Machine Learning

Dr.Hajialiasgari

Tehran University
Of
Medical Science

January 19, 2025

1. Introduction
2. Logistic Regression
3. Evaluation

What is Classification?

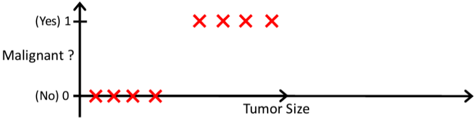
Classification is a supervised learning task where the goal is to predict the category or class label of a given input based on its features. It involves learning a decision boundary that separates different classes in the data.

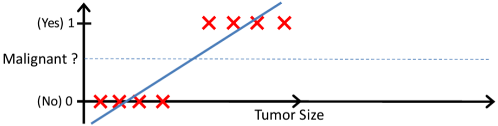
Binary Classification Examples

- **Medical Diagnosis:** Predicting whether a tumor is *benign* or *malignant*.
- **Spam Detection:** Classifying emails as *spam* or *not spam*.
- **Loan Approval:** Predicting whether a loan application is *approved* or *rejected*.
- **Weather Prediction:** Classifying if it will *rain* or *not rain* tomorrow.

Classification Problem

Can we solve this problem by Linear Regression?



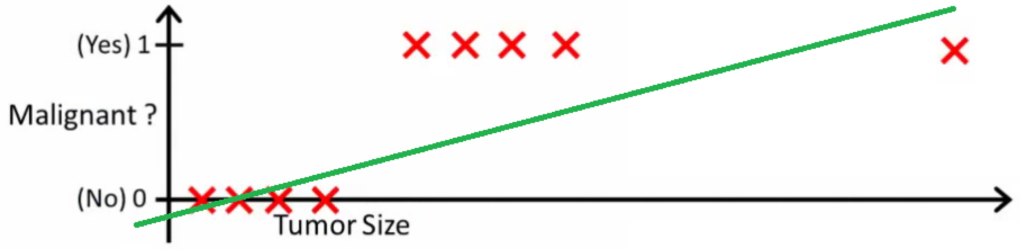


We could fit a straight line and define a threshold at 0.5:

- If $h_\theta(x) \geq 0.5$, predict $y = 1$
- If $h_\theta(x) < 0.5$, predict $y = 0$

Classification Problem

- What about now? (By adding a new data point)



- **Classification:** $y = 0$ or $y = 1$
- $h_\theta(x)$ can be > 1 or < 0
- Another drawback of using linear regression for this problem

What we need:

1. Introduction
2. **Logistic Regression**
3. Evaluation

Introduction

Problem: Distinguish if a person is *overweight* or *not overweight* based on features like age, gender, height, weight, and BMI.

Age	Gender	Height (cm)	Weight (kg)	BMI	Overweight
25	Male	175	80	25.3	0
30	Female	160	60	22.5	0
...	...	...	...	...	...
35	Male	180	90	27.3	1

Notation:

- Features of a sample: **vector** x
- Label: y

Logistic Regression: We try to find a function $\sigma(w^T x)$ that predicts the posterior probabilities $P(y = 1 \mid x)$.

Introduction (cont.)

- $\sigma(w^T x)$: probability that $y = 1$ given x (parameterized by $\mathbf{w}$)

$$P(y = 1|x, \mathbf{w}) = \sigma(\mathbf{w}^T x)$$
$$P(y = 0|x, \mathbf{w}) = 1 - \sigma(\mathbf{w}^T x)$$

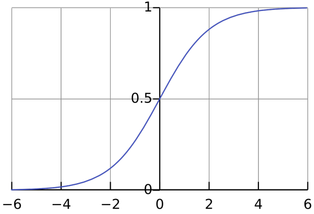
- We need to look for a function which gives us an output in the range [0, 1]. (like a probability).
- Let's denote this function with $\sigma(.)$ and call it the **activation function**.

Introduction (cont.)

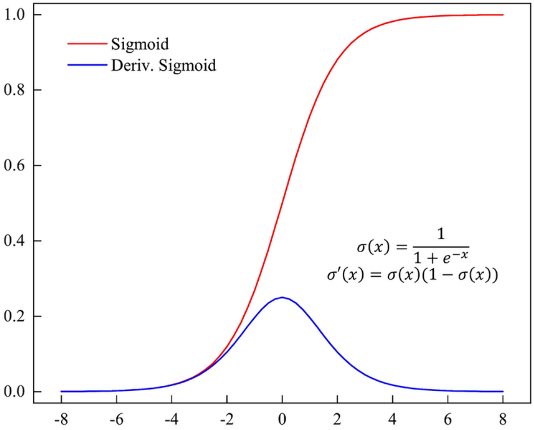
- Sigmoid (logistic) function.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- A good candidate for activation function.
- It gives us a number between 0 and 1 **smoothly**.
- It is also **differentiable**



Sigmoid function & its derivative



Introduction (cont.)

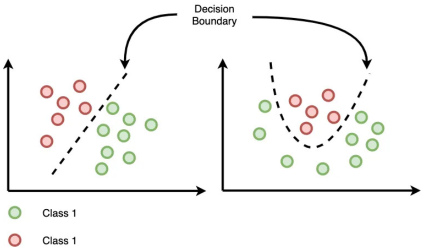
- The sigmoid function takes a number as input but we have:

$$x = [x_0 = 1, x_1, \ldots, x_d]$$
$$w = [w_0, w_1, \ldots, w_d]$$

- So we can use the **dot product** of x and w.
- We have $0 \leq \sigma(\mathbf{w}^T x) \leq 1$. which is the estimated probability of $y = 1$ on input x.
- An Example : A basketball game (Win, Lose)
  - $\sigma(\mathbf{w}^T x) = 0.7$
  - In other terms 70 percent chance of winning the game.

Decision surface

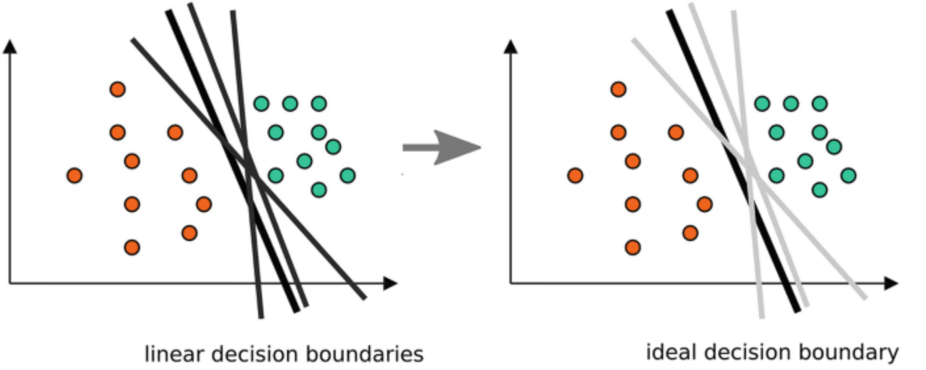
- Decision surface or decision boundary is the region of a problem space in which the output label of a classifier is ambiguous. (could be linear or non-linear)
- In binary classification it is where the probability of a sample belonging to each $y = 0$ and $y = 1$ is equal.



- Decision boundary hyperplane always has **one less dimension** than the feature space.

Decision surface (cont.)

- An example of linear decision boundaries:



Decision surface (cont.)

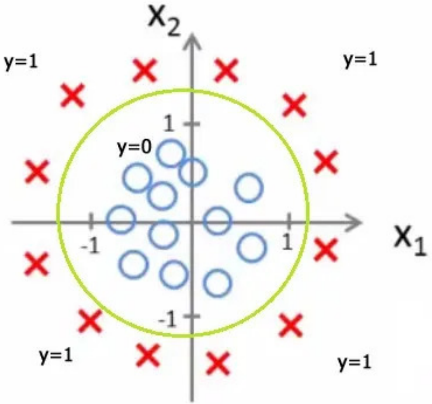
- Back to our logistic regression problem.
- Decision surface $\sigma(\mathbf{w}^T x)$ = **constant**.

$$\sigma(\mathbf{w}^T x) = \frac{1}{1 + e^{-(\mathbf{w}^T x)}} = 0.5$$

- Decision surfaces are **linear functions** of x
  - if $\sigma(\mathbf{w}^T x) \geq 0.5$ then $\hat{y} = 1$, else $\hat{y} = 0$
  - Equivalently, if $\mathbf{w}^T x + w_0 \geq 0.5$ then decide $\hat{y} = 1$, else $\hat{y} = 0$

$\hat{y}$ is the predicted label

Non-Linear Boundary Example



Predict $y = 1$ if $-1 + x_12 + x_22 \geq 0$

1. Introduction
2. Logistic Regression
3. Evaluation

Quote of the Day

It doesn't matter how beautiful your theory is ... If it doesn't agree with experiment, it's wrong.

Richard Feynman

Confusion Matrix and Accuracy

Confusion Matrix:

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Example: Cancer Detection

- **Scenario:** Predicting whether a tumor is malignant (1) or benign (0).
- **Confusion Matrix:**

	Predicted Malignant	Predicted Benign
Actual Malignant	50 (TP)	10 (FN)
Actual Benign	5 (FP)	85 (TN)

Calculations:

- **Accuracy:** $\frac{50+85}{50+85+10+5} = 0.935\ (93.5\%)$.
- **Precision:** $\frac{50}{50+5} = 0.909\ (90.9\%)$.
- **Recall:** $\frac{50}{50+10} = 0.833\ (83.3\%)$.

Precision and Recall

Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Focus: How many positive predictions are actually correct.

Recall (Sensitivity):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Focus: How many actual positives are correctly identified.

F1 Score

F1 Score:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Use: Balances Precision and Recall, especially in imbalanced datasets.

ROC Curve and AUC

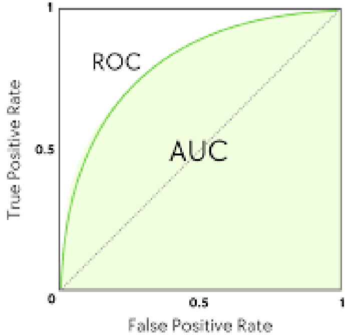
ROC Curve:

- Plots True Positive Rate (TPR) vs. False Positive Rate (FPR).
- Evaluates classifier performance at various thresholds.

AUC:

- Measures the overall ability of the model to distinguish between classes.
- Range: 0.5 (No discrimination) to 1.0 (Perfect discrimination).

ROC-AUC



Log-Loss (Logarithmic Loss)

Formula:

$$\text{Log-Loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log(p^{(i)}) + (1-y^{(i)})\log(1-p^{(i)})\right]$$

Use: Penalizes incorrect predictions with confidence.

Specificity (True Negative Rate)

Formula:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Focus: How many negatives are correctly identified.

Summary of Evaluation Metrics

Metric	Formula	Use Case
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall correctness
Precision	$\frac{TP}{TP+FP}$	Positive prediction quality
Recall	$\frac{TP}{TP+FN}$	Positive detection rate
F1 Score	$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$	Balancing Precision and Recall
Specificity	$\frac{TN}{TN+FP}$	Negative detection rate
Log-Loss	$-\frac{1}{n}\sum \log$	Probabilistic predictions

For more information and code check the related notebook

End of Classification