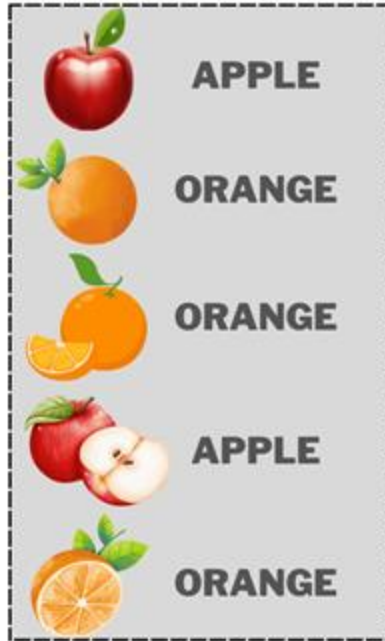


Supervised Learning

Types of Learning

- **Supervised Learning** (Today)
 - Learning from labeled training data: (**input**, **correct output**)
- **Unsupervised Learning** (later this week)
 - Learning from unlabeled training data: (**input**, **?**)
- **Semi-Supervised & Reinforcement Learning** (Later Bootcamps)
 - Semi-Supervised Learning involves learning from a mix of labeled and unlabeled data.
 - Reinforcement Learning involves learning through reward maximization.

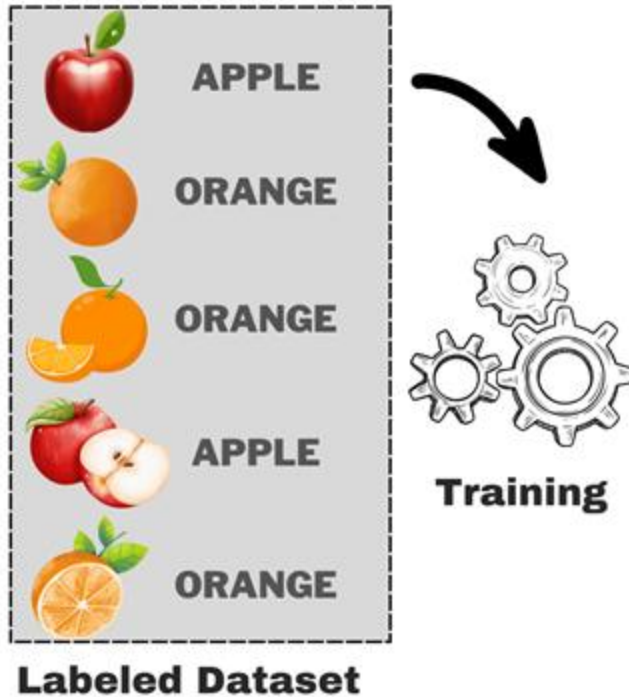
Supervised Learning



Labeled Dataset

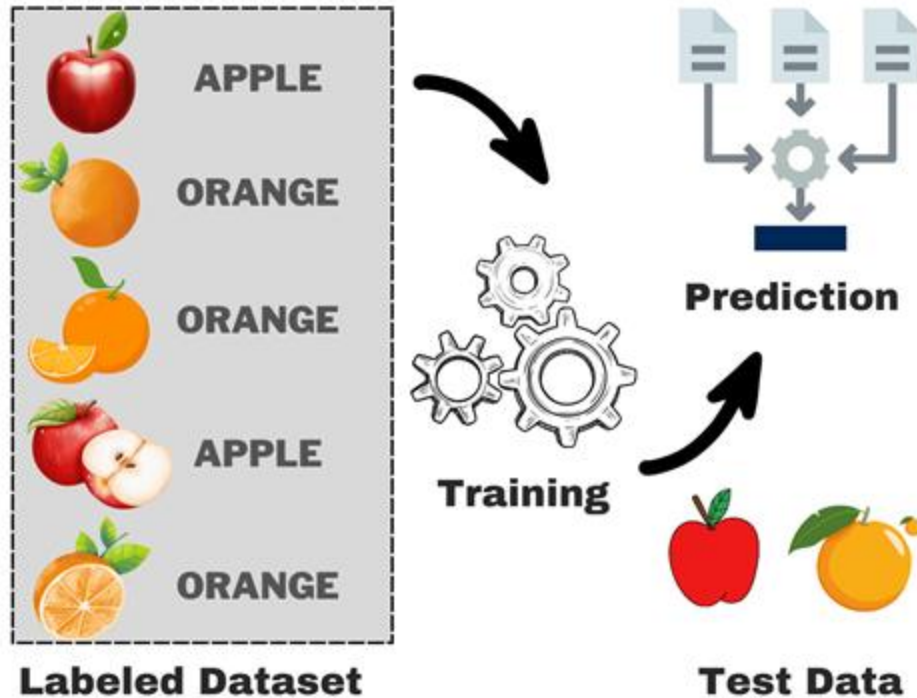
<https://www.kdnuggets.com/understanding-supervised-learning-theory-and-overview>

Supervised Learning



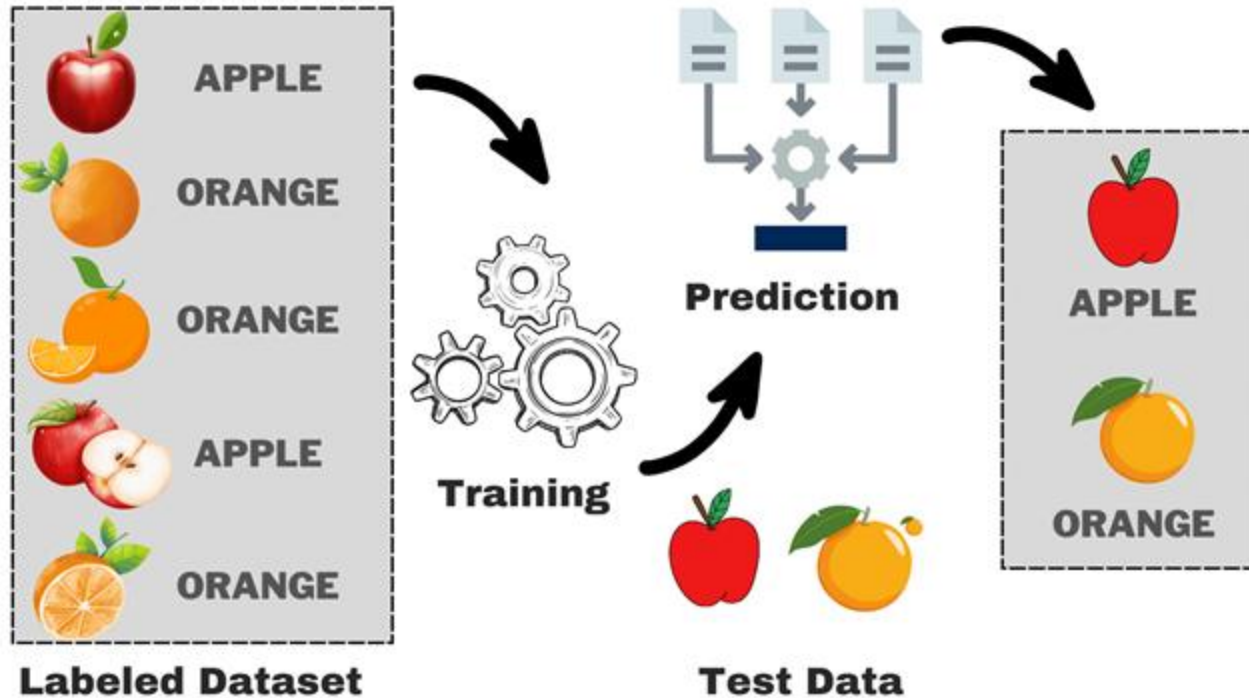
<https://www.kdnuggets.com/understanding-supervised-learning-theory-and-overview>

Supervised Learning



<https://www.kdnuggets.com/understanding-supervised-learning-theory-and-overview>

Supervised Learning



<https://www.kdnuggets.com/understanding-supervised-learning-theory-and-overview>

Basic Types of Supervised Learning

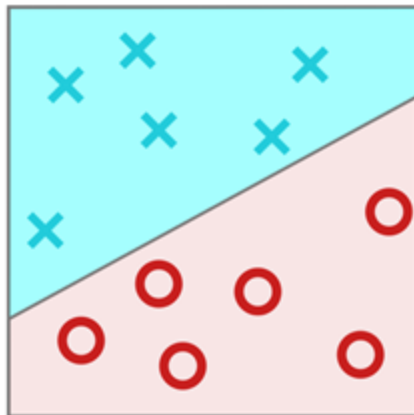
- **Regression**

- Predicting continuous numerical values based on input feature

- **Classification**

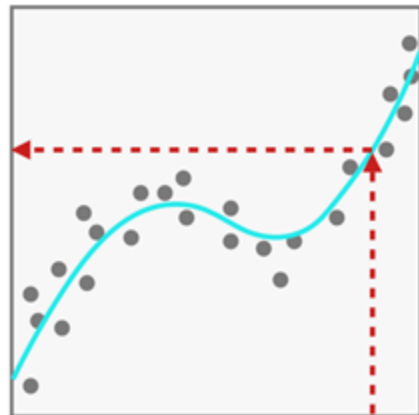
- Predicting a discrete class label based on input feature

Classification Groups observations into "classes"



Here, the line classifies the observations into X's and O's

Regression predicts a numeric value

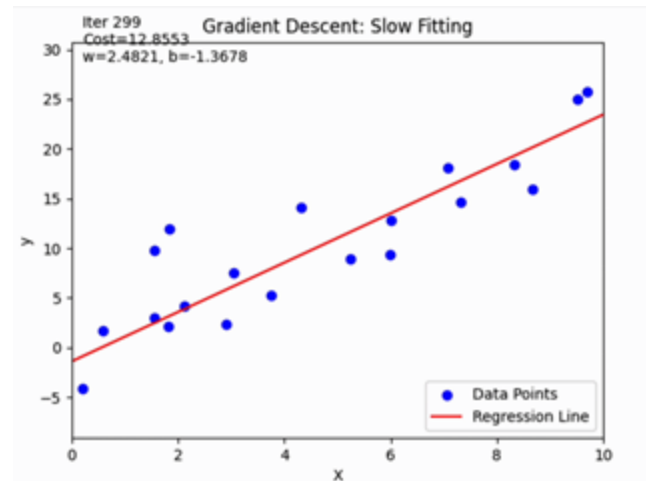


Here, the fitted line provides a predicted output, if we give it an input

<https://www.sharpsightlabs.com/blog/regression-vs-classification/>

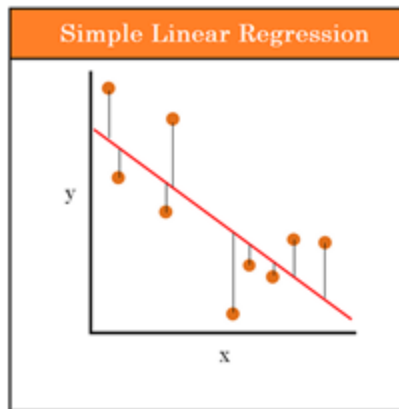
Regression

- Given data in the form:
 $[(X_1, Y_1), \dots, (X_N, Y_N)]$ where each (X_i, Y_i) corresponds to input X_i and a real valued (**continuous**) output Y_i .
- The regression model learns a **real valued function**: $\hat{y} = f(X) \in \mathbb{R}$
- To predict new output values simply plug in x into the function.

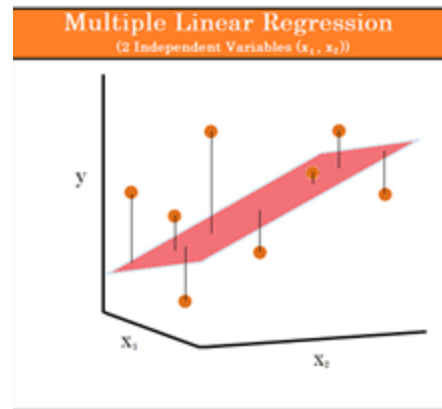


Goal of Linear Regression

Find the values for the unknown parameters (w) known as weights that minimize the average distance between all inputs and the real valued linear function (\hat{y}).



$$y = w_0 + w_1x$$



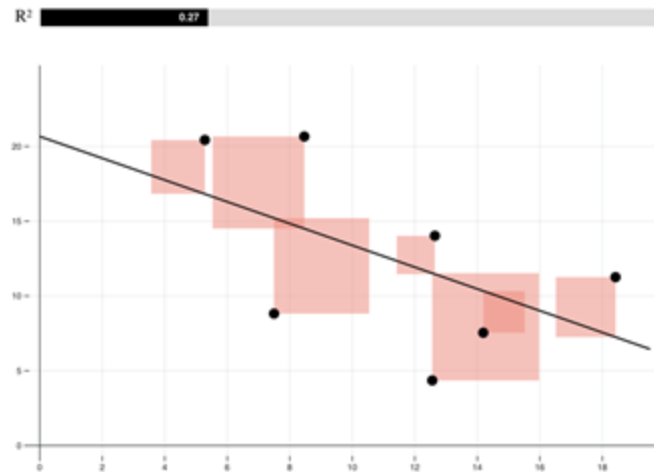
$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

Linear Regression Weights

For linear regression models, the weights are calculated by **minimizing Mean Squared Error (MSE)** which is the squared distance between each point and the line of best fit.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

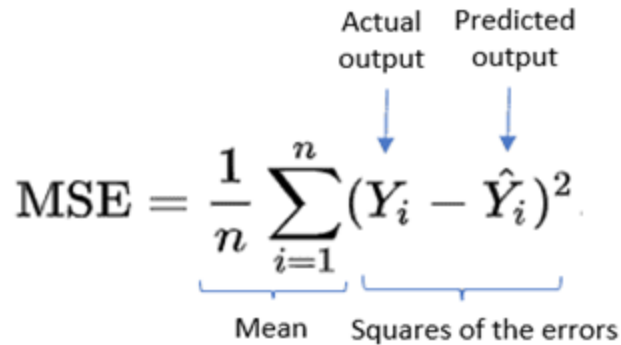
Actual output Predicted output
↓ ↓
Mean Squares of the errors



$$\hat{y} = w_0 + w_1 x$$

Advantages of Mean Squared Error

- Differentiable everywhere
- Closed form solution for linear models
- Penalizes large errors
- Nice analytical properties
- Solution to maximum likelihood

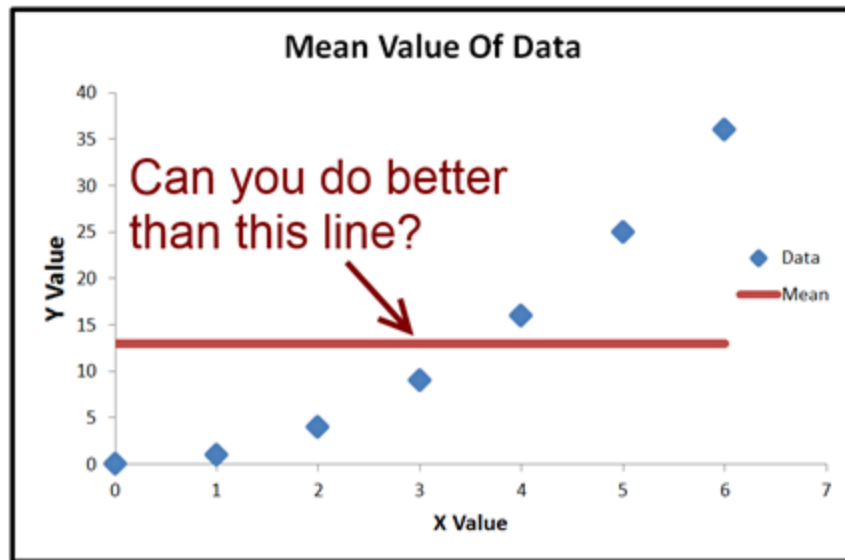


The diagram shows the Mean Squared Error (MSE) formula with annotations. At the top, 'Actual output' and 'Predicted output' are written. Blue arrows point from 'Actual output' to Y_i and from 'Predicted output' to \hat{Y}_i in the formula. Below the formula, a blue bracket under $\frac{1}{n}$ is labeled 'Mean', and another blue bracket under $\sum_{i=1}^n (Y_i - \hat{Y}_i)^2$ is labeled 'Squares of the errors'.

$$\text{MSE} = \underbrace{\frac{1}{n}}_{\text{Mean}} \sum_{i=1}^n \underbrace{(Y_i - \hat{Y}_i)^2}_{\text{Squares of the errors}}$$

Linear Regression

How would you change the mean to reduce MSE and increase the accuracy of our model?



$$\hat{y} = w_0 + w_1x$$

Linear Regression

Weights are evaluated using **Coefficient of Determination(R^2)** which shows how much of the variability in the data is explained by the independent variable.

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$



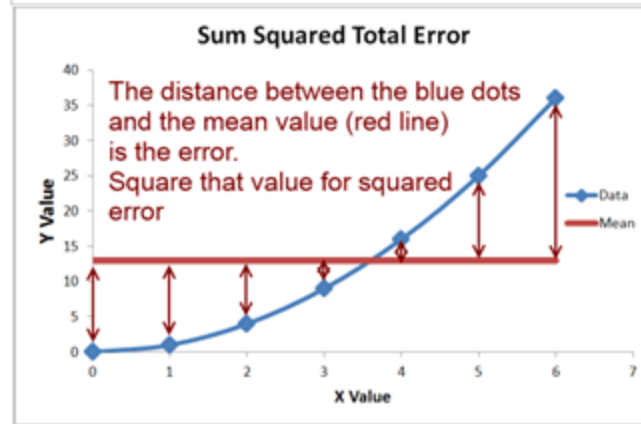
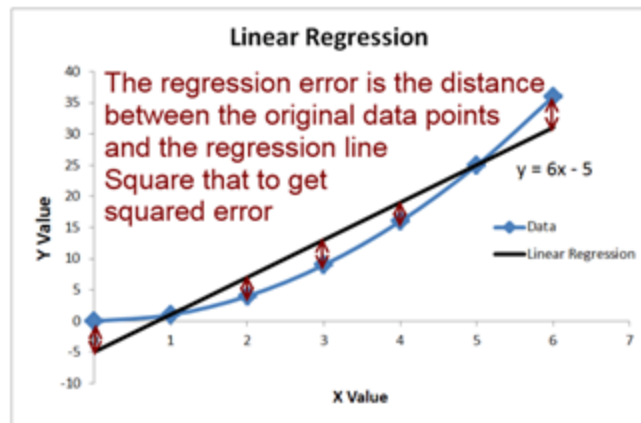
$$\sum_i (y_i - f_i)^2$$

Mean Squared Error



$$\sum_i (y_i - \bar{y})^2$$

Proportional to variance in our data.



Linear Regression: Minimizing Mean Squared Error

$$X \rightarrow y \quad \longrightarrow \quad \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \longrightarrow \quad \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1d} \\ X_{21} & X_{22} & \cdots & X_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nd} \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Linear Regression: Minimizing Mean Squared Error

→ $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$

→ $\hat{y}_i = [X_{i1} \ X_{i2} \ \dots \ X_{id}] \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} + \boxed{b}$

Input Weights Bias

→ $\hat{y}_i = X_i w + b$

Linear Regression: Minimizing Mean Squared Error

→ $X'_i = [1 \quad X_{i1} \quad X_{i2} \quad \dots \quad X_{id}]$

→ $w' = [b \quad w_1 \quad w_2 \quad \dots \quad w_d]^T$

→ $\hat{y}_i = [1 \quad X_{i1} \quad X_{i2} \quad \dots \quad X_{id}] \cdot \begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$ → $\hat{y}_i = X'_i w'$

Linear Regression: Minimizing Mean Squared Error

$$\rightarrow MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\rightarrow MSE = \frac{1}{n} \sum_{i=1}^n (y_i - X'_i w')^2$$

$$\rightarrow MSE = \frac{1}{n} \|y - X'w'\|^2$$



$$\nabla_{w'} E(w') = \frac{2}{n} X'^T (X'w' - y)$$

Linear Regression: Minimizing Mean Squared Error

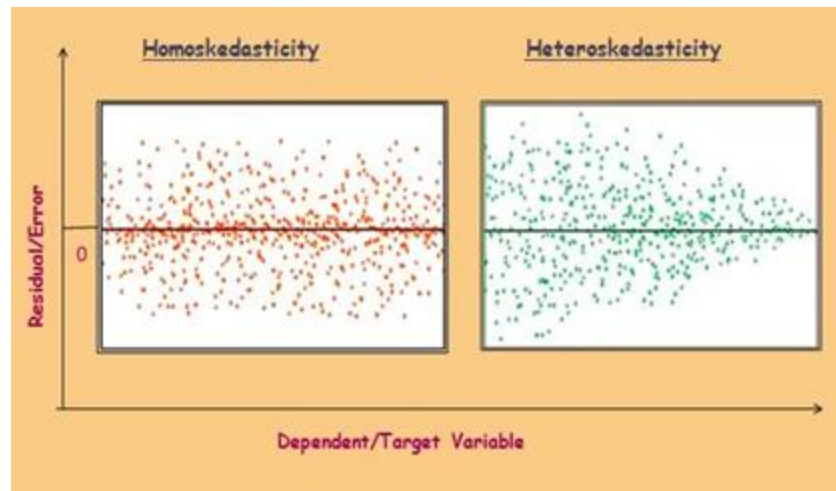
$$\longrightarrow X'^T X' w' = X'^T y \quad \longrightarrow w' = (X'^T X')^{-1} X'^T y$$

$$\longrightarrow \boxed{w' = X^+ y}$$

- The process of finding weights that minimize MSE error a linear regression model is known as **Ordinary Least Squares (OLS)**.
- We multiply these weights by any new input to predict its value.

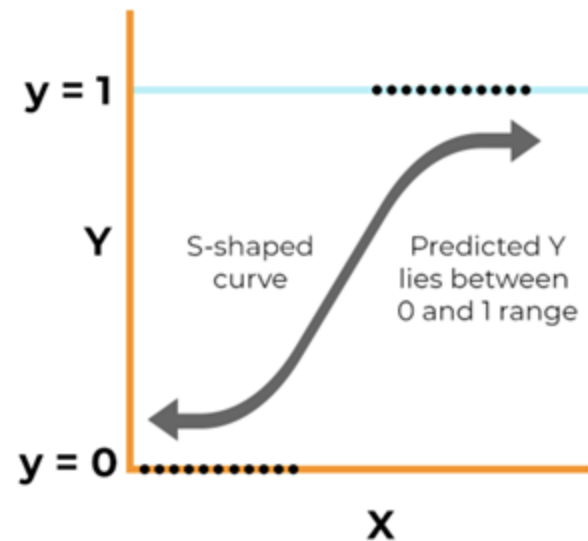
Homoskedasticity and Heteroskedasticity

- **Homoskedasticity** is achieved when variance is constant across all observations, and does not depend on the value of the explanatory variables.
- **If Homoscedasticity holds**, then the OLS weights for a linear regression model are the **Best Linear Unbiased Estimator**.



Logistic Regression

- This supervised learning model predicts the continuous likelihood of an outcome class given some input.
- Commonly used for binary classification.
- Why is it called a regression model?
- Can you think of a scenario where it is not used for classification?



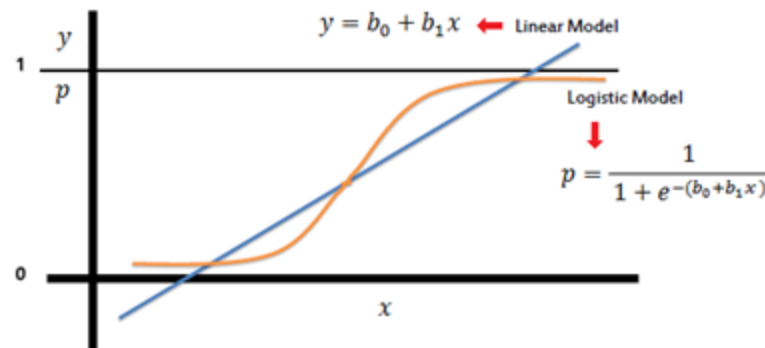
Logistic Regression With Binary Outcomes

Given a binary outcome $y \in \{0, 1\}$:

$$\hat{y} = P(y = 1 | X) = \sigma(Xw + b)$$

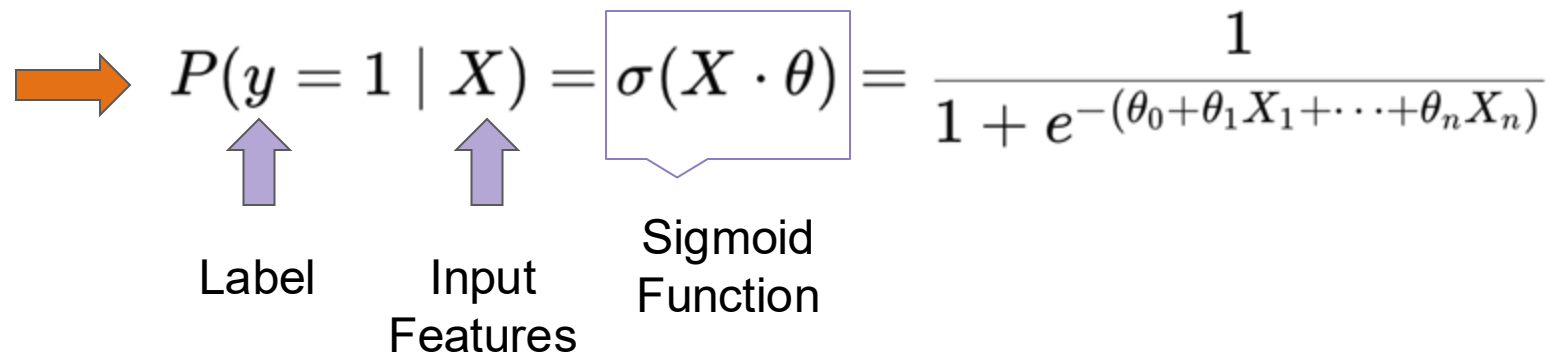
where:

- X is the feature matrix
- w is the weight vector.
- b is the bias term.
- $\sigma(z) = \frac{1}{1+e^{-z}}$ is the **sigmoid function** that maps values to probabilities.



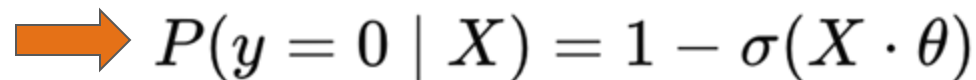
https://www.saedsayad.com/logistic_regression.htm

Logistic Regression Weight Calculation



The diagram illustrates the formula for the probability of a label given input features in logistic regression. An orange arrow points to the left of the equation. The equation is $P(y = 1 \mid X) = \sigma(X \cdot \theta) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)}}$. A purple arrow points from the label 'Label' to the y in the probability expression. Another purple arrow points from the label 'Input Features' to the X in the probability expression. A purple box highlights the sigmoid function $\sigma(X \cdot \theta)$, with a callout line pointing to the text 'Sigmoid Function' below it.

$$\text{Label} \quad \text{Input Features} \quad \text{Sigmoid Function}$$



An orange arrow points to the left of the equation $P(y = 0 \mid X) = 1 - \sigma(X \cdot \theta)$.

Logistic Regression Weight Calculation

→
$$\mathcal{L}(\theta) = \prod_{i=1}^N P(y_i | X_i)$$

→
$$\mathcal{L}(\theta) = \prod_{i=1}^N \sigma(X_i \cdot \theta)^{y_i} \cdot (1 - \sigma(X_i \cdot \theta))^{(1-y_i)}$$

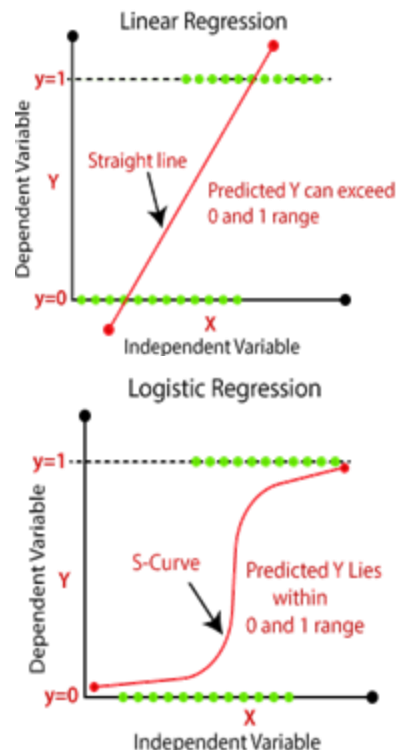
Logistic Regression Weight Calculation

→
$$\log \mathcal{L}(\theta) = \sum_{i=1}^N [y_i \log \sigma(X_i \cdot \theta) + (1 - y_i) \log(1 - \sigma(X_i \cdot \theta))]$$

→
$$\max_{\theta} \left(-\frac{1}{N} \sum_{i=1}^N [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \right)$$

Logistic Regression vs Linear Regression

- Logistic regression maps inputs to the likelihood that it belongs to a certain class.
- We use a sigmoid curve to get a range of probabilities instead of just zero or one classification.
- In logistic regression we find the weights by maximizing the log likelihood.



Logistic Regression Pros and Cons

- Pros:

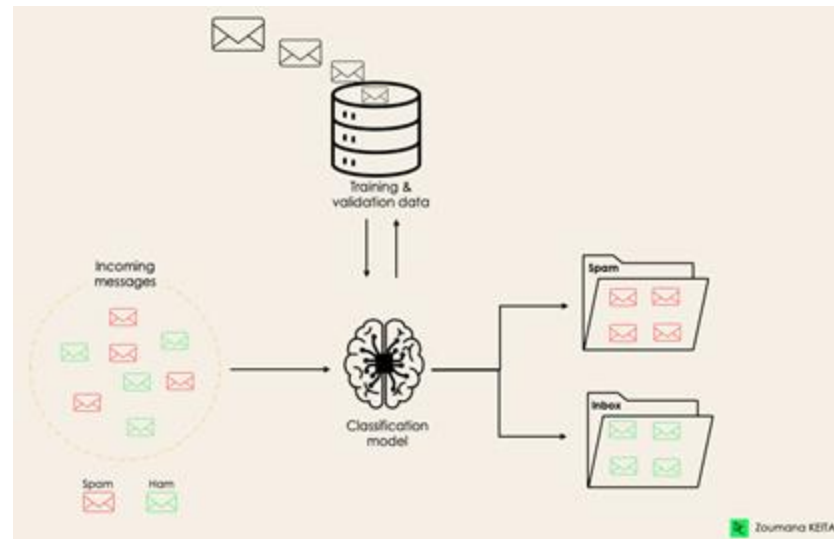
- Simple and Interpretable
- Probabilistic Output
- Computationally Efficient
- Binary Classification

- Cons:

- Assumes linearity (with the log-odds of dependent variable)
- Sensitive to outliers
- Requires independence of features

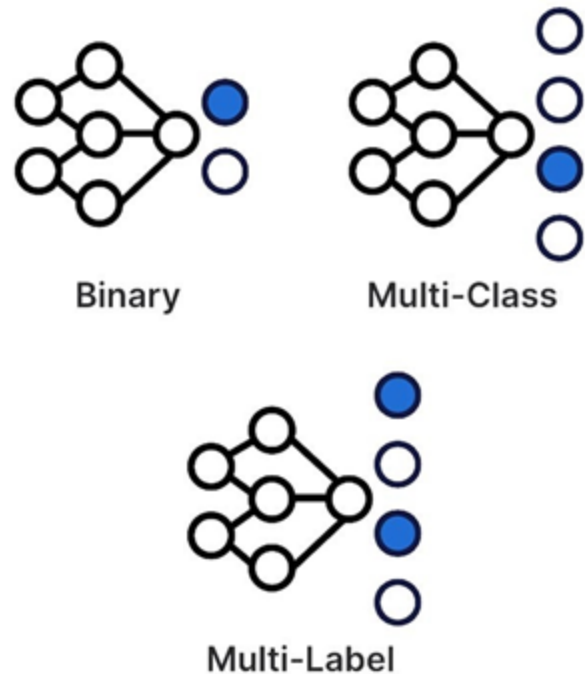
Classification

- Given data in the form:
 $[(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_N, \mathbf{Y}_N)]$ where each $(\mathbf{X}_i, \mathbf{Y}_i)$ corresponds to an input \mathbf{X}_i and a categorical (**discrete**) output \mathbf{Y}_i .
- The regression model learns a **mapping function**: $f: X \rightarrow Y$
- Here Y is a category is our set of categories.



Types of Classification

- **Binary:** We have two categories and classify each input into one of the two categories.
- **Multiclass:** We have more than two categories however, we classify each input into only one of the categories.
- **Multi-Label:** We have more than two categories, and each input can belong to multiple categories.



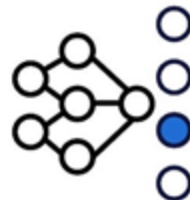
Examples

We want to design models that achieve the following goals. Which classification model should we use:

- Given an email, we want to determine whether it is spam or not.
- Given a song, determine which category of music it belongs to.
- Given an image determine which animal it is.



Binary



Multi-Class



Multi-Label

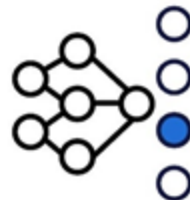
Examples

We want to design models that achieve the following goals. Which classification model should we use:

- Given an email, we want to determine whether it is spam or not.
- Given a song, determine which category of music it belongs to.
- Given an image determine which animal it is.



Binary



Multi-Class

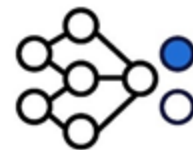


Multi-Label

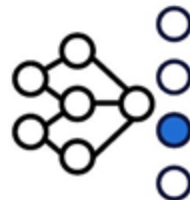
Examples

We want to design models that achieve the following goals. Which classification model should we use:

- Given an email, we want to determine whether it is spam or not.
- Given a song, determine which category of music it belongs to.
- Given an image determine which animal it is.



Binary



Multi-Class



Multi-Label

Logistic Regression For Classification Pipeline

Input Features:

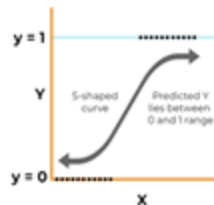
(X_1, X_2, \dots, X_n)

Linear Combination:

$$Z = w_1X_1 + w_2X_2 + \dots + w_nX_n + b$$

Sigmoid Function:

$$f(Z) = \frac{1}{1 + e^{-Z}}$$

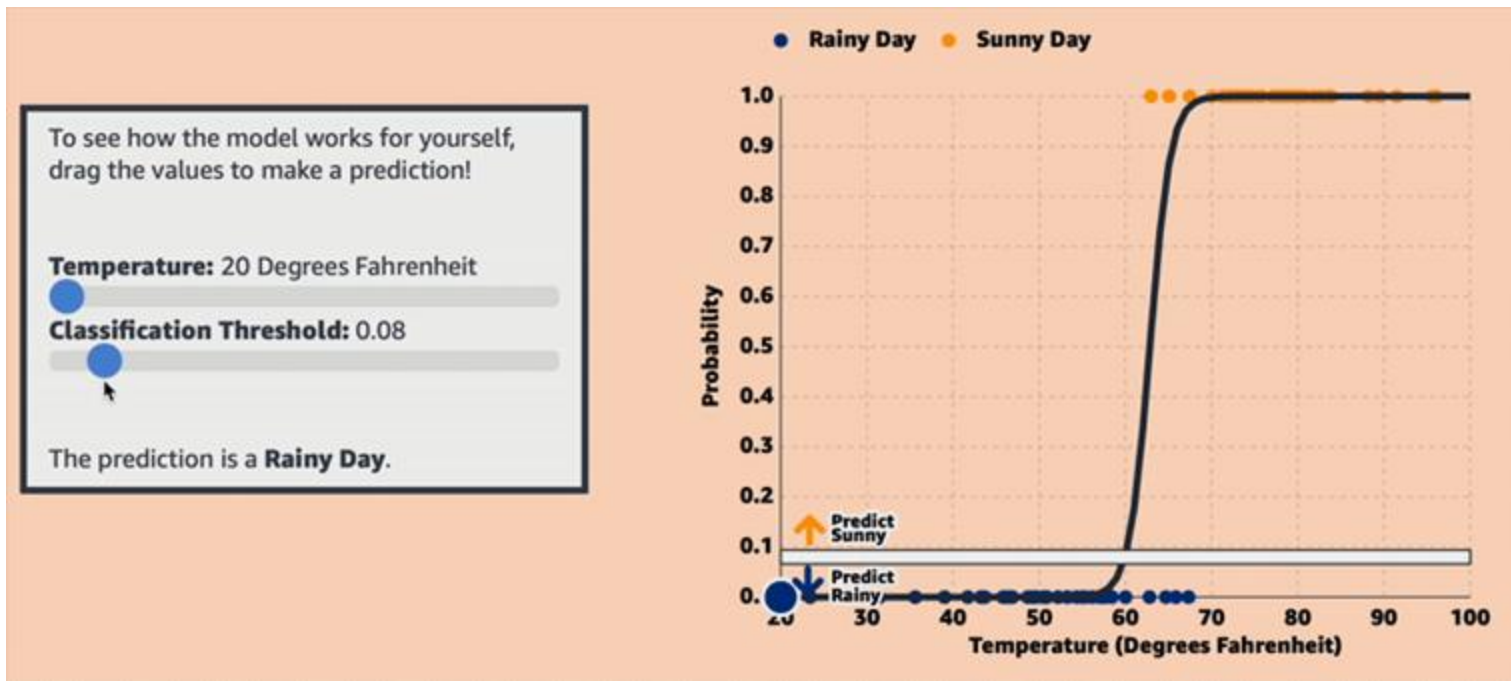


Output (Probability)

Threshold Decision

Class Label (0 or 1)

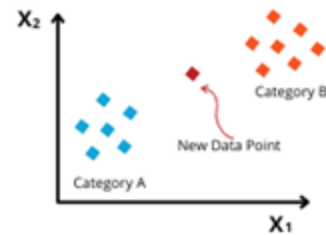
Logistic Regression for Classification Example



Popular Classification Algorithms

- **K-Nearest Neighbors (KNN):** Given a new input classify the same as the majority class of it's K closest neighbors.
- **Decision Tree (DT):** Recursively splits data into decision-based branches
- **Support Vector Machine (SVM):** Finds the optimal hyperplane to separate classes in high-dimensional space classifying new inputs into one class.

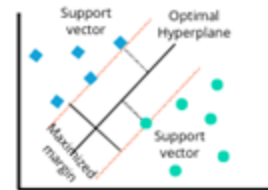
K-Nearest Neighbor



Decision Trees

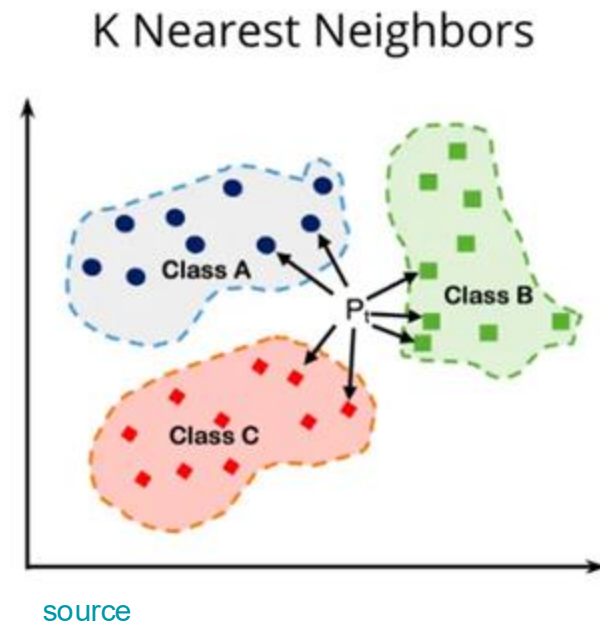


Support Vector Machine



Classification Algorithms: KNN

- **Algorithm:**
 - Define K (number of neighbors)
 - For each new data point X:
 - Compute the euclidean distance between X and all training points
 - Classify X with label of the most frequent class among its K nearest neighbors



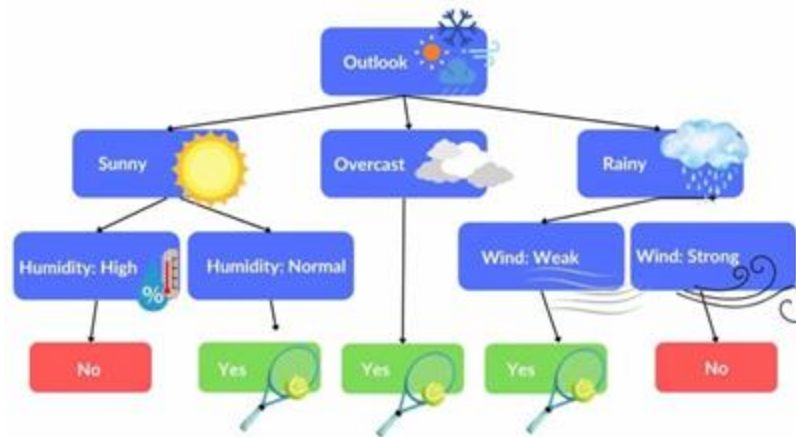
Classification Algorithms: DT

- **Algorithm:**

- Select the best feature to split on and split the dataset based on it.
- Recursively repeat the above for each subset until some stopping condition. The leaf node should be the category for
- For each new input, traverse the tree based on feature values until a leaf node classification.

X: {Outlook, Temperature, Humidity, Wind}

Y: {Play tennis, Don't play tennis}

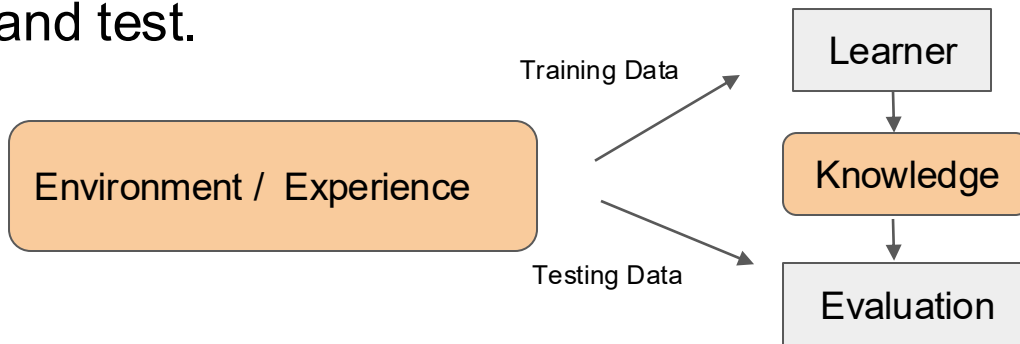


https://spotintelligence.com/2024/05/22/decision-trees-in-ml/#What_are_Decision_Trees

Framing a Learning Problem

Designing a Learning System

- Define the ideal outcome and the model's goal
- Choose how to represent the target function
- Determine the appropriate learning algorithm to infer the target function based on research.
- Define success metrics and test.



Housing Price Predictor

Goal: We want to design a model that will examine the components of a house to determine its price.

- What does our input look like?
- What does our output look like?
- What would our our target function look like?
- What kind of model is needed for this task?

https://colab.research.google.com/drive/1yQ2_Aau5MI75NkIC_8MKwMzOEneZRsU-#scrollTo=IlullIZYtZi7&uniqifier=1

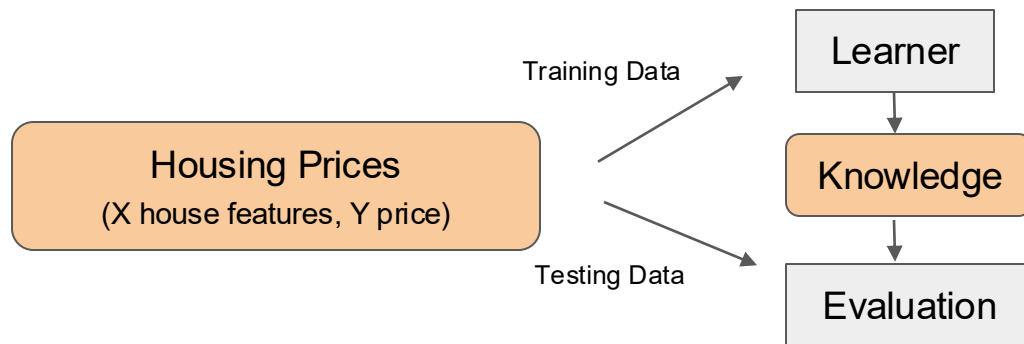
Housing Price Predictor

- What does our input look like?
 - $X: \{\text{House Features}\}$
- What does our output look like?
 - $Y: \text{Housing price (Real Value)}$
- What would our target function look like?
 - $f(\{\text{House Features}\}) \rightarrow \text{Price}$
- What kind of model is needed for this task?
 - Regression Model
- What are some success metrics we could use?



Housing Price Predictor

Success Metric: R^2

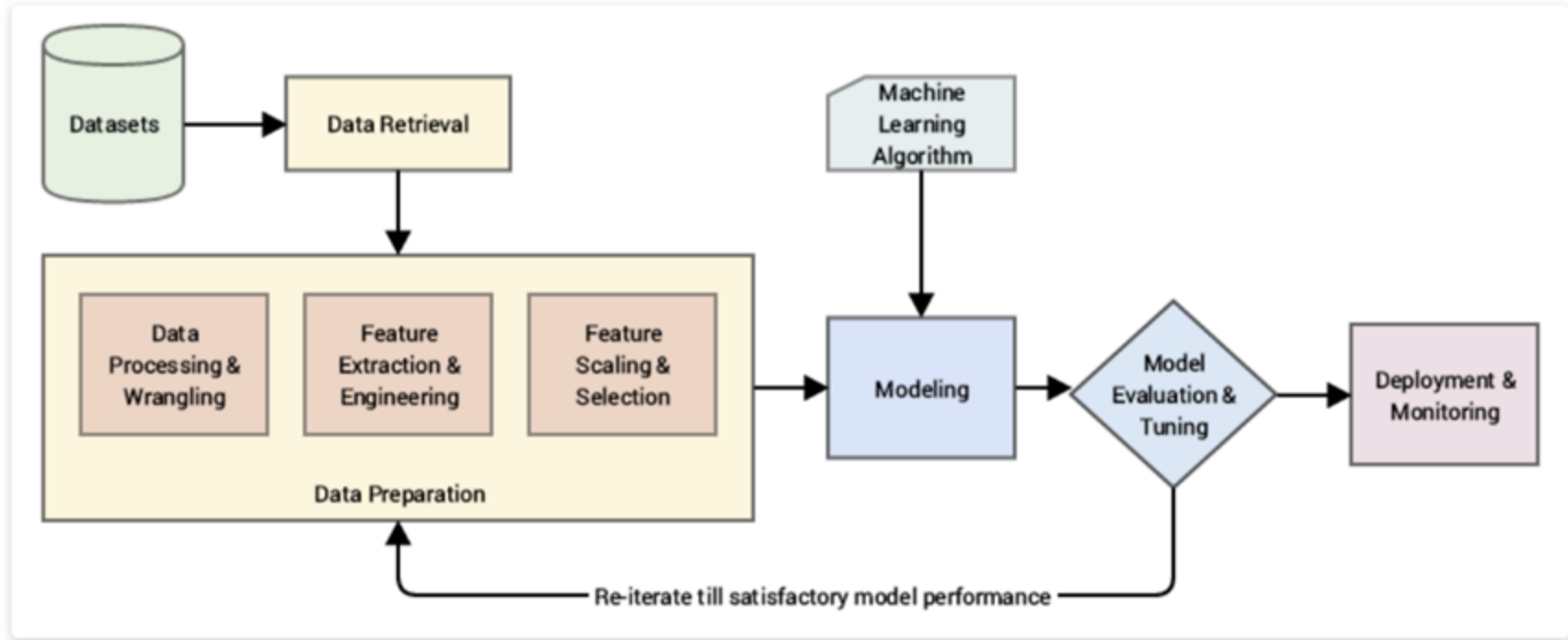


[Seeing it in Action](#)

Splitting Dataset

- Dataset is typically split into a train and test set
 - Training set is used to train the model
 - Test set is used to evaluate the model based on various metrics
- The typical split is: 80% training, 20% testing.
- We assume that each data sample was is independently drawn from the same overall distribution of data making all data points **Independent and Identically Distributed (iid)**

Overall Machine Learning Pipeline



Leakage and the Reproducibility Crisis in ML-based Science



Paper (Patterns, 2023)

July '22 online workshop

We argue that there is a reproducibility crisis in ML-based science. We compile evidence of this crisis across fields, identify data leakage as a pervasive cause of reproducibility failures, conduct our own reproducibility investigations using in-depth code-review, and propose a solution.

Top

List of failures

Taxonomy

Model info sheets

Case study

Terminology

Citation

About us

Context

Many quantitative science fields are [adopting](#) the paradigm of predictive modeling using machine learning. We welcome this development. At the same time, as researchers whose interests include the strengths and limits of machine learning, we have concerns about reproducibility and overoptimism.

There are many reasons for caution:

- Performance evaluation is notoriously tricky in machine learning.
- ML code tends to be complex and as yet lacks standardization.
- Subtle pitfalls arise from the differences between explanatory and predictive modeling.
- The hype and overoptimism about commercial AI may spill over into ML-based scientific research.
- Pressures and publication biases that have led to past reproducibility crises are also present in ML-based science.

Given these reasons, we view reproducibility difficulties as the expected state of affairs until best practices become better.

<https://reproducible.cs.princeton.edu/>