Features and Labels

Description of features and labels in the context of machine learning.

Component	Description	Role	Example
Features	Individual measurable properties or characteristics of the data used as input to the machine learning model.	Input to the model (independent variables).	For predicting house prices: square footage, number of bedrooms, location.
Label	The value or outcome the model aims to predict or categorize based on the features.	Output that the model predicts (dependent variable).	For predicting house prices: actual house price.

Here is an example of a small dataset for predicting house prices, showing the feature columns and the label.

Area (m²)	Bedrooms	Location	House Age (years)	Price (label)
120 m²	3	Suburban	10	250,000 PLN
150 m²	4	Urban	5	350,000 PLN
80 m²	2	Suburban	20	150,000 PLN
200 m²	5	Urban	2	500,000 PLN
100 m²	3	Rural	30	180,000 PLN

Features:

- **Area (m²)**: Represents the size of the house in square meters.
- **Bedrooms**: The number of bedrooms in the house.
- **Location**: Describes whether the house is located in an urban, suburban, or rural area.
- **House Age (years)**: Indicates the age of the house in years.

Label:

- **Price (label)**: The actual price of the house, which the model aims to predict based on the features.

The above small dataset demonstrates how **features (input)** are used to predict the **label (output)** in machine learning.

Split into Training, Validation, and Test Data

In the context of machine learning, data splitting is crucial for evaluating model performance and preventing overfitting. The data is typically divided into three sets:

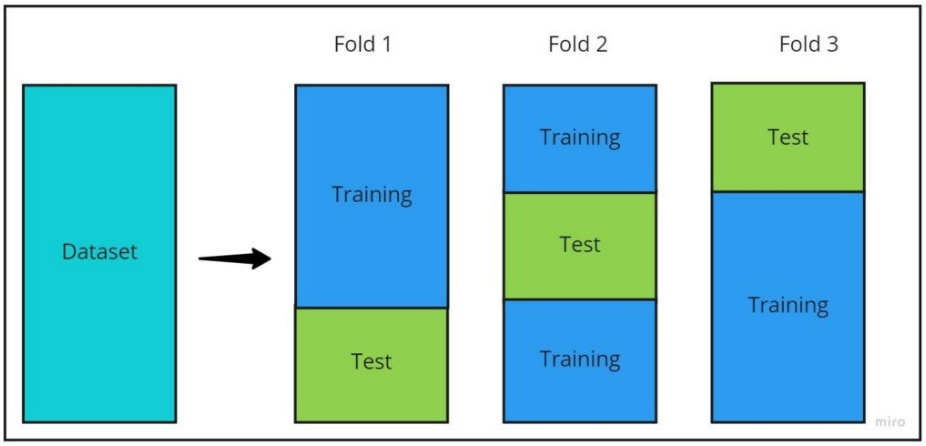
1. **Training set**: This subset is used to train the model. It contains the majority of the data, and the model learns patterns and relationships from it.
2. **Validation set**: This set is used for tuning the model's hyperparameters and making decisions about the model's architecture. It helps evaluate the model's performance during training and can guide adjustments without overfitting on the training data.
3. **Test set**: This final subset is reserved for assessing the model's performance after training and validation. It provides an unbiased evaluation of how the model will perform on unseen data.

This three-way split helps ensure that the model generalizes well to new data. When we say that a model generalizes on new data **(the test data set)**, it means that the model can make accurate predictions on unseen data **(the test data set)** (data it was not trained on – **the train data set**).

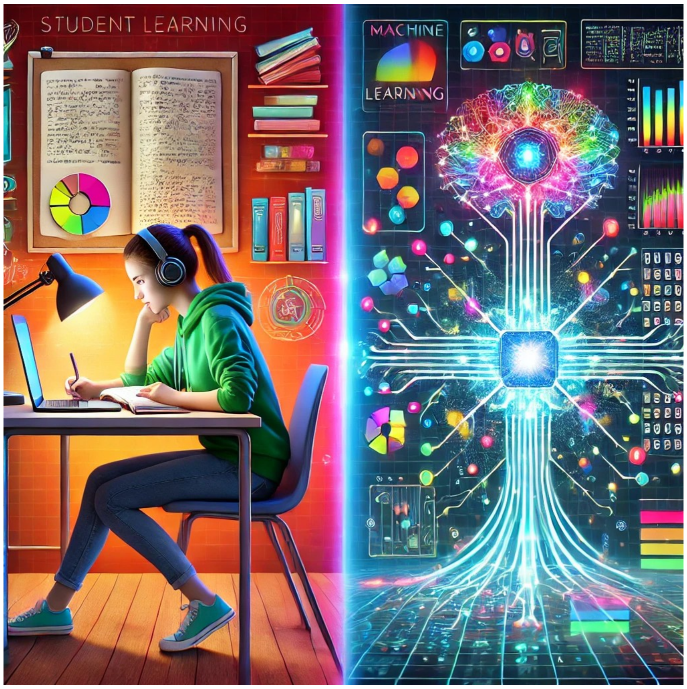
In the context of data splitting there is a technique called Cross-validation

Cross-validation is used to improve the evaluation process and make it more reliable by mitigating issues like overfitting or model variance based on a single train-test split.

Rather than performing a single split of the data, cross-validation splits the data multiple times into different training and testing subsets. The most common approach is k-fold cross-validation, where the data is divided into k equal-sized "folds".



Splitting the datasets works in Machine Learning like Studying for Exam



Machine Learning Model	Student Learning Process
🔢 **Training Phase (Train Set)** → The model learns from labeled data.	📚 **Training Phase (Homework & Practice)** → The student studies by solving practice problems.
📊 **Testing Phase (Test Set)** → The model predicts on unseen data.	📝 **Testing Phase (Final Exam)** → The student takes an exam with unseen questions.
🎯 **Goal:** Learn patterns to make accurate predictions on new data.	🎯 **Goal:** Understand concepts to answer new questions correctly.

⚠️ Common Pitfalls (for both models & students!)

- **Overfitting (Memorization)** → The student memorizes answers but struggles with new questions.
- **Underfitting (Lack of Learning)** → The student doesn’t study enough and performs poorly.
- **Good Generalization** → The student understands the subject and applies knowledge to new problems.