

->Why should we have validation and test dataset along with training dataset. Why can't we stick to just training and testing datasets?

<-A validation set helps tune hyperparameters and prevent overfitting, while the test set evaluates final model performance. Without a validation set, tuning on the test set introduces bias, making it unreliable for evaluation. Using all three datasets ensures effective training, tuning, and unbiased performance assessment on unseen data.

<- ? **Training set:** The model learns.

? **Validation set:** The model is tuned and selected.

? **Test set:** The model is evaluated.

->What happens when your model reaches the maximum accuracy? How would you know it's a good time to stop the training process?

<- When your model reaches maximum accuracy, it means further training is not improving performance and may even cause overfitting. You can stop training when:

- 1.Validation loss stops decreasing** (early stopping).
2. Validation accuracy plateaus** or starts dropping.
3. Training loss continues decreasing** while validation performance worsens.

These signals indicate the model has learned the optimal patterns and continuing training could harm generalization to unseen data.

<- Stop training when validation accuracy plateaus or validation loss stops decreasing, signaling the model has learned the optimal patterns. Continuing after this may lead to overfitting, where the model performs well on training data but poorly on unseen data. Use early stopping to prevent this.

->What is a parameter and a hyper parameter?

<- A **parameter** is a value that the model learns from the training data, like weights in a neural network or coefficients in a regression model. A **hyperparameter** is a setting that you configure before training, such as learning rate, number of layers, or regularization strength, which influences the training process.

->Difference between L2 and L1 regularization?

<- L2 regularization penalizes the square of the coefficients, shrinking them but keeping all features. L1 regularization penalizes the absolute value of the coefficients, driving some to zero, effectively performing feature selection. L2 helps reduce overfitting, while L1 creates a sparse model by eliminating less important features.

Difference between Lasso and Ridge Regression?

- **Lasso (L1 Regularization):** Shrinks coefficients and can set some to zero, performing feature selection.
- **Ridge (L2 Regularization):** Shrinks coefficients but does not eliminate any, keeping all features.

Lasso creates sparse models, while Ridge retains all predictors.

Why would we want to use regularization?

Regularization helps prevent **overfitting**, where a model performs well on training data but poorly on unseen data. By adding a penalty to large coefficients, regularization forces the model to focus on important features and generalize better. It also improves model simplicity, making it more robust to noise and irrelevant data.

What is an epoch?

An **epoch** is one complete pass through the entire training dataset during the training process of a machine learning model. In each epoch, the model updates its parameters based on the data it has seen, allowing it to learn and improve. Multiple epochs are often needed for the model to converge and achieve optimal performance.

->Match each term to a corresponding statement.

Question 6 options:

Lasso regression Ridge regression Produces a model with fewer weights.

Produces a model with smaller weights.

1. L1 regularization

2. L2 regularization

<-☐ Lasso regression: Produces a model with fewer weights. (L1 regularization)

☐ Ridge regression: Produces a model with smaller weights. (L2 regularization)

How would you know when your model has reached its maximum accuracy based on validation data?

Question 4 options:

Validation accuracy begins to trend upwards.

Validation loss begins to trend upwards. (correct)

This cannot be determined from validation data.

Validation loss begins to trend downwards

->Match each choice to a corresponding statement.

Chosen during model construction. (hyperparam)

Too many can lead to overfitting. (hyper)

Updated during the training loop.(param)

It can be chosen to turn on L1 or L2 regularization. (hyper)

1. Parameter

2. Hyperparameter

<-Parameter: Updated during the training loop.

Hyperparameter: Chosen during model construction.

Why should we split data into training, testing, and validation sets? What is the danger of only splitting into training and testing sets?

<-Splitting data into training, testing, and validation sets is essential for effective model development:

- **Training Set:** Used for model training, learning patterns in data.
- **Testing Set:** Evaluates model performance, assessing generalization to unseen data.
- **Validation Set:** Aids in model selection and hyperparameter tuning, preventing overfitting to testing data.

Only splitting into training and testing sets risks overfitting during hyperparameter tuning and model selection, as the testing set may inadvertently influence these decisions.

<- Splitting data into training, validation, and testing sets ensures robust model performance. The training set learns, the validation set tunes hyperparameters and prevents overfitting, and the testing set evaluates generalization. Without a validation set, tuning could bias the test results, risking overfitting and inaccurate performance estimates.