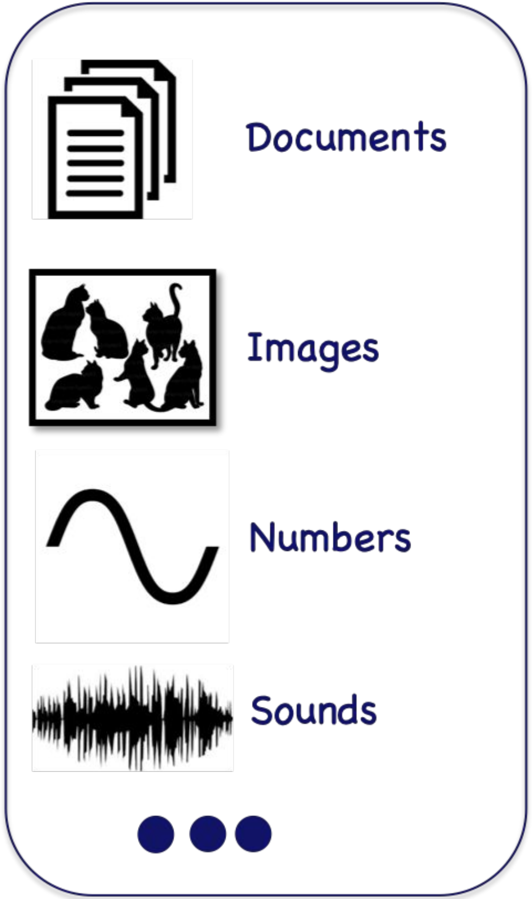
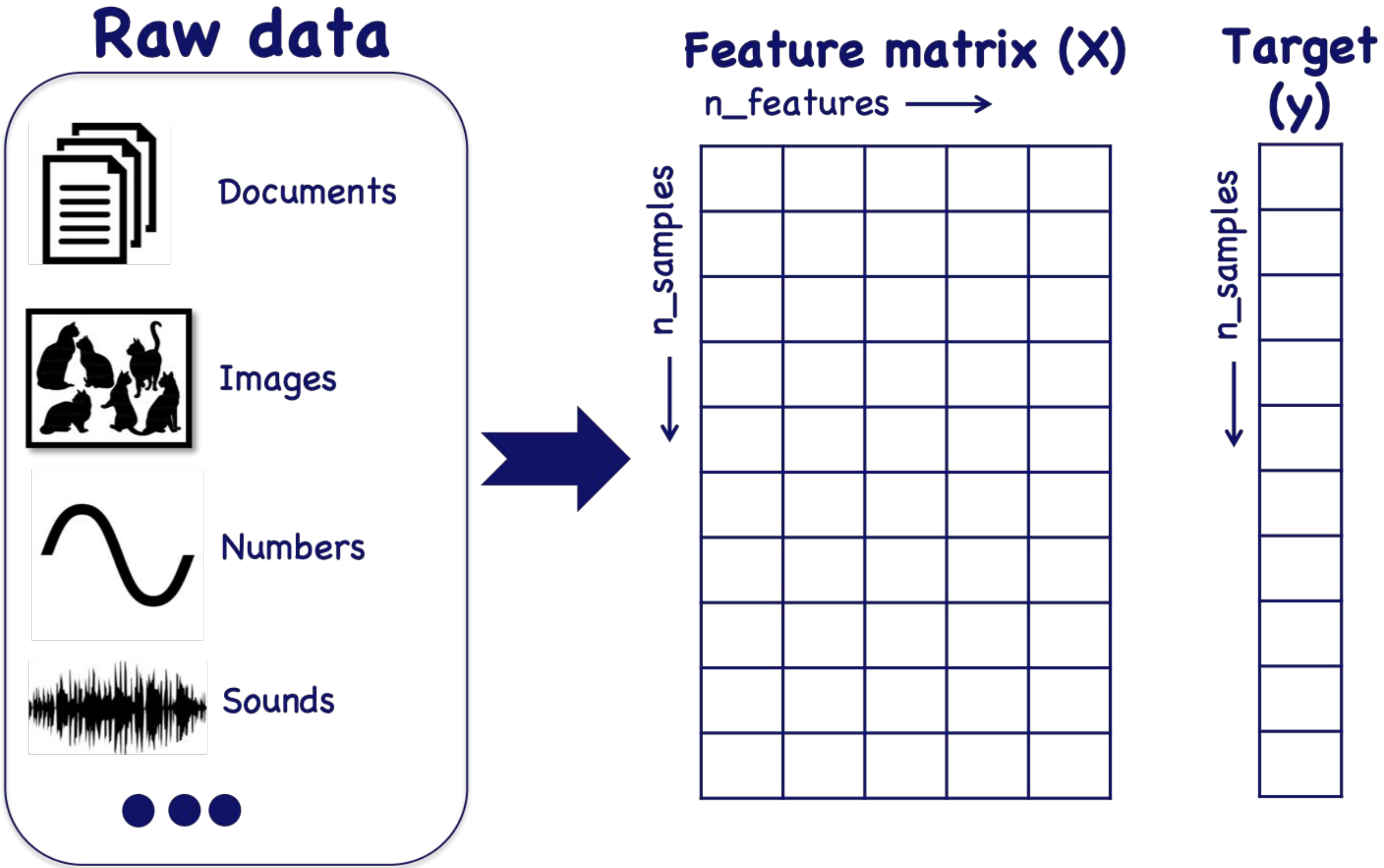
Artificial Intelligence Club Week 3 Slides

October 15, 2025

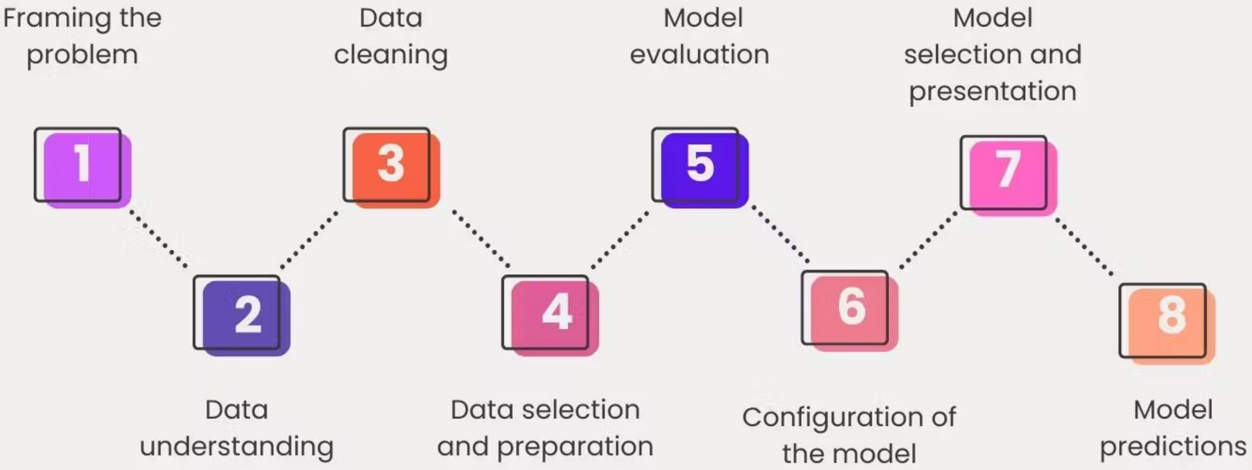
Quick Review from Week 2!

Raw data




Raw data
Documents
Images
Numbers
Sounds
Feature matrix (X)
n_features
n_samples
Target (y)
n_samples

How Statistics is Used in Machine Learning



Turing

Supervised Learning Basics

Goal: learn a mapping from features (inputs) to labels (outputs) using labeled examples

Training data: pairs of (features, target) used to fit a model

Choose a model and a loss function; minimize average loss on the training set

Tasks: regression (numbers) and classification (categories)

Metrics: Root mean squared error for regression or log loss for classification

Bayesian vs Frequentist

Lebron is about to shoot a free throw, what are the chances he makes it?

Frequentist: So far in the game, he shot 5/10 free throws. This means his free throw percentage is 50%, so **50% chance** he makes it!

Bayesian: However, Lebron has also shot 200 free throws so far this season, scoring 170 of them. This information can be used to come up with a better prediction. We need to include this information/data, also called a **prior**.

We use a Beta function to do that, where:

Beta(prior success + data success, prior misses + data misses) -> Beta(170 + 5, 30 + 5) =

Beta(175, 35) = 175 / 175 + 35=0.833

Thus, the predicted probability (posterior mean) is 83%, giving us an 83% chance he makes it!

Bayesian vs Frequentist

The **Frequentist** standpoint emphasizes how Lebrons free throw percentage changes with each **game**, thus we try to predict the chances of his next shot going in with data from the **current game.**

The **Bayesian** standpoint emphasizes that historical data can also guide our prediction. It becomes even more powerful when we include **scaling**, a way to add more **human emphasis** on how much influence we think the historical data should have. Say Lebron is sick this game, so the prior is less reliable:

Beta(scaled prior success + data success, scaled prior misses + data misses) $\rightarrow$
Beta((170 × 0.5) + 5, (30 × 0.5) + 5) = Beta(90, 20) =0.818

This gives us an 81.8% chance he makes it, slightly lower than using the full historical data, since we scaled the prior's influence to 50%. The scaling acts like a "confidence dial" on how much we trust the past versus what we've just observed.

We do this all the time naturally!

Our brains automatically consider historical data, when it comes to making future predictions!

But these ideas also need to be formalized and proven mathematically

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

Maximum Likelihood Estimate (MLE)

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

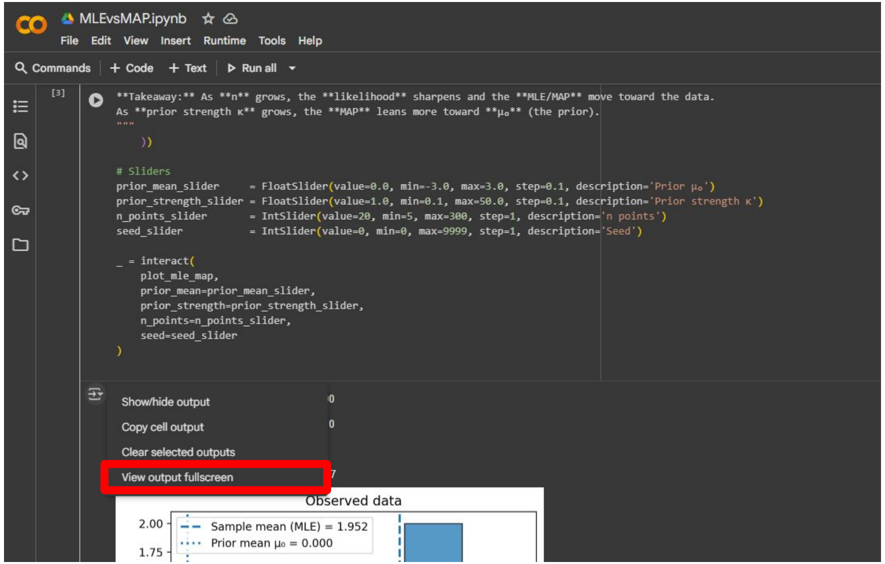
Maximum *a posteriori* (MAP) estimate

$$\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})\underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

Maximum Likelihood Estimation VS Maximum a Posteriori

https://colab.research.google.com/drive/16TUgd1C4QxBQFYOi0fq1dTOvWbgcNeLT?usp=sharing

(Make sure to hit view output fullscreen)



Resources

- ## Linear Algebra:

  Easier: https://youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&si=qQnVeyJd58BkU4AV

  More complex: https://youtu.be/N1Pvj4CZT1M?si=PbvkwWiJlulsgfLD

  In machine learning: https://www.visual-design.net/post/linear-algebra-for-machine-learning

- ## Statistics:

  Easier: https://www.youtube.com/watch?v=NIqeFYUhSzU

  More complex: https://www.youtube.com/watch?v=WB8eYZSZyaE

- ## Supervised Learning:

  https://www.geeksforgeeks.org/machine-learning/supervised-machine-learning/

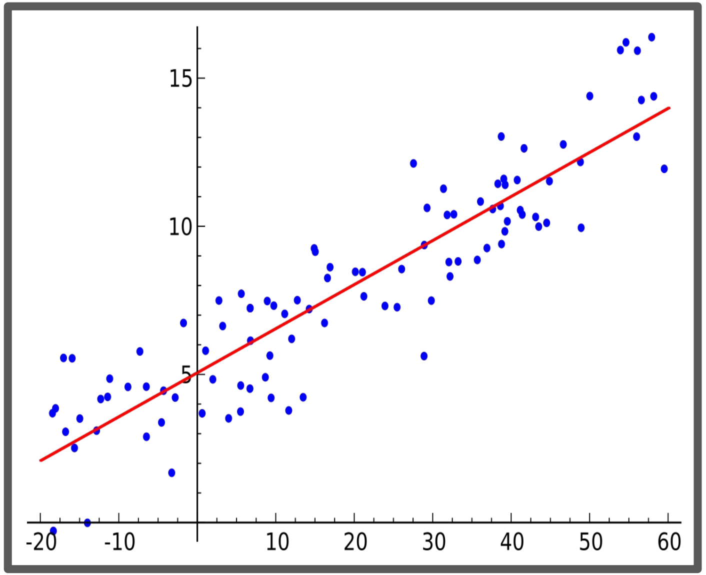
  https://www.youtube.com/watch?v=wvODQqb3D_8

Survey!

Feedback

Connections + Pizza + Talk to Officers

Linear Regression



Model: prediction = weighted sum of features + bias

MLE/Ordinary Least Squares: choose coefficients that minimize the sum of squared errors

Assumptions: linear relationship and independent errors

Evaluation: use root mean squared error

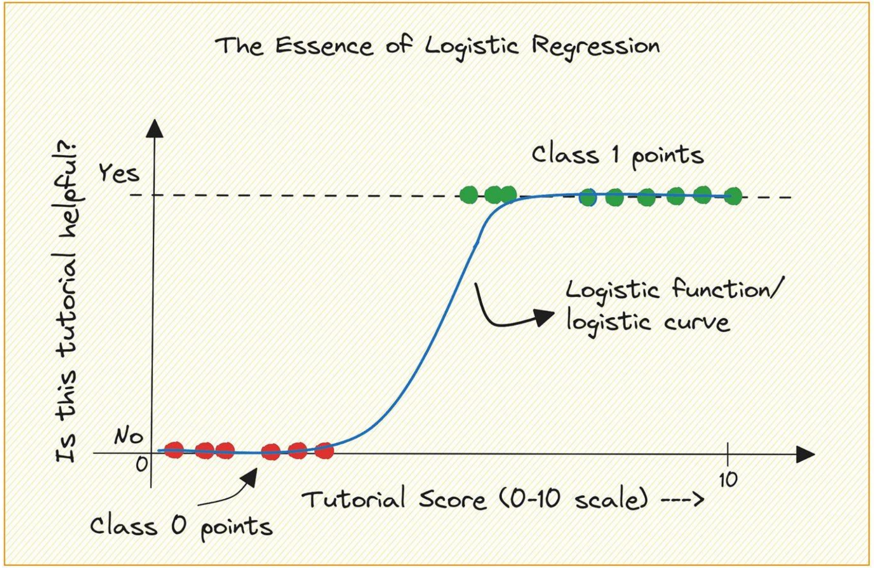
Linear Regression Practice

https://www.mladdict.com/linear-regression-simulator

Logistic Regression

Model: probability of the positive class = sigmoid(weighted sum of features)

Fitting: maximize the log-likelihood (or minimize log loss) using gradient-based optimization



Outputs: 0-1 predicted probabilities; choose a threshold (often halfway) to produce class labels

Evaluation: accuracy and log loss

Logistic Regression Practice

https://mlu-explain.github.io/logistic-regression/

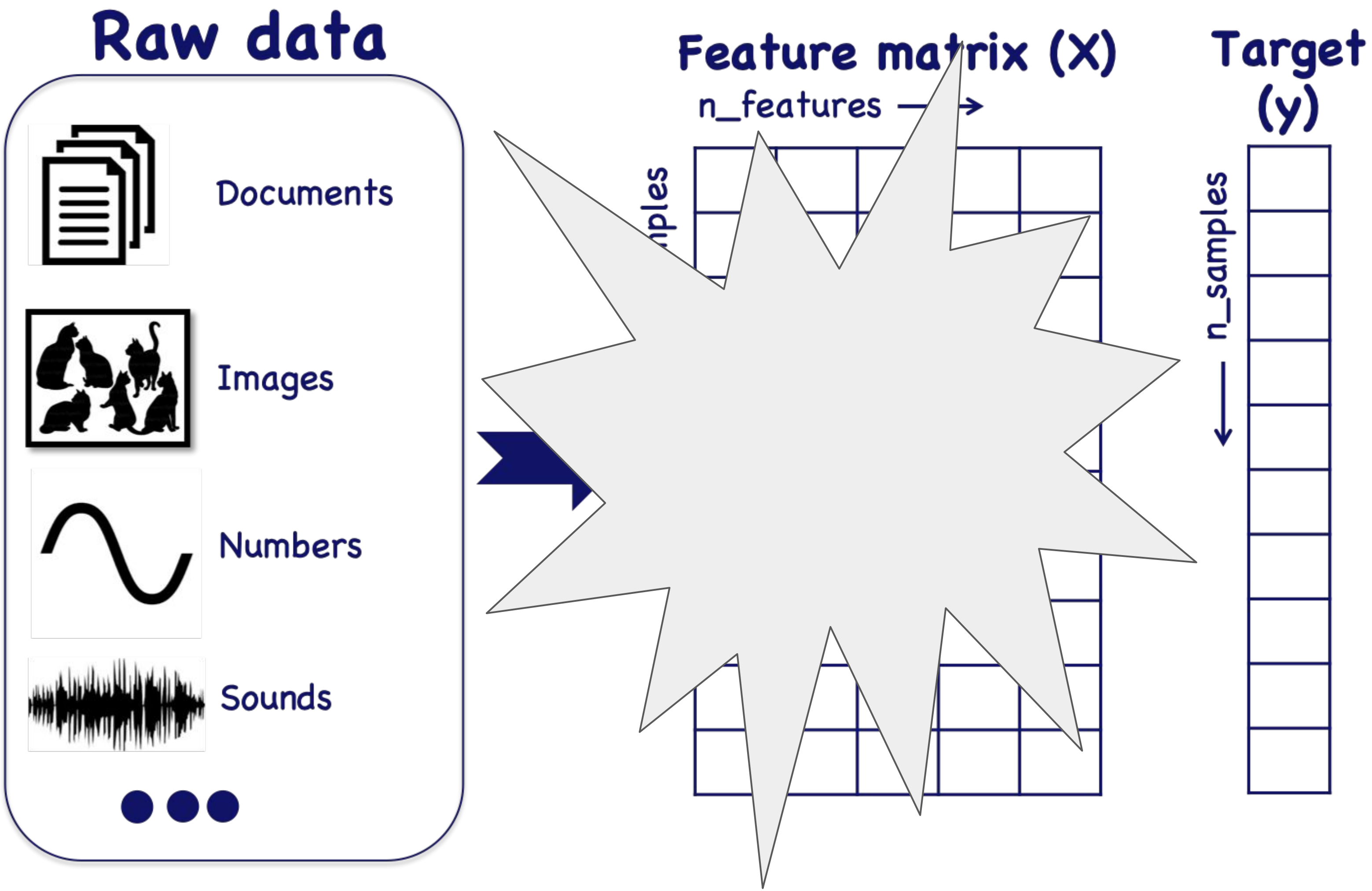
Cross-Validation for Regression & Classification

k-fold CV: split data into k folds; train on k−1 folds and validate on the remaining fold; average the validation score

Use appropriate metrics: RMSE/MAE for regression; accuracy/ROC-AUC/log loss for classification

Hyperparameter search: try multiple settings (e.g., regularization strength) via grid or randomized search, pick the best by CV score

Finalize: retrain the chosen model on all training data with the selected settings; report performance on a held-out test set



Raw data
Documents
Images
Numbers
Sounds
Feature matrix (X)
n_features
Target (y)
n_samples