, making it easily interpretable.

MAE: Working Principle

Calculate Errors

Compute the difference between each predicted value and the corresponding actual value.

Take Absolute Values

Convert all errors to positive values by taking their absolute values, ensuring all deviations contribute positively to the final metric.

Calculate Average

Sum all absolute errors and divide by the number of observations to get the average error magnitude across all predictions.

MAE: Numerical Example

Actual (y_i)	Predicted ($\hat{y}_i$)	Error ($y_i - \hat{y}_i$)	Absolute Error $\lvert y_i - \hat{y}_i \rvert$
10	12	-2	2
15	13	2	2
8	7	1	1
20	22	-2	2
12	10	2	2

MAE = (2 + 2 + 1 + 2 + 2) / 5 = 9 / 5 = 1.8

This means that, on average, our predictions are off by 1.8 units from the actual values.

MAE: Advantages and Limitations

Advantages

- Intuitive and easy to understand
- Less sensitive to outliers than MSE
- Directly interpretable in the original units
- Consistent with L1 norm optimization

Limitations

- Doesn't penalize large errors as heavily as MSE
- Not differentiable at zero, which can complicate optimization
- May not be ideal when large errors are particularly undesirable
- Doesn't provide direction of errors (positive or negative bias)

Mean Squared Error (MSE)

Formula

$MSE = (1/n) \times \Sigma(y_i - \hat{y}_i)^2$

Where:

- n = number of observations
- y_i = actual value
- $\hat{y}_i$ = predicted value

Interpretation

MSE measures the average of squared differences between predicted and actual values.

Lower values indicate better model performance.

MSE is in squared units of the target variable, making it less directly interpretable than MAE.

It penalizes larger errors more heavily due to the squaring operation.

MSE: Working Principle

Calculate Errors

Compute the difference between each predicted value and the corresponding actual value.

Square the Errors

Square each error value, which both eliminates negative values and gives more weight to larger errors.

Calculate Average

Sum all squared errors and divide by the number of observations to get the average squared error across all predictions.

MSE: Numerical Example

Actual (y_i)	Predicted ($\hat{y}_i$)	Error ($y_i - \hat{y}_i$)	Squared Error $(y_i - \hat{y}_i)^2$
10	12	-2	4
15	13	2	4
8	7	1	1
20	22	-2	4
12	10	2	4

MSE = (4 + 4 + 1 + 4 + 4) / 5 = 17 / 5 = 3.4

This means that, on average, our predictions have a squared error of 3.4 units². Note that this value is not directly interpretable in the original units of the target variable.

MSE: Advantages and Limitations

Advantages

- Penalizes larger errors more heavily
- Mathematically convenient for optimization (differentiable)
- Consistent with maximum likelihood estimation for Gaussian noise
- Widely used in statistical and machine learning applications

Limitations

- Not in the same units as the target variable
- More sensitive to outliers than MAE
- Less intuitive to interpret than MAE
- May overemphasize the impact of outliers in some applications

Root Mean Squared Error (RMSE)

Formula

RMSE = $\sqrt{[(1/n) \times \Sigma(y_i - \hat{y}_i)^2]}$

Or simply:

RMSE = $\sqrt{MSE}$

Where:

- n = number of observations
- y_i = actual value
- $\hat{y}_i$ = predicted value

Interpretation

RMSE is the square root of MSE, bringing the metric back to the same units as the target variable.

Lower values indicate better model performance.

RMSE represents a kind of average error, but with higher weight given to larger errors.

It's more interpretable than MSE while maintaining the property of penalizing large errors more heavily.

RMSE: Working Principle

Calculate MSE

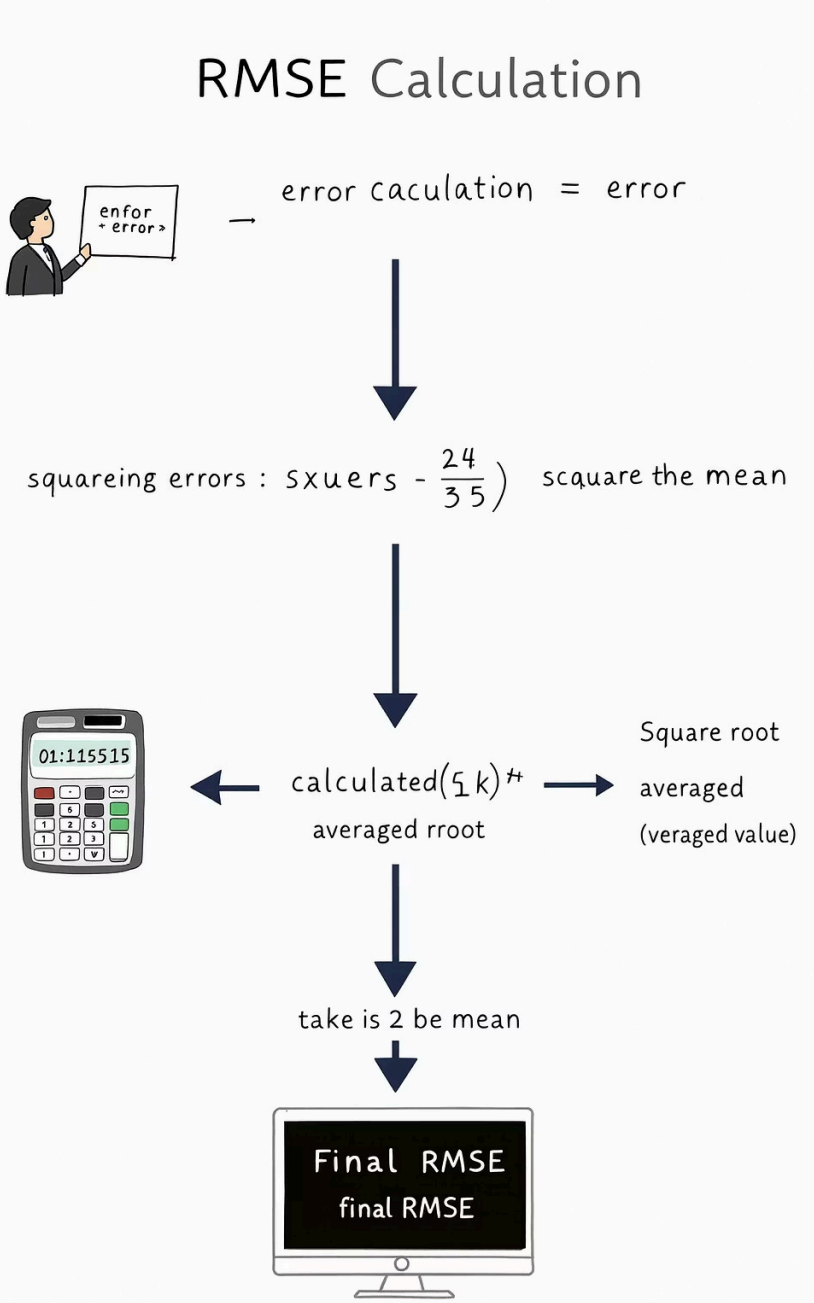
First compute the Mean Squared Error as described previously.

Take Square Root

Apply the square root operation to the MSE value to convert back to the original units of measurement.

Interpret Result

The resulting value represents a kind of "standard deviation of errors," giving more weight to larger errors while maintaining the original units.



RMSE: Numerical Example

Using the same data from our MSE example:

MSE = 3.4

RMSE = √MSE = √3.4 ≈ 1.84

This means that, on average, our predictions have an error of about 1.84 units, with larger errors being penalized more heavily in this calculation.

Note that RMSE (1.84) is slightly larger than MAE (1.8) for this dataset. This is typically the case, as RMSE gives more weight to larger errors.

The difference between RMSE and MAE can also provide insight into the variance of the error distribution. A larger difference suggests more variability in the errors.

RMSE: Advantages and Limitations

Advantages

- Same units as the target variable, making it interpretable
- Penalizes large errors more than MAE
- Widely used and recognized in various fields
- Consistent with standard deviation concept

Limitations

- More sensitive to outliers than MAE
- Not as intuitive as MAE for non-technical stakeholders
- May not be ideal for certain error distributions
- Square root operation can complicate some mathematical analyses

Comparing MAE, MSE, and RMSE

Metric	Formula	Units	Outlier Sensitivity	Interpretability
MAE	$(1/n) \times \Sigma\|y_i - \hat{y}_i\|$	Same as target	Less sensitive	High
MSE	$(1/n) \times \Sigma(y_i - \hat{y}_i)^2$	Squared target	More sensitive	Low
RMSE	$\sqrt{[(1/n) \times \Sigma(y_i - \hat{y}_i)^2]}$	Same as target	More sensitive	Medium

When choosing between these metrics, consider your specific application needs. Use MAE when all errors should be treated equally, and RMSE when larger errors should be penalized more heavily. MSE is often used during model training due to its mathematical properties.

R-squared (Coefficient of Determination)

Formula

$R^2 = 1 - (SSres / SStot)$

Where:

- $SSres = \Sigma(y_i - \hat{y}_i)^2$ (Sum of squared residuals)
- $SStot = \Sigma(y_i - \bar{y})^2$ (Total sum of squares)
- $\bar{y}$ = mean of actual values

Interpretation

R^2 represents the proportion of variance in the dependent variable that is predictable from the independent variables.

It ranges from 0 to 1 (or 0% to 100%), where:

- $R^2 = 1$: Perfect fit, all variance explained
- $R^2 = 0$: Model doesn't explain any variance
- $R^2 < 0$: Can occur with poor models (worse than using the mean)

R-squared: Working Principle

Calculate Total Variance

Compute the total sum of squares (SStot), which represents how much the actual values vary around their mean.

Calculate Unexplained Variance

Compute the residual sum of squares (SSres), which represents the variance not explained by the model.

Calculate Proportion Explained

Determine what proportion of the total variance is explained by the model by comparing SSres to SStot.

R-squared: Numerical Example

Actual (y_i)	Predicted ($\hat{y}_i$)	Mean ($\bar{y}$)	$(y_i - \hat{y}_i)^2$	$(y_i - \bar{y})^2$
10	12	13	4	9
15	13	13	4	4
8	7	13	1	25
20	22	13	4	49
12	10	13	4	1

$SS_{res} = 4 + 4 + 1 + 4 + 4 = 17$

$SS_{tot} = 9 + 4 + 25 + 49 + 1 = 88$

$R^2 = 1 - (17/88) = 1 - 0.193 = 0.807$ or 80.7%

This means our model explains about 80.7% of the variance in the target variable.

R-squared: Advantages and Limitations

Advantages

- Scale-independent (always between 0 and 1)
- Intuitive interpretation as "percentage of variance explained"
- Allows comparison across different models and datasets
- Widely recognized and used in various fields

Limitations

- Always increases when more variables are added (never decreases)
- Can be misleadingly high with overfitted models
- Doesn't indicate whether coefficients and predictions are biased
- Not suitable for comparing models with different target variables

Adjusted R-squared

Formula

Adjusted $R^2 = 1 - [(1 - R^2)(n - 1) / (n - p - 1)]$

Where:

- R^2 = standard R-squared value
- n = number of observations
- p = number of predictors (independent variables)

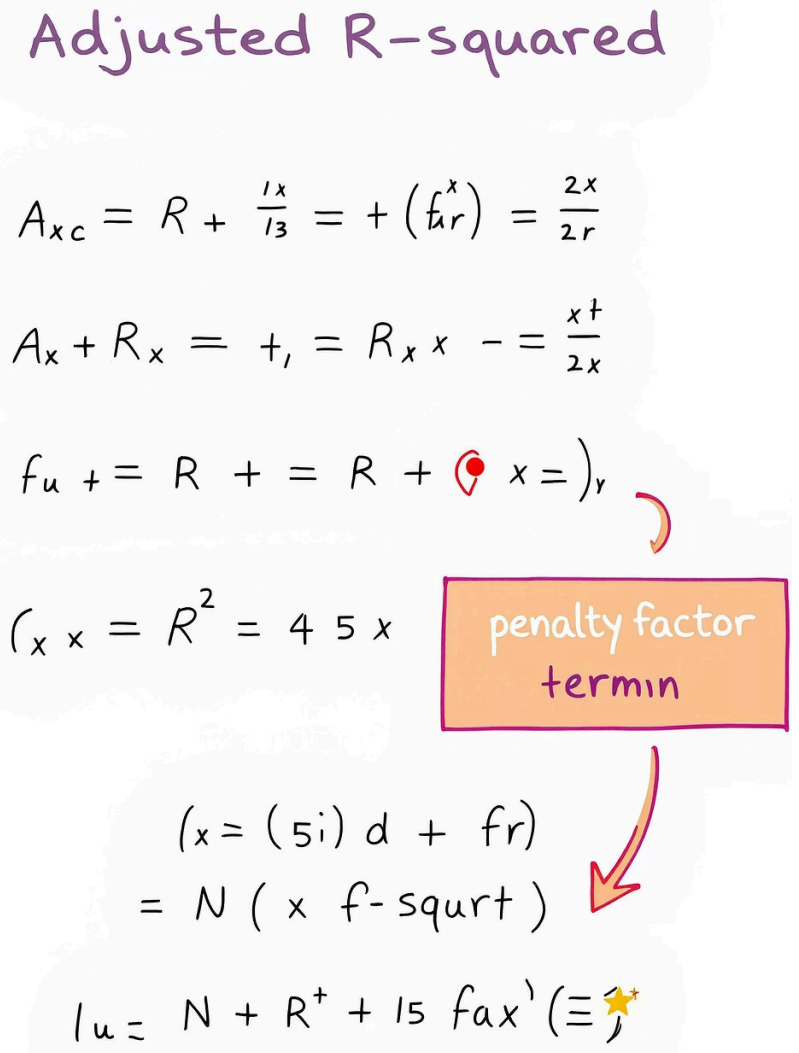
Interpretation

Adjusted R^2 modifies the standard R^2 to account for the number of predictors in the model.

It penalizes the addition of predictors that don't improve the model significantly.

Like R^2, it ranges from 0 to 1 (or can be negative for very poor models).

Higher values indicate better fit, adjusted for model complexity.



Adjusted R-squared: Working Principle

Calculate Standard R^2

First compute the regular R-squared value as described previously.

Apply Complexity Penalty

Adjust the R^2 value by applying a penalty based on the number of predictors relative to the sample size.

Interpret Result

The resulting value represents the proportion of variance explained, adjusted for model complexity, allowing fairer comparison between models with different numbers of predictors.

Adjusted R-squared: Numerical Example

Using our previous example:

$R^2 = 0.807$

n = 5 (observations)

p = 1 (predictors)

Adjusted $R^2 = 1 - [(1 - 0.807)(5 - 1) / (5 - 1 - 1)]$

$= 1 - [(0.193)(4) / 3]$

$= 1 - [0.772 / 3]$

$= 1 - 0.257$

$= 0.743$ or 74.3%

Note that the Adjusted R^2 (74.3%) is lower than the standard R^2 (80.7%) because it accounts for the number of predictors relative to the sample size.

If we were to compare this model with another model that has more predictors, the Adjusted R^2 would provide a fairer comparison by penalizing unnecessary complexity.

Adjusted R-squared: Advantages and Limitations

Advantages

- Penalizes unnecessary complexity
- Better for comparing models with different numbers of predictors
- Helps prevent overfitting by discouraging addition of irrelevant variables
- More realistic assessment of model performance than standard R^2

Limitations

- Still doesn't guarantee the best model for prediction
- Can still increase with irrelevant predictors if they happen to fit noise in small samples
- Doesn't address multicollinearity issues
- May still be misleading if model assumptions are violated

Mean Absolute Percentage Error (MAPE)

Formula

MAPE = $(1/n) \times \Sigma|(y_i - \hat{y}_i) / y_i| \times 100\%$

Where:

- n = number of observations
- y_i = actual value
- $\hat{y}_i$ = predicted value

Interpretation

MAPE expresses accuracy as a percentage of the error.

Lower values indicate better model performance.

It's scale-independent, making it useful for comparing across different datasets.

A MAPE of 10% means that, on average, the predictions are off by 10% of the actual values.

MAPE: Working Principle

Calculate Percentage Errors

For each observation, compute the absolute difference between actual and predicted values, then divide by the actual value to get a percentage.

Take Absolute Values

Ensure all percentage errors are positive by taking their absolute values.

Calculate Average

Sum all percentage errors and divide by the number of observations to get the mean absolute percentage error.

MAPE: Numerical Example

Actual (y_i)	Predicted ($\hat{y}_i$)	Error ($y_i - \hat{y}_i$)	$\lvert(y_i - \hat{y}_i) / y_i\rvert$	$\lvert(y_i - \hat{y}_i) / y_i\rvert \times 100\%$
10	12	-2	0.2	20%
15	13	2	0.133	13.3%
8	7	1	0.125	12.5%
20	22	-2	0.1	10%
12	10	2	0.167	16.7%

MAPE = (20% + 13.3% + 12.5% + 10% + 16.7%) / 5 = 72.5% / 5 = 14.5%

This means that, on average, our predictions are off by 14.5% of the actual values.

MAPE: Advantages and Limitations

Advantages

- Scale-independent, allowing comparison across different datasets
- Easy to interpret as a percentage
- Widely used in business forecasting
- Communicates error in relative terms, which is often more meaningful

Limitations

- Undefined or infinite when actual values are zero
- Puts a heavier penalty on negative errors than positive ones
- Biased towards predictions that are lower than actual values
- Can be heavily influenced by small actual values

Symmetric Mean Absolute Percentage Error (SMAPE)

Formula

SMAPE = $(1/n) \times \Sigma[2 \times |y_i - \hat{y}_i| / (|y_i| + |\hat{y}_i|)] \times 100\%$

Where:

- n = number of observations
- y_i = actual value
- $\hat{y}_i$ = predicted value

Interpretation

SMAPE is a variation of MAPE that treats over-forecasting and under-forecasting more symmetrically.

It ranges from 0% to 200%, with lower values indicating better performance.

A SMAPE of 10% indicates that the average absolute error is 10% of the sum of the absolute actual and predicted values.

SMAPE: Working Principle

Calculate Absolute Errors

Compute the absolute difference between each predicted and actual value.

Normalize by Sum of Absolutes

Divide the absolute error by the sum of the absolute actual and predicted values, then multiply by 2 for scaling.

Calculate Average

Sum all normalized errors and divide by the number of observations to get the symmetric mean absolute percentage error.

SMAPE: Numerical Example

Actual (y_i)	Predicted ($\hat{y}_i$)	$\lvert y_i - \hat{y}_i \rvert$	$\lvert y_i \rvert + \lvert \hat{y}_i \rvert$	$2 \times \lvert y_i - \hat{y}_i \rvert / (\lvert y_i \rvert + \lvert \hat{y}_i \rvert) \times 100\%$
10	12	2	22	18.2%
15	13	2	28	14.3%
8	7	1	15	13.3%
20	22	2	42	9.5%
12	10	2	22	18.2%

SMAPE = (18.2% + 14.3% + 13.3% + 9.5% + 18.2%) / 5 = 73.5% / 5 = 14.7%

This means that, on average, the absolute error is 14.7% of the sum of the absolute actual and predicted values.

SMAPE: Advantages and Limitations

Advantages

- More symmetric treatment of over and under predictions
- Bounded between 0% and 200%
- Less sensitive to outliers than MAPE
- Handles zero or near-zero actual values better than MAPE

Limitations

- Still undefined when both actual and predicted values are zero
- Not as intuitive to interpret as MAPE
- Can still be biased in certain scenarios
- Less commonly used than MAPE in some industries

Residual Analysis

What are Residuals?

Residuals are the differences between observed values and predicted values: $e_i = y_i - \hat{y}_i$

They represent the portion of the dependent variable that the model fails to explain.

Why Analyze Residuals?

- Verify model assumptions (normality, homoscedasticity)
- Identify patterns the model missed
- Detect outliers and influential points
- Check for autocorrelation in time series data

Residual Plots

Residuals vs. Fitted Values

Plots residuals against predicted values to check for non-linearity and heteroscedasticity. Ideally, points should be randomly scattered around zero with no discernible pattern.

Q-Q Plot

Compares the distribution of residuals to a normal distribution. Points should roughly follow a straight line if residuals are normally distributed.

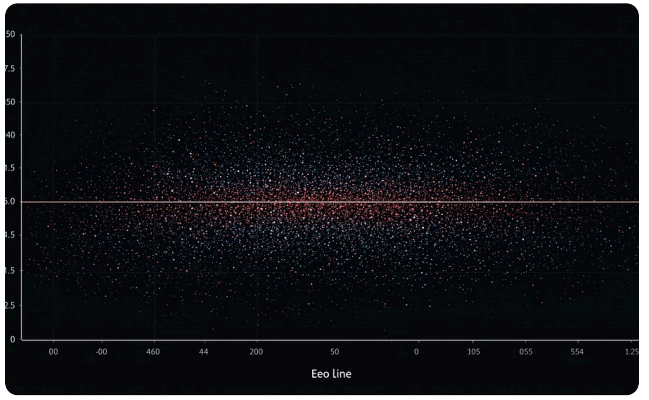
Scale-Location Plot

Plots the square root of standardized residuals against fitted values to check for homoscedasticity. Points should be randomly scattered with a constant spread.

Residuals vs. Leverage

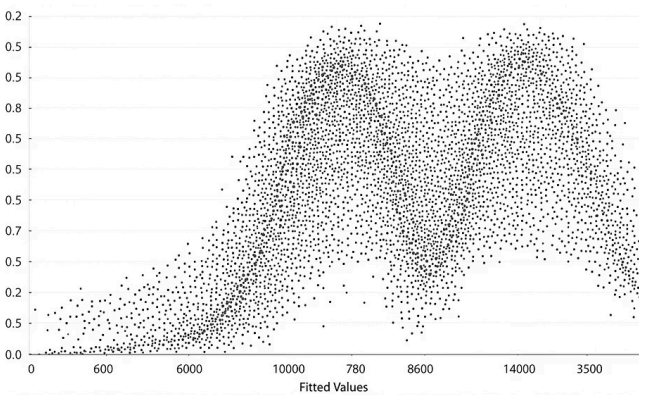
Helps identify influential observations that might disproportionately affect the model. Points outside Cook's distance contours may be influential outliers.

Interpreting Residual Patterns



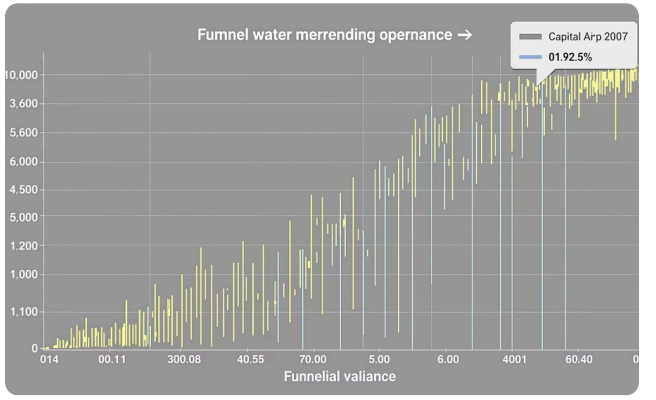
Good Model Fit

Random scatter of residuals around zero with no discernible pattern indicates a well-specified model that captures the underlying relationships in the data.



Non-linearity

U-shaped or inverted U-shaped patterns suggest that a linear model is not appropriate. Consider adding polynomial terms or using non-linear models.



Heteroscedasticity

Fan or funnel shapes indicate that variance of residuals changes with the predicted values. Consider transformations or weighted regression.

Learning Curves

What are Learning Curves?

Learning curves plot model performance (training and validation error) as a function of training set size or training iterations.

They help diagnose bias-variance tradeoff issues and determine if more data would help improve the model.

Interpreting Learning Curves

- High training and validation error: Underfitting (high bias)
- Low training error but high validation error: Overfitting (high variance)
- Converging curves with high error: Need a more complex model
- Converging curves with low error: Well-fitted model
- Gap between curves that doesn't narrow with more data: Consider regularization

Cross-Validation for Regression

Split Data

Divide the dataset into k equal-sized folds (typically 5 or 10).

Train and Evaluate

For each fold, train the model on k-1 folds and evaluate on the remaining fold.

Aggregate Results

Calculate the average performance metric across all k iterations to get a more robust estimate of model performance.

Assess Stability

Examine the variance of performance across folds to assess model stability.

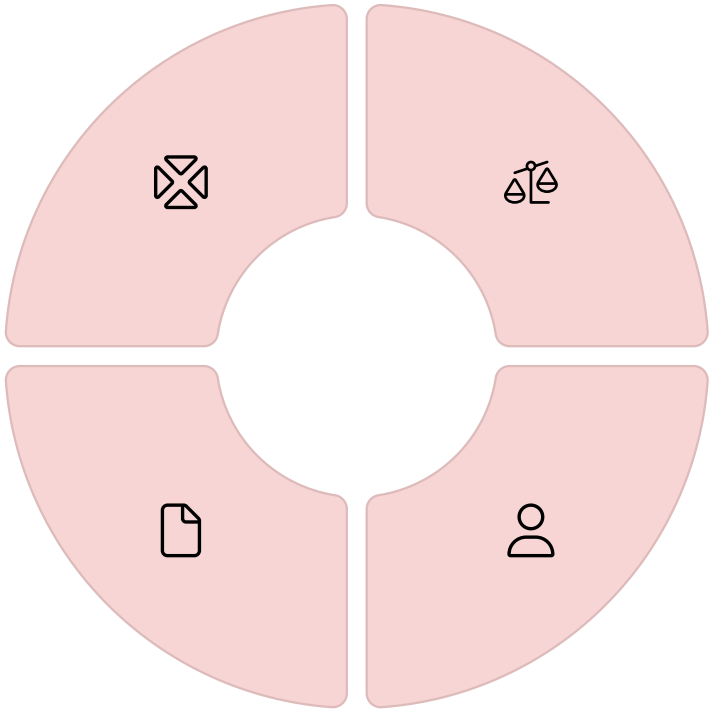
Choosing the Right Metric

Business Context

Consider what errors mean in your specific domain. Are all errors equally important, or are some more costly than others?

Audience

Consider who will be interpreting the results. Some metrics are more intuitive for non-technical stakeholders.



Scale Sensitivity

If comparing models across different scales or units, consider scale-independent metrics like R^2 or percentage errors.

Outlier Sensitivity

If your data contains outliers, consider how different metrics handle them. MAE is less sensitive to outliers than MSE/RMSE.

Practical Implementation

Python Implementation

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

# Calculate metrics
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)

# Calculate MAPE
def mape(y_true, y_pred):
    return np.mean(np.abs((y_true - y_pred) / y_true)) *
100
```

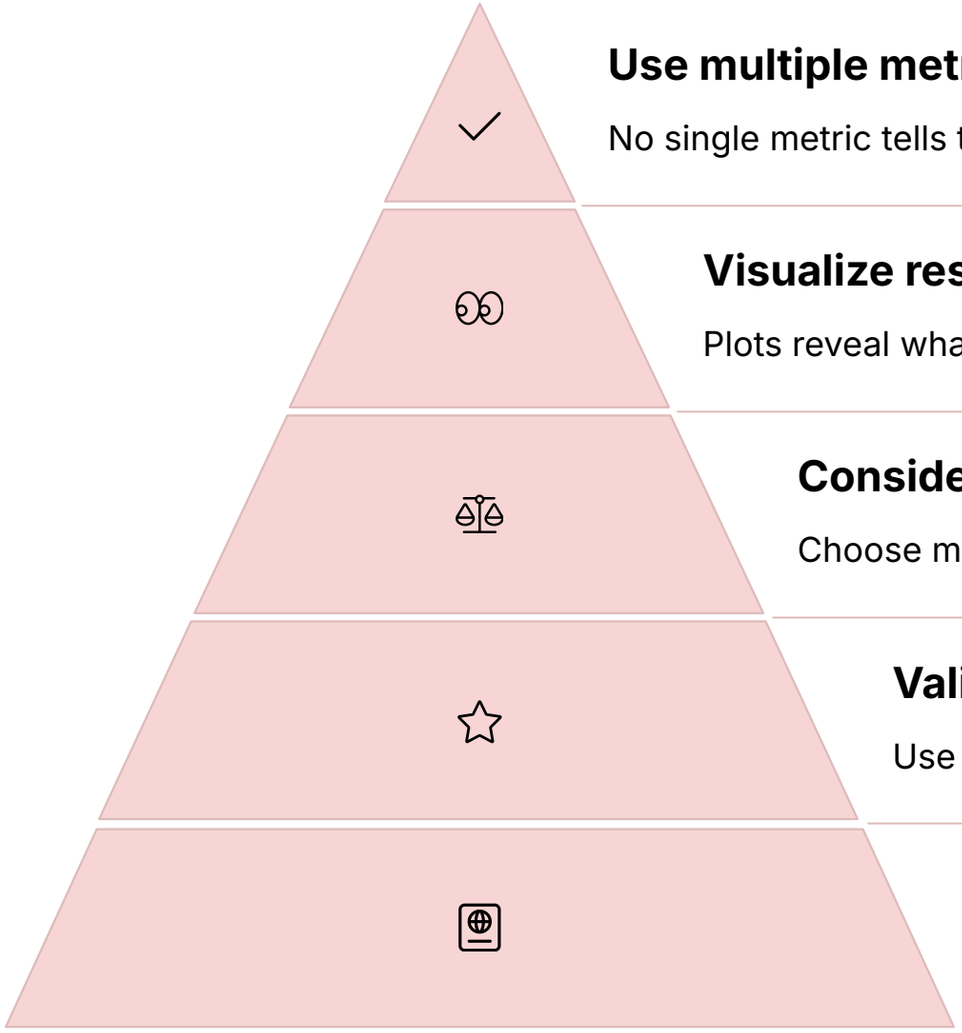
R Implementation

```
# Calculate metrics
mae <- mean(abs(y_true - y_pred))
mse <- mean((y_true - y_pred)^2)
rmse <- sqrt(mse)
r2 <- 1 - sum((y_true - y_pred)^2) /
      sum((y_true - mean(y_true))^2)

# Calculate MAPE
mape <- mean(abs((y_true - y_pred) / y_true)) * 100

# Using built-in functions
library(Metrics)
mae <- mae(y_true, y_pred)
rmse <- rmse(y_true, y_pred)
r2 <- cor(y_true, y_pred)^2
```

Summary and Best Practices



Use multiple metrics

No single metric tells the whole story

Visualize residuals

Plots reveal what summary statistics might miss

Consider context

Choose metrics aligned with business objectives

Validate thoroughly

Use cross-validation for robust performance estimates

Document assumptions

Be transparent about model limitations

Remember that the goal of evaluation is not just to measure performance, but to guide improvement. Use these metrics to identify specific weaknesses in your models and address them systematically. Always consider the practical implications of errors in your specific domain when selecting and interpreting metrics.