Performance Metrics

And their relation to Loss Functions

Loss Functions

- A loss function is used in machine learning to quantify how well a machine learning model is performing.
- It measures the inconsistency between the predicted value and the actual value.
- This function is what the model wants to minimize during training.
- Also known as *Cost Function*

Performance Metrics

- Performance metrics are used to evaluate how well a machine learning model has learned and is predicting.
- They give insight into the quality of the model’s outputs.
- Also known as *Evaluation Metrics*

Performance Metrics

A mathematical function can serve the dual role of a loss function and a performance metric, provided that it meets the necessary conditions for both use-cases.

Performance Metrics

A mathematical function can serve the dual role of a loss function and a performance metric, provided that it meets the necessary conditions for both use-cases.

What are the necessary conditions?

Performance Metrics

- Performance Metric - Interpretable
- Loss Function - The choice of a loss function should be compatible with the optimization algorithm being used.

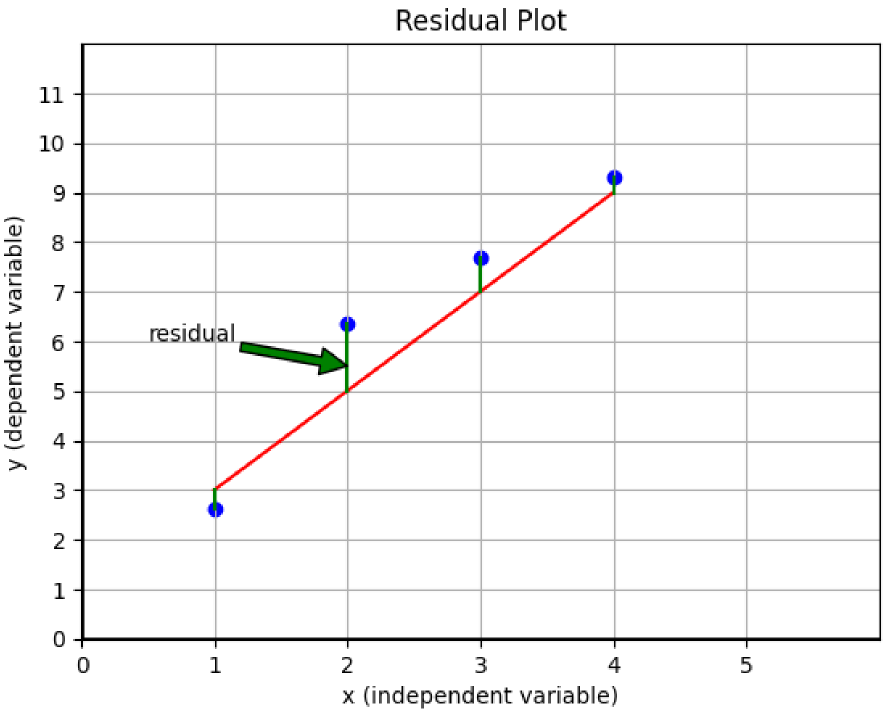
Performance Metric vs. Loss Function: Comparison

- Performance Metric - Can have multiple metrics per a single problem (*this is often the case*)
- Loss function - a single function for a given training process

Performance Metric vs. Loss Function: Comparison

	Usage	Requirements	Number per Problem
Performance Metric	Assess the quality of the model	Interpretable	Multiple per problem
Loss function	Train the model	Has to be suitable for the optimization algorithm	Single per training

Example - Regression



Loss Functions Examples

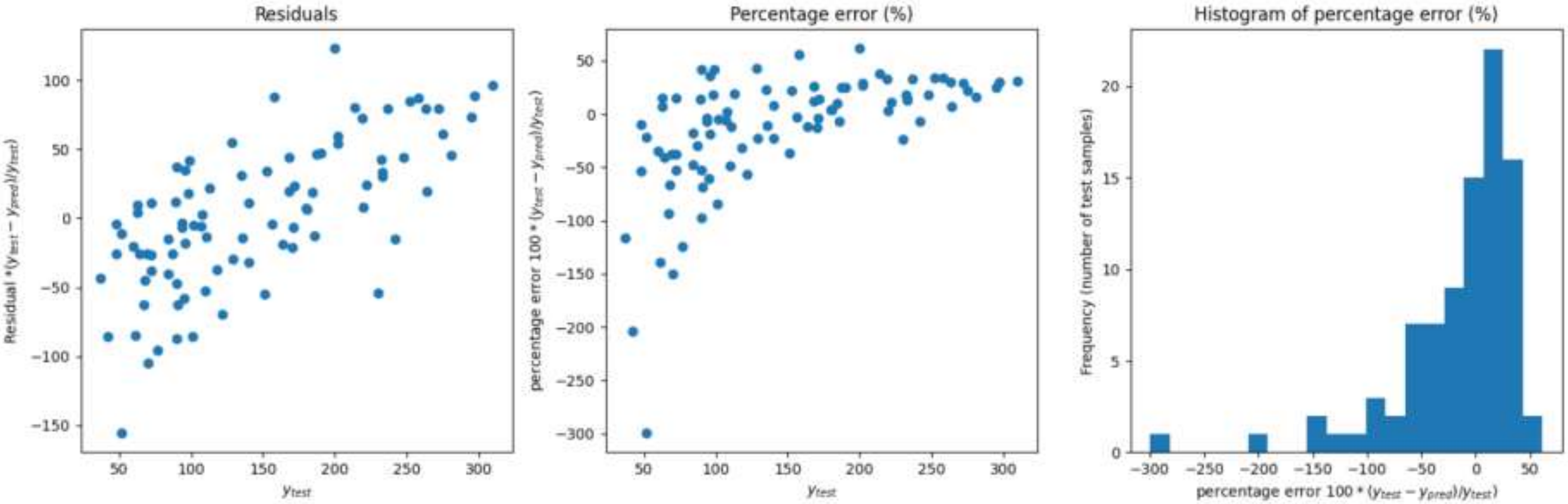
- **Mean Squared Error (MSE)**: This is the average of the squared differences between the predicted and actual values. It's probably the most common loss function for regression. Equation: MSE = 1/n * Σ(actual - prediction)^2
- **Mean Absolute Error (MAE)**: This is the average of the absolute differences between the predicted and actual values. It's less sensitive to outliers than the MSE. Equation: MAE = 1/n * Σ|actual - prediction|
- **Huber Loss**: This is a combination of MSE and MAE. It is quadratic for small error values and linear for large error values. The point where it switches from quadratic to linear is determined by a hyperparameter, delta (δ).

Performance Metrics - Examples

- **Mean Absolute Error (MAE)**: This is the average of the absolute differences between the predicted and actual values. It's less sensitive to outliers than the MSE. Equation: MAE = 1/n * Σ|actual - prediction|
- **Mean Absolute Percentage Error (MAPE)**: This is the average of the absolute percentage differences between the predicted and actual values. Equation: MAPE = 100/n * Σ|(actual - prediction) / actual|
- **R-squared (Coefficient of Determination)**: This measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It's a statistical measure between 0 and 1 which calculates the goodness of fit of the regression model.

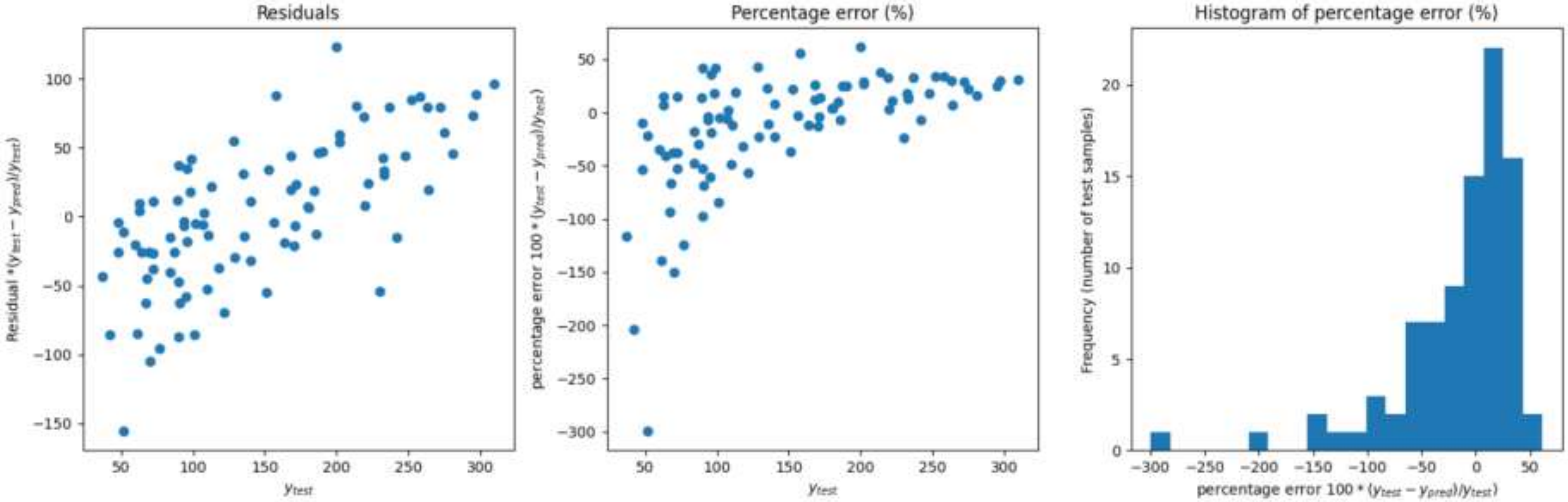
An opportunity to practice model evaluation and reporting:

- **Part 1**: evaluate the model’s predictions from multiple perspectives. Opt for detailed (e.g. single sample) analysis when needed.



An opportunity to practice model evaluation and reporting:

- **Part 1**: evaluate the model’s predictions from multiple perspectives. Opt for detailed (e.g. single sample) analysis when needed.
- **Part 2**: Based on the detailed analysis - Provide high level summary, insights and guidelines to the user/stakeholder:
  - where does the model succeed and when does it fail?
  - what should a user expect in terms of goodness of fit and types of errors (for those cases that matter to the user)?



Methods to gauge the accuracy of a Linear Regression Model

Category	Method	What it tells you	Typical use-notes
Hold-out evaluation (HOA)	**Train / validation / test split**	Performance on data that the model never saw during training.	The simplest guard against overfitting; reserve ≥ 20 % of the data for the final test set.
Cross-validation (CV)	**k-Fold CV** (e.g., k = 5 or 10)	Average error over k different train/test partitions; variance of the scores.	Preferred when the dataset is small; gives a more stable estimate than a single split.
Point-error metrics (same units as target) **(PEM)**	**MAE** (Mean Absolute Error)	Typical absolute dollar error; less sensitive to outliers.	Easy to interpret ("average miss is $X").
	MSE / RMSE (Mean / Root-Mean-Squared Error)	Penalises large errors more than small ones. RMSE is in the same units as the target.	The de-facto benchmark loss for regression, especially if you optimised MSE during training.
	MAPE (Mean Absolute Percentage Error)	Average relative error (%)	Only meaningful when the target is strictly positive and errors scale with magnitude.
Goodness-of-fit (GOF)	**R^2 (coefficient of determination)**	Fraction of variance in yy explained by the model (1 = perfect).	Can be misleading if the relationship is non-linear or if you compare models with different target transformations.
	Adjusted R^2	R^2 corrected for the number of predictors.	Use when comparing models with different numbers of features.
Statistical inference (FS)	**F-statistic & overall p-value**	Whether the regression explains significantly more variance than a constant model.	Available from stats packages (statsmodels, R). Less common in pure ML pipelines, but useful for interpretability.
	t-tests for coefficients	Whether each slope differs significantly from 0 given the sample size and noise.	Helps decide if a feature's influence is statistically meaningful.
Residual diagnostics (RD)	**Residual plot**	Reveals non-linearity, heteroscedasticity, or outliers.	Look for random scatter around 0; patterns imply model mis-specification.
	Q–Q plot / Shapiro–Wilk test	Check normality of residuals (a linear-regression assumption for inference).	Important if you'll compute confidence intervals or prediction intervals.
	Breusch–Pagan / White test	Detect heteroscedasticity (variance of residuals changes with xx).	If present, consider transforming variables or using weighted least squares.
Model-selection criteria (MSC)	**AIC / BIC**	Trade-off between fit quality and model complexity.	Lower values are better; handy when choosing polynomial degree or subset of features.
Learning-curve diagnostics (LCD)	**Train vs. validation error vs. sample size**	Detect high bias (both errors high) vs. high variance (train ≪ val).	Guides decisions on gathering more data, adding regularisation, or simplifying the model.

Strategy	Description	How It Adapts to LLMs
Code Walkthroughs	Students explain their code step-by-step to peers or instructor.	Ensures genuine understanding; LLM-generated code must still be explained in own words.

1. Study the Category/Method you were assigned
2. Apply the method to your Univariate Linear Regression Experiment (Lab 1)
3. Prepare three talking points about the most important sections of your code

When you are ready:

You		Your peer
4a Present the code to your peer using the talking points	4b	Assess the accuracy of the code and the clarity of the walkthrough. Write notes.

4c

5. Be ready to present both your notes and reflections **as pairs**

Any questions?