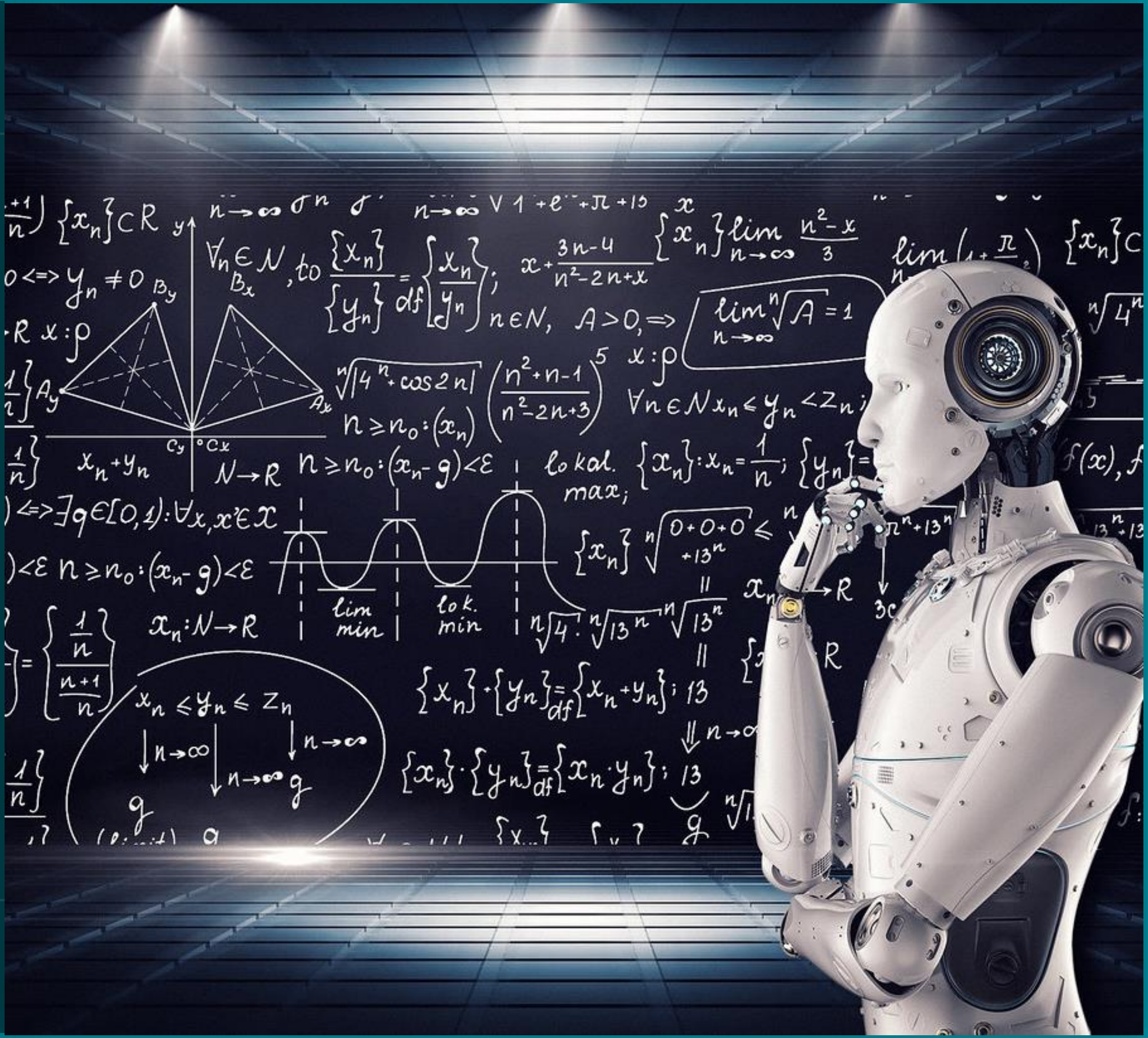
Introduction to Machine Learning

Supervised learning



HOSPITAL DA LUZ
LEARNING HEALTH
TRAINING, RESEARCH & INNOVATION CENTER

Summary

26 Jun

- Introduction to supervised learning
- Linear regression
  - Simple linear regression
  - Multiple linear regression
  - Gradient descent

27 Jun

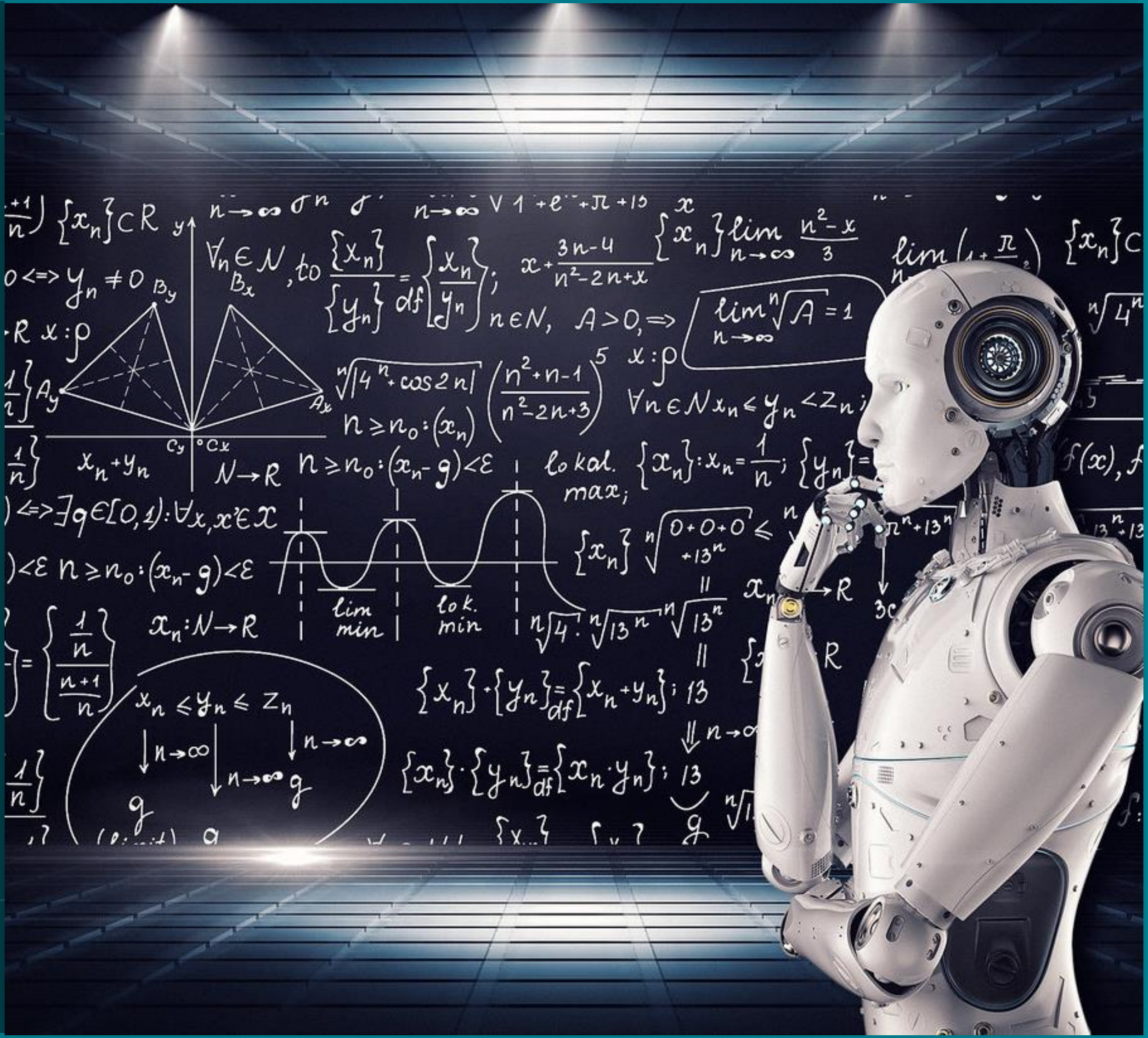
- Classification
  - Logistic regression
  - KNN

2 Jul

- Non-linear world
  - Polynomials and variants
  - Tree-based methods

Introduction to Machine Learning

Classification

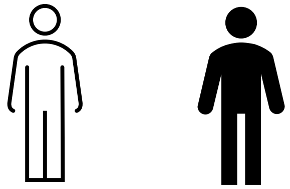


HOSPITAL DA LUZ
LEARNING HEALTH
TRAINING, RESEARCH & INNOVATION CENTER

What is classification?

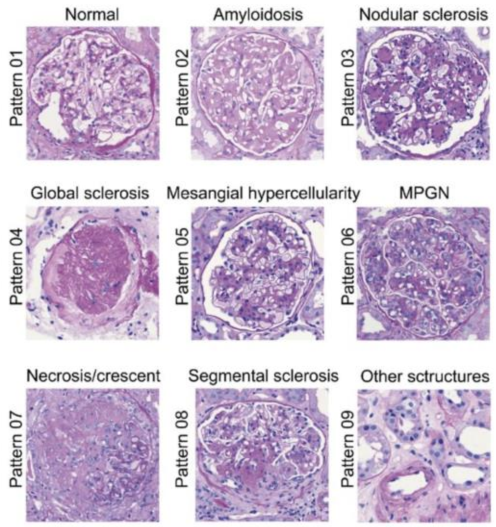
- Assigning observations to one of a **finite set of classes**

Binary classification



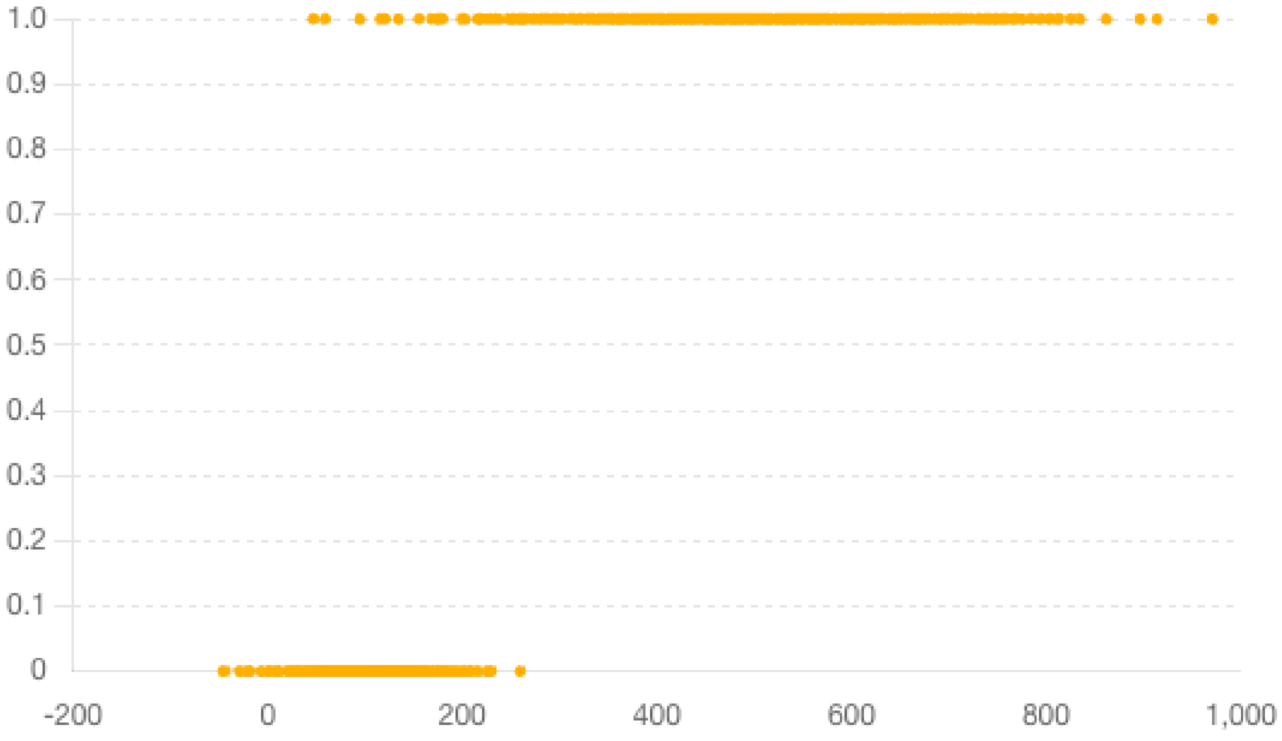
Sick Healthy

Multiclass classification



How to classify?

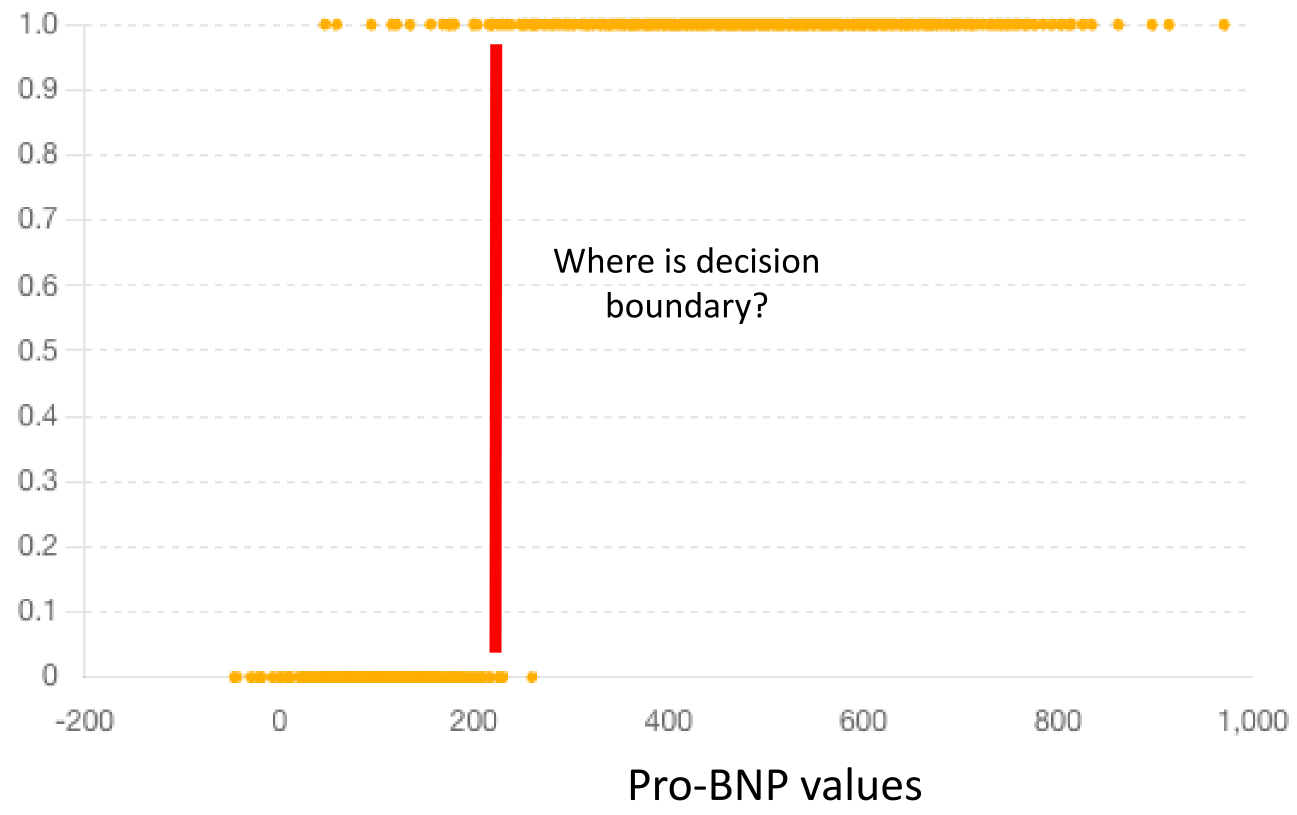
Heart Failure 1 = yes
Heart Failure 0 = no



Pro-BNP values

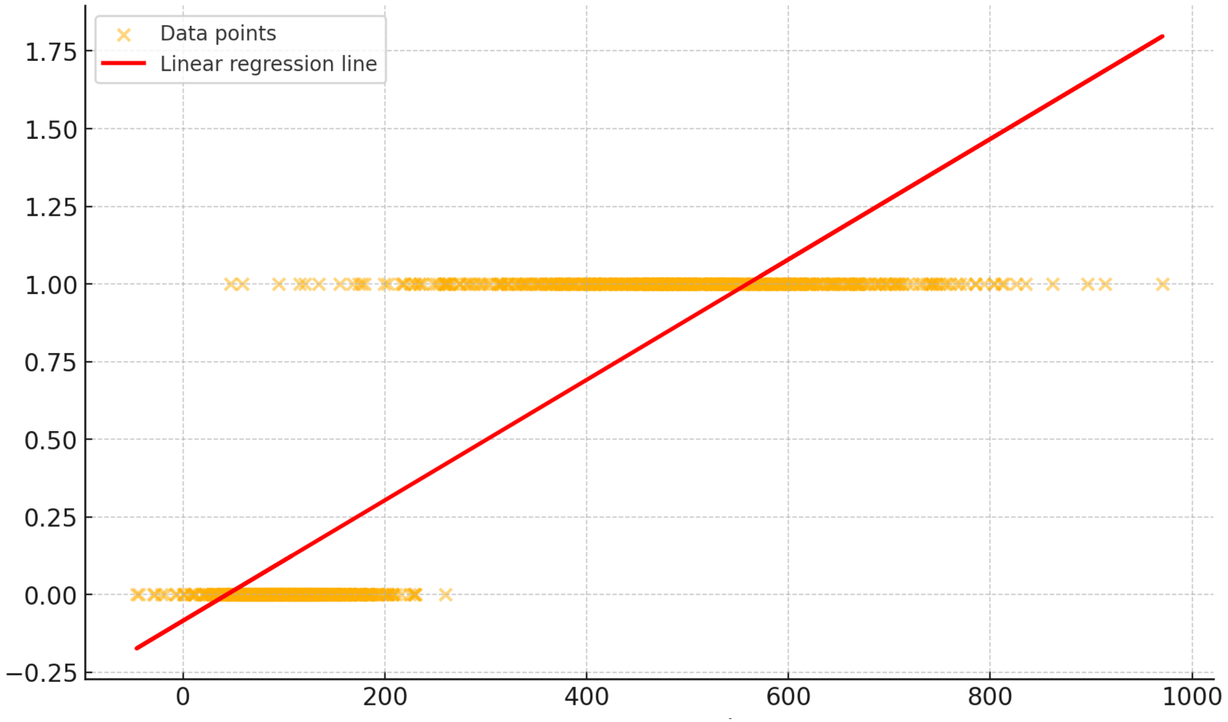
How to classify?

Heart Failure 1 = yes
Heart Failure 0 = no



Would a linear regression help?

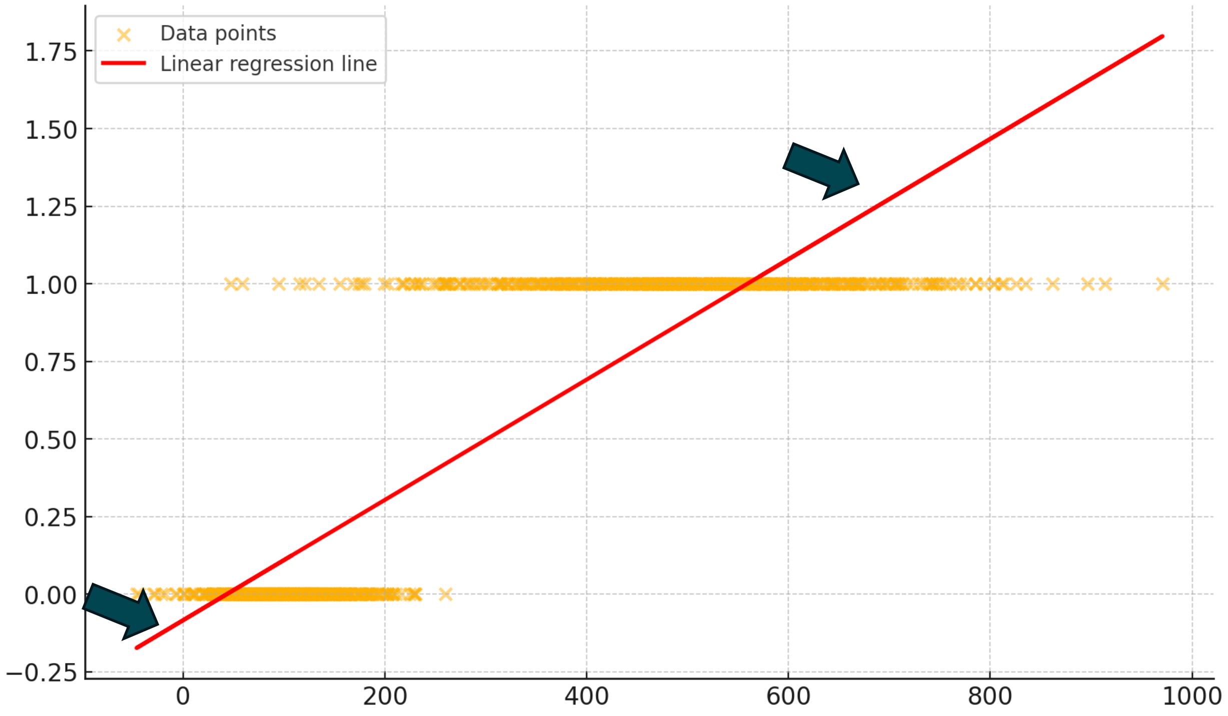
Heart Failure 1 = yes
Heart Failure 0 = no



Pro-BNP values

Would a linear regression help?

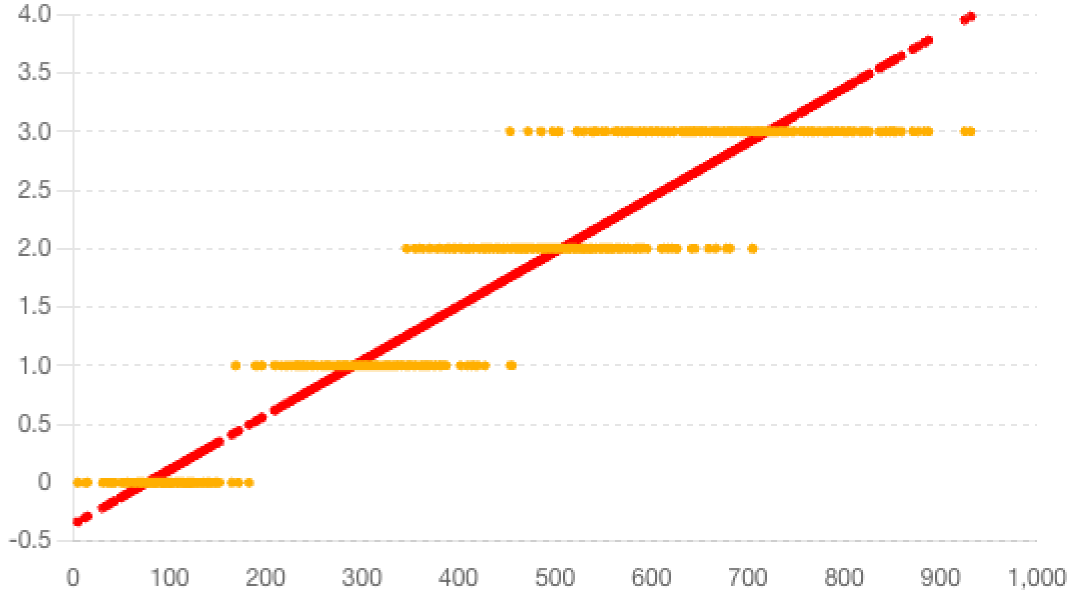
Heart Failure 1 = yes
Heart Failure 0 = no



Pro-BNP values

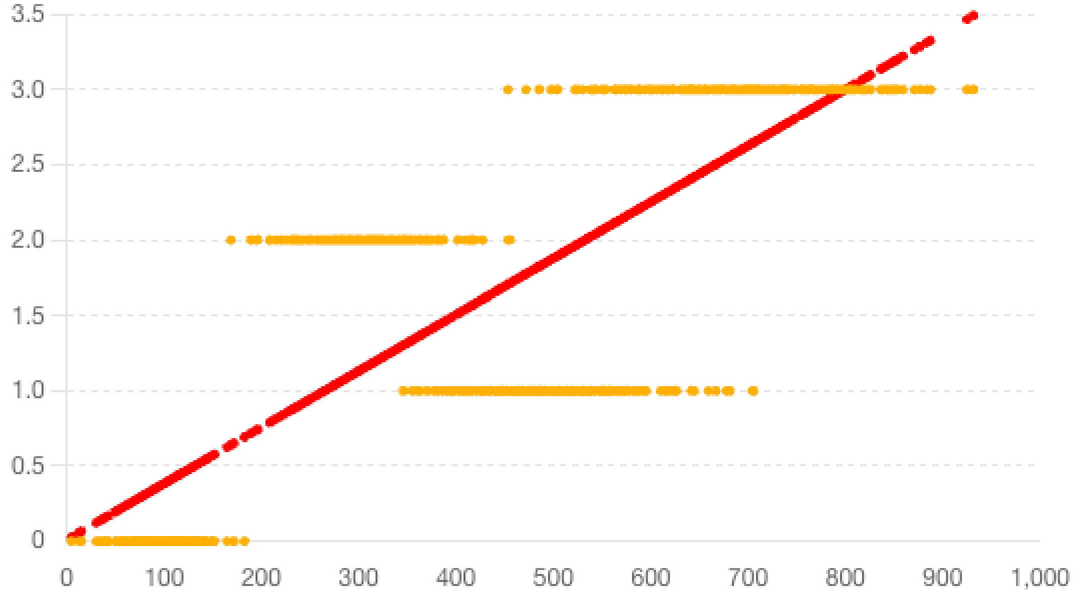
Would a linear regression help?

(0: No HF, 1: Preserved EF, 2: Mid Range EF, 3: Reduced EF)



- Coefficient: 0.00466
- Intercept: -0.35743

(0: No HF, 2: Preserved EF, 1: Mid Range EF, 3: Reduced EF)



- Coefficient: 0.00374
- Intercept: 0.00989

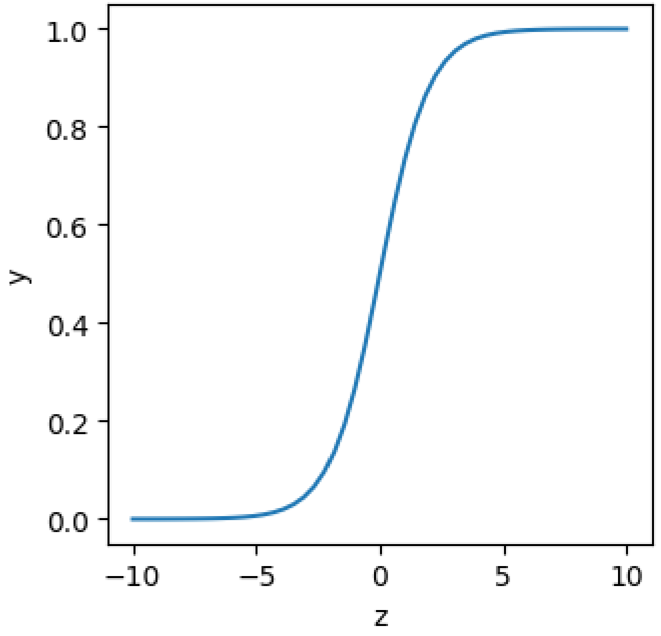
HOSPITAL DA LUZ LEARNING HEALTH
TRAINING, RESEARCH & INNOVATION CENTER

Logistic regression

- Sigmoid function

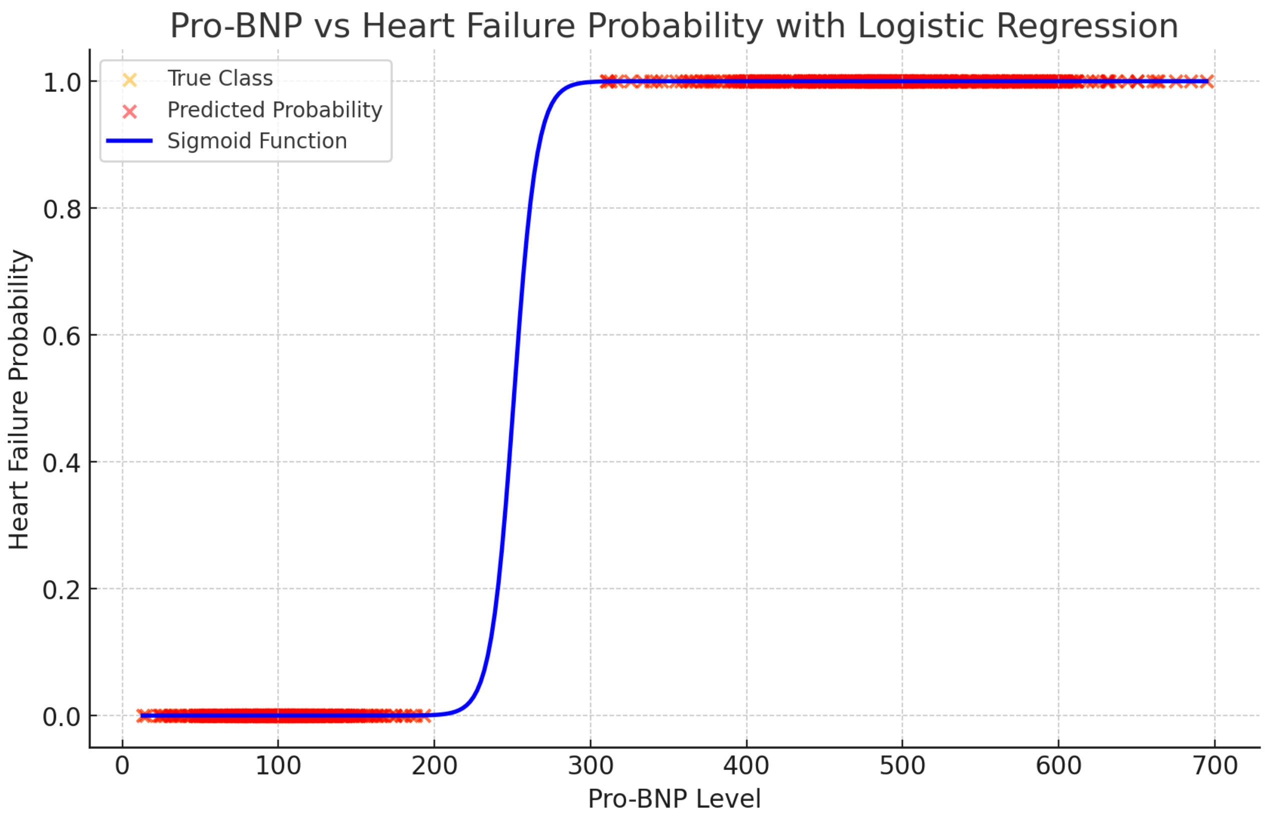
$$y = \frac{1}{1 + e^{-z}}$$

$$z = \beta 0 + \beta 1 X$$



We are still dealing with limited parameters!

Logistic regression



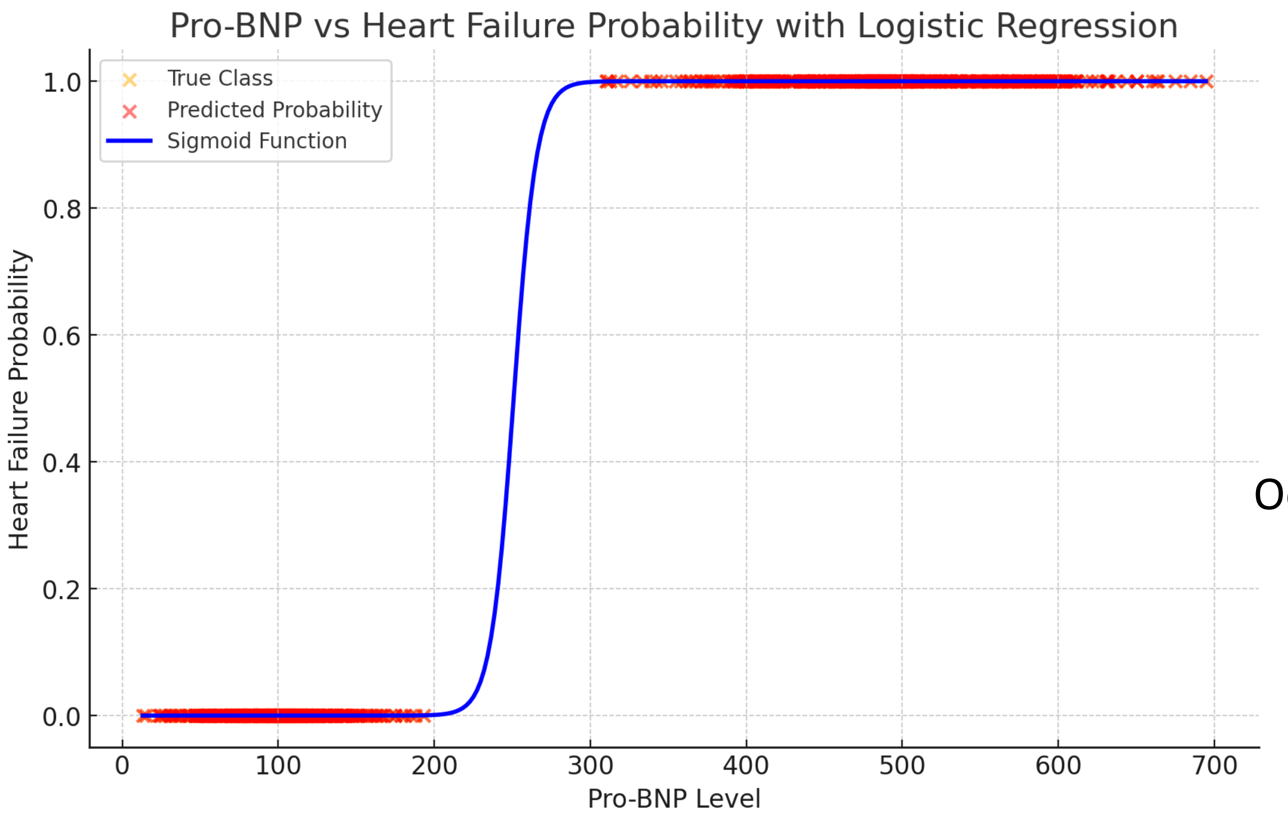
P(*y* = HF | *x* = pro-BNP)

$$P(y=1|X)=\frac{1}{1+e^{-(B_0+B_1X)}}$$

- B_0 (intercept): -33.9817
- B_1 (coefficient for pro-BNP): 0.1355

Now we have a probability between 0 and 1!

Logistic regression



P(y = HF | x = pro-BNP)

$$P(y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$$

$$\downarrow$$

Odds ratio $\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}.$

$$\downarrow$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

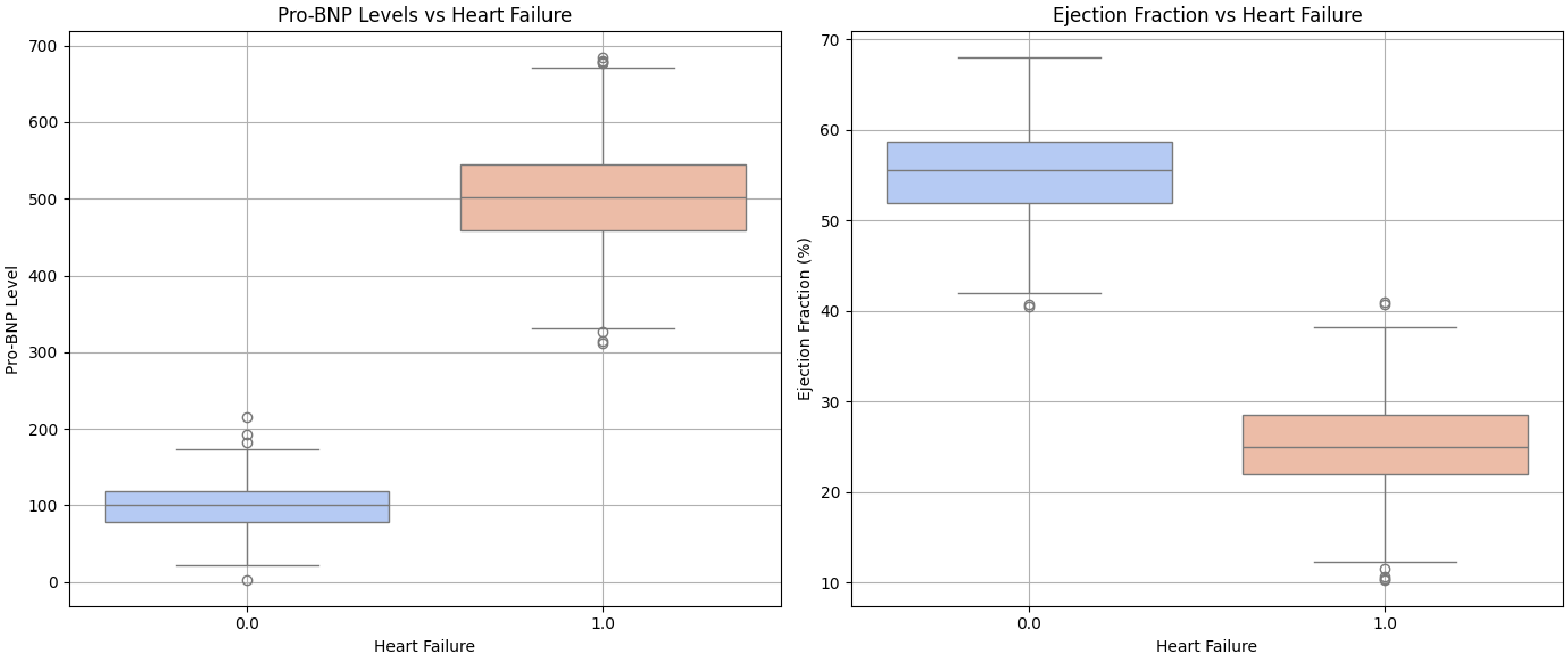
Multiple logistic regression

- Using multiple predictors

$$\hat{y} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}}$$

$$\log\left(\frac{\hat{y}}{1 - \hat{y}}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Multiple logistic regression



Multiple logistic regression

Log Odds Formula:

$$\log\left(\frac{\hat{y}}{1-\hat{y}}\right) = \beta_0 + \beta_1 \cdot \text{Pro\_BNP\_Level} + \beta_2 \cdot \text{Ejection\_Fraction}$$

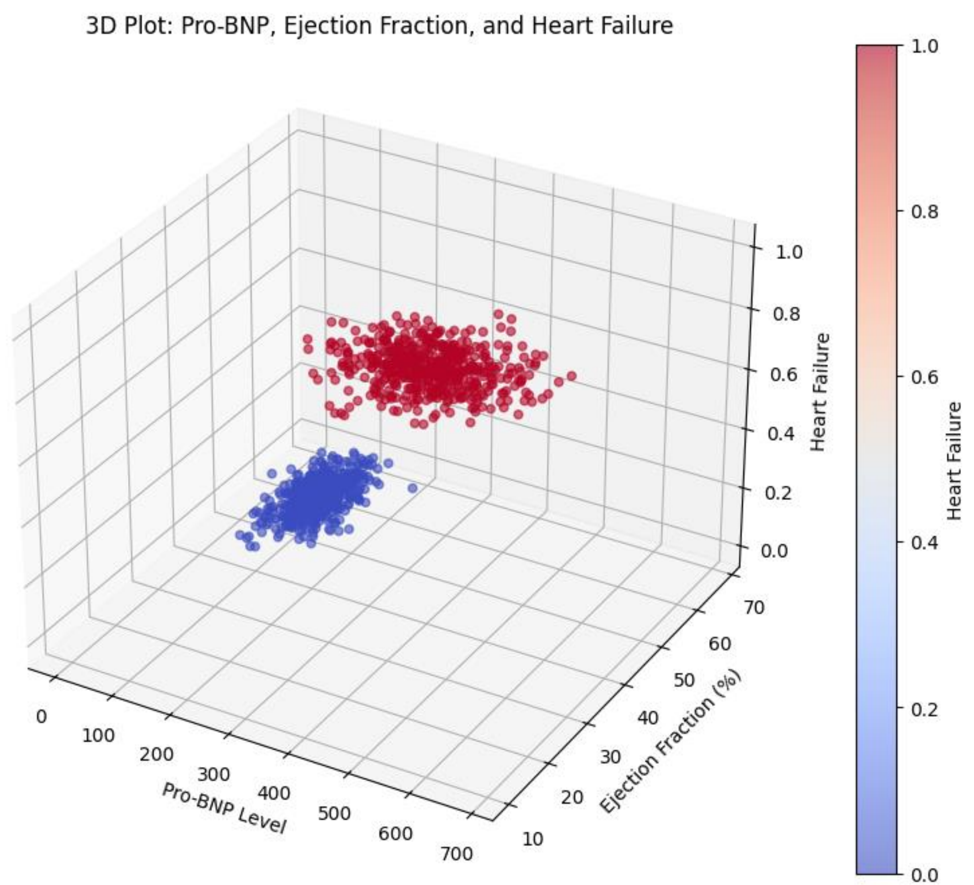
Where:

- $\hat{y}$ is the predicted probability of heart failure.
- β_0 is the intercept term.
- β_1 is the coefficient for Pro-BNP Level.
- β_2 is the coefficient for Ejection Fraction.

Probability Prediction Formula:

$$\hat{y} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{Pro\_BNP\_Level} + \beta_2 \cdot \text{Ejection\_Fraction})}}$$

Multiple logistic regression



Estimating regression coefficients or fitting the model

- Log-likelihood cost function

$$\log L(\beta) = \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

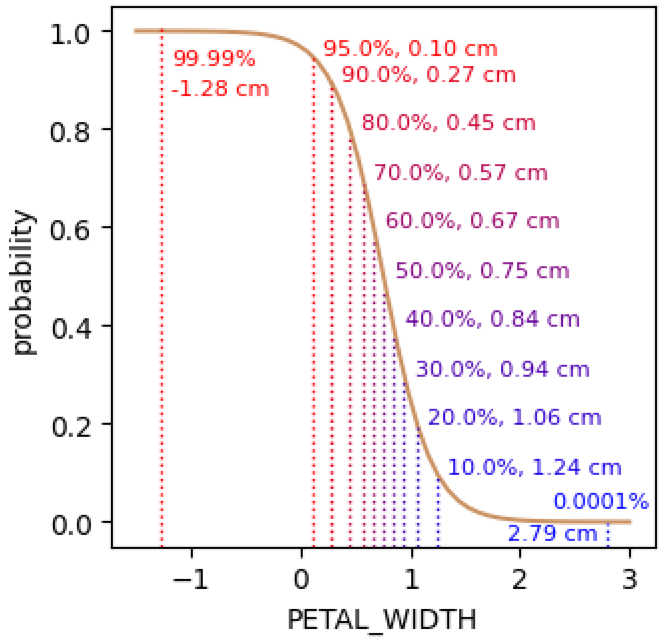
Goal: maximize the log-likelihood cost function (ie. Minimizing the absolute value)

From probabilities to classification

- Different metrics for assessing accuracy
- How are observations "correctly classified"?

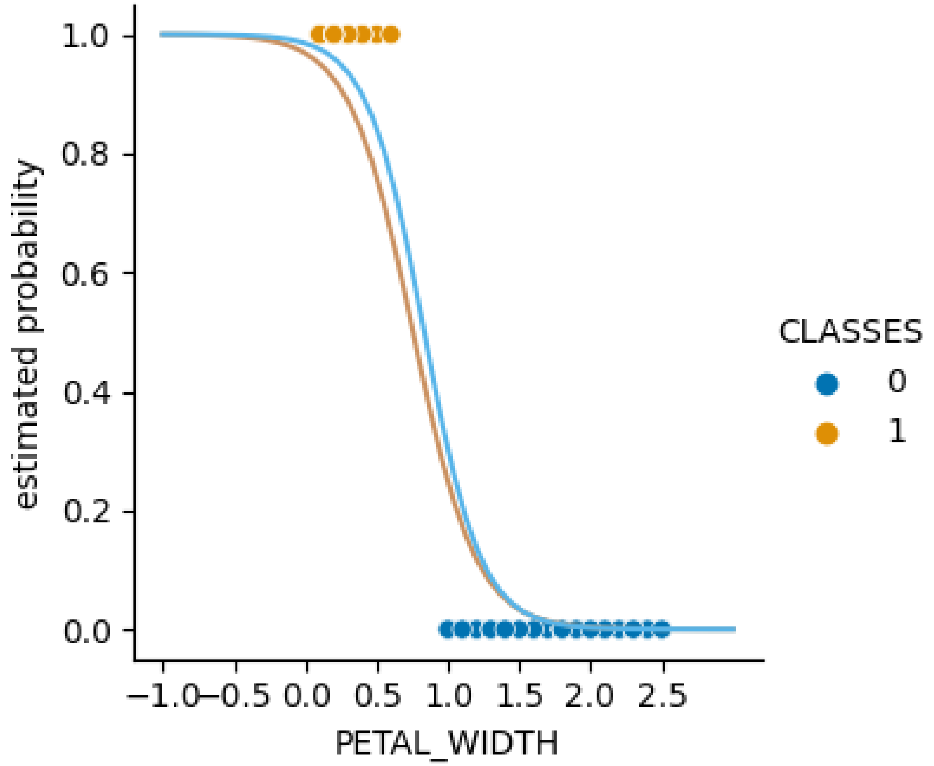
Decision boundary

$$\hat{p} = 0.5 = \frac{1}{1 + e^{-(3.42 - 4.53 * petal\_width)}}$$

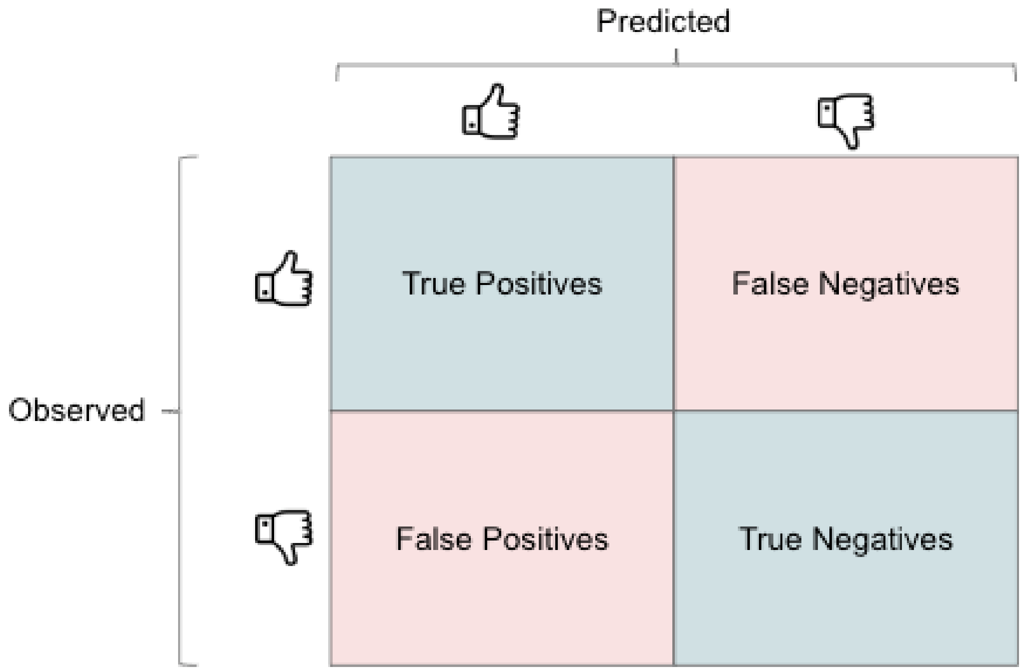


Decision boundary

- Decision boundary depends on the context / application
- Important to understand if datasets are unbalanced



Classification evaluation



Accuracy

Sensitivity /Recall

Precision / True positive value

Specificity

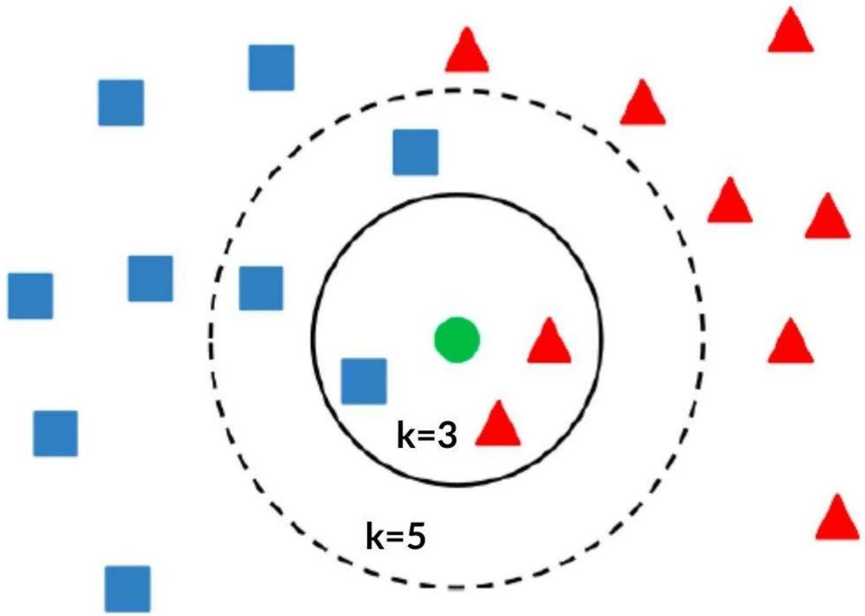
F1 score

...

KNN

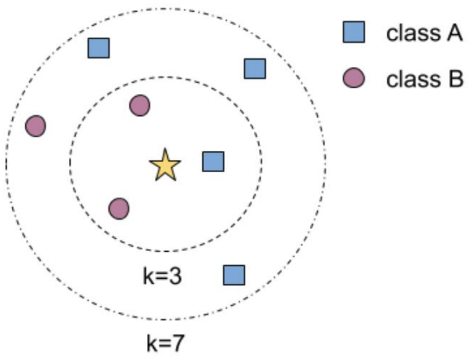
- Simple, non-parametric, lazy learning algorithm used for classification and regression tasks

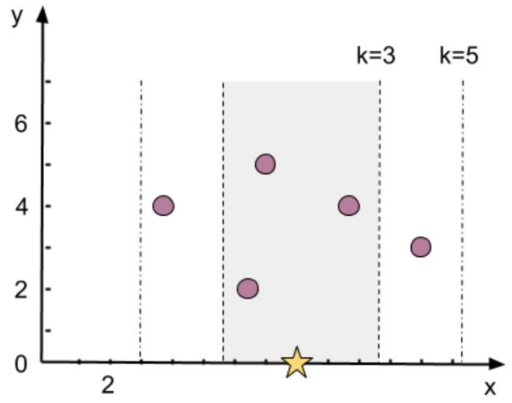
- Classifies a data point based on how its neighbors are classified. In regression, it predicts the value based on the average of its neighbors.



KNN

1. Choose the number of neighbors (K): The number of nearest neighbors to consider.
2. Calculate Distance: Use distance metrics (e.g., Euclidean, Manhattan) to find the K nearest neighbors.
3. Vote for Classification/Calculate for Regression:
   - **Classification**: The class most common among the neighbors is assigned to the data point.
   - **Regression**: The average of the neighbor values is taken as the prediction.





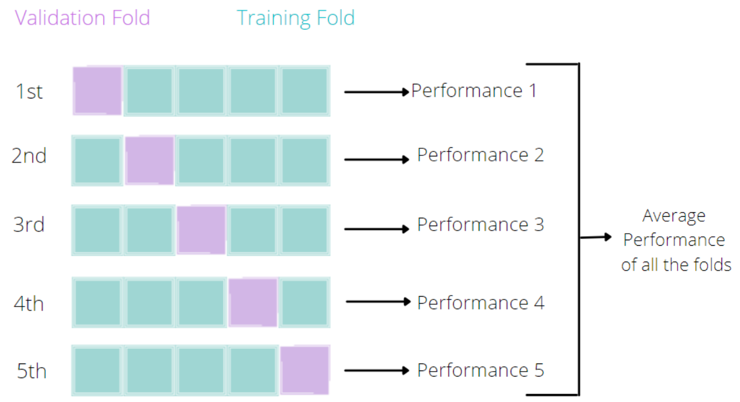
Regression with kNN

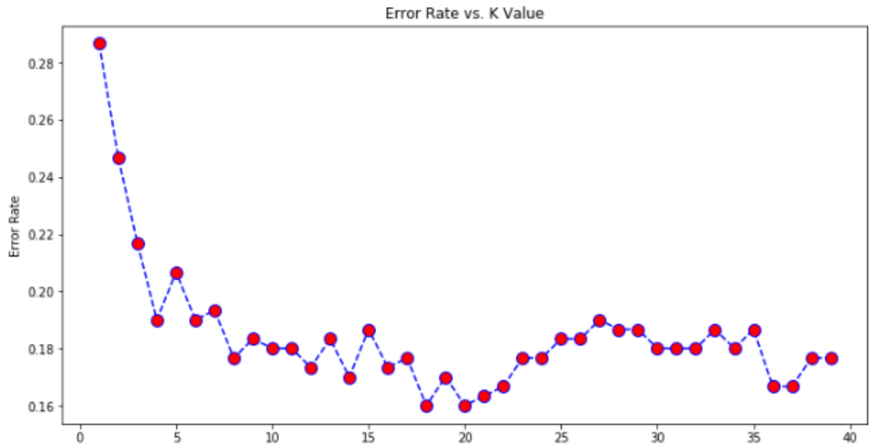
KNN

- How to choose K?

  1. K-fold cross validation

  2. Elbow method





KNN: recap

Pros

- Simple and easy to understand
- No assumptions about data distribution
- Effective with a small number of input variables.

Cons

- Computationally expensive, especially with large datasets
- Sensitive to the scale of data and irrelevant features
- Performance depends on the choice of K and distance metric
- Not really "learning"