KAZAKH-BRITISH TECHNICAL UNIVERSITY

Introduction to Machine Learning
Week 9

Olivier JAYLET

School of Information Technology and Engineering

Classification problems

Some examples:

- spam detection in e-mails (ham or spam)
- fraud detection in online transactions (fraudulent, safe)
- tumor diagnosis (malignant, benign)
- heart rhythm diagnosis (normal sinus rhythm, atrial fibrillation)
- ...

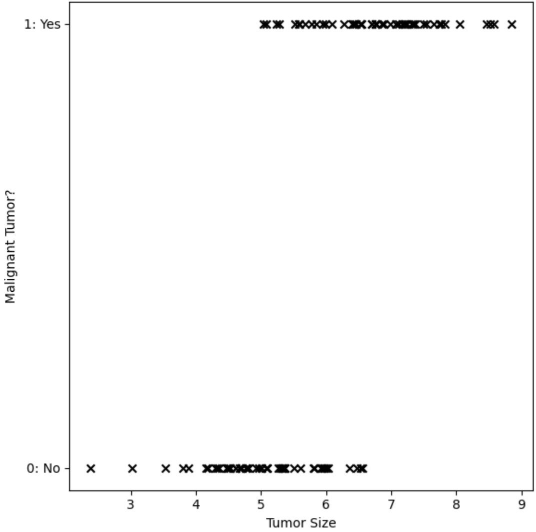
Classification problems

In a classification problem, the variable to be explained (or target variable) y is categorical, we note :

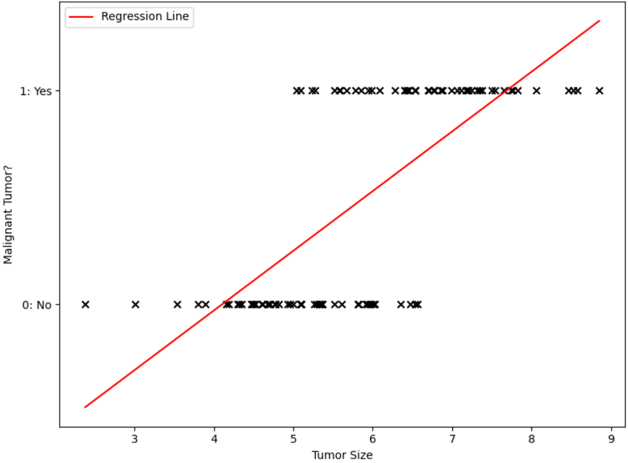
- **Binary classification**: y
  - 0 : negative class (ex: normal sinus rhythm)
  - 1 : positive class (ex: atrial fibrillation)
- **Multiclass classification**:

$$y \in \{0, 1, 2, \ldots, K\}, \; K \geq 2$$

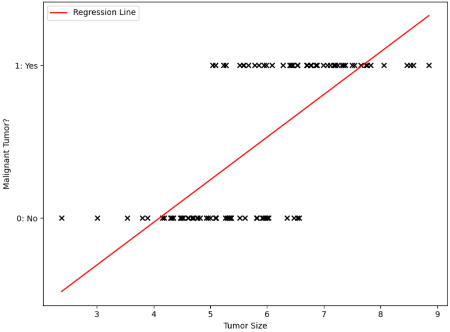
Example of binary classification



Fit a linear regression on a classification



Fit a linear regression on a classification problem



Let $f(x) = \theta^T x$, we define the following **classification rule**:

- if $f(x) \geq 0.5$, we predict $y = 1$
- if $f(x) < 0.5$, we predict $y = 0$

Predicted Probabilities Properties

The predicted probabilities should :

- be in range [0,1]
- add up to 1
- interpretable and easy to understand
- Well calibrated (monotonically related to the true (or empirical) probabilities)

The log model

We want to choose a function $f_\theta(x)$ such that :

$$0 \leq f_\theta(x) \leq 1 \tag{1}$$

We set:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{2}$$

$$p(X) = \frac{e^z}{1 + e^z} = \frac{e^z/e^z}{(1/e^z) + (e^z/e^z)} \tag{3}$$

This simplifies to:

$$p(X) = \frac{1}{1 + e^{-z}} \tag{4}$$

where $e^{-z} = \frac{1}{e^z}$.

Representation of the model

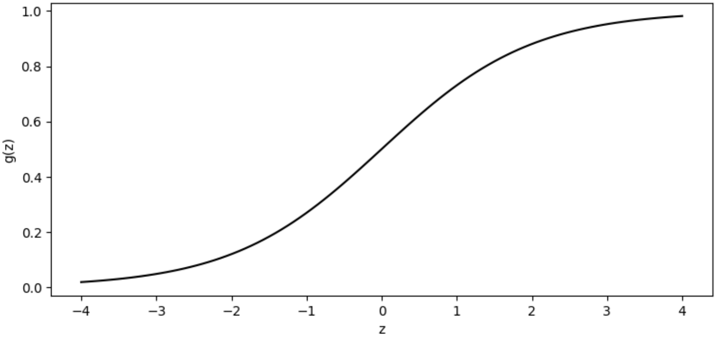
$$f_\theta(x) = \sigma(\theta^T x)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ σ is the **sigmoid** function, that is :

$f_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ **logistic** function.

Sigmoid function

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



Interpretation of $f_\theta(x)$

$f_\theta(x)$ is the **estimated probability** that $y = 1$ for the observation x.

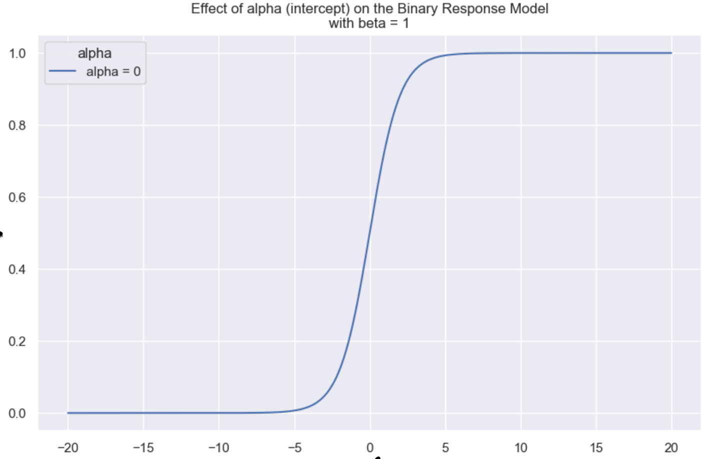
Example: $f_\theta(x) = 0.8$ means that the individual has a 80% chance of having a malignant tumor.

We denote:

$$f_\theta(x) = P(y = 1|x;\theta)$$
with,
$$P(y = 0|x;\theta) + P(y = 1|x;\theta) = 1$$
$$P(y = 0|x;\theta) = 1 - P(y = 1|x;\theta)$$

Effect of varying α = β_0 = Constant

$\sigma(z)$



x

z

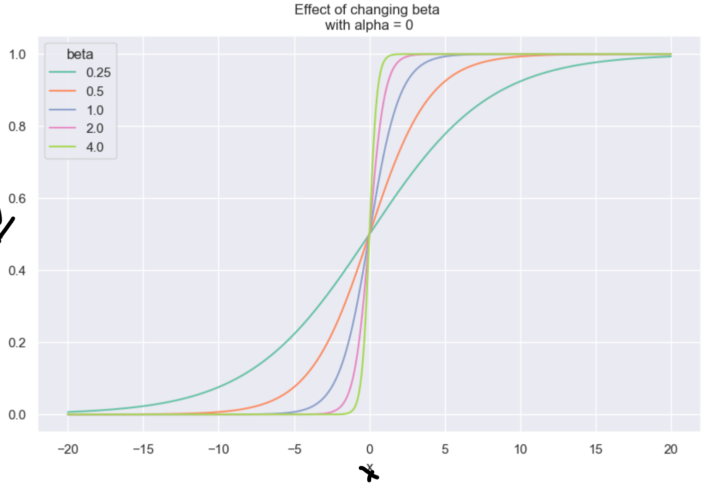
Effect of varying α $= \beta_0 =$ constant

$\sigma(z)$

z

Effect of varying β

$\sigma(z)$



z

Classification rule

- Predict $y = 1$ if $f_\theta(x) \geq 0.5$
  - as $\sigma(z) \geq 0.5$ when $z \geq 0$
  - we have: $\sigma(\theta^T x) \geq 0.5$ when $\theta^T x \geq 0$

- Predict $y = 0$ if $f_\theta(x) < 0.5$
  - as $\sigma(z) < 0.5$ when $z < 0$
  - we have: $\sigma(\theta^T x) < 0.5$ when $\theta^T x < 0$

Reminder Probability

We have:

$$P(y = 1|x; \theta) = f_\theta(x)$$
$$P(y = 0|x; \theta) = 1 - f_\theta(x)$$

That can be written as:

$$p(y|x; \theta) = f_\theta(x)^y (1 - f_\theta(x))^{1-y}$$

Maximum likelihood estimation

We try to find the value of θ that **maximizes the likelihood**:

$$L(\theta) = \prod_{i=1}^{n} P(y_i|x_i;\theta) = \prod_{i=1}^{n} f_\theta(x_i)^{y_i}\left(1 - f_\theta(x_i)\right)^{1-y_i} \quad (5)$$

Maximum log-likelihood estimation

We try to find the value of θ that **maximizes the likelihood**:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} P(y_i | x_i; \theta) = \prod_{i=1}^{n} f_\theta(x_i)^{y_i} \left(1 - f_\theta(x_i)\right)^{1-y_i} \tag{6}$$

We consider instead the **log-likelihood**:

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{7}$$

Cost function

By defining the cost function $\mathcal{L}(\theta)$ as follows:

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{8}$$

We search for the vector of parameters θ which **minimizes** $\mathcal{L}(\theta)$.

Odds

The probability of the positive class is given by :

$$P(Y = 1|x) = \frac{e^z}{1 + e^z} \tag{9}$$

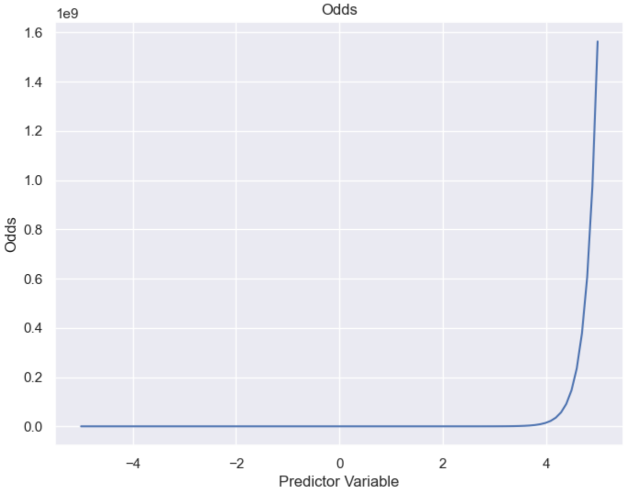
And we can compute the odds by :

$$\frac{P(Y = 1|x)}{P(Y = 0|x)} = e^{\theta^T x} \tag{10}$$

Odds are the probabilities that one event will happen instead of another.
Values of odds :

- close to 0 : low probability
- going to ∞ : very high probability

Odds plot


Odds
1e9
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Odds
−4
−2
0
2
4
Predictor Variable

Interpreting Odds

Example: Logistic regression model for disease probability based on age

$$\text{odds} = e^{-2.5+0.05\cdot\text{age}}$$

- Odds at age 40: $e^{-2.5+0.05\cdot 40} \approx 0.607$
- Odds at age 50: $e^{-2.5+0.05\cdot 50} \approx 1$

Challenge: Interpreting odds directly can be complicated due to the exponential relationship.

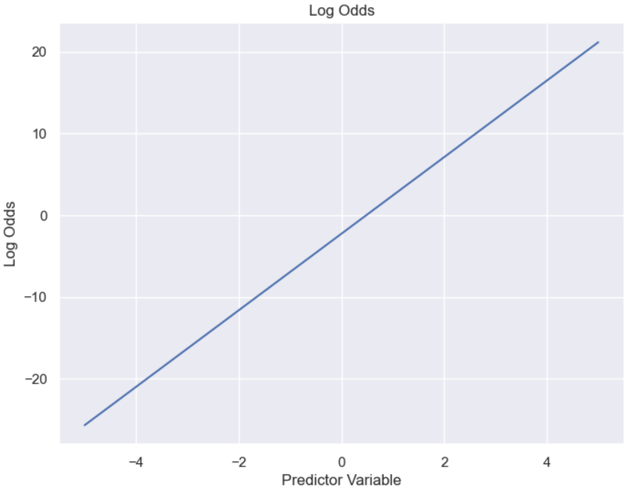
Transformation log-odds

Since the odds are exponential, we can linearize them by taking their log(.).

$$\log\left(\frac{P(Y=1|x)}{P(Y=0|x)}\right) = \log\left(\frac{p}{1-p}\right) = \theta^T x \tag{11}$$

- A log odds of 0 corresponds to a 50% predicted probability
- A positive log odds corresponds to a probability greater than 50%,
- A negative log odds corresponds to a probability lower than 50%,

Log-Odds plot



Interpreting Log-Odds

Example: Same logistic regression model, but using log-odds

$$\text{log-odds} = -2.5 + 0.05 \cdot \text{age}$$

- Log-odds at age 40: $-2.5 + 0.05 \cdot 40 = -0.5$
- Log-odds at age 50: $-2.5 + 0.05 \cdot 50 = 0$

Advantage: The log-odds increase linearly with age by 0.05 per year

Multiclass classification

In a multiclass classification problem we have:

$$y \in \{1, 2, \ldots, K\}, K > 2$$

Example of multiclass classification:

- 'Bad', 'average' and 'good' students
- Nationalities : Kazakh, Tatar, Russian, Ukrainian, Uzbek, Uighur, etc...

Multiclass classification

Two different approach :

- One Versus rest (All)
- Softmax

One Versus Rest (OVR)

Lets assume we have K different categories. In OVR strategy, we train K classifiers with $y \in \{0, 1\}$, where each classifier considers another k as the positive class.
We then get k classification models:

$$
\begin{array}{ll}
f_1(x_j; \theta_1) & \text{Positive class} : 0 \\
f_2(x_j; \theta_2) & \text{Positive class} : 1 \\
f_3(x_j; \theta_3) & \text{Positive class} : 2 \qquad \arg\max_k \; f_k(x_j; \theta_k) \\
\vdots & \vdots \quad \vdots \\
f_k(x_j; \theta_k) & \text{Positive class} : k
\end{array}
$$

This method works, but we lose a valid probabilistic interpretation.

Softmax

- Lets create z as a vector of scores (one for each class):
  $z = [z_1, z_2, \ldots, z_K] = \left[\theta_1^T x,\ \theta_2^T x,\ \ldots, \theta_K^T x\right].$
- These scores z are **not probabilities** yet—they need to be normalized.
- Apply softmax to z to create probabilities:
  $p_k = \text{softmax}(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}},$
  where p_k is the probability for class k.
- The softmax of an input vector
  $z = [z_1, z_2, \ldots, z_K]$ is thus a vector itself:
  $\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{K} \exp(z_i)} \quad \frac{\exp(z_2)}{\sum_{i=1}^{K} \exp(z_i)} \quad \cdots \quad \frac{\exp(z_K)}{\sum_{i=1}^{K} \exp(z_i)}\right]$

Confusion matrix

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

This matrix gives four different informations :

- tp : The model predicted a *True* event while it was actually *True*.
- tn : The model predicted a *False* event while it was actually *False*.
- fp : The model predicted a *False* event while it was actually *True*.
- fn : The model predicted a *True* event while it was actually *False*.