COESIONE ITALIA 21-27 EMILIA-ROMAGNA

Cofinanziato dall'Unione europea

GUARDA AVANTI

Big Data, nuove competenze per nuove professioni

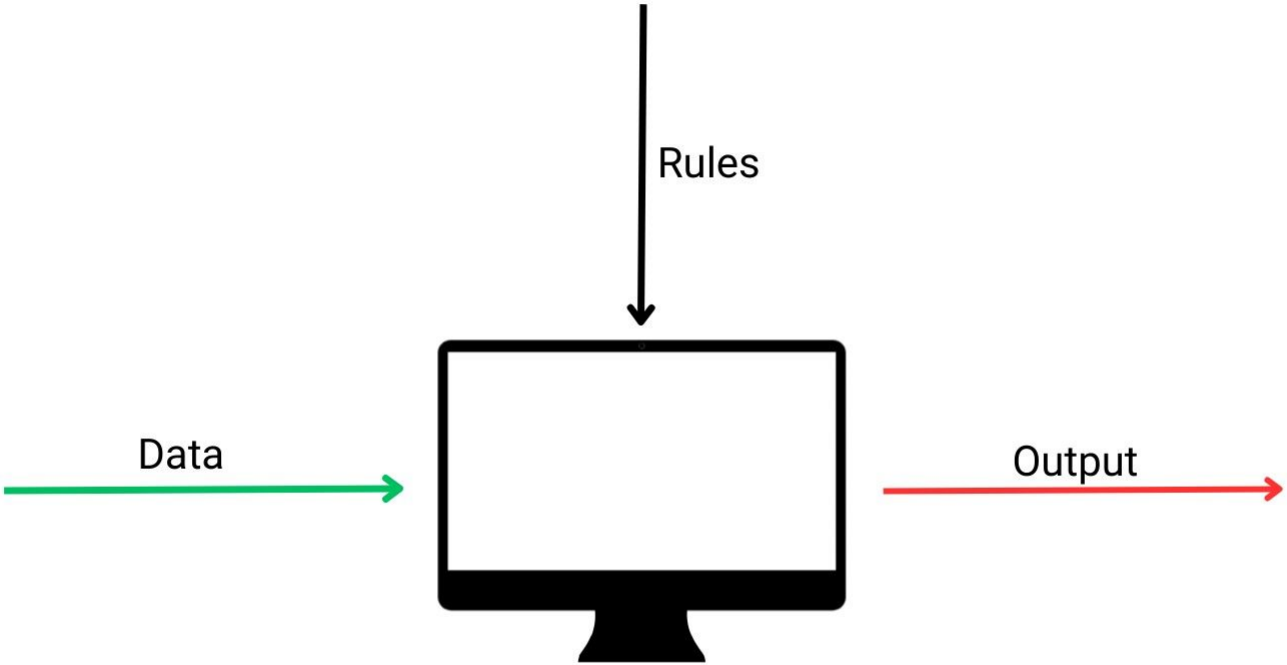
(Progetto rivolto a laureati in tutte le aree disciplinari, co-finanziato dal Fondo Sociale Europeo Plus 2021-2027 Regione Emilia-Romagna)

DATA LAB

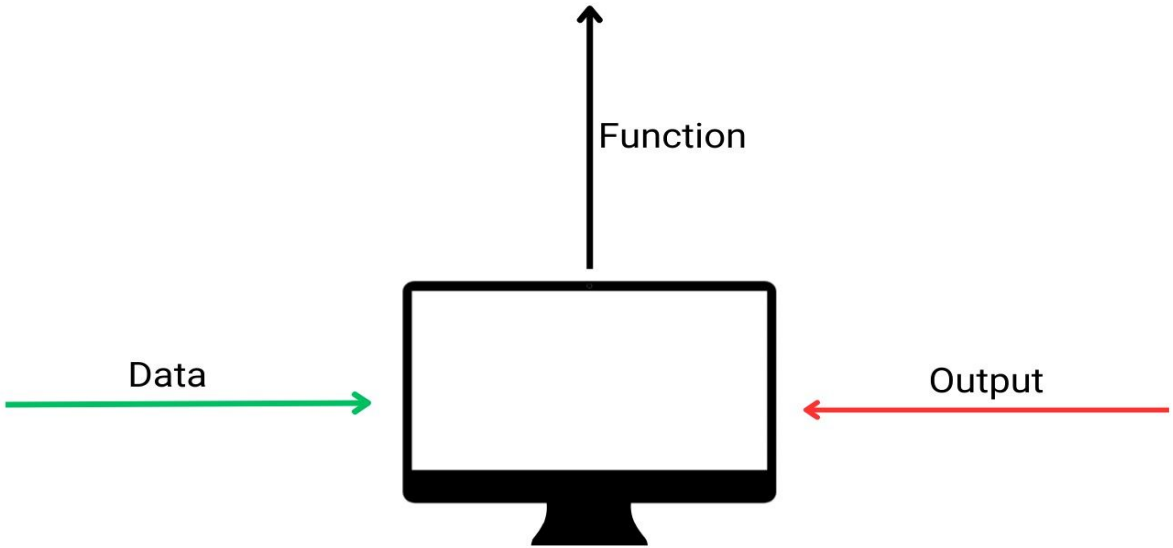
Programma della lezione

- Panoramica del Machine Learning
- 2 modelli di machine learning
  - KNN
  - SVM
- Metriche per valutare il modello

Programmazione tradizionale



Machine Learning



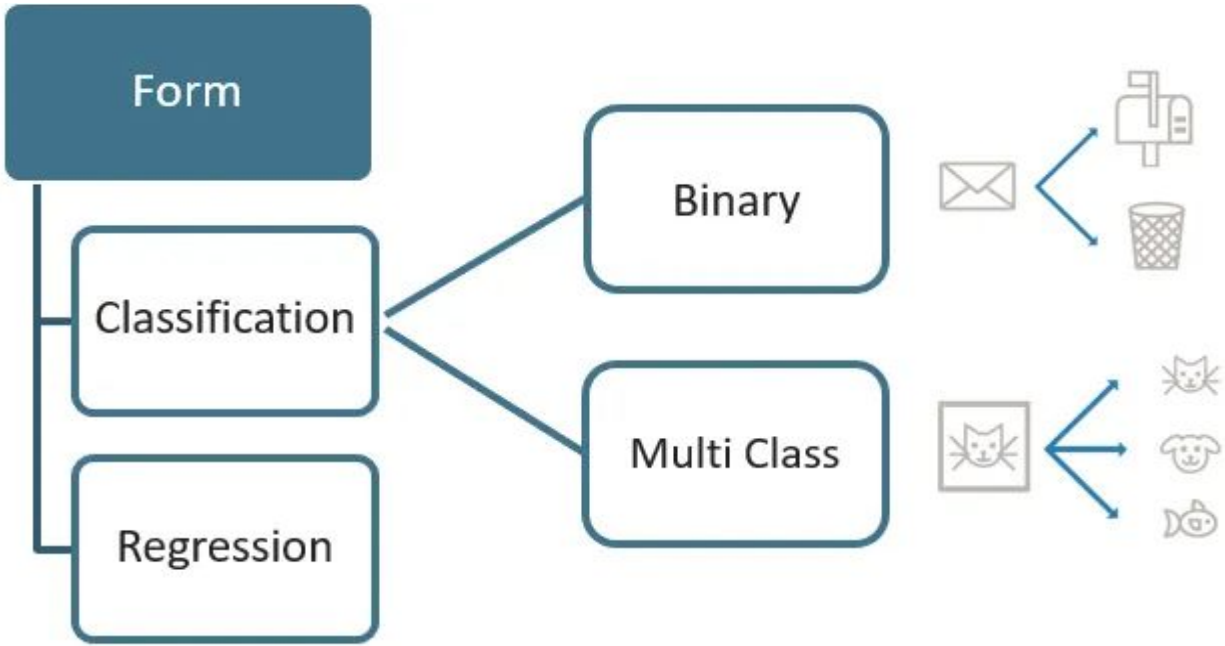
Machine Learning

$$y = f(x)$$

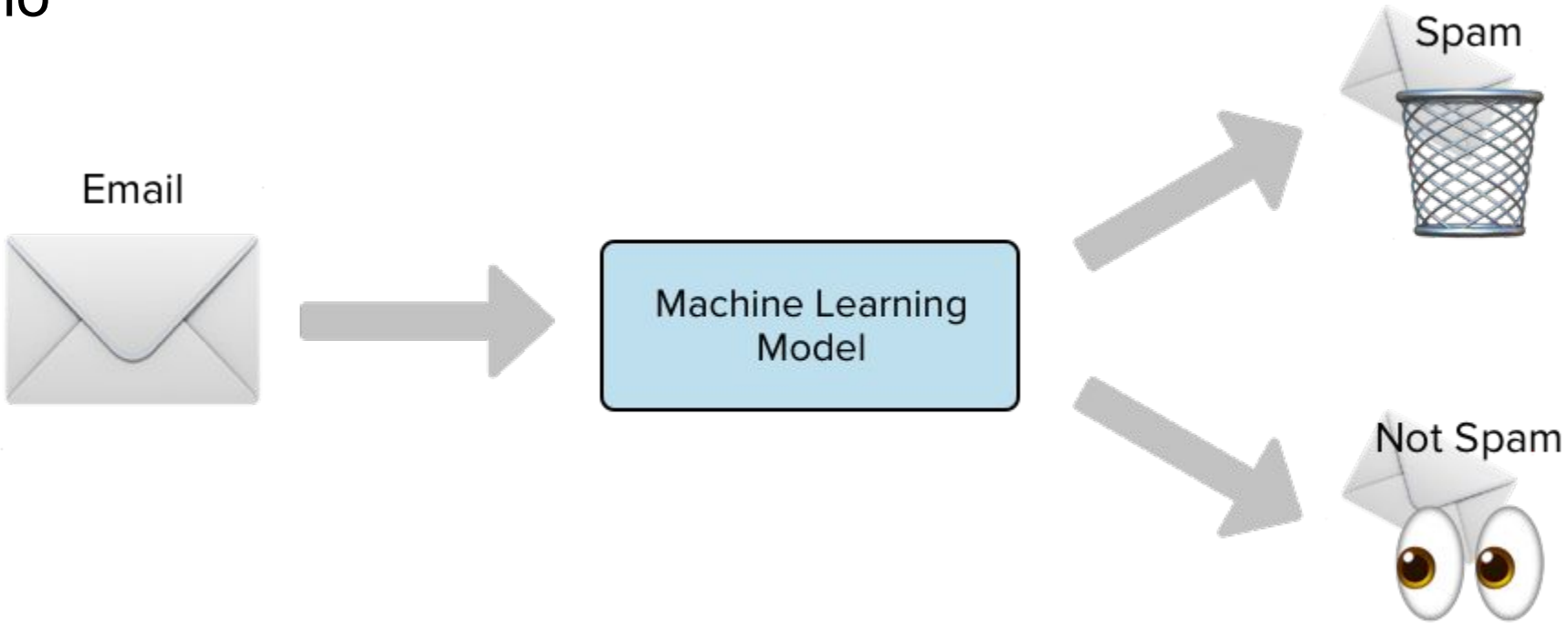
Machine Learning

$$y = f(x) + \varepsilon$$

Esistono due tipi di Supervised Learning

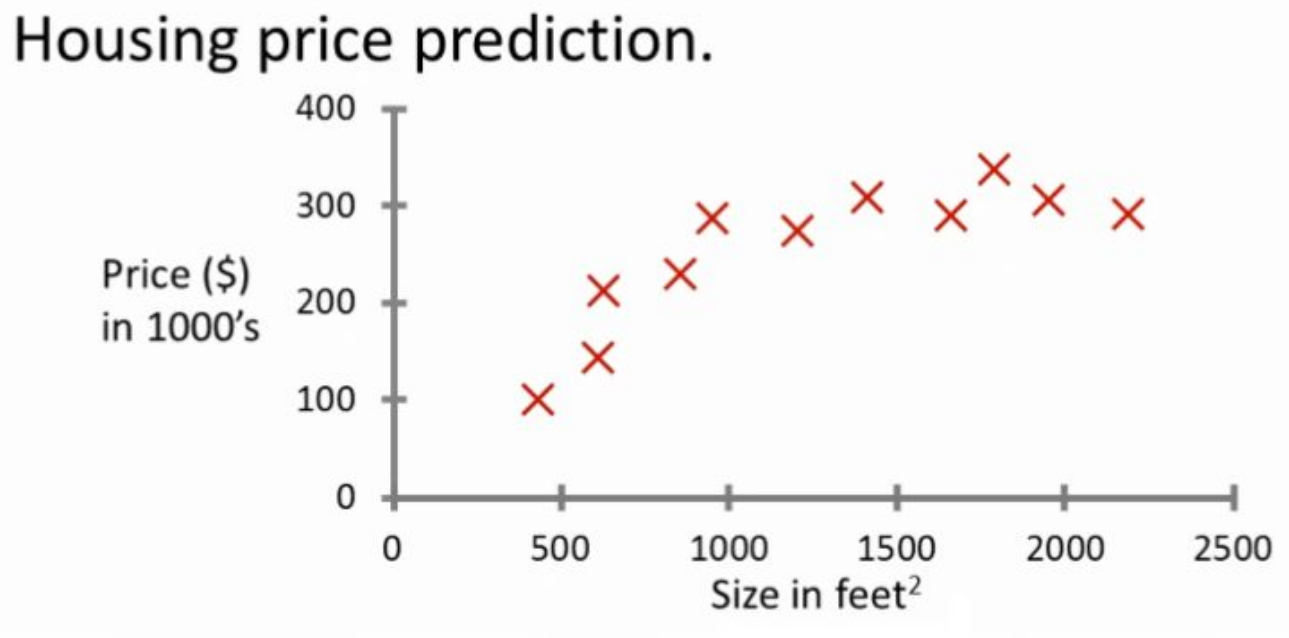


Esempio: vogliamo prevedere se un’email è uno spam o no



È un problema di regressione o classificazione?

Esempio: vogliamo prevedere il prezzo di una casa, data la sua dimensione

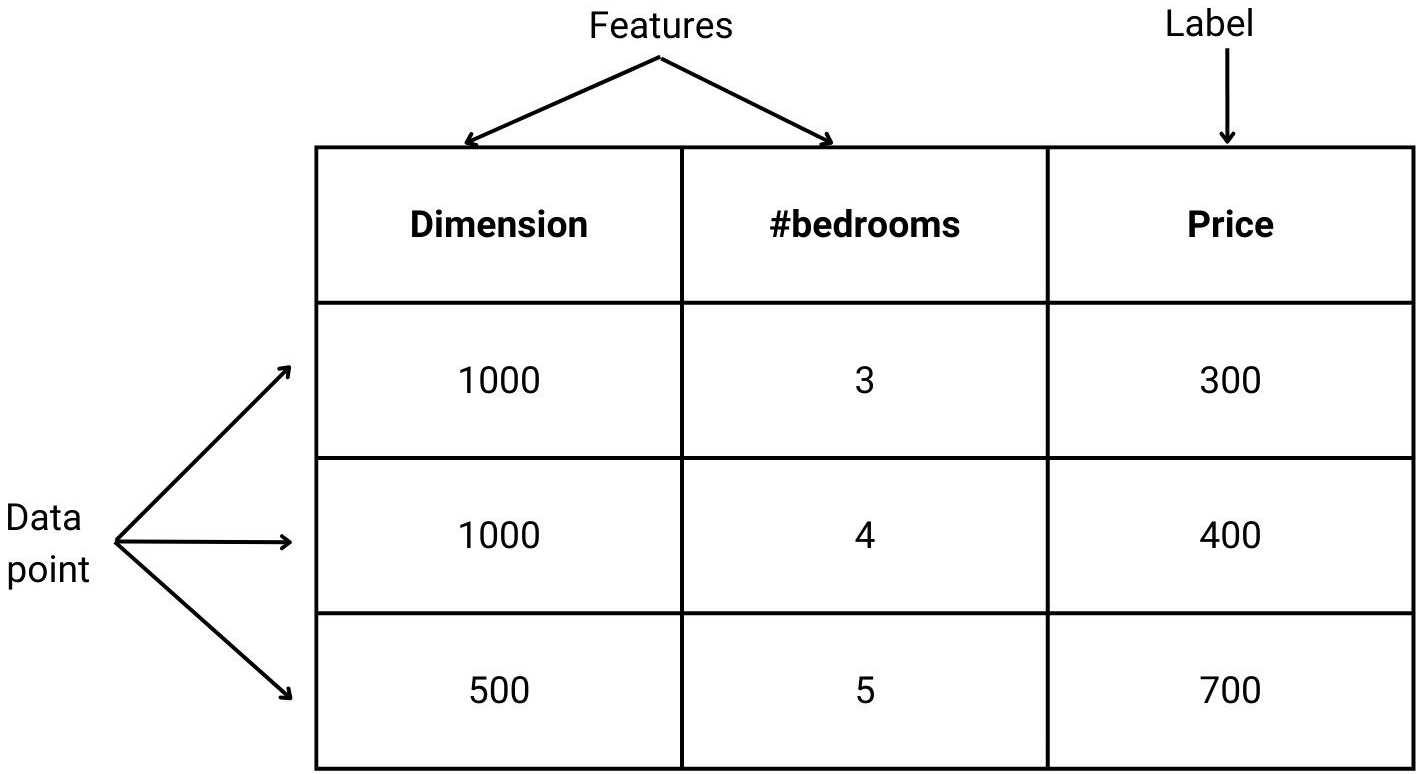


È un problema di regressione o classificazione?

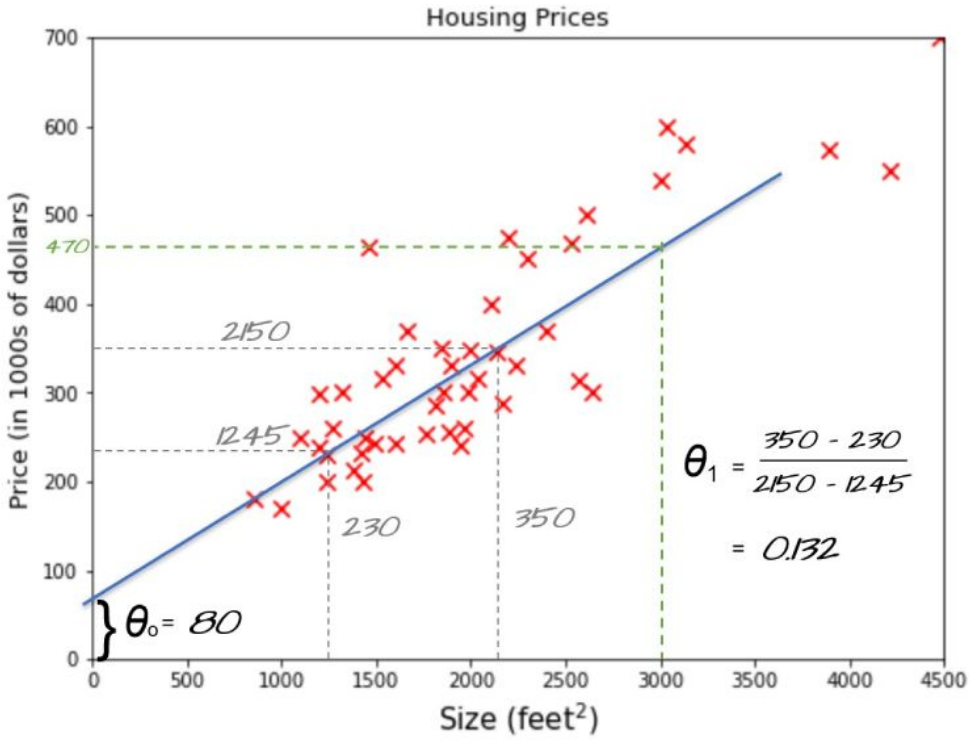
Alcune notazioni:

- Il valore che vogliamo prevedere (prezzo) si chiama **label/target**
- I valori che uso come input (dimensione) si chiamano **features**
- Ogni campione/record si chiama **data point**

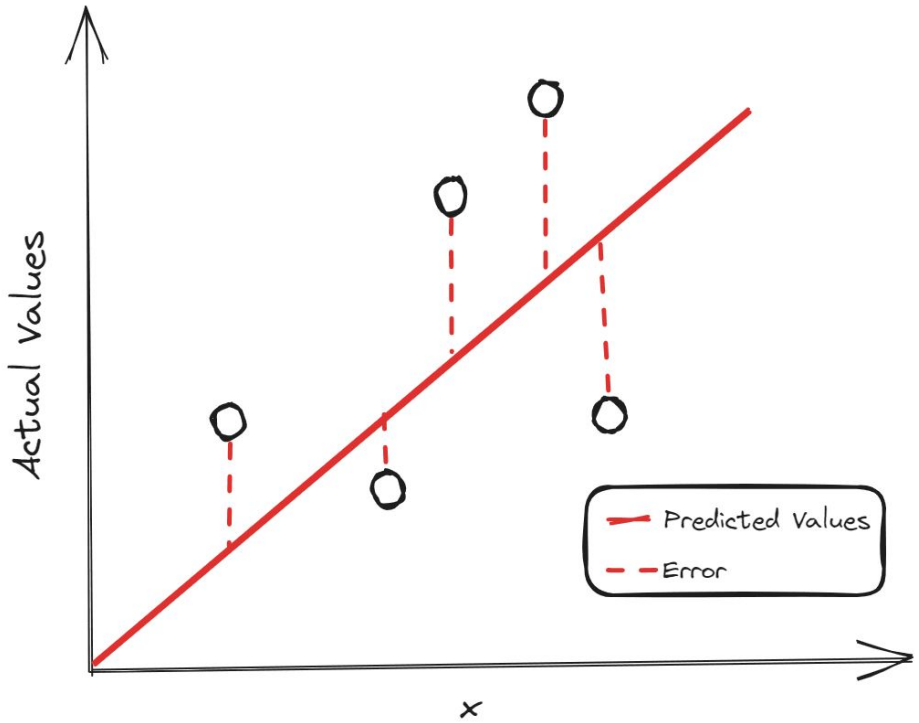
Dimension	#bedrooms	Price
1000	3	300
1000	4	400
500	5	700


Features
Label
Dimension
#bedrooms
Price
1000
3
300
1000
4
400
500
5
700
Data point

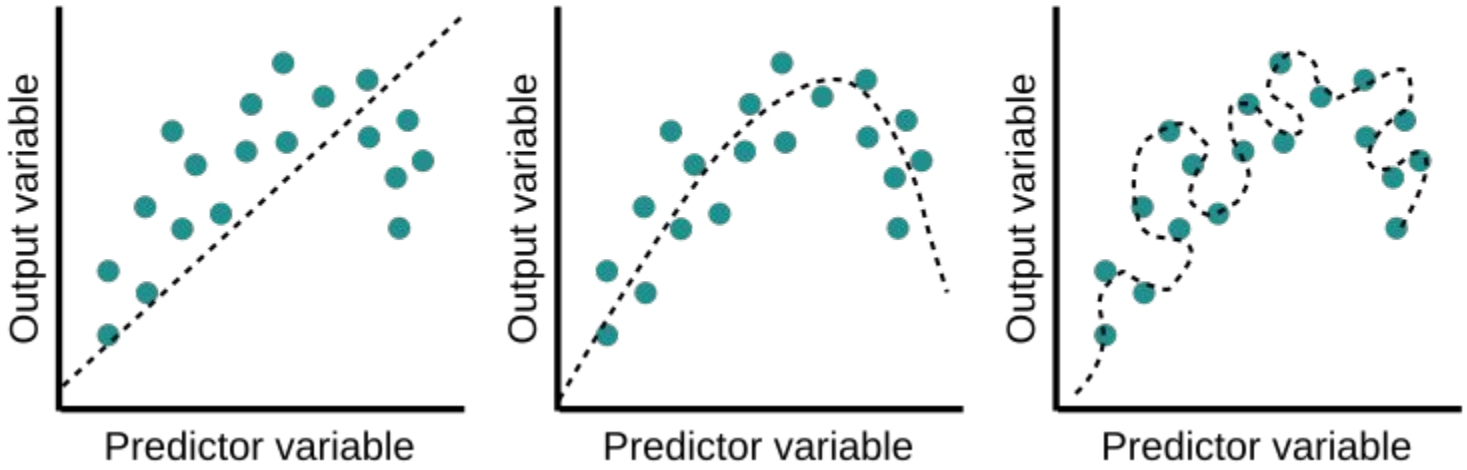
Esempio: funzione di fitting



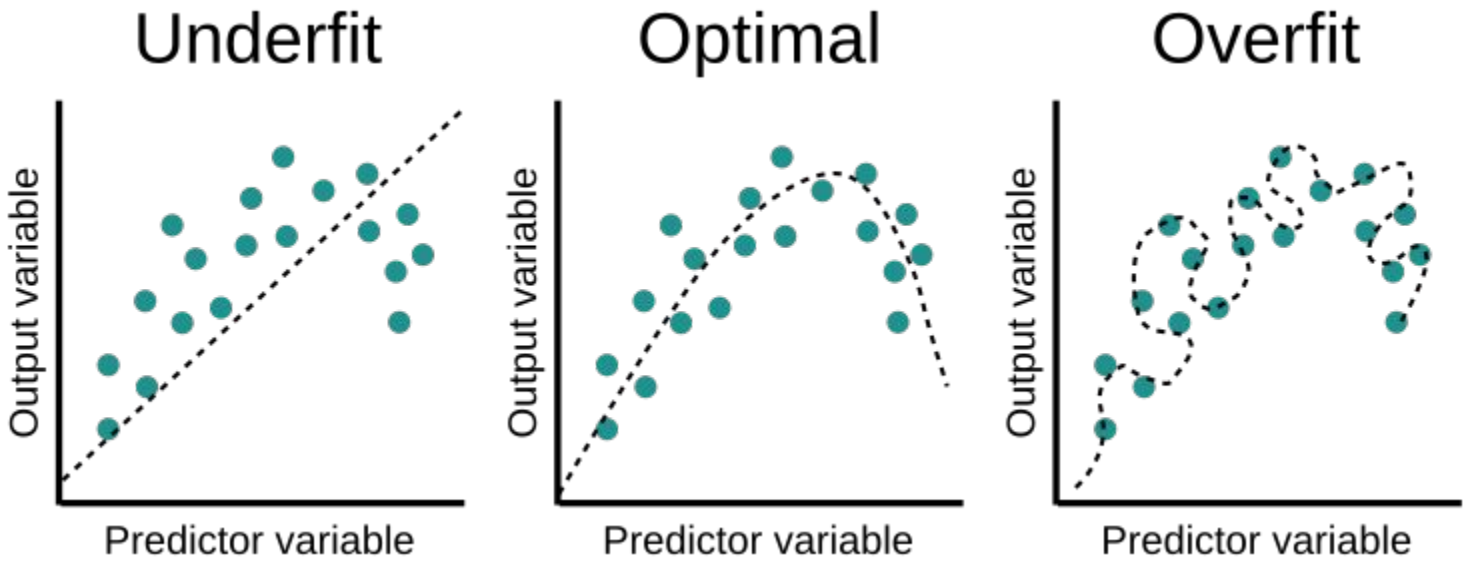
Come valutiamo il nostro modello? Funzione di costo



Quali di questi modelli ha il costo migliore?



Quali di questi modelli ha il costo migliore?



Soluzione: dividiamo i nosti dati in un “training set” e un “test set”

Notazione

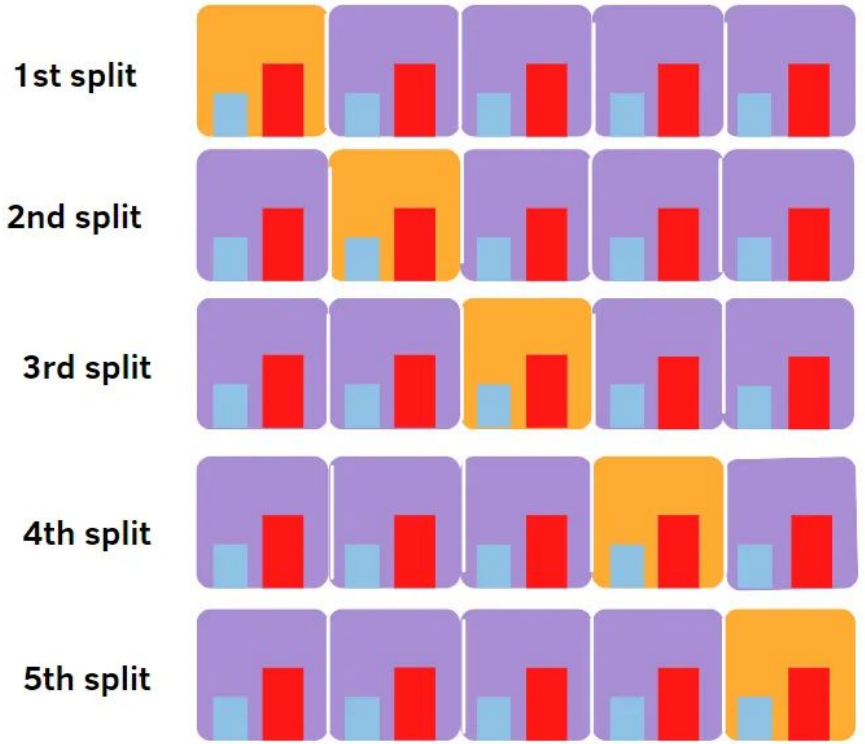
- **Train set** è un sottoinsieme del dataset usato per allenare il modello (di solito 75% del dataset)
- **Test set** è un sottoinsieme del dataset usato per valutare per il modello addestrato (di solito 25% del dataset)
  - per valutare la performance del modello su dati che non ha mai visto
  - Per valutare la sua abilità di generalizzare
- **Cross Validation** è la tecnica che divide il dataset in training e test sets

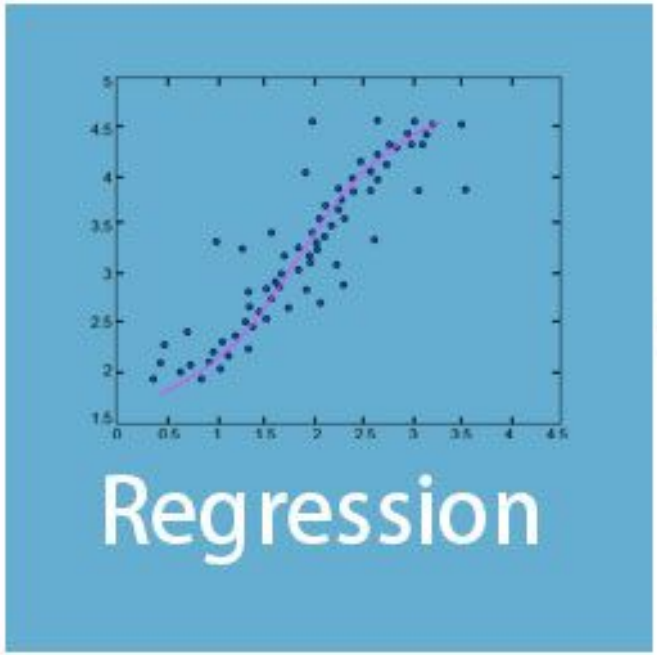
K fold cross validation

- Divido il dataset in k parti, k-1 per allenare il modello e la rimanente parte per valutarlo
- Ripeto questa operazione k volte

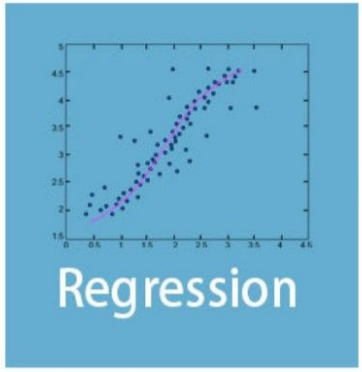
Stratified k fold Cross Validation

- Permette di mantenere la stessa percentuale di osservazioni per ogni livello categorico della variabile target
- Questo è molto utile quando si risolvono problemi di classificazione



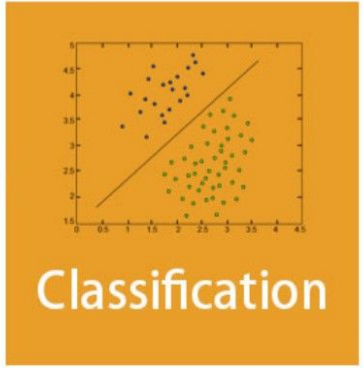

Regression
VS
Classification

Supervised Learning



KNN

SVM

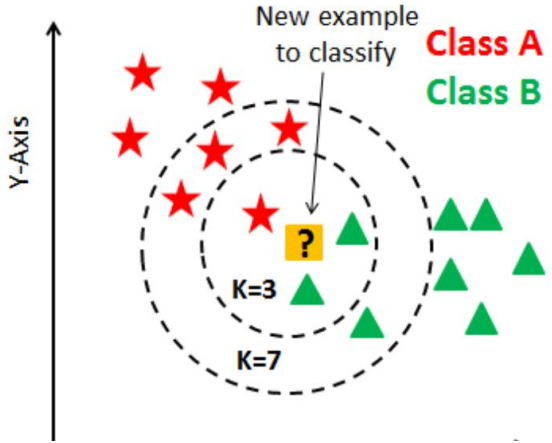


KNN

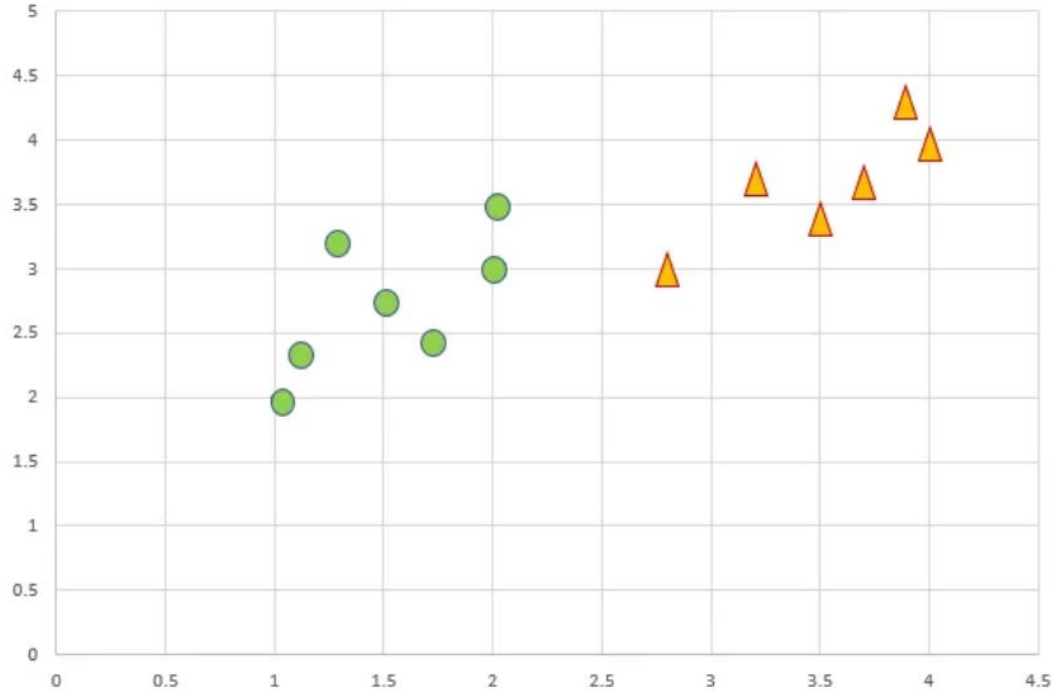
SVM

K Nearest Neighbors

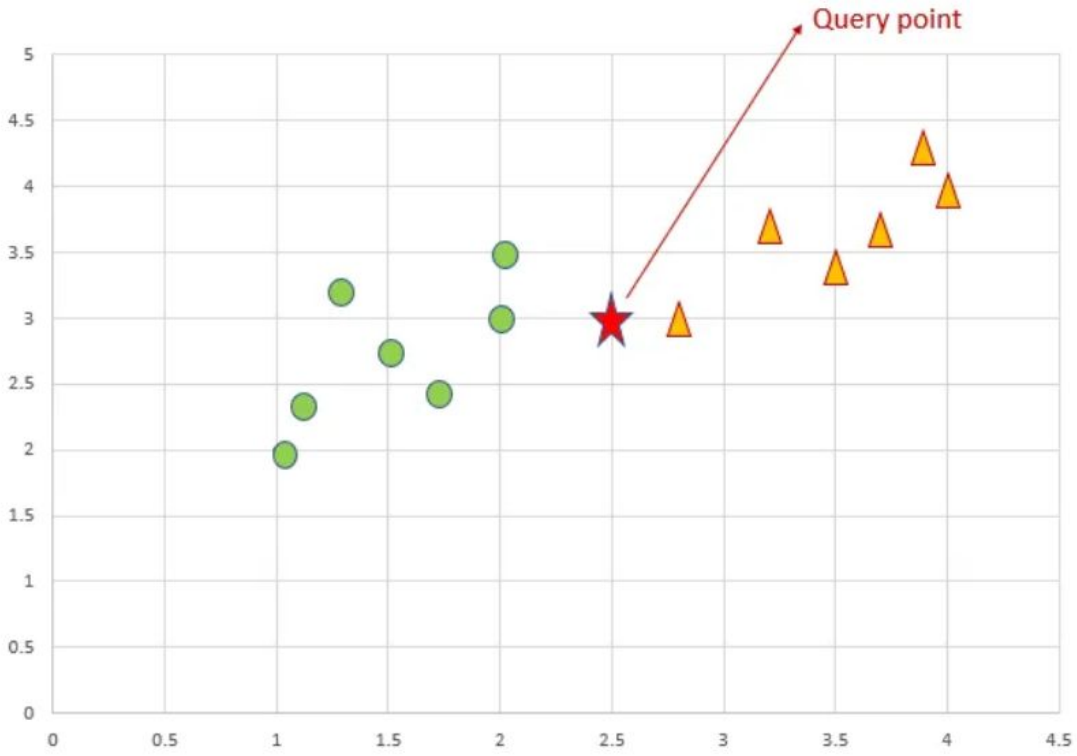
- È un modello statistico utilizzato per:
  - **Classificazione**: trova i k punti più vicini al punto preso in considerazione e prevede la classe in base al voto voto di maggioranza dei punti più vicini
  - **Regressione**: trova i k punti più vicini e prevede il valore calcolando il valore medio dei punti più vicini



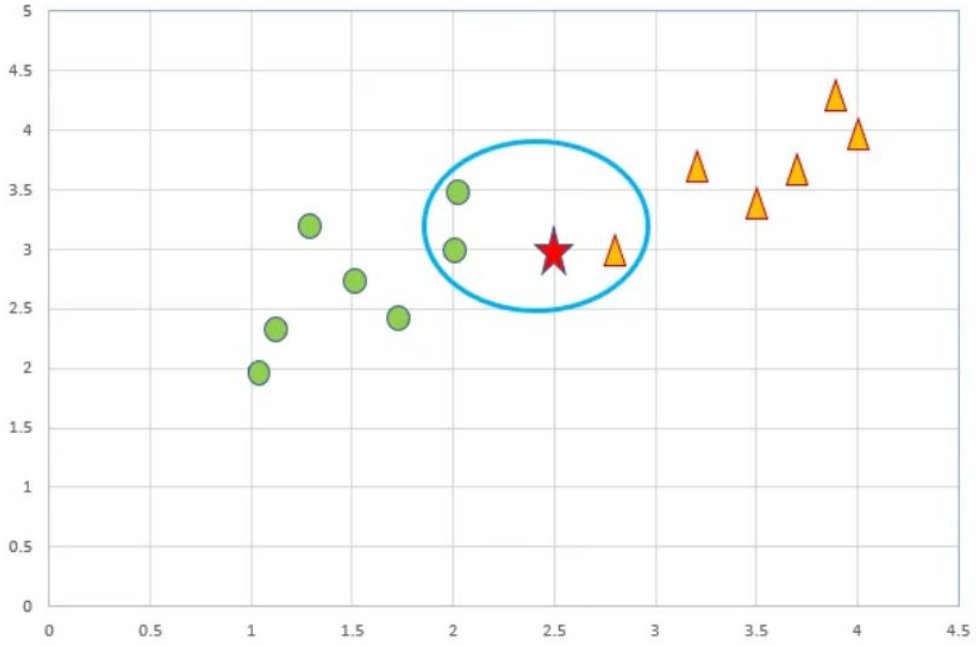
Esempio: vogliamo classificare i dati in due classi



Vogliamo prevedere la classe di un nuovo punto

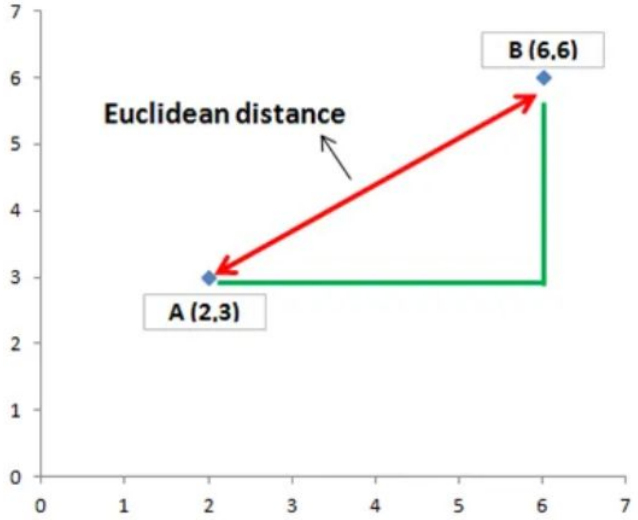


Step 1: Bisogna scegliere il valore di k



Esempio: k=3

Step 2: Definire la metrica di distanza



$$\textit{Euclidean distance}\ (a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Step 2: Definire la metrica di distanza

Distance functions

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

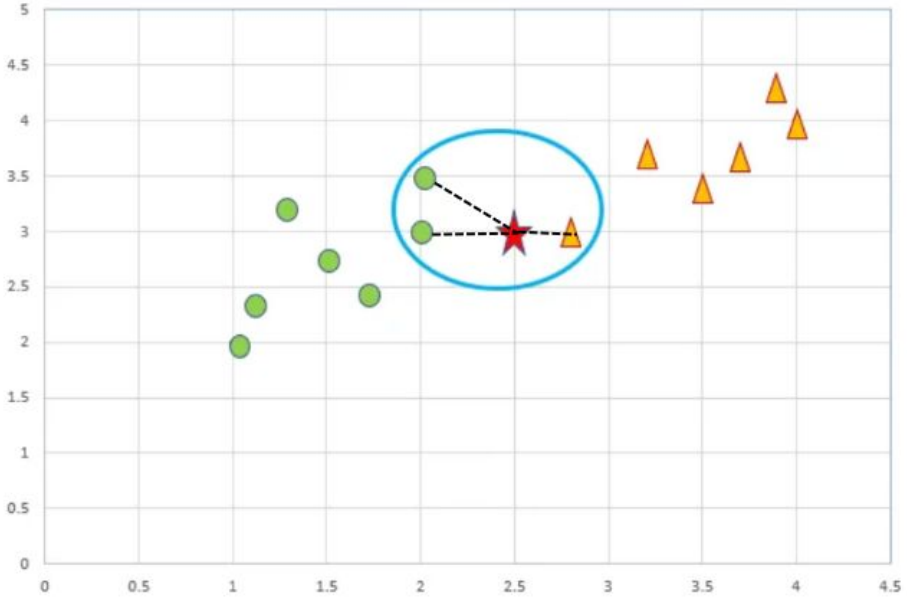
Manhattan

$$\sum_{i=1}^{k}|x_i - y_i|$$

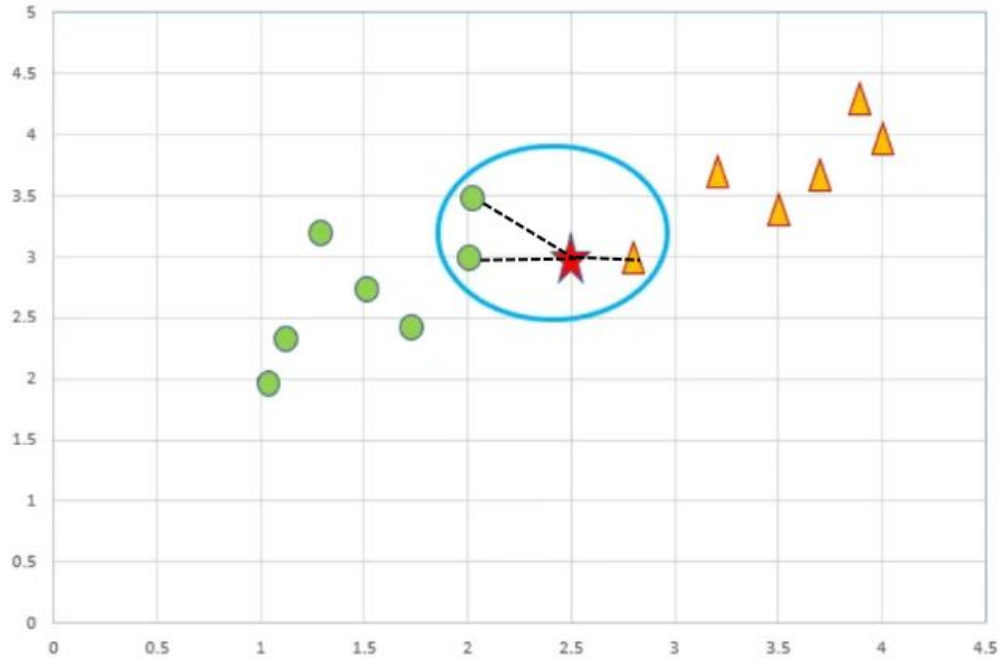
Minkowski

$$\left(\sum_{i=1}^{k}\left(|x_i - y_i|\right)^q\right)^{1/q}$$

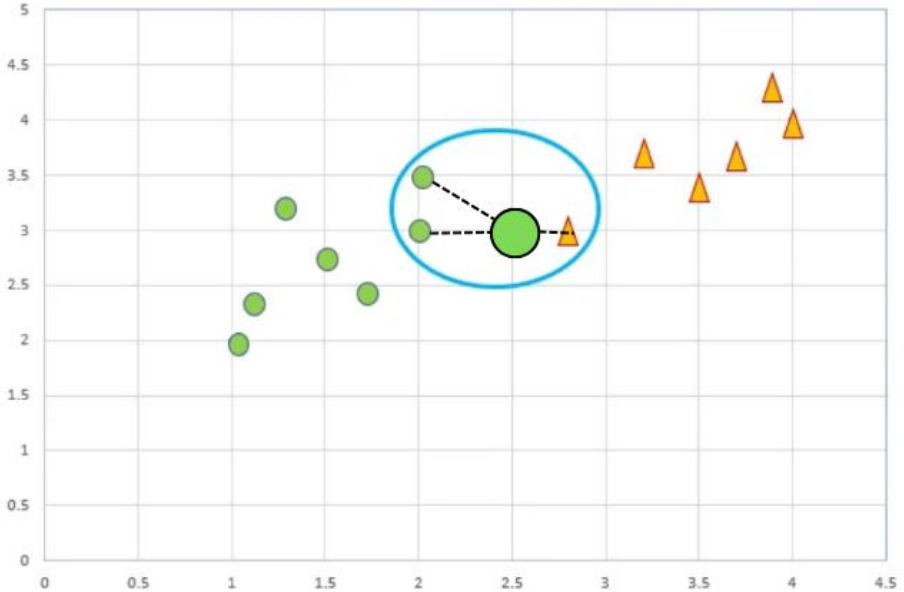
Step 3: Calcolo la distanza da tutti i punti nel training set



Step 4: Ordino le distanze e seleziono i k punti più vicini



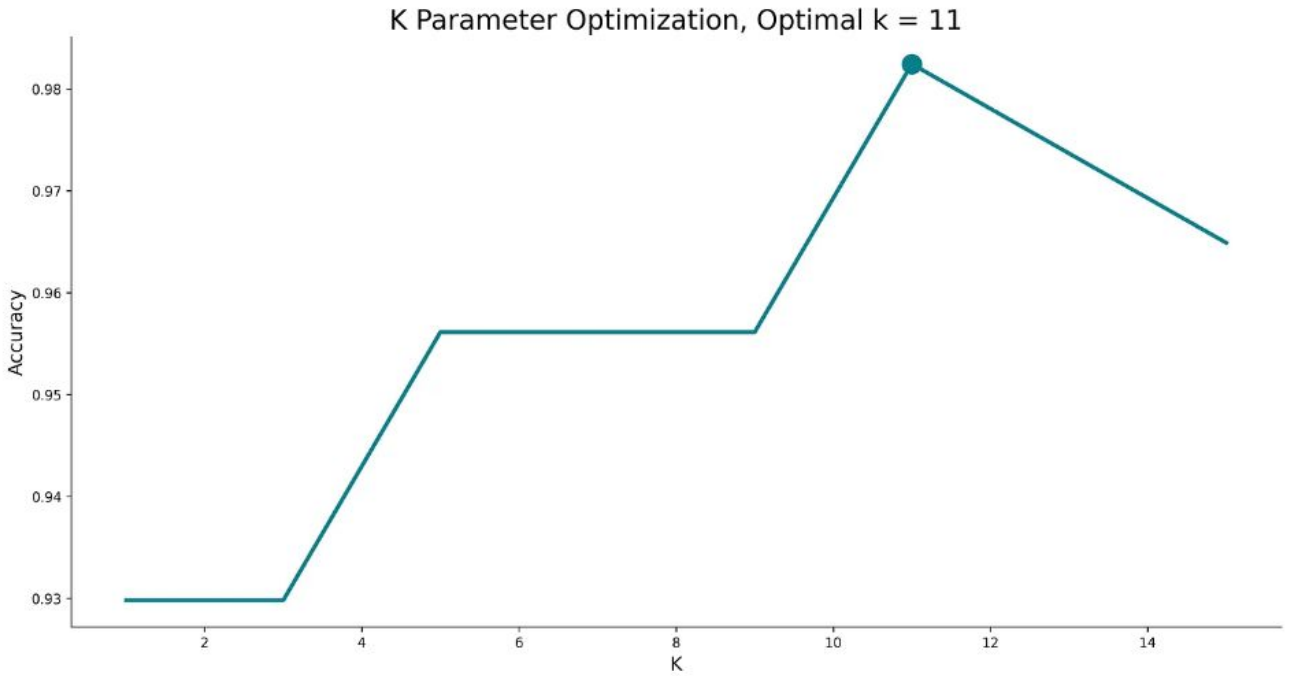
Step 5: Assegno la classe più frequente dei k punti al nuovo punto



Riassunto delle fasi

1. Preprocessing: Normalizzazione
2. **Scegliere il valore di k**
3. **Definire la metrica di distanza**
4. Calcolare la distanza da tutti i punti nel training set
5. Ordinare le distanze e selezionare i K punti più vicini
6. Assegna
    a. Classe più frequente tra i k punti più vicini alla nuova osservazione
    b. Media (o mediana) dei valori più vicini alla nuova osservazione

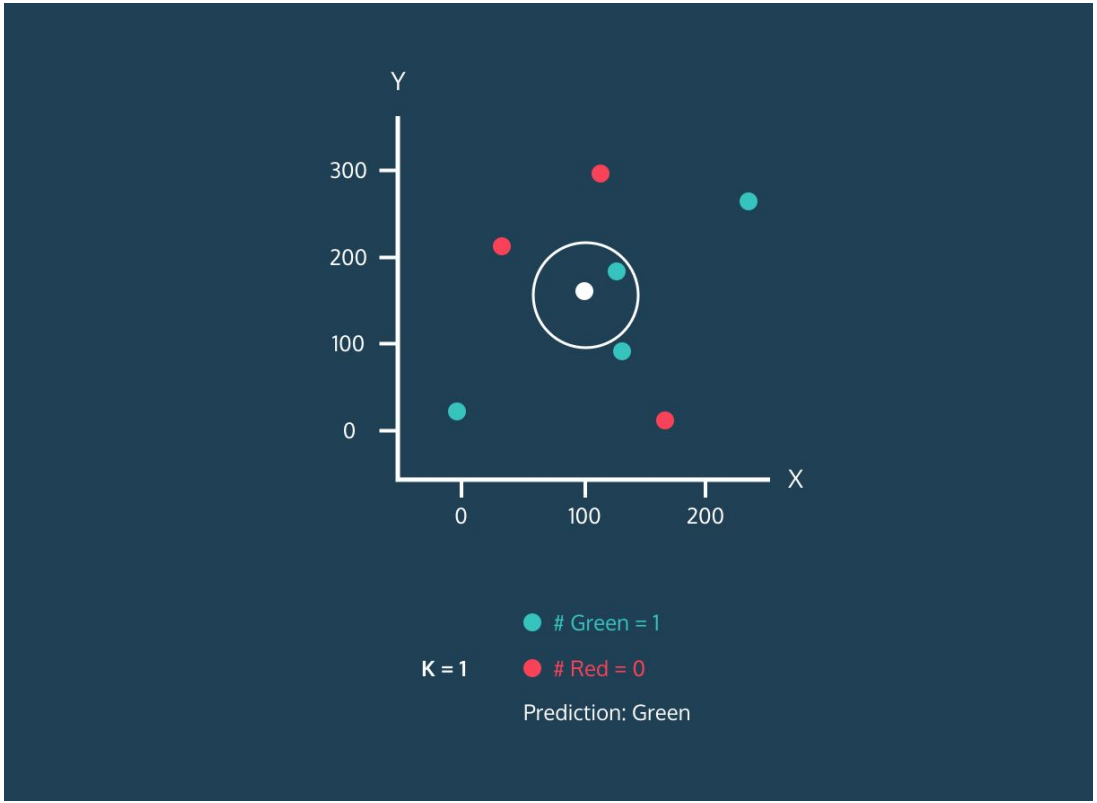
Come trovo il valore di k ottimale?



Vantaggi

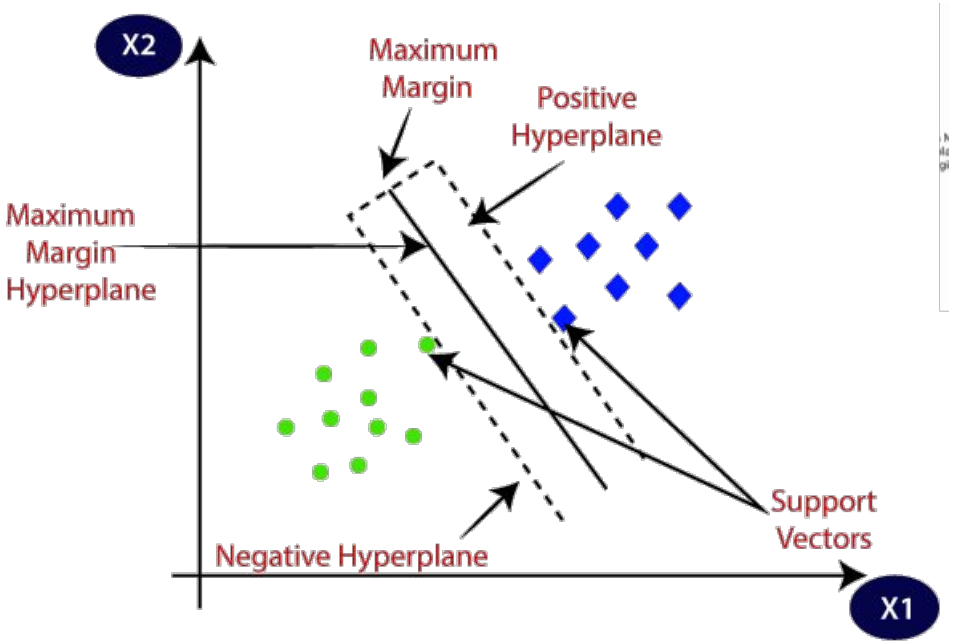
- È intuitivo e facile
- Versatile
- Funziona bene con dati non lineari
- Ma … ci sono anche svantaggi
  - Lento quando ci sono tanti dati
  - Sensibile agli outlier
  - Tende a non funzionare bene quando ci sono tante dimensioni (variabili)
  - Occupa tanta memoria!

Risultati al variare di k

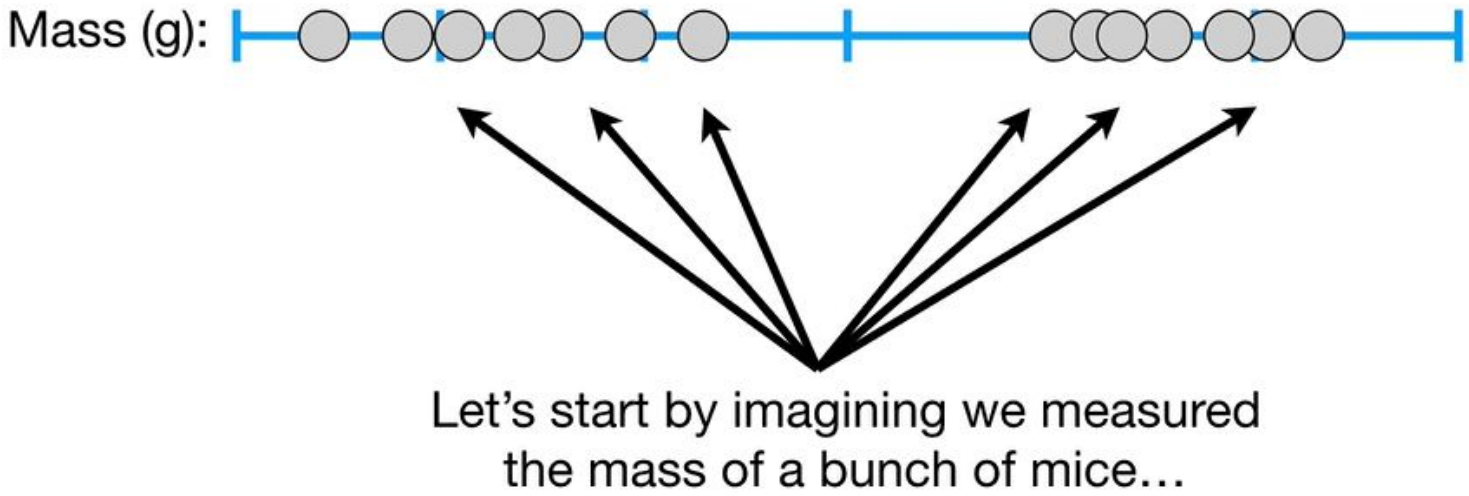


Support Vector Machines

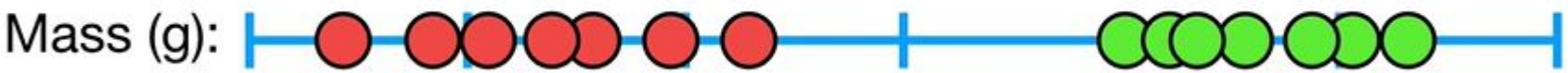
- È un modello supervisionato utilizzato sia per la classificazione che per la regressione
- Viene utilizzato sia per dati separabili linearmente che per dati separabili in modo non lineare



Esempio: vogliamo i classificare i topi in obesi e non obesi



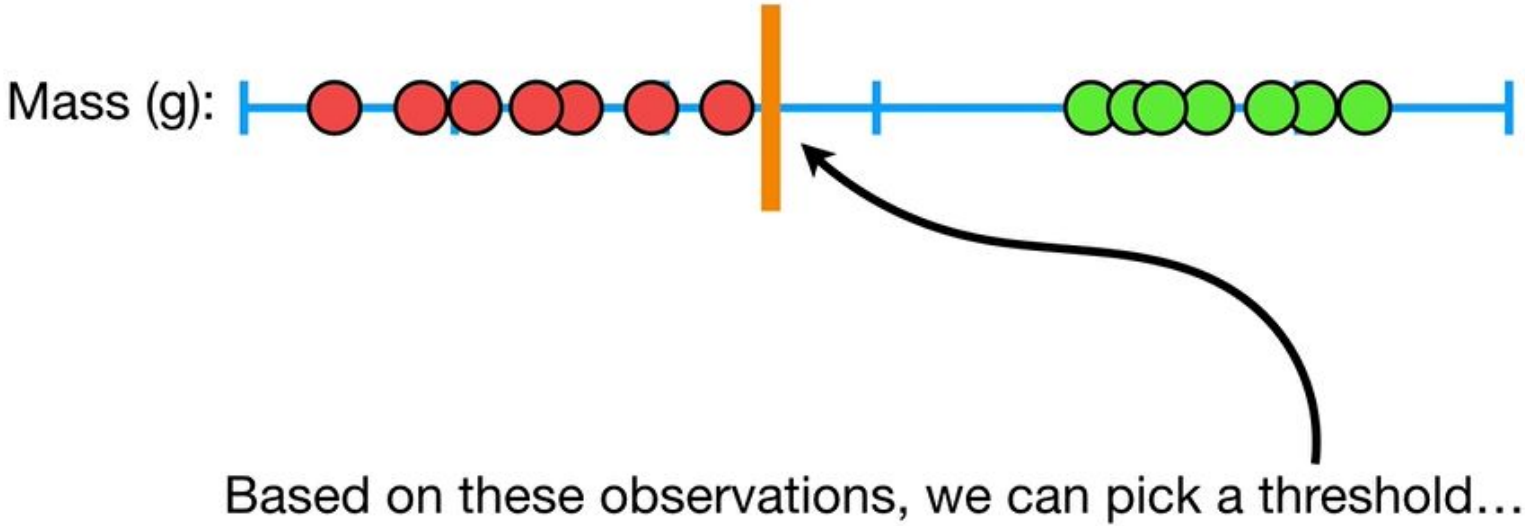
Esempio: vogliamo i classificare i topi in obesi e non obesi



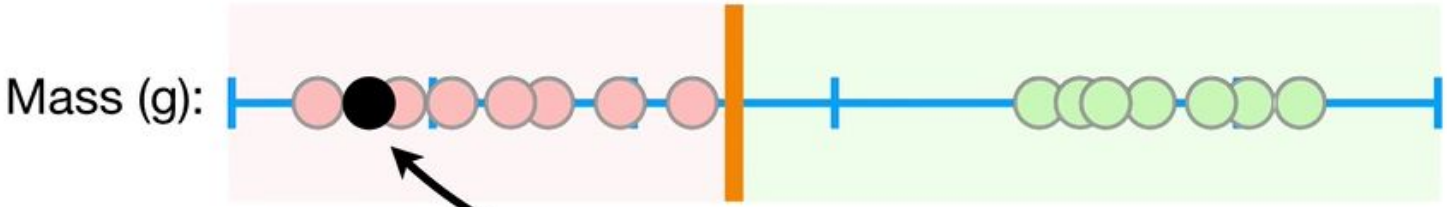
Dove:

- Rosso = non obeso
- Verde = obeso

Scegliamo un valore soglia

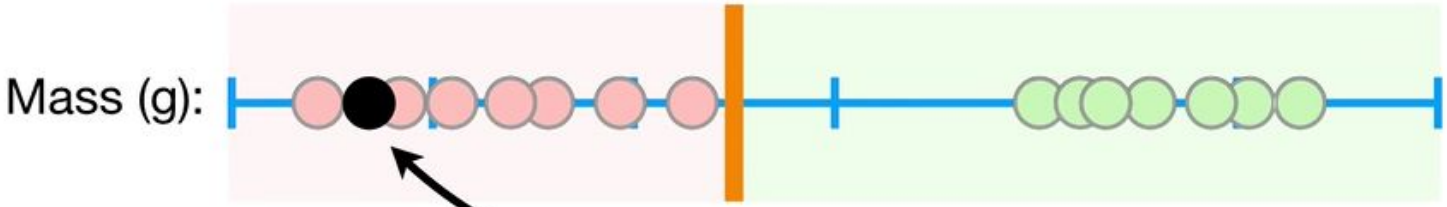


Aggiungiamo una nuova osservazione



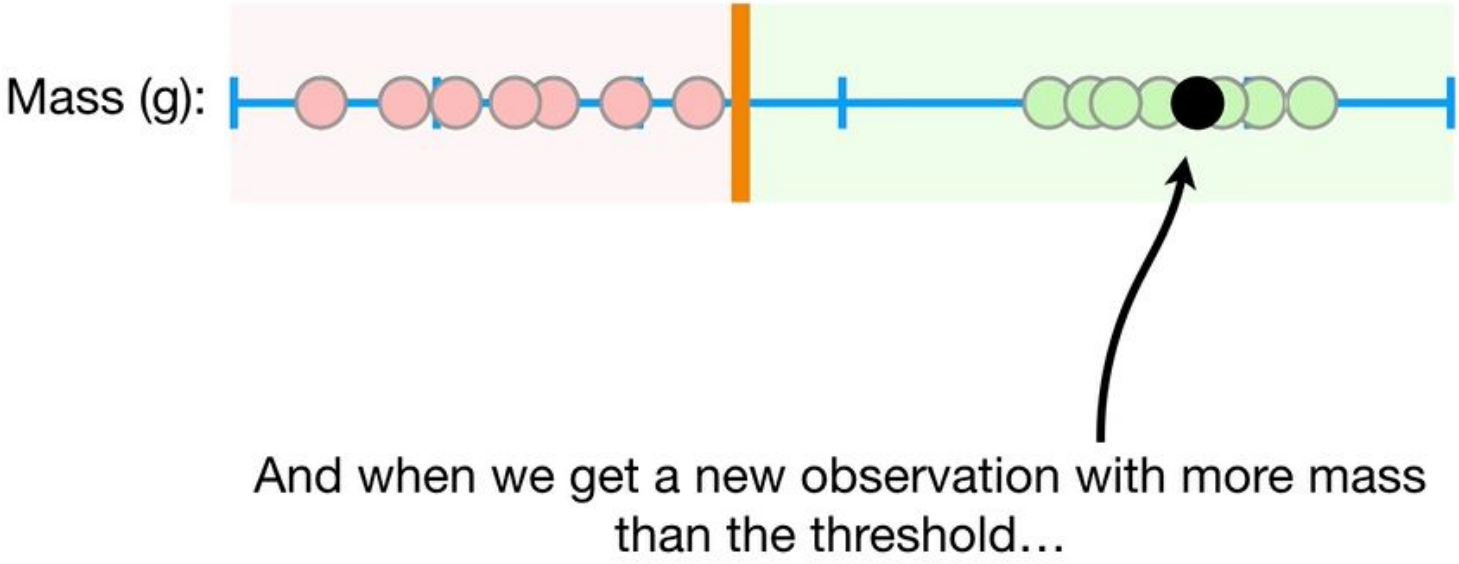
…and when we get a new observation that has less mass than the threshold…

Classifichiamo l’osservazione come …

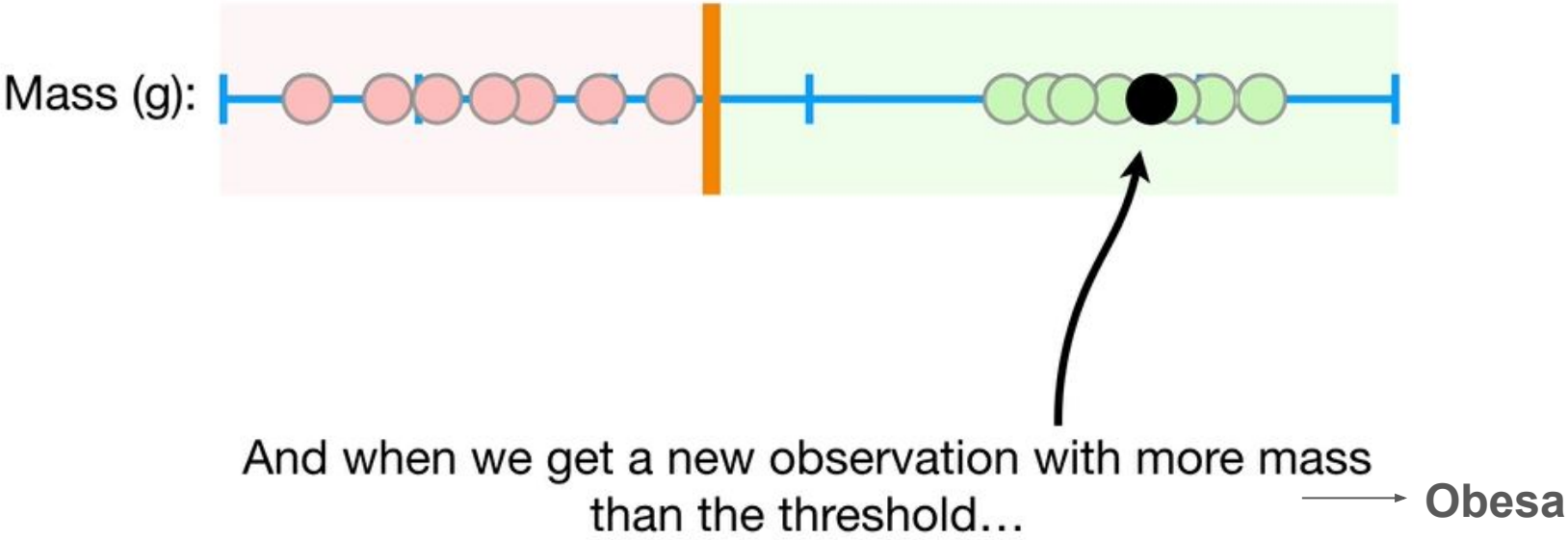


…and when we get a new observation that has less mass than the threshold… → **Non obesa**

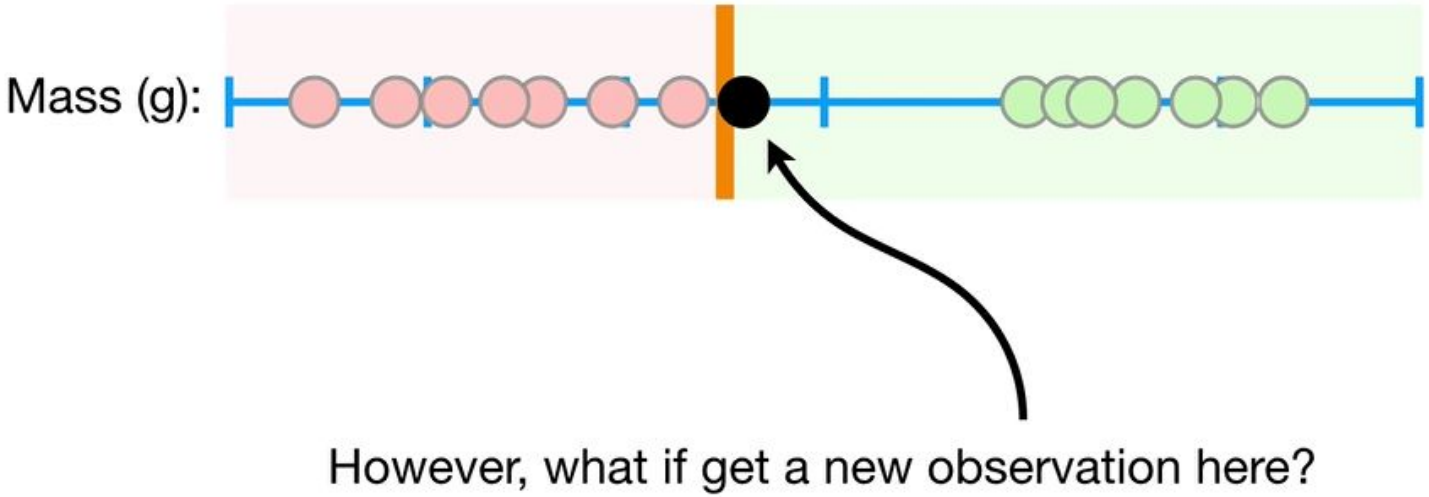
Riceviamo un’altra nuova osservazione



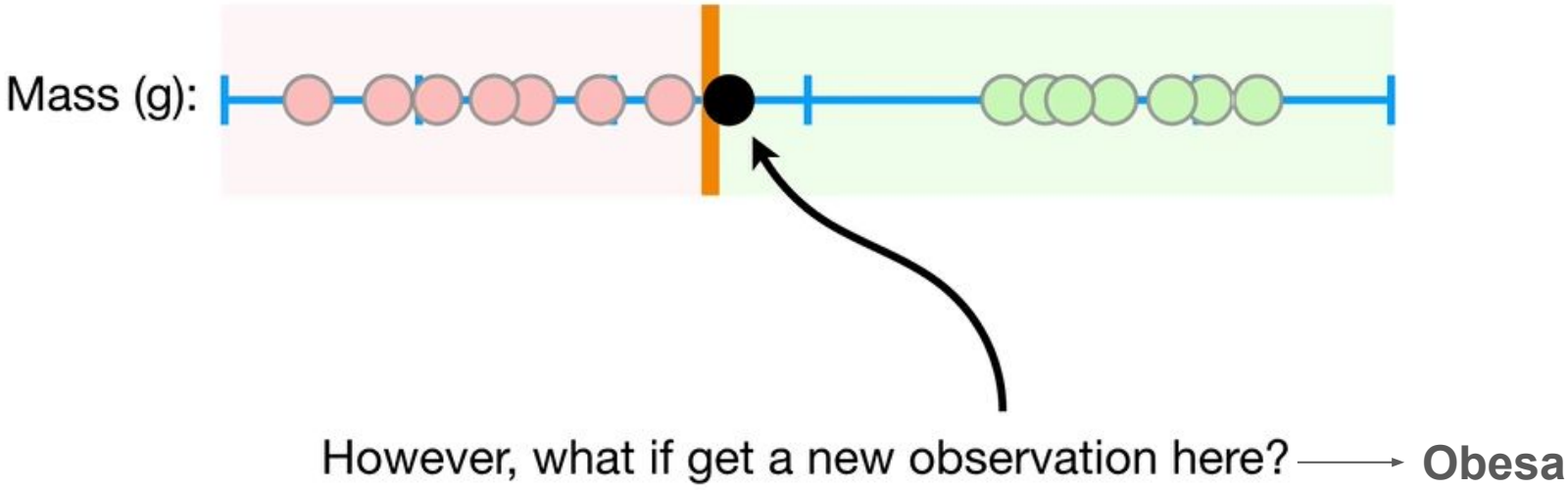
Classifichiamo l’osservazione come …



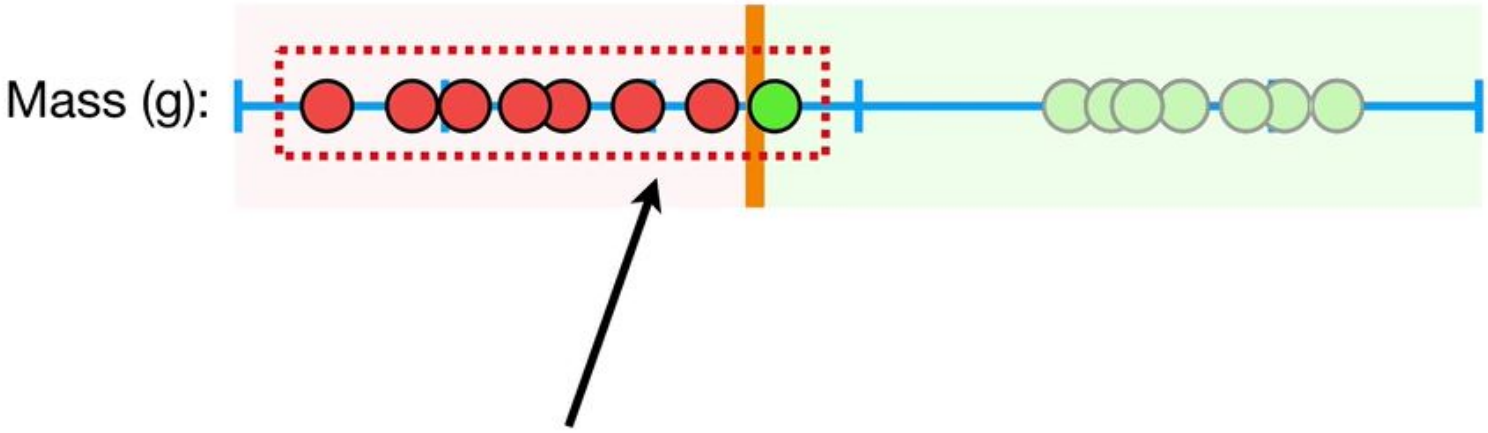
Cosa succede se abbiamo una nuova osservazione qui?



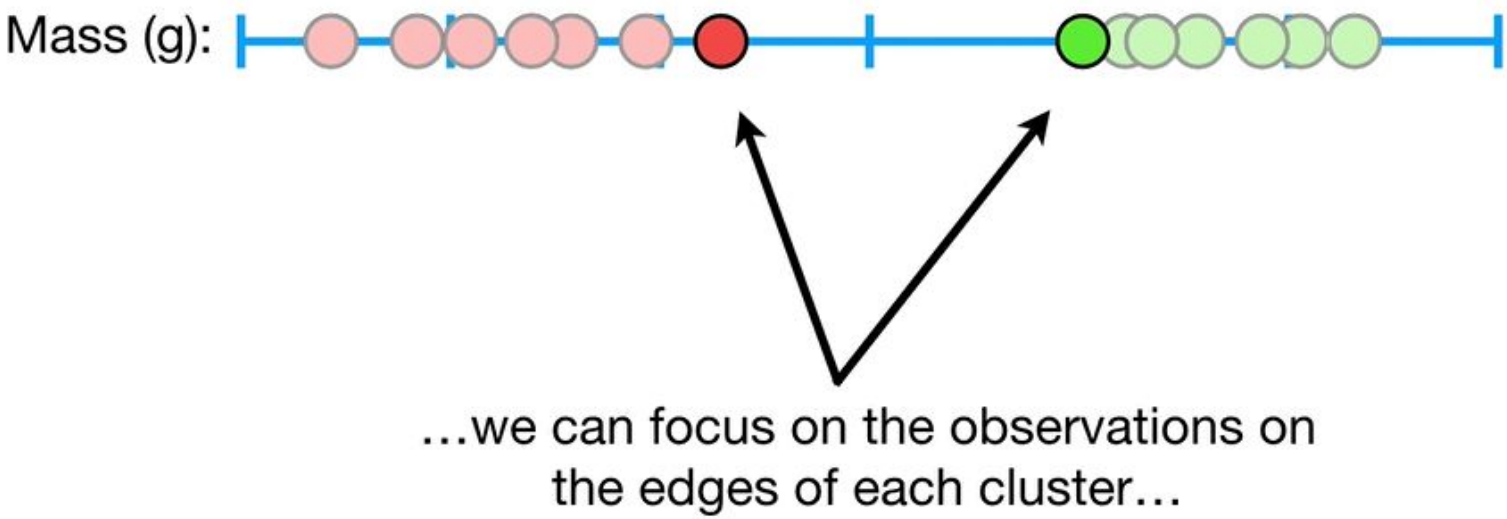
Cosa succede se abbiamo una nuova osservazione qui?



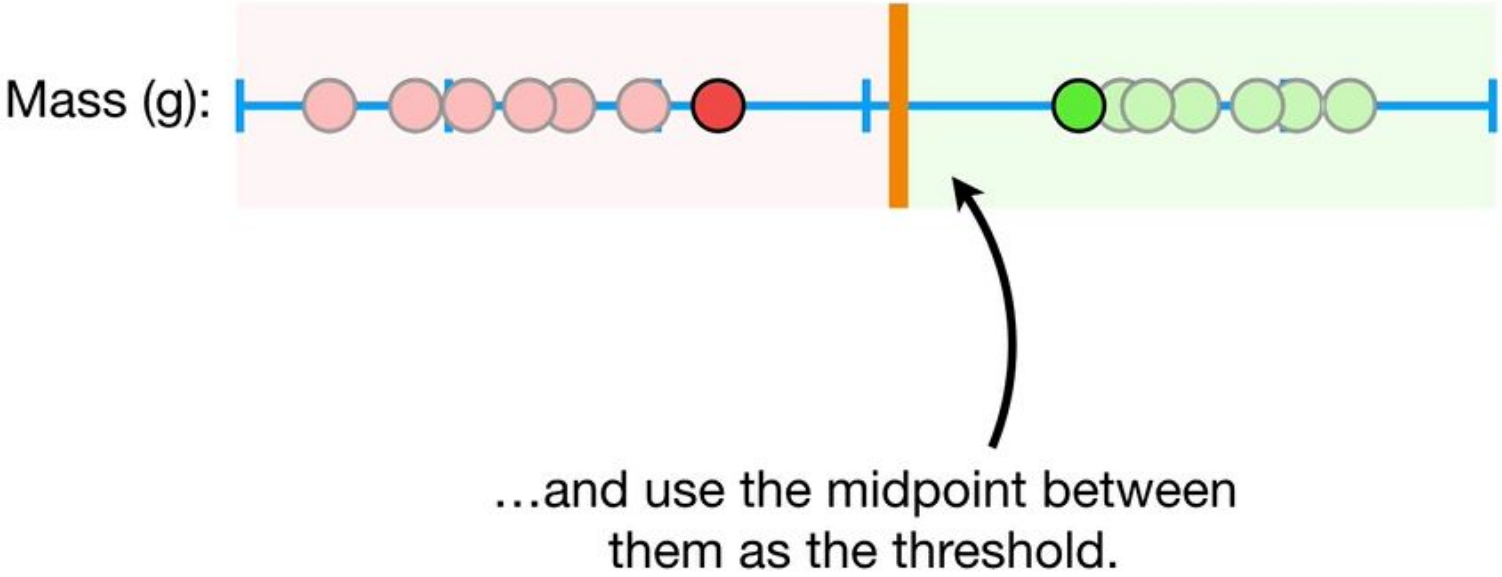
Ma ha senso?


Mass (g):
But that doesn't make sense, because it is much closer to the observations that are **not obese**.

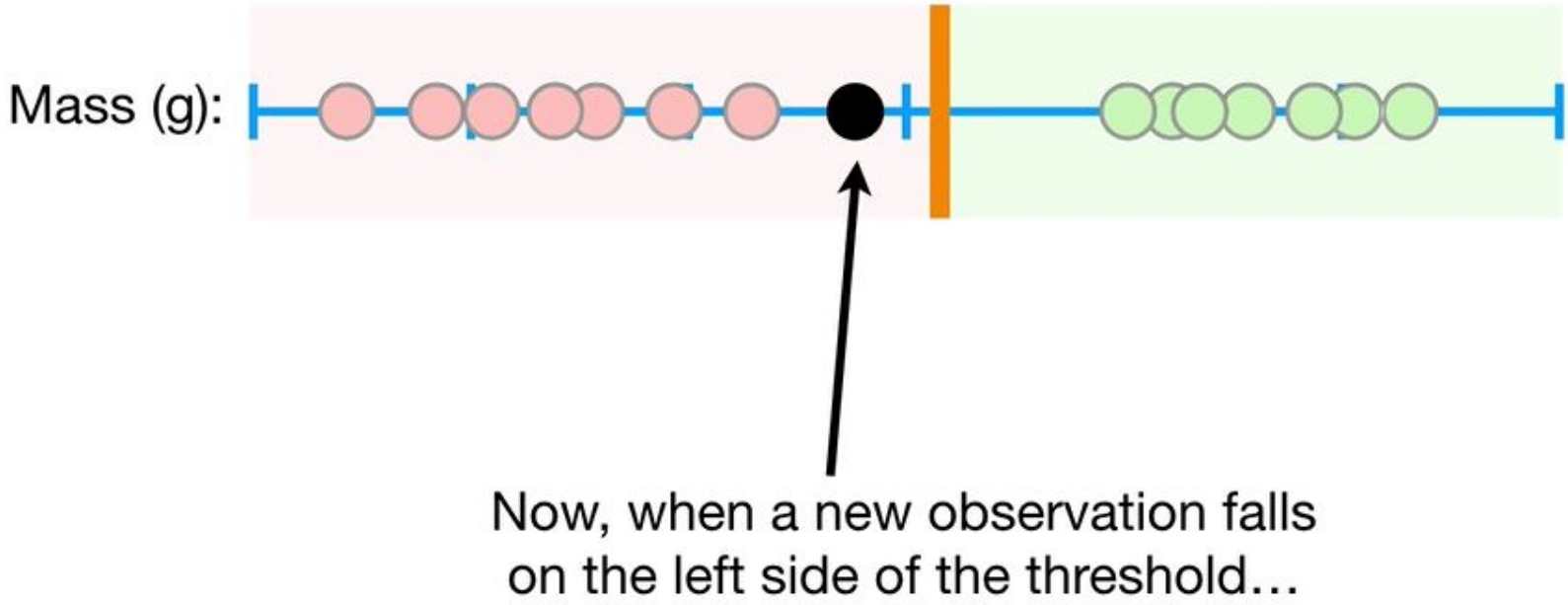
Alternativa: Ci focalizziamo sulle osservazioni al limite di ciascun gruppo



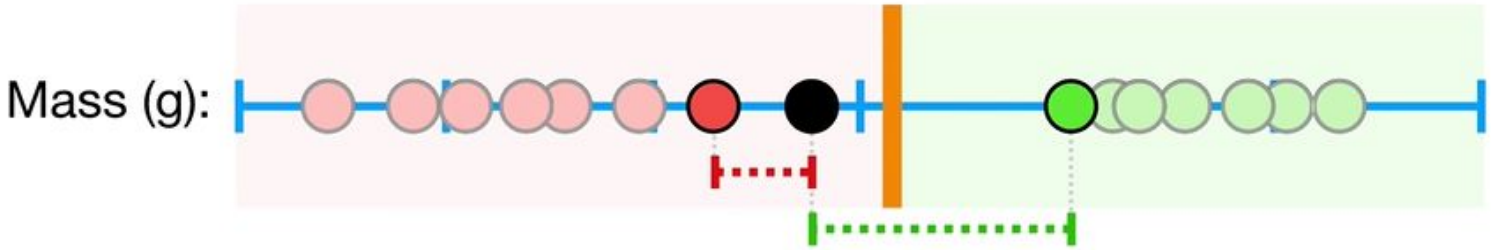
Alternativa: e … scegliamo il punto medio tra loro



Riceviamo un’altra nuova osservazione



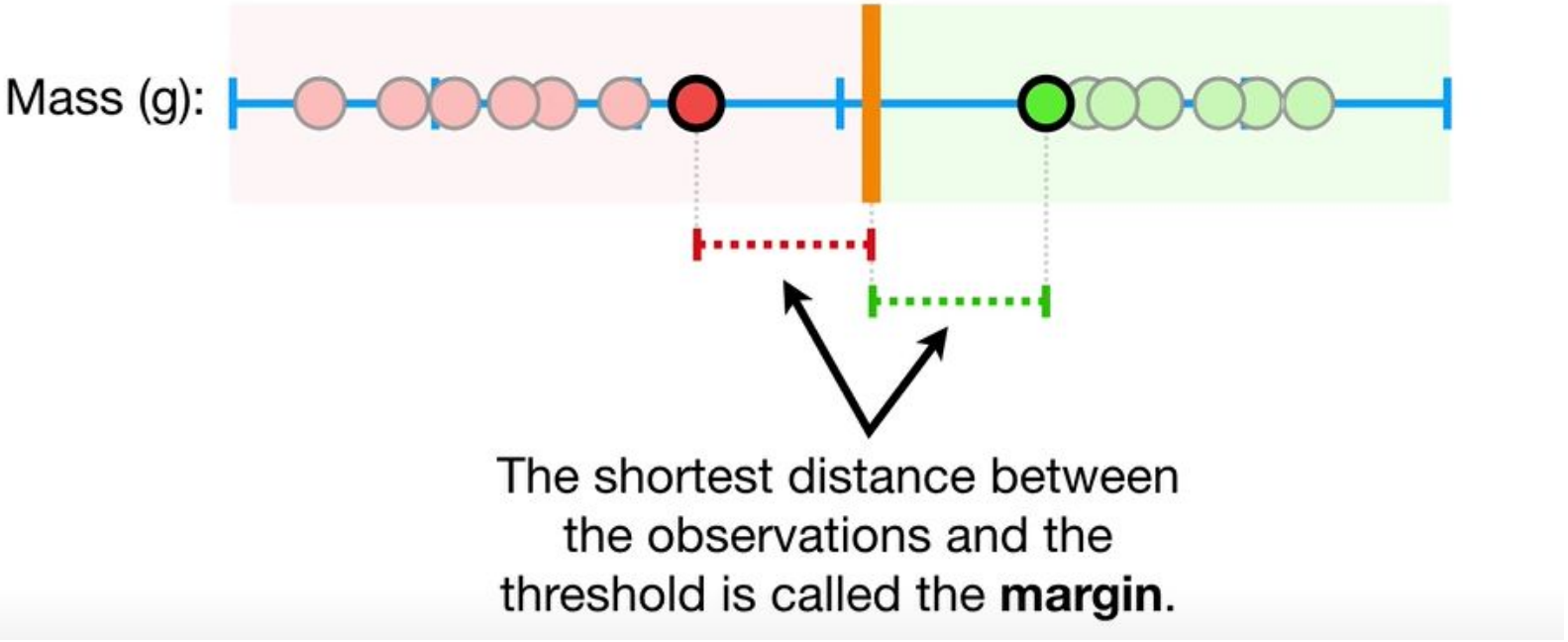
È più vicina alle osservazioni non obese



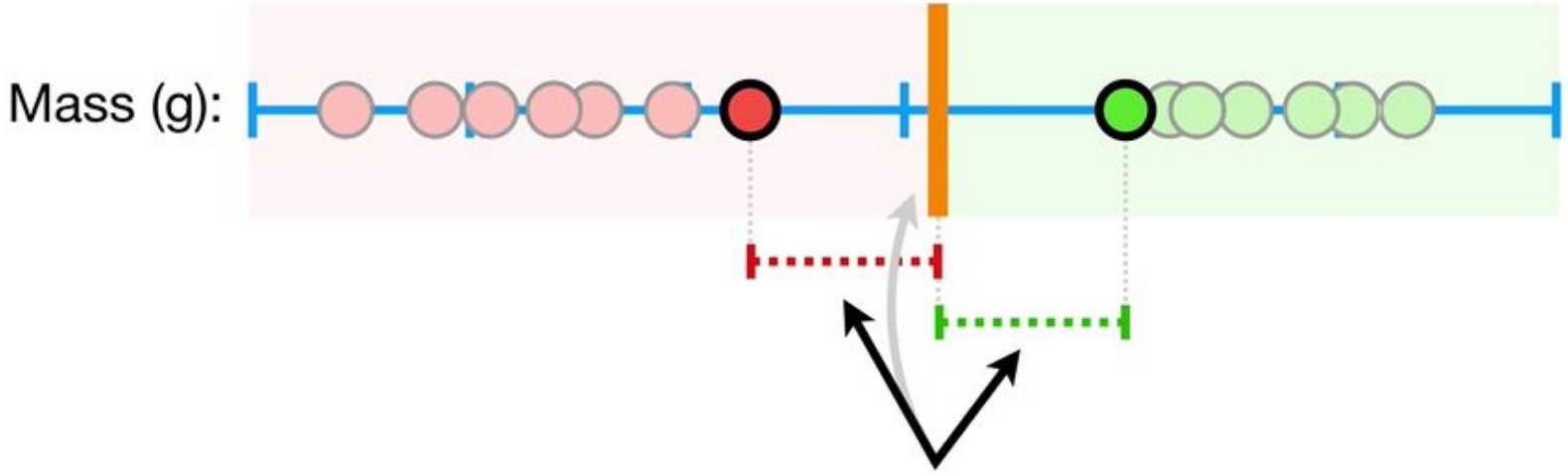
…than it is to the ***obese*** observations.

→ **Non Obesa**

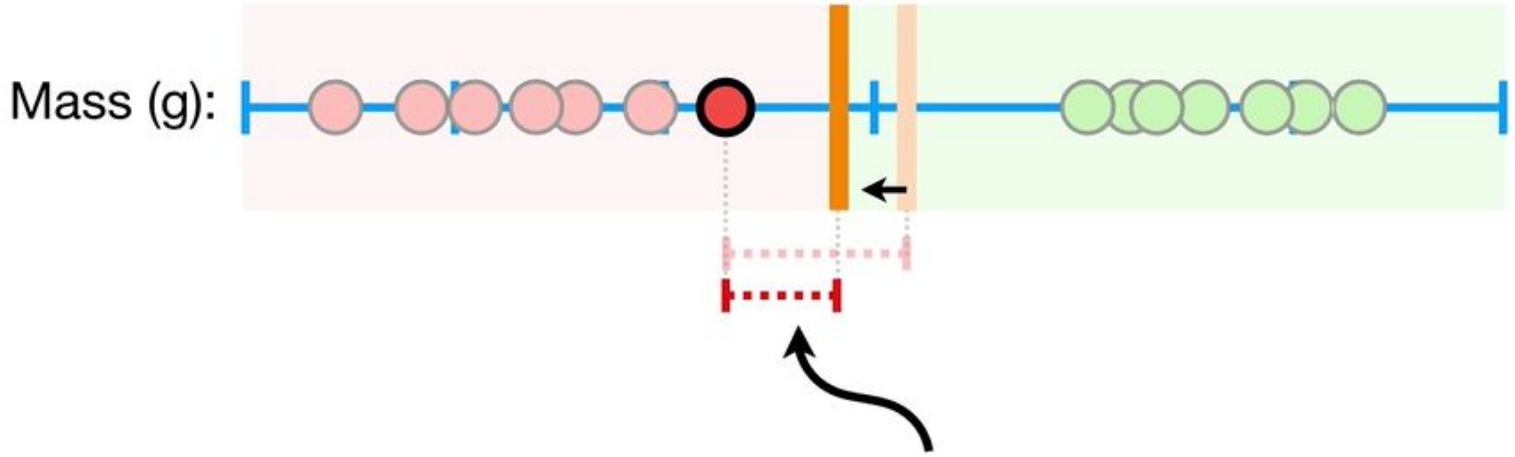
Definizione di margine



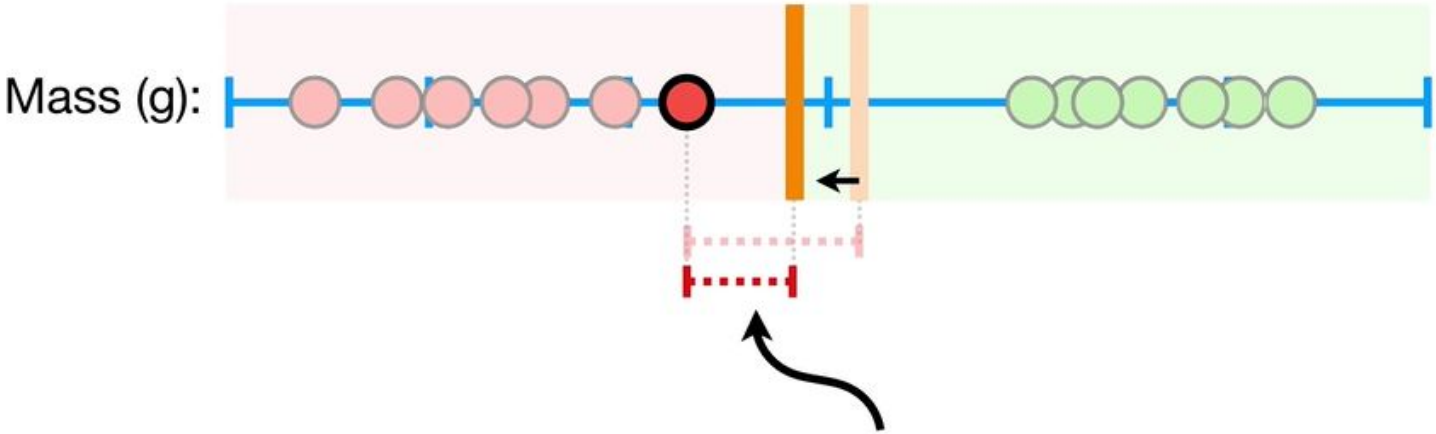
Definizione di margine



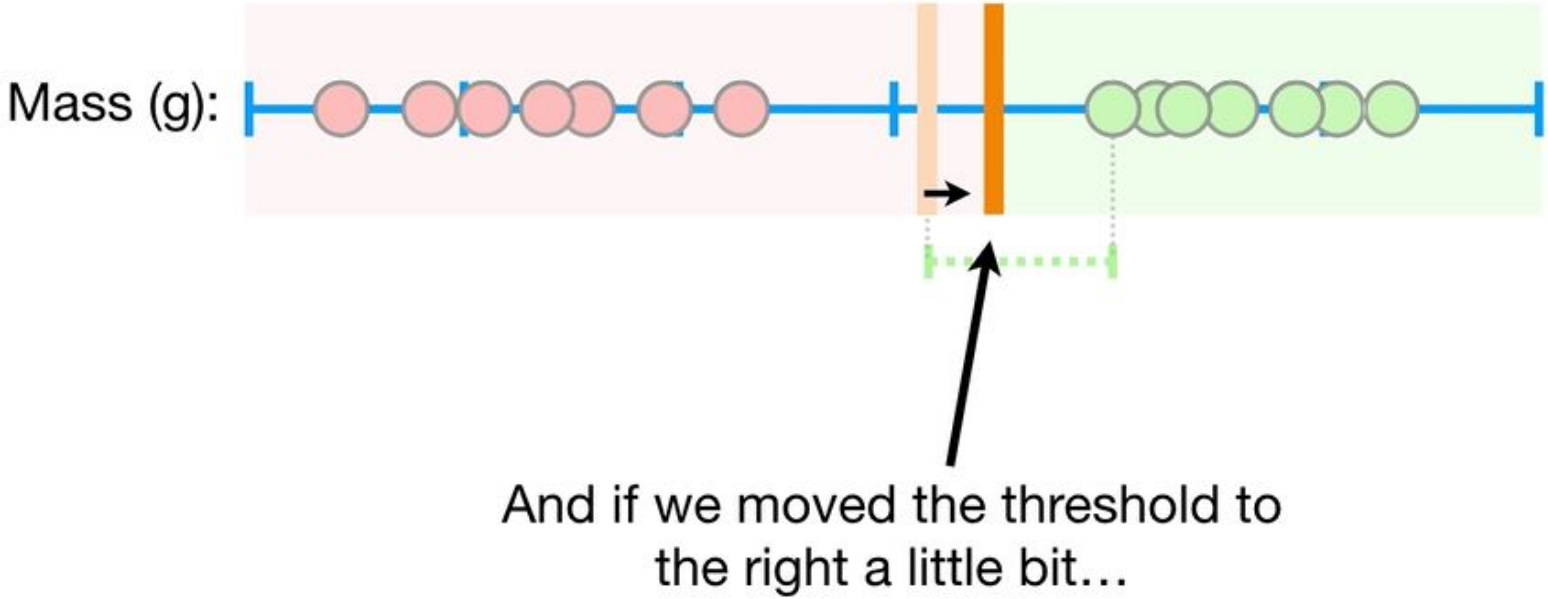
Definizione di margine



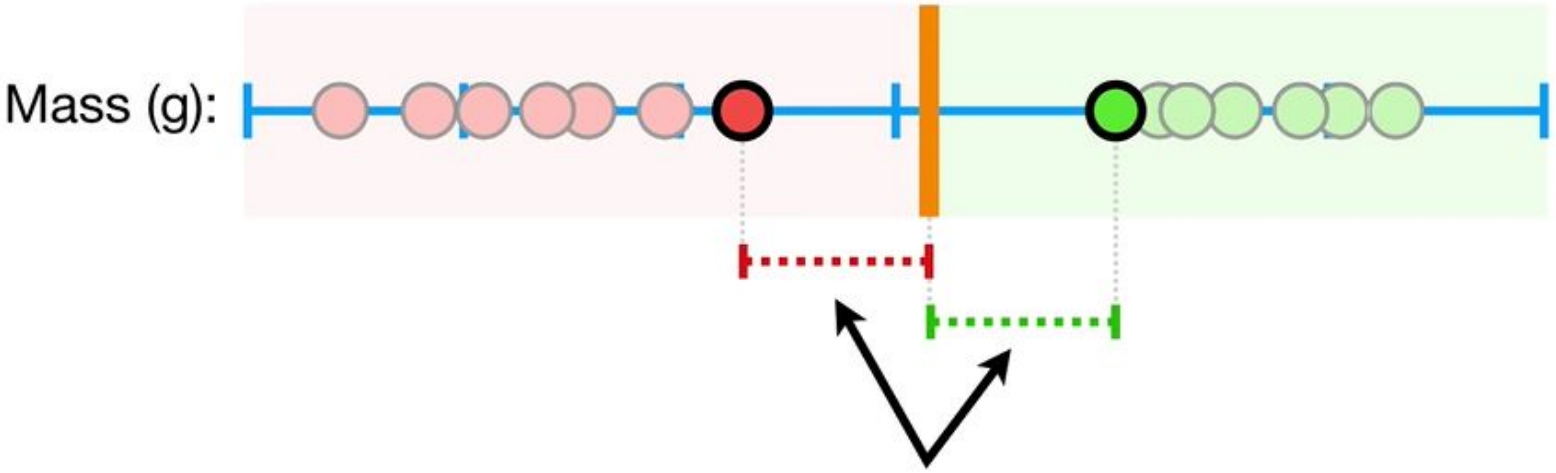
Definizione di **margine**



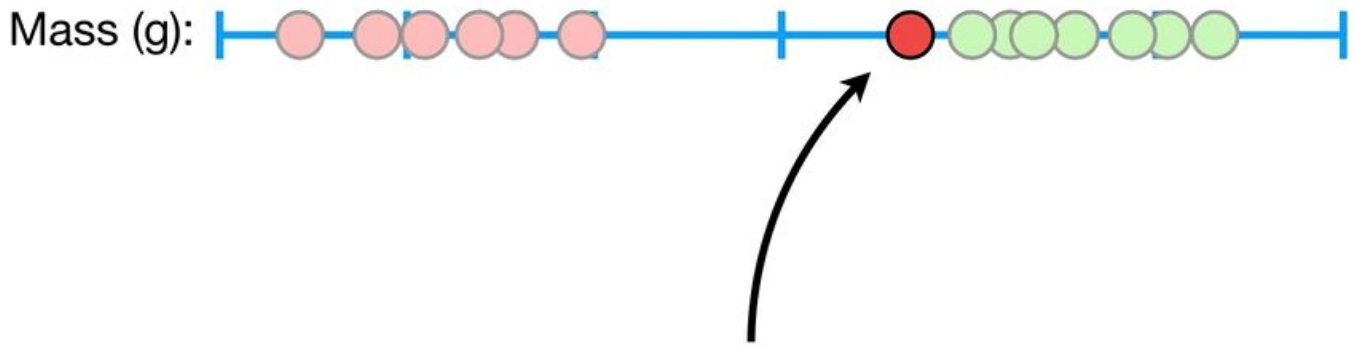
Definizione di margine



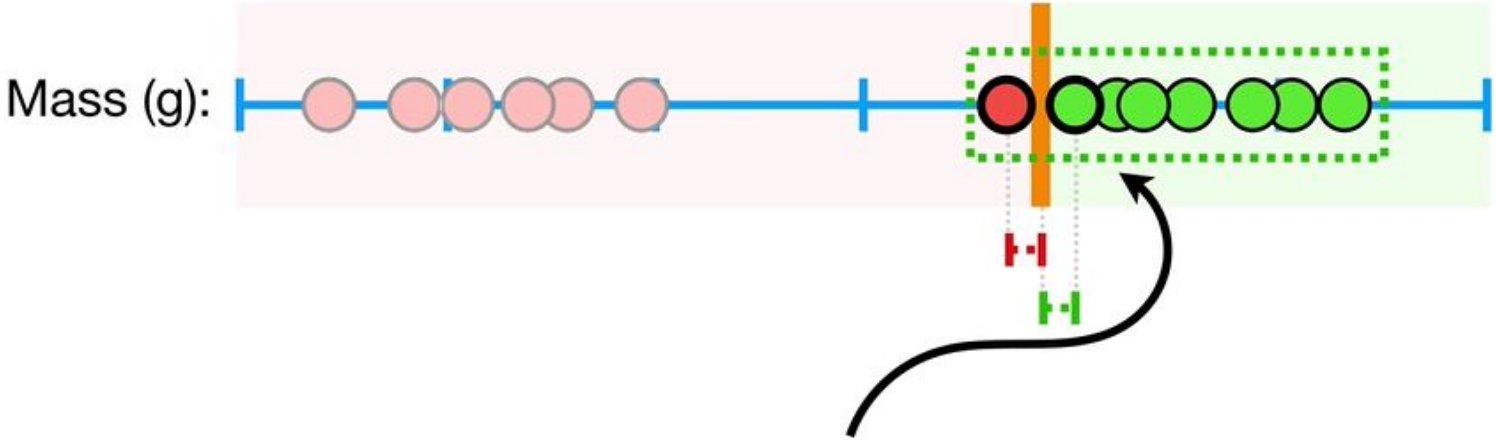
Vogliamo la threshold che massimizza il margine



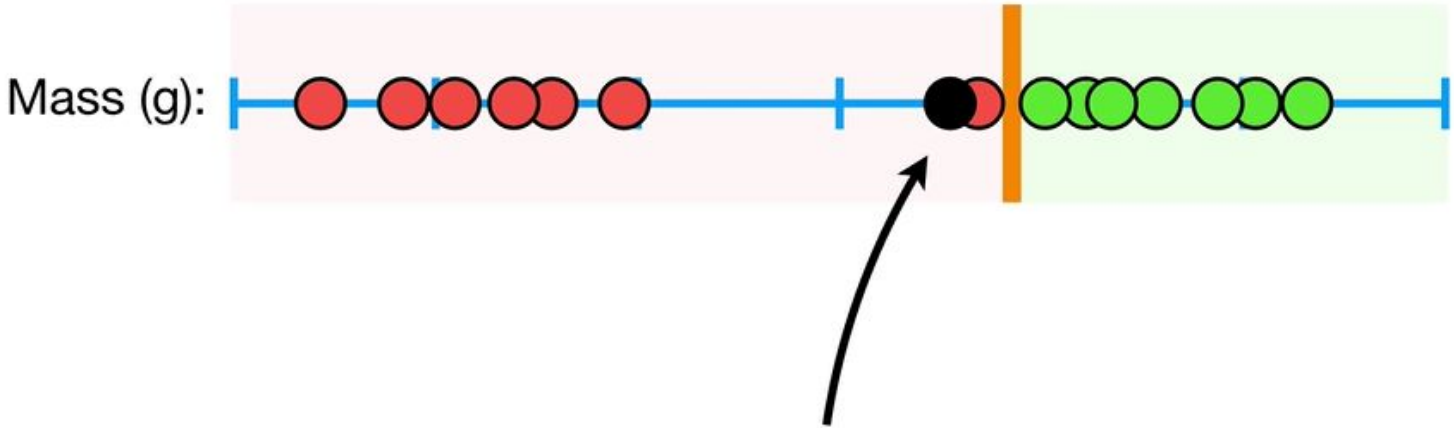
Ma cosa succederebbe se avessimo un outlier?



Ma cosa succederebbe se avessimo un outlier?



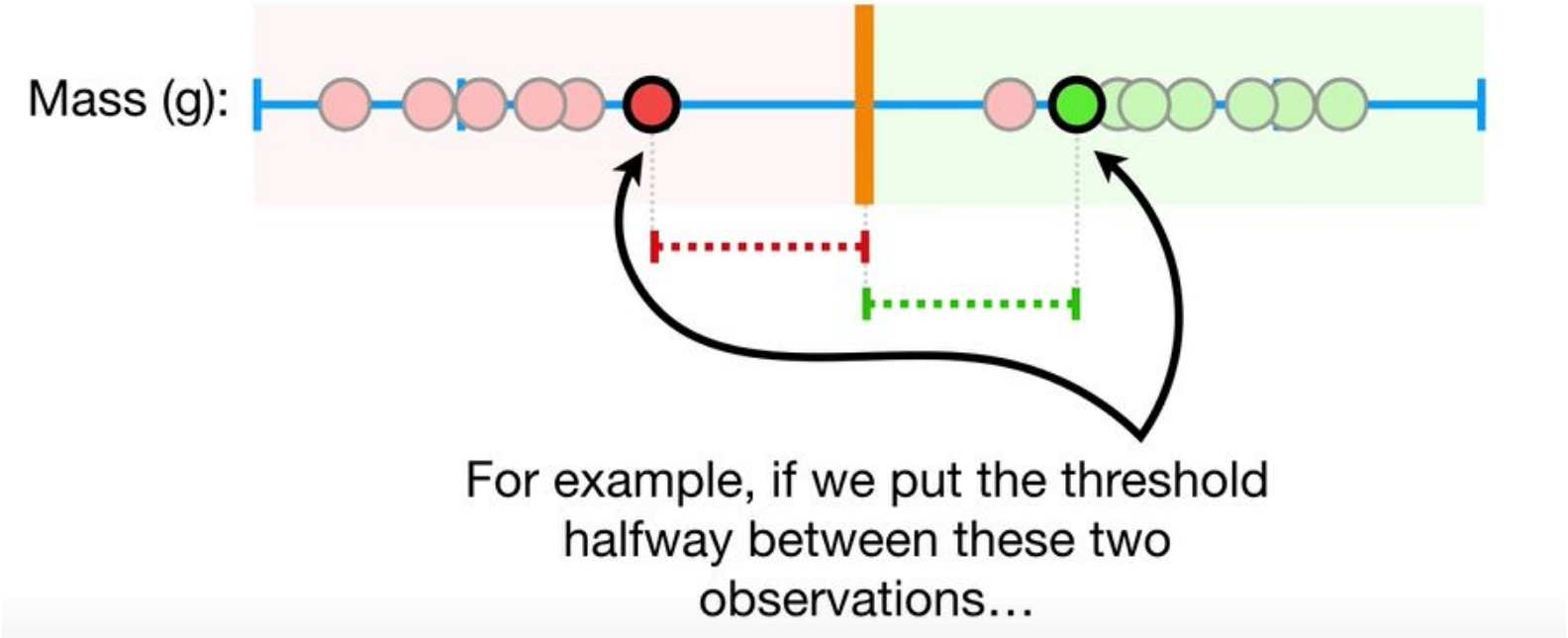
Ma cosa succederebbe se avessimo un outlier?



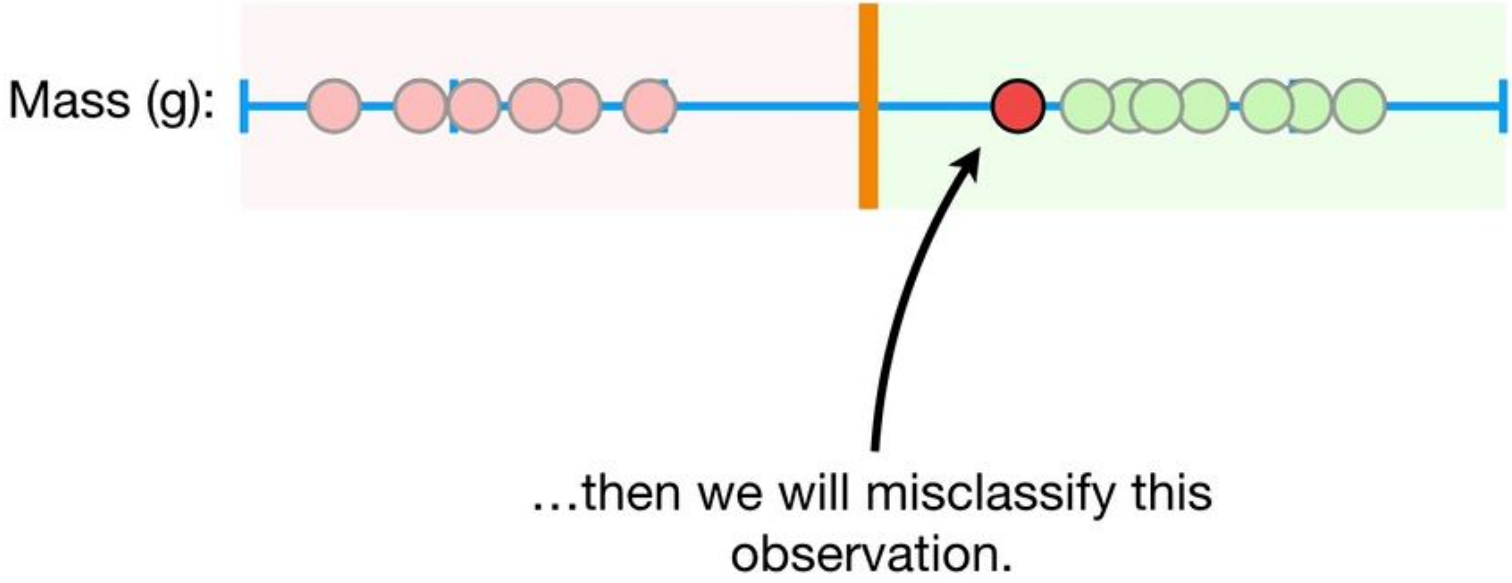
Come possiamo risolvere il problema?

To make a threshold that is not so sensitive to outliers we must **allow misclassifications**.

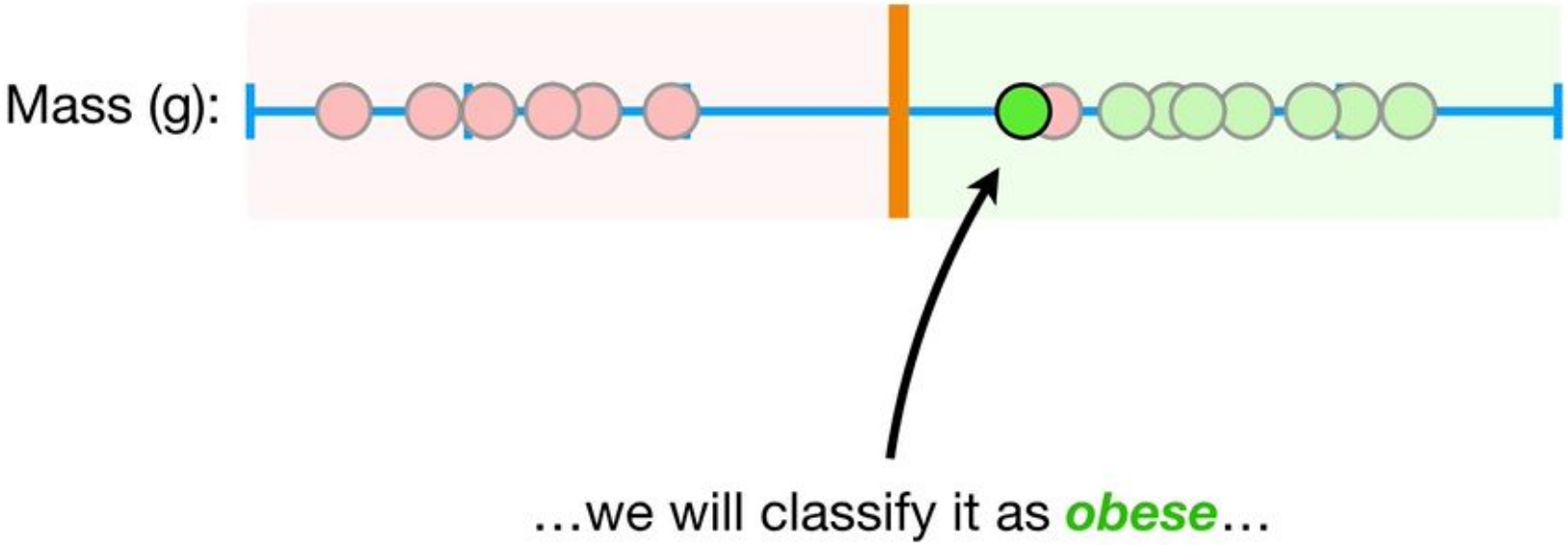
Soluzione per risolvere il problema



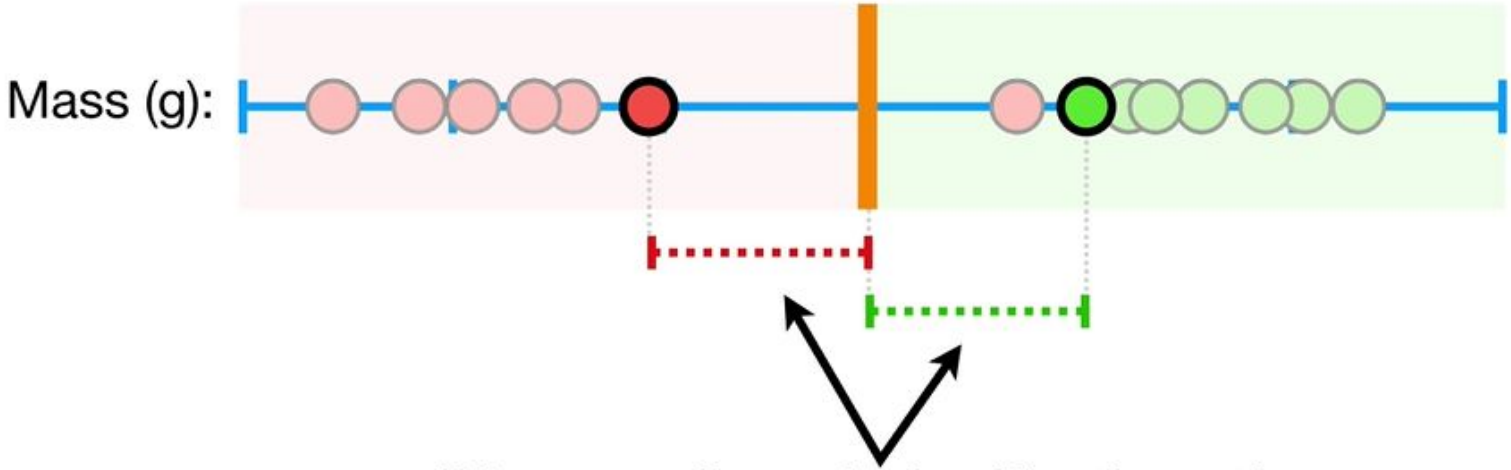
Soluzione per risolvere il problema



Soluzione per risolvere il problema



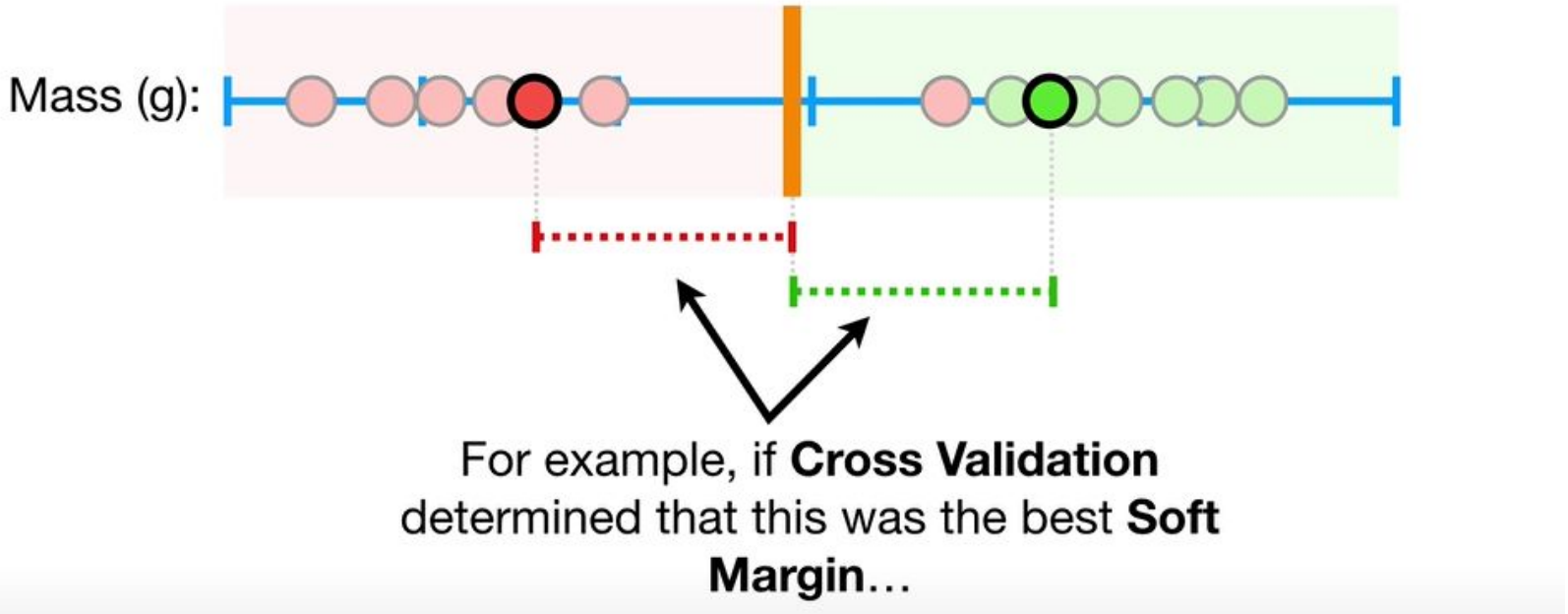
Definizione di **Soft Margin**



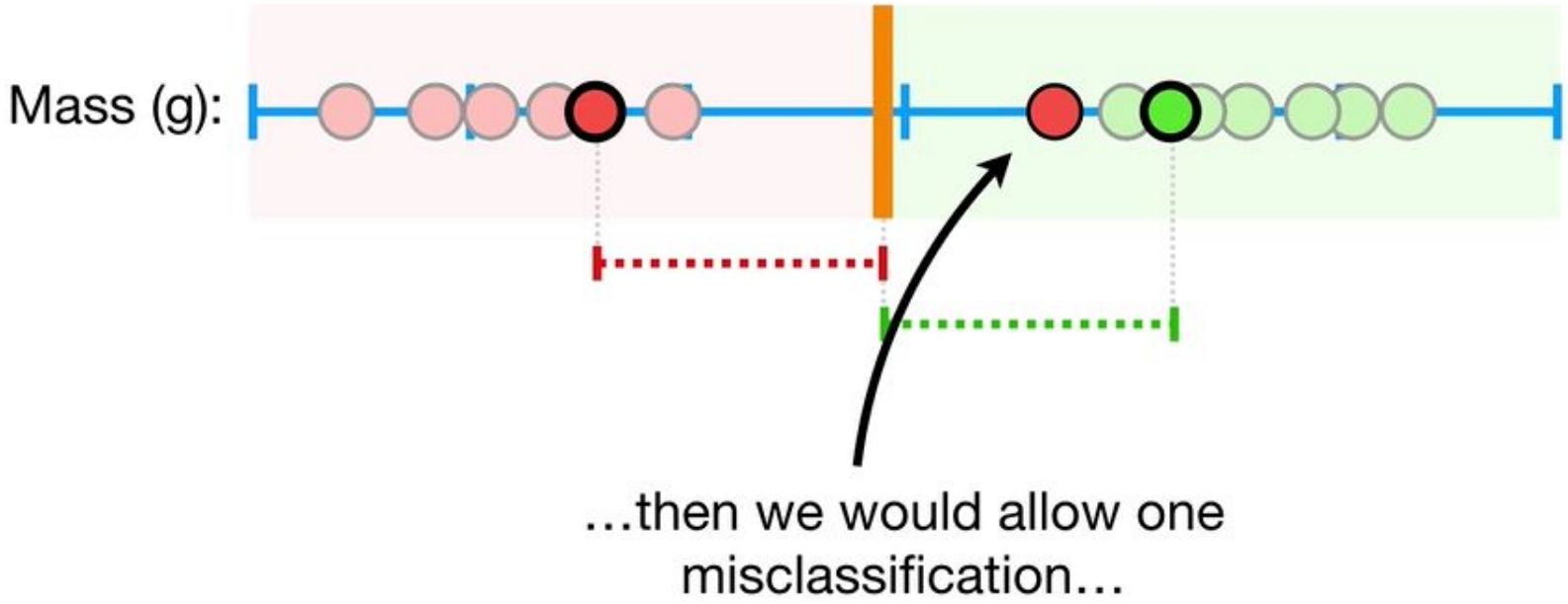
Ma come sappiamo quale soft margin è il migliore?

- Si usa la **k-fold cross validation** per determinare quante classificazioni errate e quante osservazioni sono permesse all'interno del Soft Margin per ottenere la migliore classificazione!

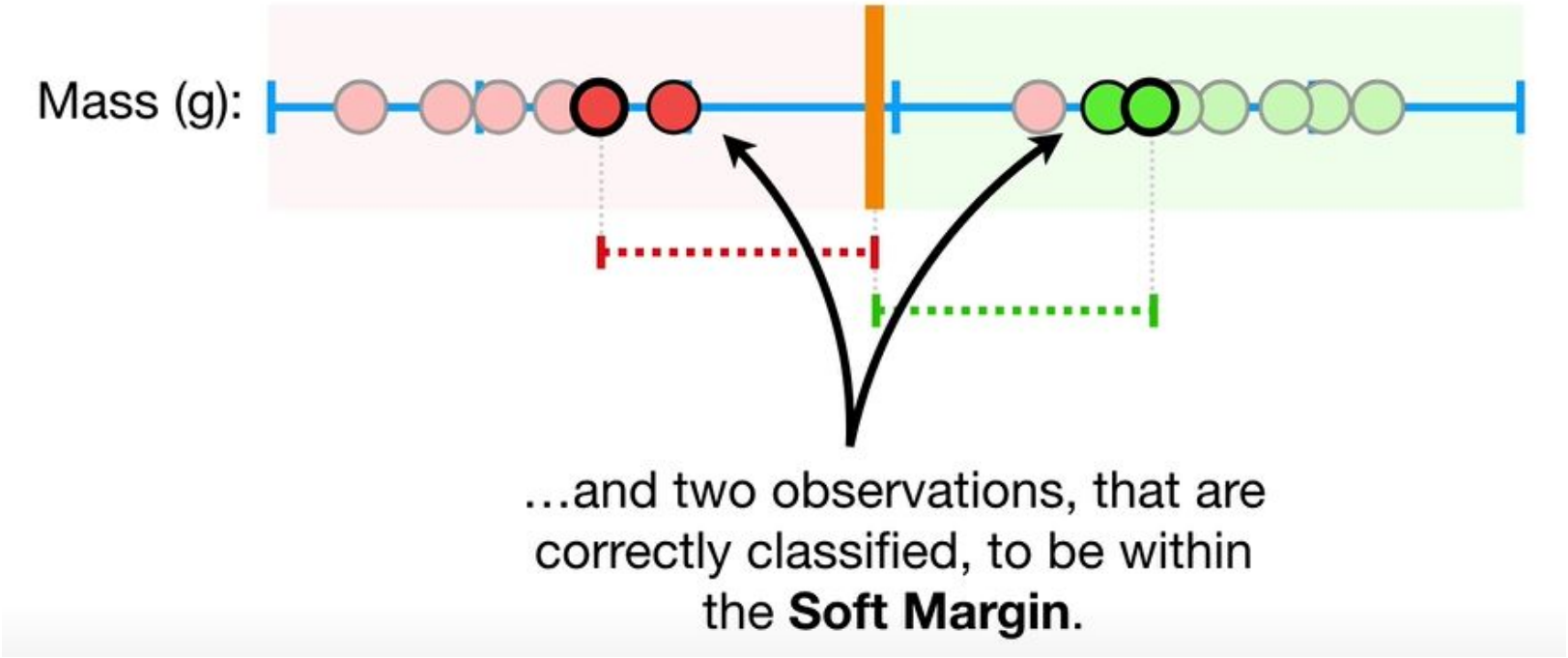
Soft Margin dopo la k-fold cross validation



Soft Margin dopo la k-fold cross validation

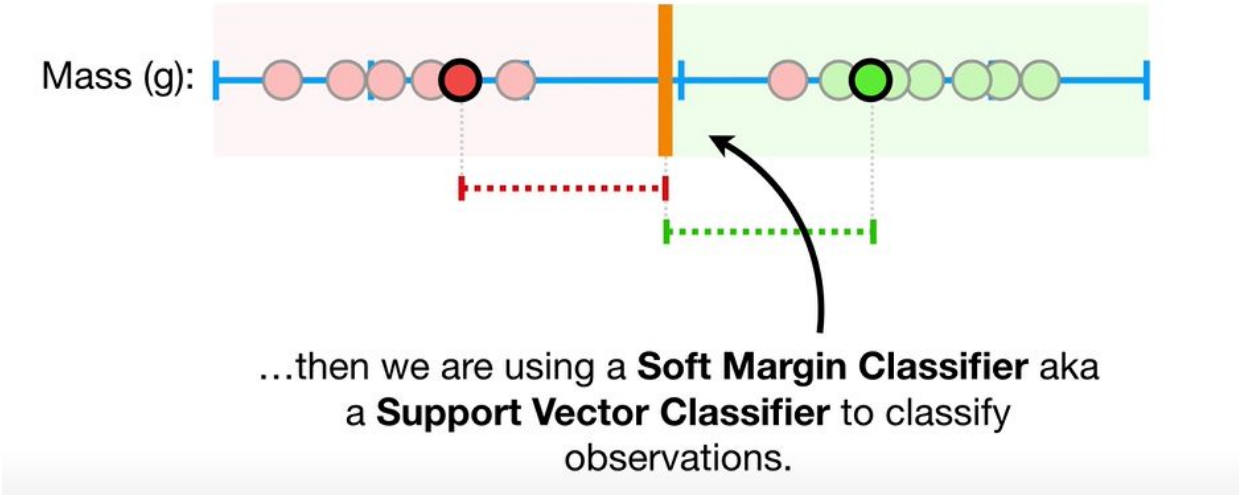


Soft Margin dopo la k-fold cross validation

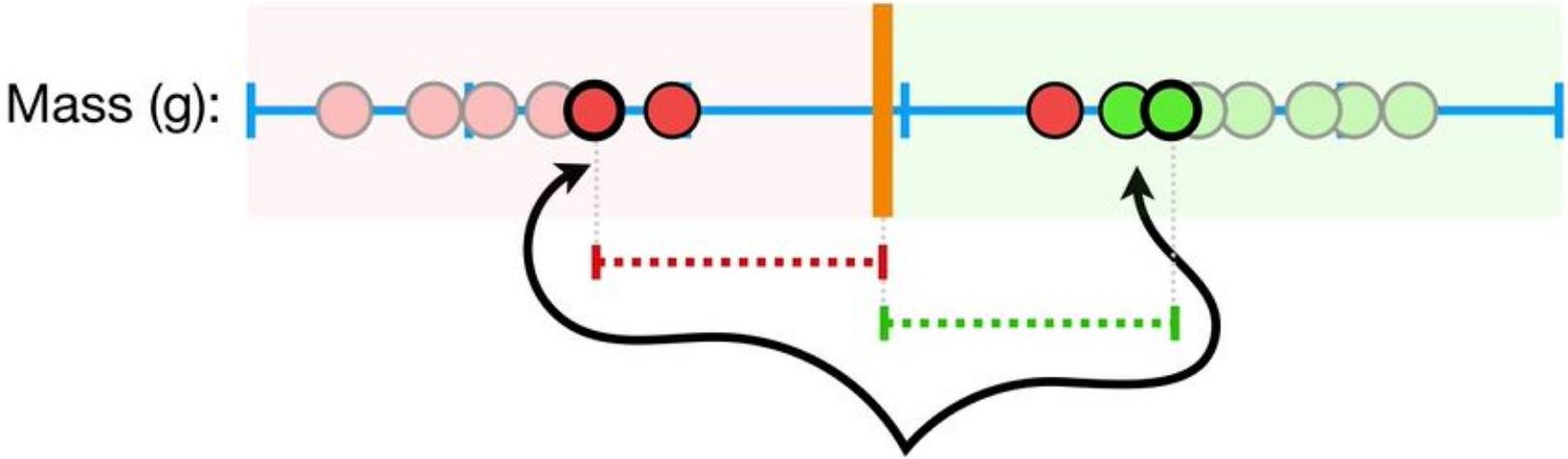


Notazione

- Quando usiamo il **margine** standard, abbiamo il **maximum margin classifier**
- Quando usiamo il **Soft Margin**, abbiamo il **soft margin classifier**, chiamato anche support vector classifier per *classificare le osservazioni*

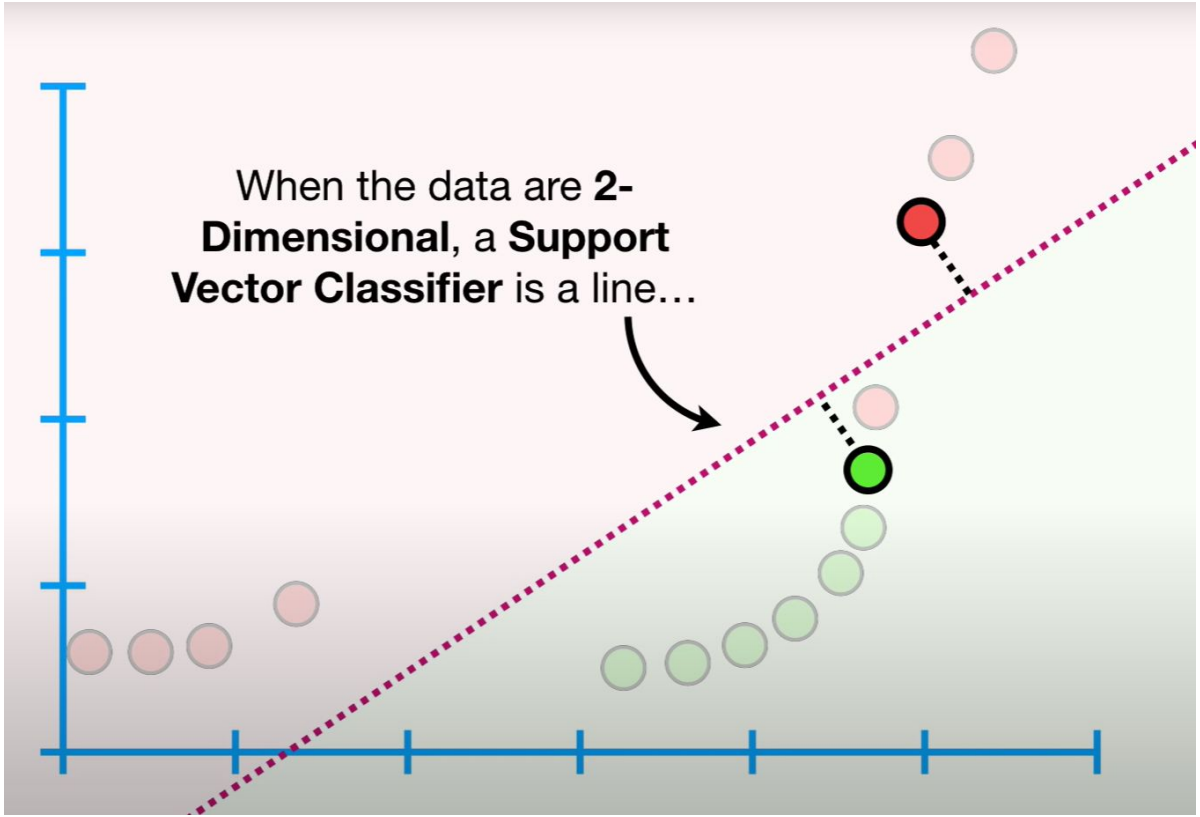


Definizione di **support vectors**

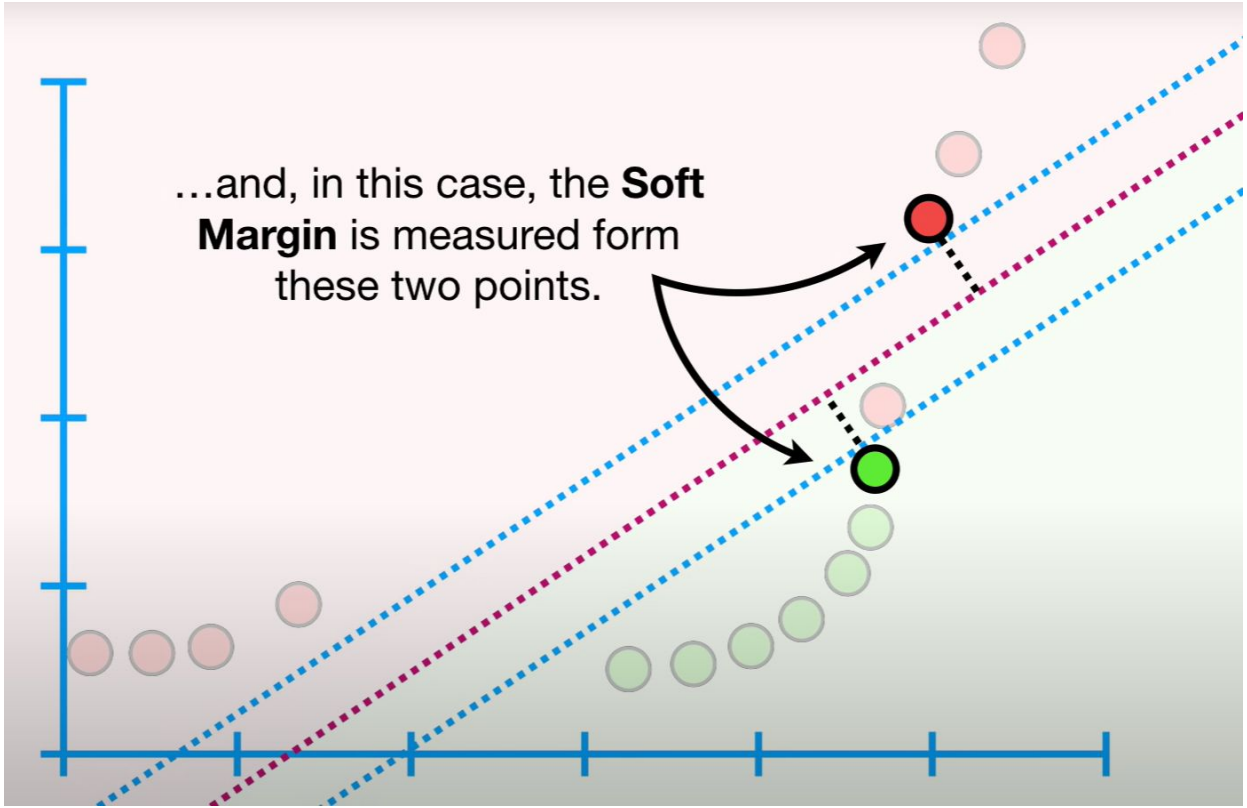


The name **Support Vector Classifier** comes from the fact that the observations on the edge *and within* the **Soft Margin** are called **Support Vectors**.

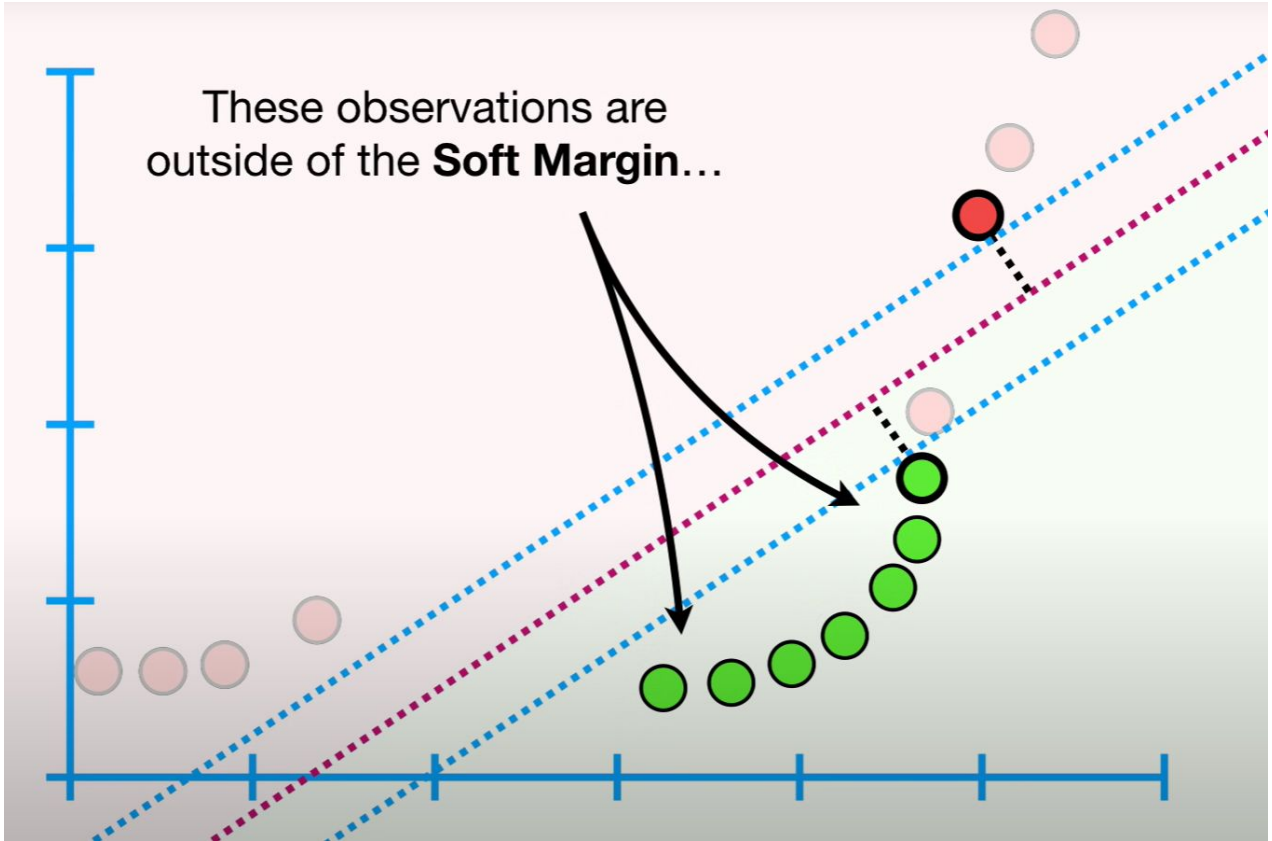
Quando i dati sono bidimensionali …



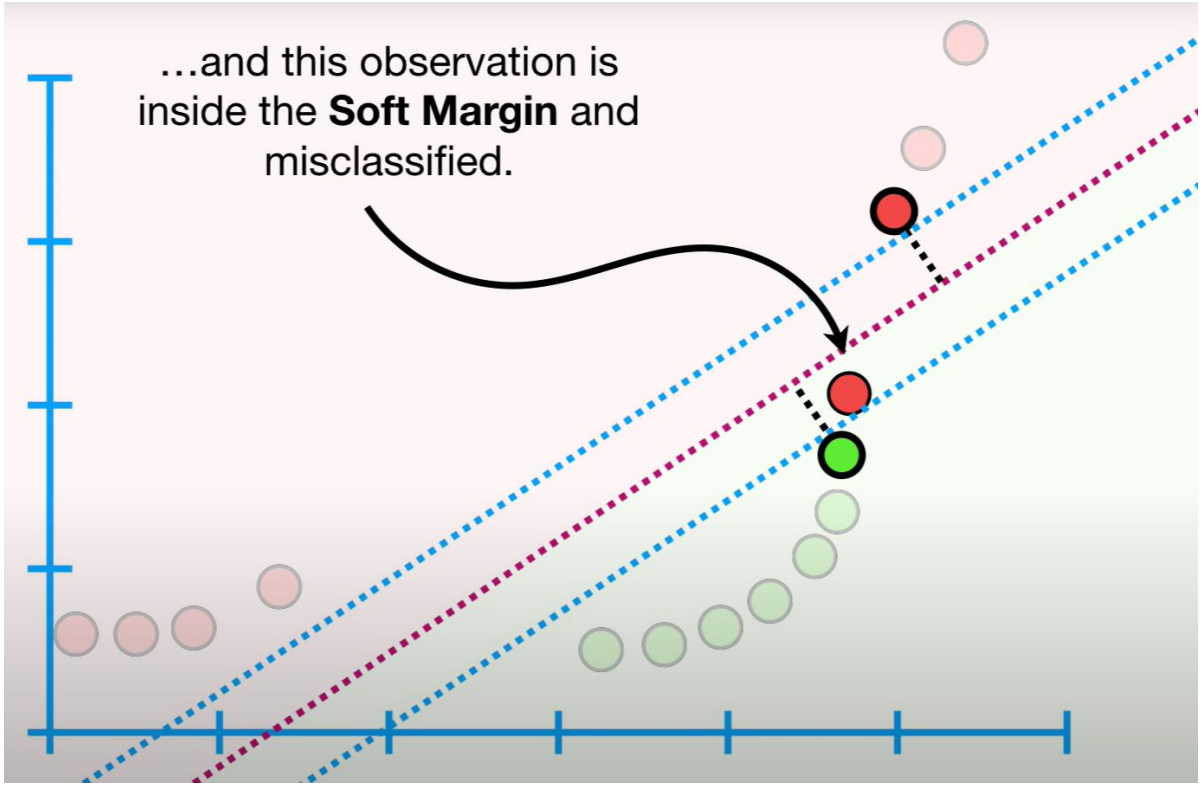
Quando i dati sono bidimensionali …



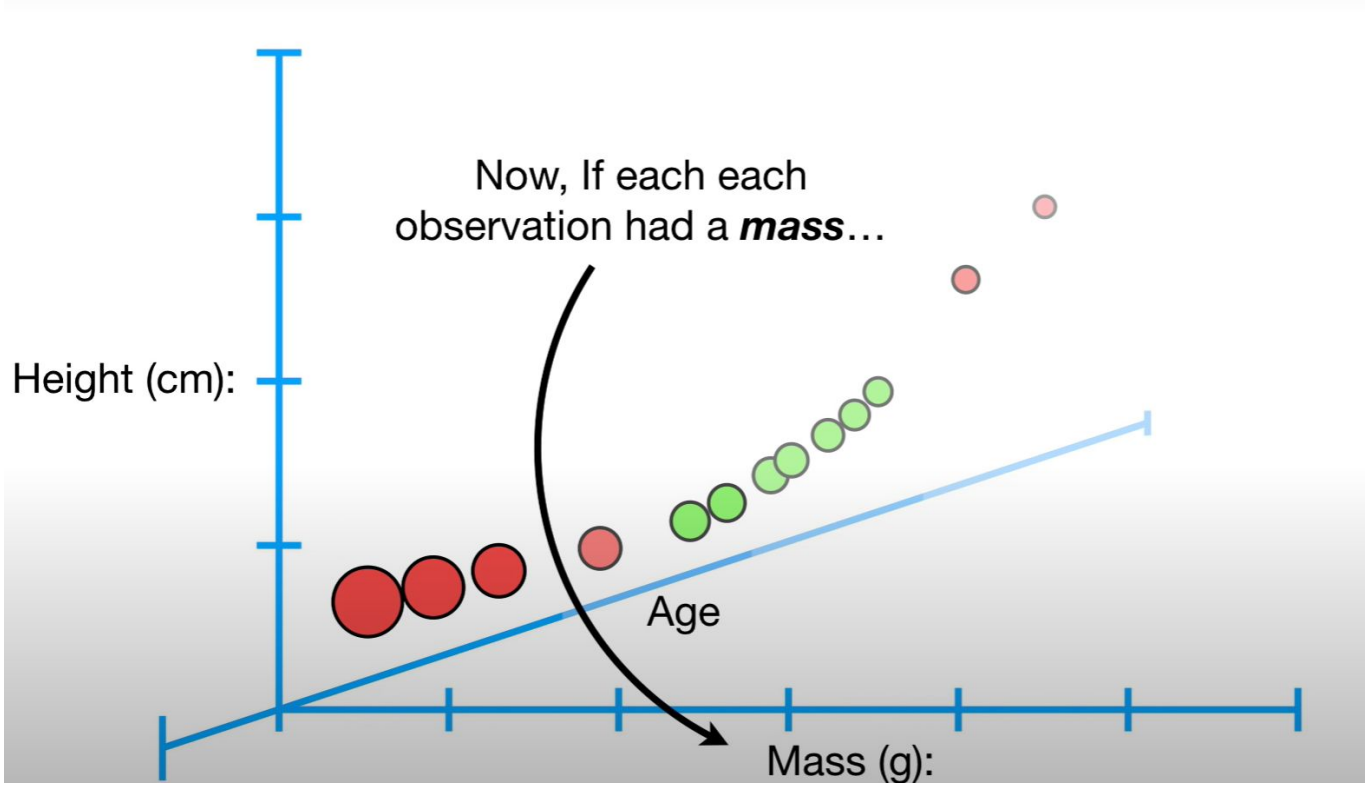
Quando i dati sono bidimensionali …



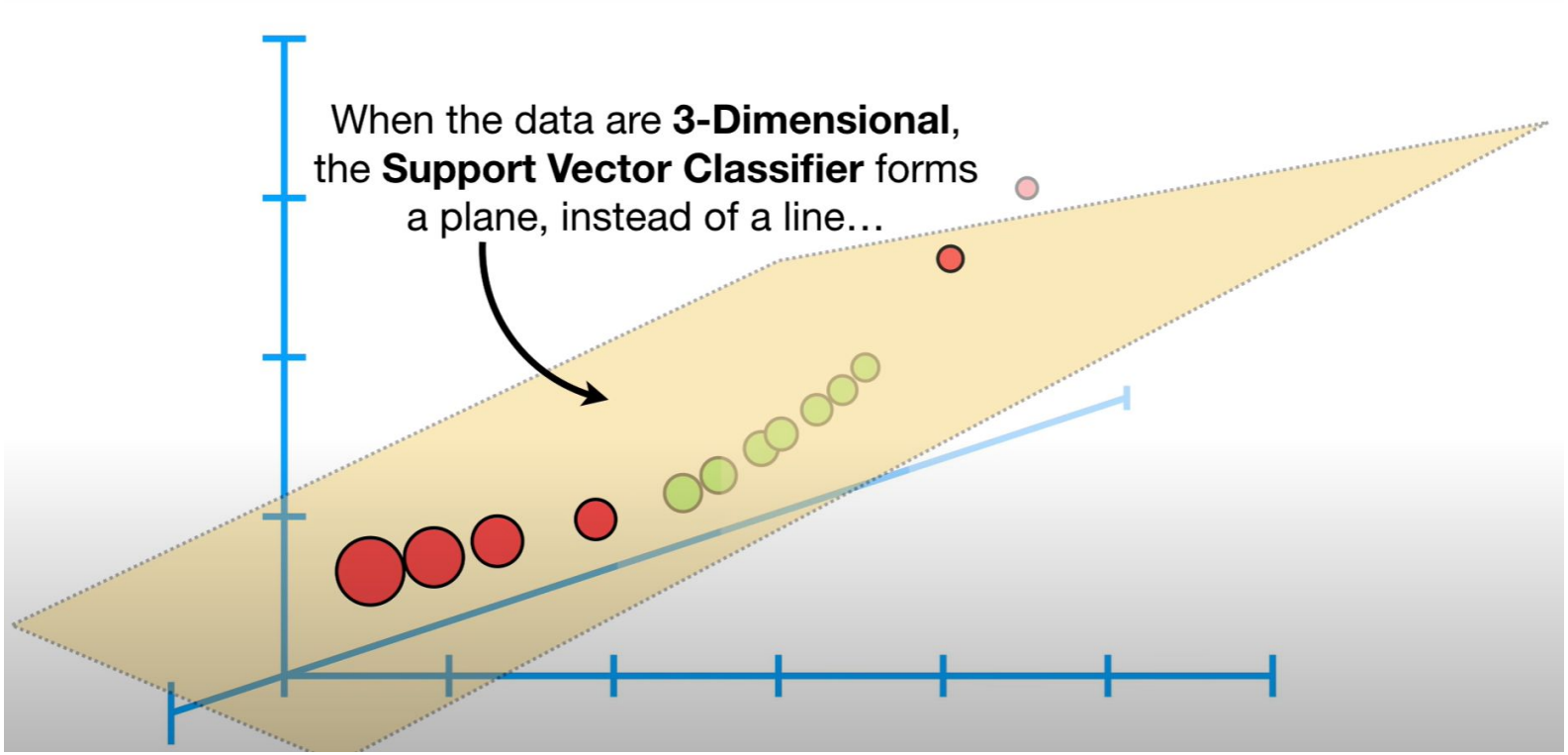
Quando i dati sono bidimensionali …



Quando i dati sono tridimensionali …

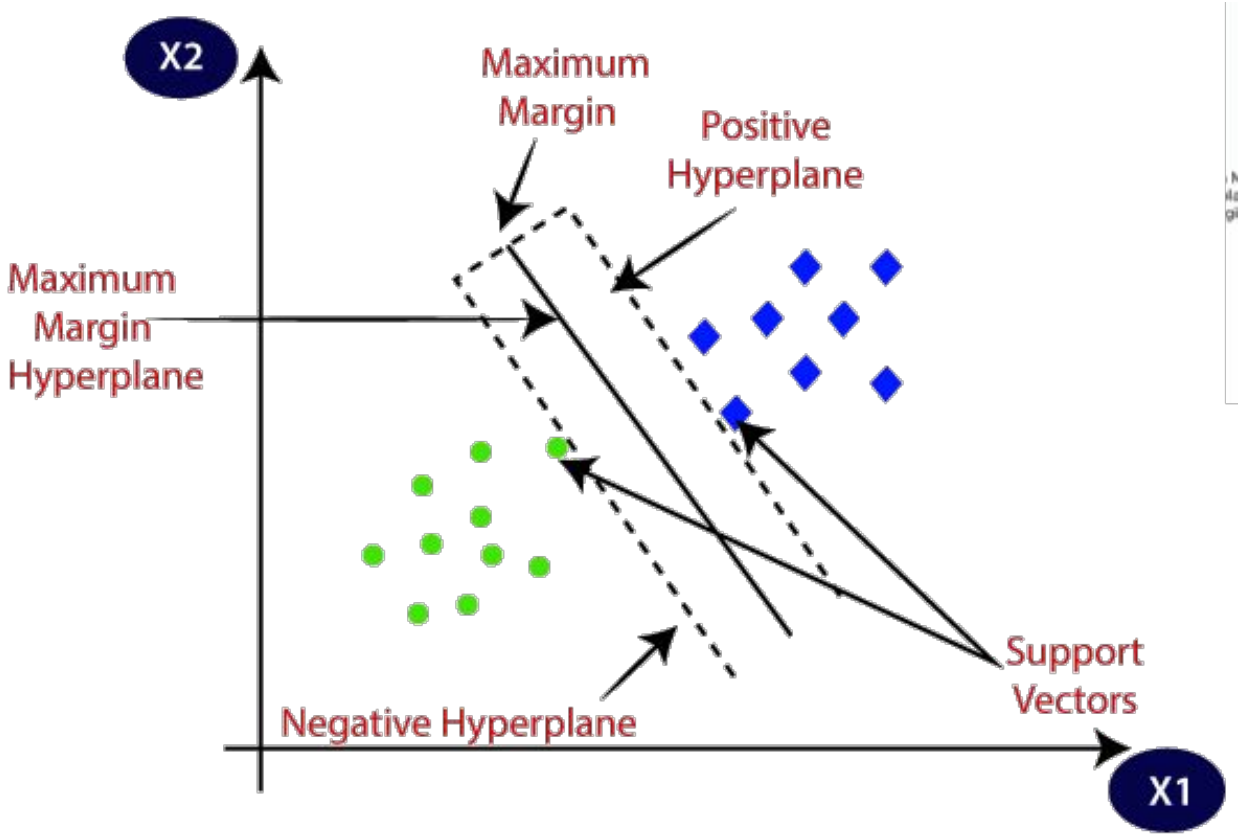


Quando i dati sono tridimensionali …



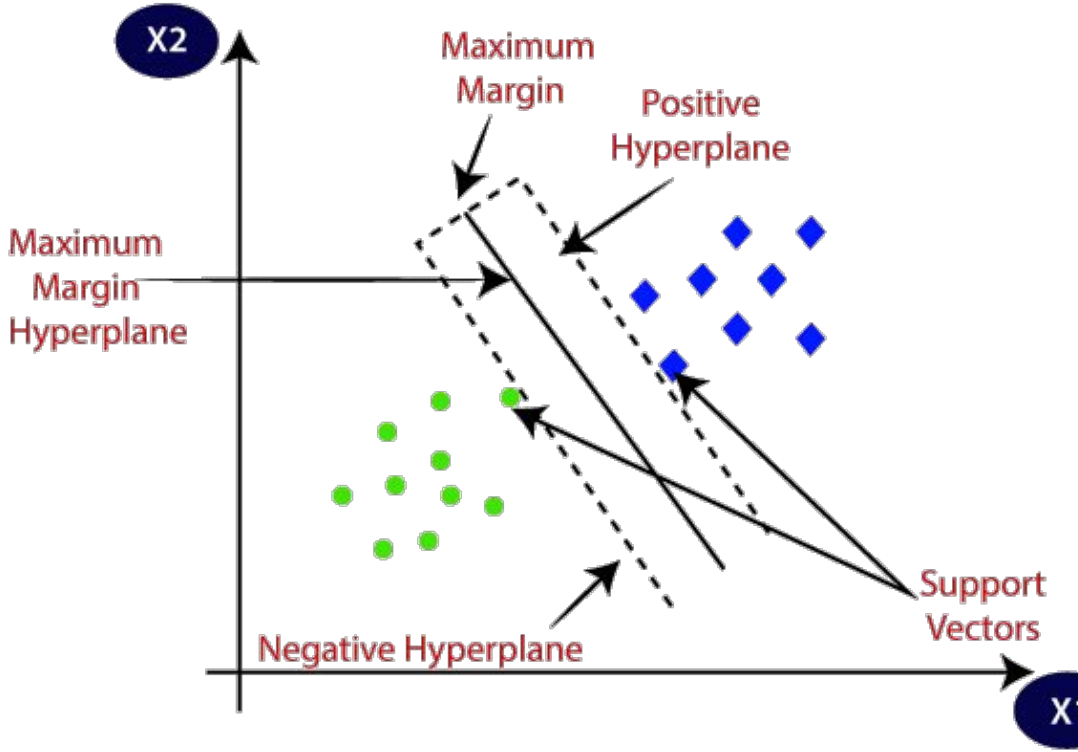
E quando i dati hanno dimensione >= 4?

Quando i dati hanno 4 o più dimensioni, il Support Vector Classifier è un **iperpiano** (hyperplane)

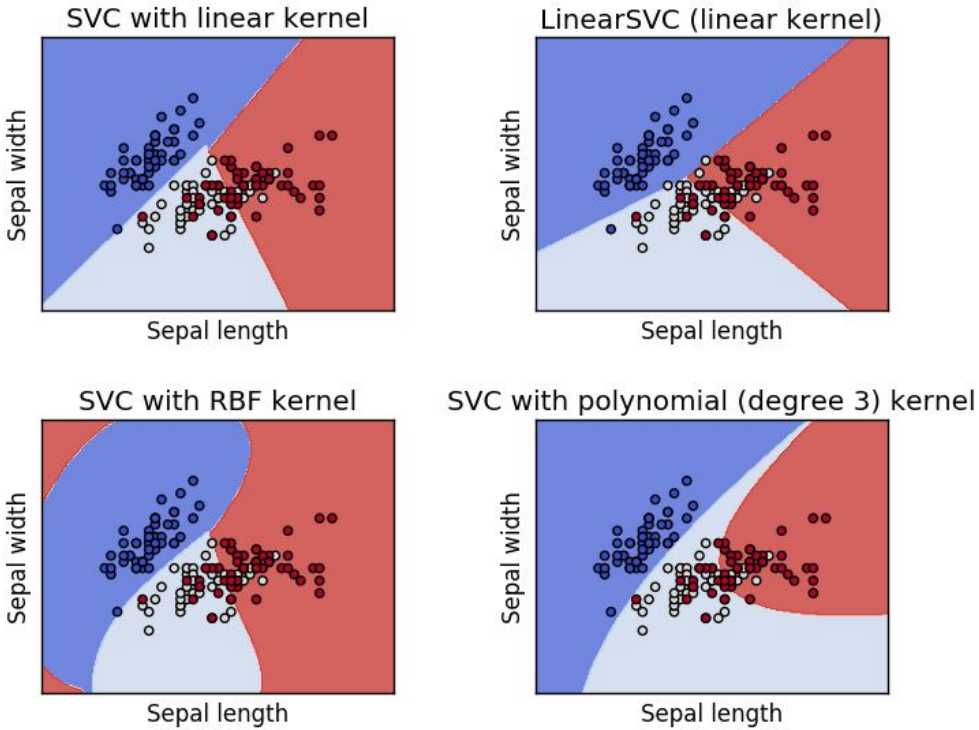

X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

Riassunto del SVM

- SVM cerca un iperpiano che massimizzi il margine tra le due classi
- Supporta anche una tecnica, chiamata **Kernel trick**, per i trasformare i dati non linearmente separabili in *dati di dimensione superiore* che possono essere linearmente separabili



Risultati con diversi kernel



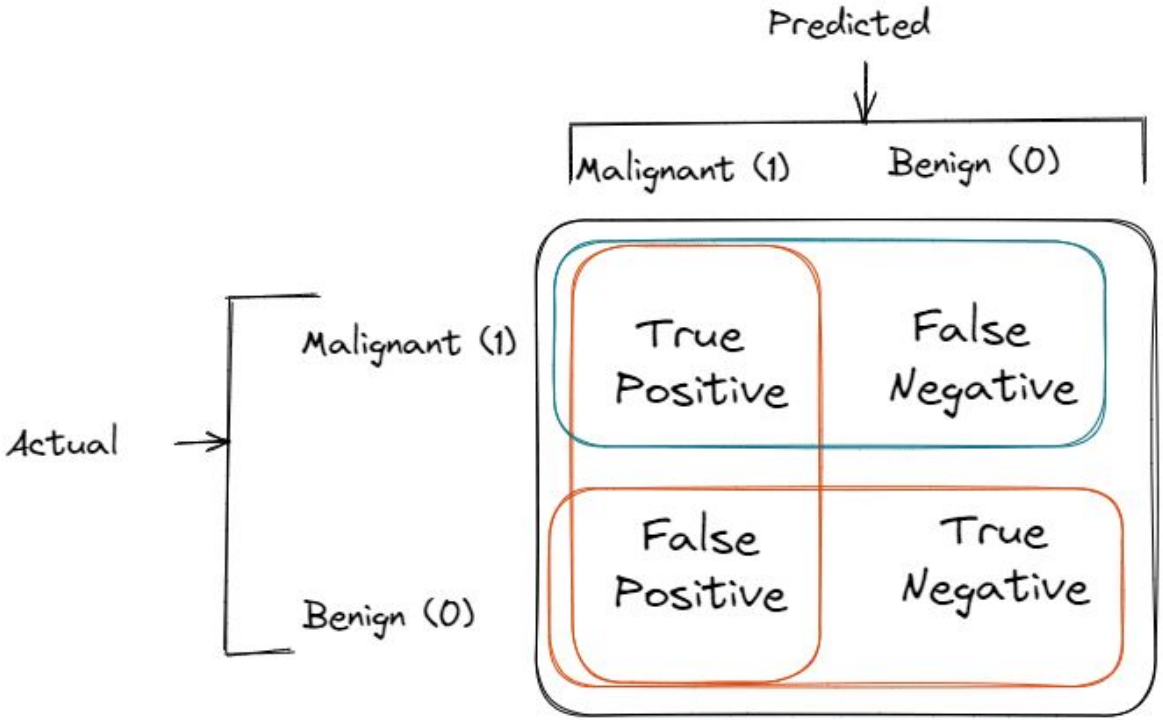
Vantaggi di SVM

- È efficace in dati ad alta dimensionalità
- Versatile grazie al kernel trick
- Robusto perché riesce a gestire outliers
- Flessibile perché permette classificazioni errate
- Ma … ha anche svantaggi
  - Non scala molto bene con grandi volumi di dati
  - Sensibile alla scelta di iperparametri
  - Non molto intuitivo da interpretare

Come valuto le performance del mio modello

- Classificazione
  - Accuratezza
  - Precision
  - Recall
  - F1-Score
- Regressione
  - MAE
  - MSE
  - RMSE

Confusion Matrix



Accuracy

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

- Si usa quando il dataset è bilanciato
  - Esempio: la classe ‘tumore benigno’ deve avere lo stesso numero di osservazioni della classe ‘tumore maligno’
- Ci poniamo la domanda: Il nostro modello ha fatto le previsioni corrette per entrambe le classi?

Precision

$$\text{Precision} = \text{True Positives} / (\text{True Positives} + \text{False Positives})$$

- Si usa quando
  - il dataset è **sbilanciato**
  - Si vuole minimizzare il numero di falsi positivi
- Falso positivo significa che il nostro modello dice che il paziente ha un tumore maligno, ma in realtà non ha un tumore maligno

Recall

$$\text{Recall} = \text{True Positives} / (\text{True Positives} + \text{False Negatives})$$

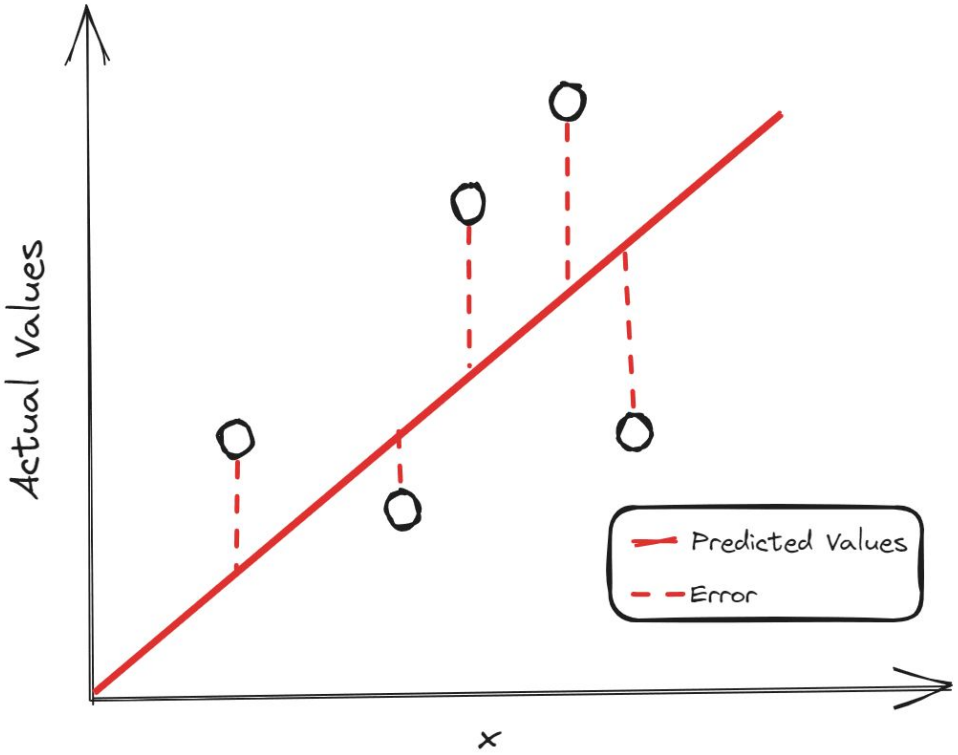
- Si usa quando
  - il dataset è **sbilanciato**
  - Si vuole minimizzare il numero di falsi negativi
- Falso negativo significa che il paziente ha un tumore maligno, ma il modello non lo ha identificato

F1-score

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) \ / \ (\text{Precision} + \text{Recall})$$

- Si usa quando
  - il dataset è **sbilanciato**
  - Si vuole una misura che riassume Precision e Recall

Regressione



MAE

$$MAE = \frac{1}{n} \sum_{i=1}^{n} | y_i - \hat{y}_i |$$

- Differenza in valore assoluto tra i valori delle previsioni e i valori attuali della variabile target

MSE

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

- Differenza al quadrato tra i valori delle previsioni e i valori attuali della variabile target
  - Gli errori grandi peseranno di più

RMSE

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

- Radice quadrata del MSE
  - Gli errori grandi peseranno di più

Resources:

- KNN tutorial
- SVM video di StatQuest
- Statistical Learning di Trevor Hastie