

Dimensione Cognitiva

2. Macchine che Imparano

Introduzione al Machine Learning

Giovanni Della Lunga
giovanni.dellalunga@unibo.it

A lezione di Intelligenza Artificiale

Siena - Giugno 2025

- 1 Algoritmi Tradizionali vs Machine Learning
- 2 Tutto comincia con una Retta...
- 3 Il Processo di Addestramento
- 4 Dalla Regressione alla Classificazione
- 5 Un Salto Concettuale: Dai Numeri alla Geometria

Algoritmi Tradizionali vs Machine Learning

Definizione

Un algoritmo tradizionale è una sequenza di istruzioni precise e predefinite che il computer esegue passo dopo passo per risolvere un problema specifico.

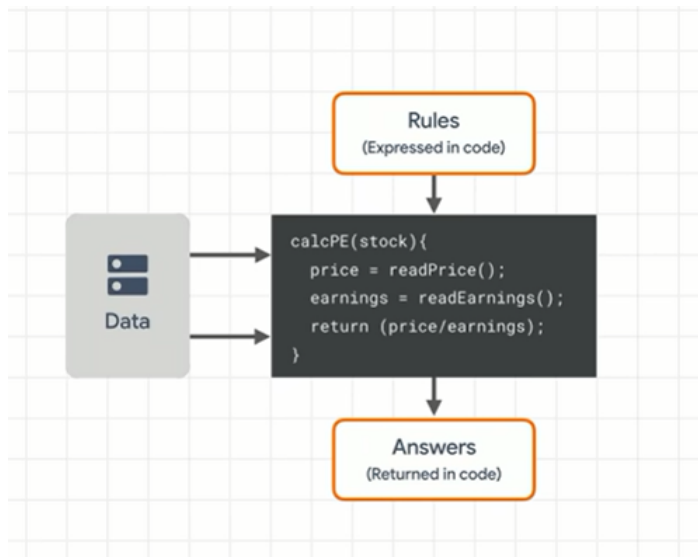
Caratteristiche:

- Le regole sono scritte esplicitamente dal programmatore
- Il comportamento è completamente spiegabile e prevedibile
- Infatti la logica è trasparente e verificabile

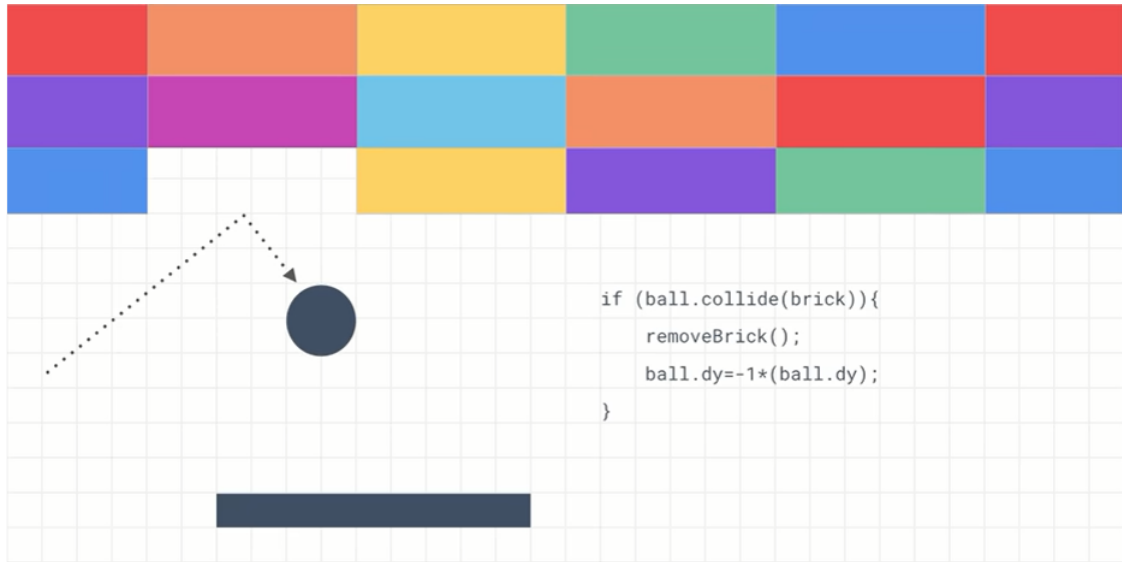
Esempio concreto - Calcolo dello sconto:

- SE il cliente spende più di 100€ ALLORA applica sconto 10%
- ALTRIMENTI nessuno sconto
- Il programmatore ha definito esattamente quando e come applicare lo sconto

Algoritmo Tradizionale: Dalle Regole al Codice



Algoritmo Tradizionale: Dalle Regole al Codice



Un gioco più complicato ...

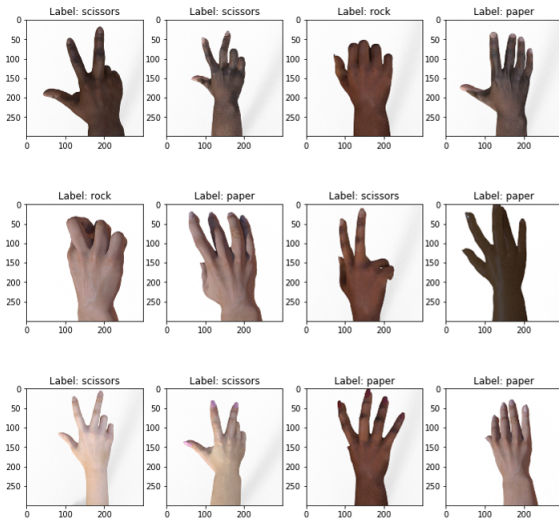


Image shape: (300, 300, 3)

```
[[[254 254 254]
 [253 253 253]
 [254 254 254]
 ...
 [251 251 251]
 [250 250 250]
 [250 250 250]]]
```

```
[[[254 254 254]
 [254 254 254]
 [253 253 253]
 ...
 [250 250 250]
 [251 251 251]
 [249 249 249]]]
```

```
[[[254 254 254]
 [254 254 254]
 [254 254 254]
 ...
 [251 251 251]
 [250 250 250]
 [252 252 252]]]
```

...

Definizione

Il Machine Learning è un approccio in cui l'algoritmo scopre automaticamente le regole analizzando grandi quantità di dati, senza che queste regole vengano programmate esplicitamente.

Caratteristiche:

- Le regole emergono dall'analisi dei dati
- Il comportamento può variare in base ai dati di addestramento
- L'algoritmo può gestire situazioni non previste dal programmatore
- La logica interna è spesso complessa e non direttamente interpretabile

Esempio concreto - Rilevamento frodi:

- L'algoritmo analizza milioni di transazioni passate
- Identifica automaticamente pattern sospetti
- Impara a distinguere transazioni normali da quelle fraudolente
- Non esistono regole esplicite scritte dal programmatore

Confronto Diretto: Riconoscimento di Spam

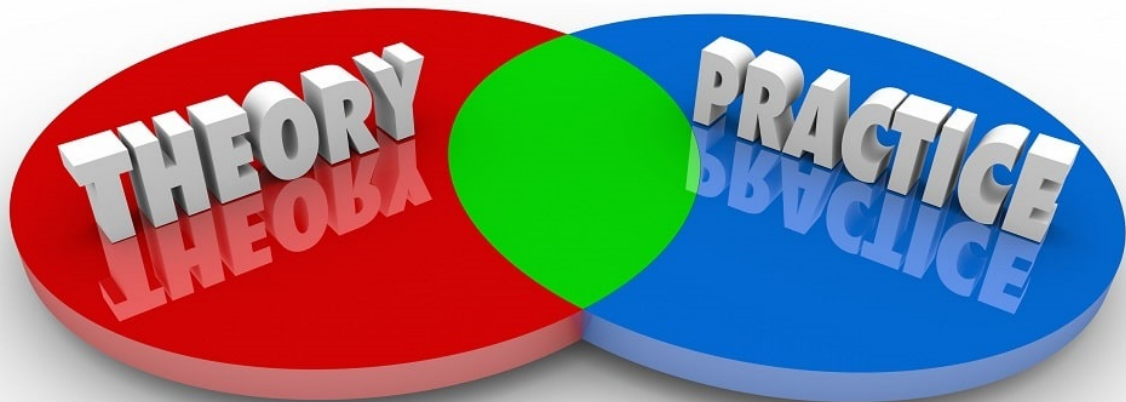
Approccio Tradizionale

- Il programmatore scrive regole:
 - Se contiene "GRATIS" → spam
 - Se ha più di 5 punti esclamativi → spam
 - Se mittente sconosciuto → spam
- Ogni regola è esplicita
- Facile da capire ma limitato
- Non si adatta a nuovi tipi di spam

Machine Learning

- L'algoritmo analizza:
 - 100.000 email spam
 - 100.000 email legittime
- Scopre automaticamente pattern:
 - Frequenza di certe parole
 - Struttura del testo
 - Caratteristiche del mittente
- Si adatta a nuovi tipi di spam

Tutto comincia con una Retta...



Il Problema: Valutare un Appartamento

Esempio pratico

Come può un computer determinare il prezzo di un appartamento?

- Analizzando migliaia di vendite passate
- Identificando le caratteristiche che influenzano il prezzo
- Creando un modello predittivo

Il Problema: Valutare un Appartamento

Caratteristiche dell'appartamento:

- Superficie (m^2)
- Numero di stanze
- Piano
- Zona della città
- Età dell'edificio

Domanda:

Quanto vale questo appartamento?

Approccio tradizionale:

- Perizia manuale
- Confronto con vendite simili
- Esperienza dell'agente

Soluzione: Regressione Lineare

Idea base: Trovare una relazione matematica tra caratteristiche e prezzo

Modello Semplificato (una variabile)

$$\text{Prezzo} = a \times \text{Superficie} + b$$

Modello Completo (più variabili)

$$\text{Prezzo} = a_1 \times \text{Superficie} + a_2 \times \text{Stanze} + a_3 \times \text{Piano} + b$$

L'algoritmo impara:

- I **coefficienti** a_1, a_2, a_3, \dots (quanto influisce ogni caratteristica)
- L'**intercetta** b (prezzo base)

Come Funziona l'Apprendimento?

- 1 **Training:** Raccogliamo dati di appartamenti già venduti
 - Es: 90m², 3 stanze, 2° piano → venduto a €180.000
- 2 **Algoritmo:** Trova la retta che meglio approssima i dati
 - Minimizza l'errore tra prezzi reali e predetti
- 3 **Predizione:** Utilizziamo il modello per nuovi appartamenti
 - Es: 75m², 2 stanze, 1° piano → prezzo stimato?

Esempio Numerico

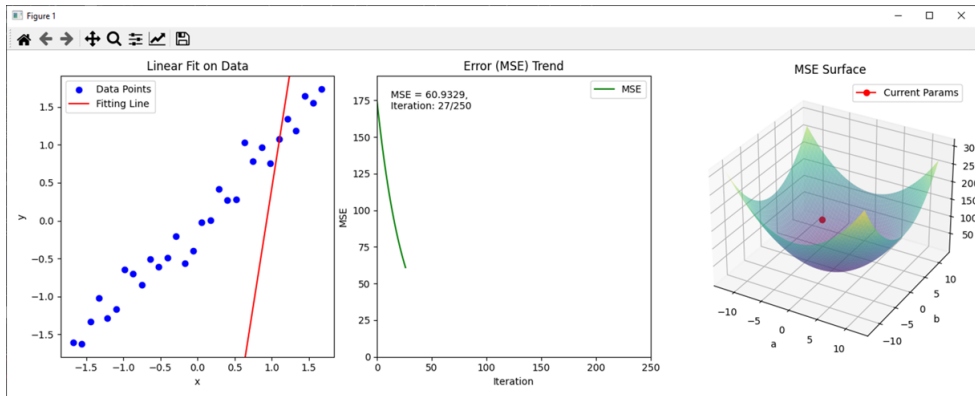
Se il modello impara: $\text{Prezzo} = 2000 \times \text{Superficie} + 15000 \times \text{Stanze} + 5000$

Per 75m², 2 stanze: $\text{Prezzo} = 2000 \times 75 + 15000 \times 2 + 5000 = \text{€185.000}$

Gli Ingredienti Fondamentali del Machine Learning

Un esempio semplice, ma completo!

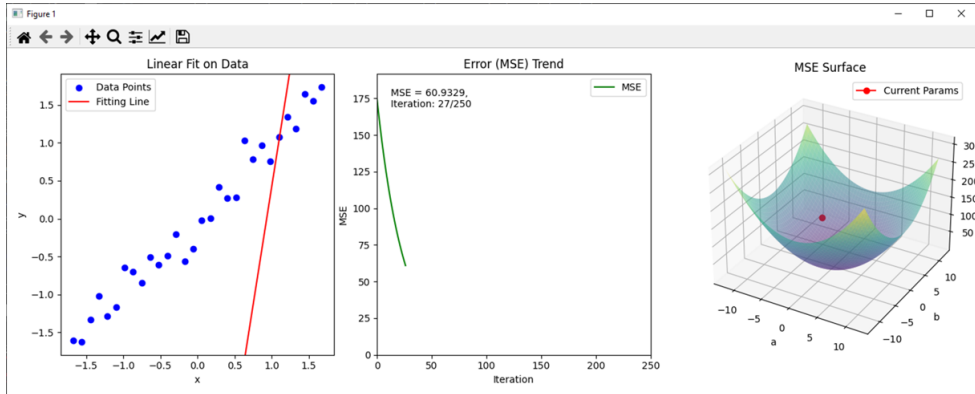
Sebbene l'esempio della regressione lineare sia molto semplice, esso contiene **tutti** gli ingredienti fondamentali del Machine Learning:



Gli Ingredienti Fondamentali del Machine Learning

Data-Driven (Guidato dai Dati)

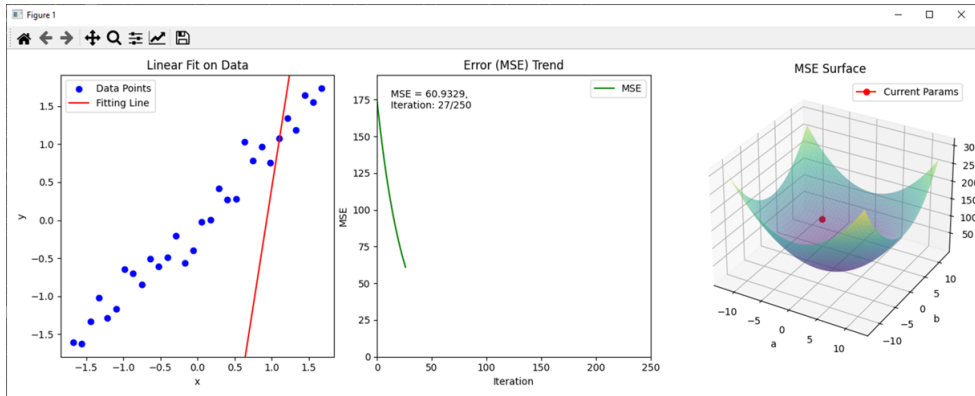
- Le decisioni non sono programmate manualmente
- Il modello impara direttamente dai dati storici di vendita



Gli Ingredienti Fondamentali del Machine Learning

Funzione di Errore

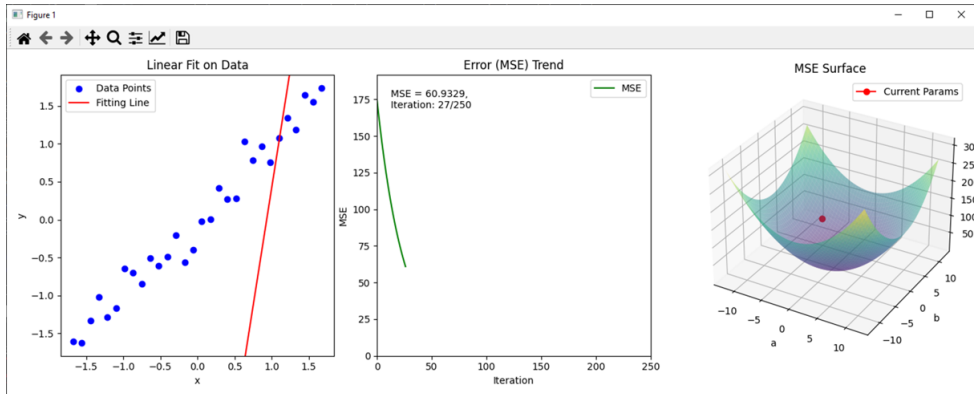
- Misuriamo quanto le nostre predizioni si discostano dalla realtà
- Es: Errore Quadratico Medio (MSE)



Gli Ingredienti Fondamentali del Machine Learning

Metodo di Ottimizzazione

- Algoritmo che cerca i parametri che minimizzano l'errore
- Es: Gradient Descent, Least Squares



Il Processo di Addestramento

Che cos'è l'addestramento

L'addestramento è il processo computazionale attraverso cui un algoritmo di machine learning analizza un dataset per identificare pattern statistici e costruire un modello matematico capace di fare predizioni su dati nuovi.

Componenti essenziali:

- 1 **Dataset di addestramento:** Insieme di esempi con input e output desiderati
- 2 **Algoritmo di apprendimento:** Procedura matematica che trova i pattern
- 3 **Funzione di costo:** Misura quanto l'algoritmo sbaglia
- 4 **Ottimizzazione:** Processo per ridurre gli errori

Fasi del Processo di Addestramento

1. Inizializzazione

- L'algoritmo inizia con parametri casuali
- Non sa ancora come risolvere il problema

2. Presentazione dei dati

- L'algoritmo riceve un esempio dal dataset
- Prova a fare una predizione con i parametri attuali

3. Calcolo dell'errore

- Confronta la sua predizione con la risposta corretta
- Calcola numericamente quanto ha sbagliato

4. Aggiornamento dei parametri

- Modifica leggermente i suoi parametri interni
- L'obiettivo è ridurre l'errore per esempi simili

5. Iterazione

- Ripete il processo per tutti gli esempi nel dataset
- Continua per molti cicli (epoche) fino a convergenza

Esempio Dettagliato: Predizione Prezzi Case

Dataset: 10.000 case con caratteristiche e prezzi reali

Input per ogni casa:

- Superficie (mq), Numero stanze, Età, Distanza dal centro

Output: Prezzo di vendita

Processo di addestramento:

- 1 L'algoritmo inizia con una formula con parametri casuali:

$$\text{Prezzo} = a \times \text{Superficie} + b \times \text{Stanze} + c \times \text{Eta} + d$$

- 2 Per la prima casa (100mq, 3 stanze, 10 anni): predice 150.000 EUR
- 3 Il prezzo reale era 200.000 EUR → errore di 50.000 EUR
- 4 Aggiusta i coefficienti a, b, c, d per ridurre questo errore
- 5 Ripete per tutte le 10.000 case
- 6 Dopo molte iterazioni, la formula diventa accurata

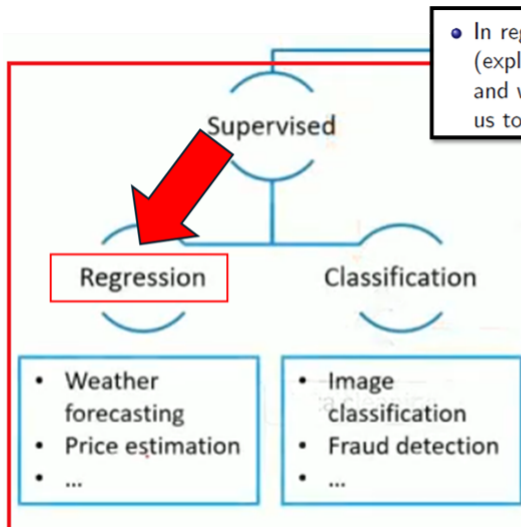
Un po' di Lessico: Features e Labels

The diagram illustrates a dataset table with four rows and five columns. A bracket above the first four columns is labeled 'Features', and a bracket above the last column is labeled 'Label'. A bracket to the left of the rows is labeled 'Rows', and a bracket below the columns is labeled 'Columns'.

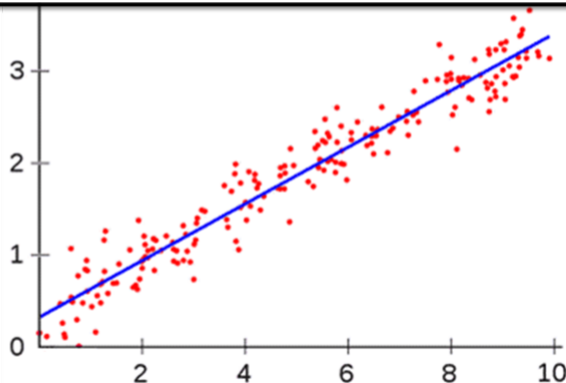
| Size | Beds | Baths | Zip | Price |
|------|------|-------|-------|-------|
| 1100 | 1 | 1 | 64576 | 1.29 |
| 1900 | 3 | 1.5 | 78321 | 2.14 |
| 2800 | 3 | 3 | 98712 | 3.10 |
| 3400 | 4 | 3.5 | 25721 | 3.75 |

Dalla Regressione alla Classificazione

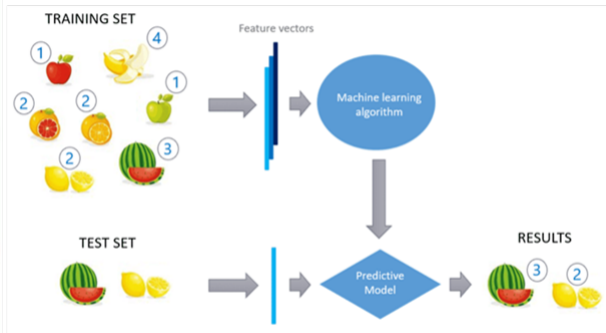
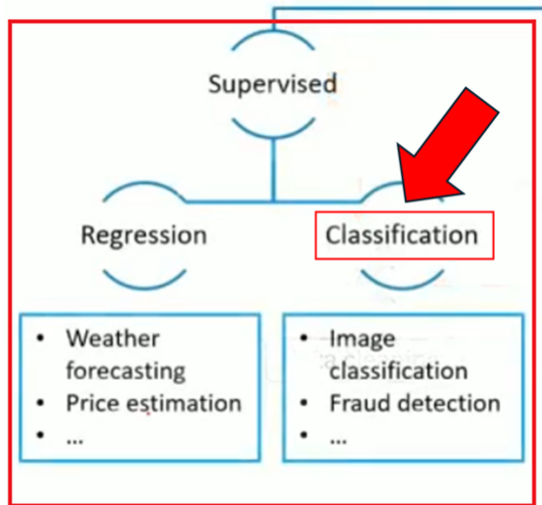
Dalla Regression alla Classificazione



- In regression analysis, we are given a number of predictor (explanatory) variables and a continuous response variable (outcome), and we try to find a relationship between those variables that allows us to predict an outcome.



Dalla Regressione alla Classificazione



Dalla Regressione alla Classificazione

Regressione

- Predice valori **continui**
- Output: numeri reali
- Esempio: prezzo €185.000

Formula

$$y = w_1x_1 + w_2x_2 + b$$

dove $y \in \mathbb{R}$

Classificazione

- Predice **categorie**
- Output: classi discrete
- Esempio: "Spam" o "Non Spam"

Stesso Principio!

$$\text{Decisione} = f(w_1x_1 + w_2x_2 + b)$$

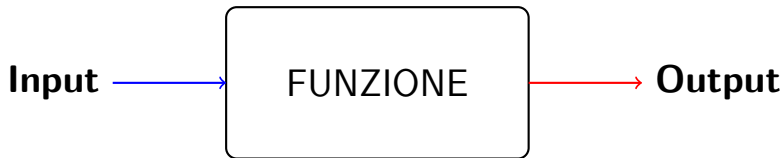
dove f trasforma in categorie

Concetto Chiave

Anche nella classificazione cerchiamo i **pesi ottimali** w_1, w_2, \dots che minimizzano l'errore!

Che cos'è una funzione?

Una funzione è come un **operatore**:



- Riceve un **input** (dato di ingresso)
- Esegue una **trasformazione**
- Produce un **output** (risultato)

Che cos'è una funzione?

$$f(x) = y$$

- f è il nome della funzione
- x è l'input (variabile indipendente)
- y è l'output (variabile dipendente)

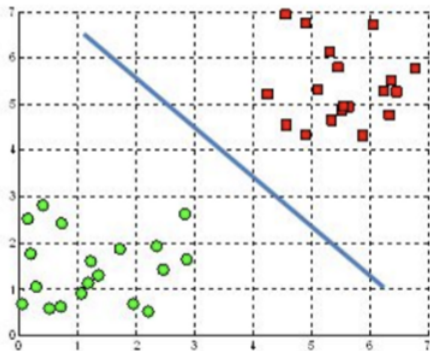
Esempio semplice:

$$f(x) = 2x + 1$$

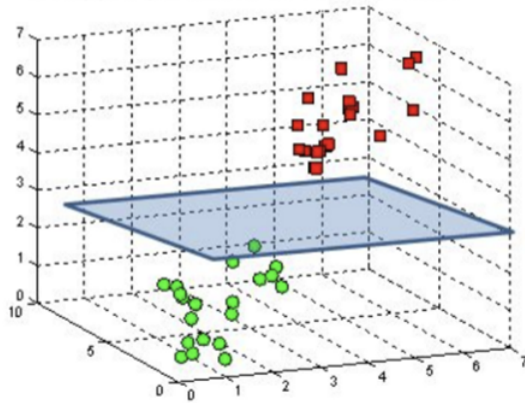
- Se $x = 3$, allora $f(3) = 2 \cdot 3 + 1 = 7$
- Se $x = 0$, allora $f(0) = 2 \cdot 0 + 1 = 1$

Dalla Regressione alla Classificazione

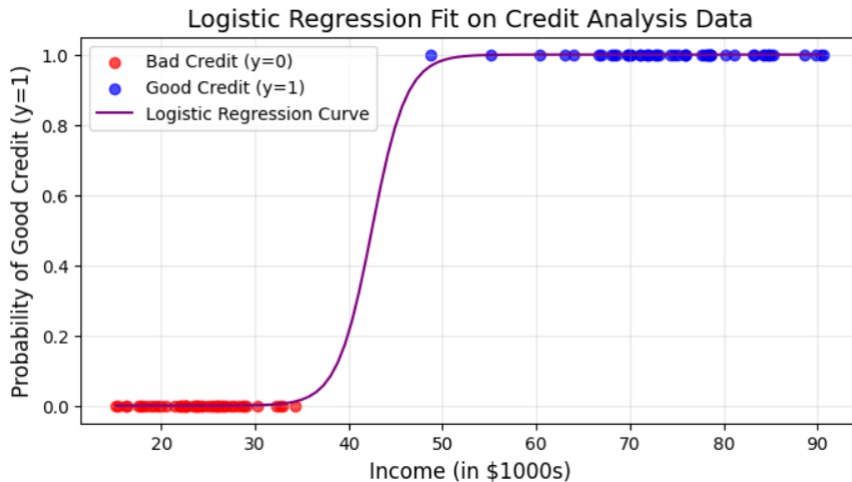
A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane



Dalla Regressione alla Classificazione



Esempio: Rilevamento Email Spam

Problema: Classificare automaticamente le email come "Spam" o "Non Spam"

Caratteristiche dell'Email (Features)

- x_1 : Numero di parole "GRATIS"
- x_2 : Numero di punti esclamativi
- x_3 : Presenza di link sospetti (0 o 1)
- x_4 : Lunghezza dell'email

Modello Lineare

$$\text{Score} = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 + b$$

Esempio: Rilevamento Email Spam

Problema: Classificare automaticamente le email come "Spam" o "Non Spam"

Caratteristiche dell'Email (Features)

- x_1 : Numero di parole "GRATIS"
- x_2 : Numero di punti esclamativi
- x_3 : Presenza di link sospetti (0 o 1)
- x_4 : Lunghezza dell'email

Regola di Decisione:

- Se $\text{Score} > 0 \rightarrow \text{SPAM}$
- Se $\text{Score} \leq 0 \rightarrow \text{NON SPAM}$

Supponiamo che l'algoritmo impari questi pesi:

Modello Appreso

$$\text{Score} = +3.2 \cdot x_1 + +1.8 \cdot x_2 + +5.1 \cdot x_3 + -0.01 \cdot x_4 + 0.5$$

Interpretazione dei Pesi:

- $w_1 = +3.2$: Ogni "GRATIS" aumenta molto la probabilità di spam
- $w_2 = +1.8$: I punti esclamativi sono indicatori di spam
- $w_3 = +5.1$: I link sospetti sono il segnale più forte di spam
- $w_4 = -0.01$: Email più lunghe tendono a essere meno spam

Supponiamo che l'algoritmo impari questi pesi:

Modello Appreso

$$\text{Score} = +3.2 \cdot x_1 + +1.8 \cdot x_2 + +5.1 \cdot x_3 + -0.01 \cdot x_4 + 0.5$$

Esempio Concreto

Email con: 2 "GRATIS", 5 "!", 1 link sospetto, 200 parole

$$\text{Score} = 3.2 \times 2 + 1.8 \times 5 + 5.1 \times 1 - 0.01 \times 200 + 0.5 = 18.9 > 0 \rightarrow \text{SPAM}$$

Come Trovare i Pesi Ottimali?

Stesso processo della regressione, ma con funzione di errore diversa!

1 Dati di Training

- Migliaia di email già etichettate: (features, label)
- Es: $([2, 5, 1, 200], \text{"Spam"})$, $([0, 1, 0, 50], \text{"Non Spam"})$

2 Funzione di Errore

- Non più errore quadratico, ma **Cross-Entropy Loss**
- Penalizza classificazioni sbagliate

3 Ottimizzazione

- Gradient Descent (come nella regressione!)
- Cerca i pesi w_1, w_2, w_3, w_4, b che minimizzano l'errore

L'Unità Fondamentale del Machine Learning

Regressione e Classificazione: Stessa Filosofia

Entrambi i problemi seguono lo stesso schema fondamentale:

1. Dati

Input + Output
di training



2. Modello

Combinazione lineare
 $\sum w_i x_i + b$



3. Ottimizzazione

Trova i pesi w_i
migliori

Differenze Principali

- **Regressione:** Output continuo, Errore Quadratico
- **Classificazione:** Output discreto, Cross-Entropy Loss
- **Entrambi:** Cercano pesi ottimali Minimizzando l'Errore!

Un Salto Concettuale: Dai Numeri alla Geometria

I Dati Come Punti nello Spazio

Intuizione Chiave: Ogni dato può essere rappresentato come un **punto** in uno spazio multidimensionale

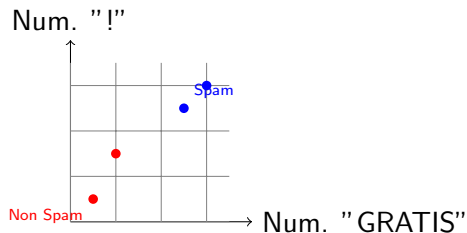
Esempio: Email Spam (2 caratteristiche)

- Email A: 3 "GRATIS", 5 "!" → Punto (3, 5)
- Email B: 0 "GRATIS", 1 "!" → Punto (0, 1)
- Email C: 8 "GRATIS", 12 "!" → Punto (8, 12)

I Dati Come Punti nello Spazio

Intuizione Chiave: Ogni dato può essere rappresentato come un **punto** in uno spazio multidimensionale

Spazio 2D



Osservazione

- Email non-spam tendono a raggrupparsi in una zona
- Email spam si raggruppano in un'altra zona
- Esiste una **separazione naturale!**

Il principio si estende a qualsiasi numero di caratteristiche!

Email con 4 Caratteristiche

Ogni email diventa un punto in uno spazio a 4 dimensioni:

$$\text{Email} = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$$

dove:

- x_1 = Numero "GRATIS"
- x_2 = Numero "!"
- x_3 = Link sospetti (0/1)
- x_4 = Lunghezza email

Il principio si estende a qualsiasi numero di caratteristiche!

Esempi Concreti

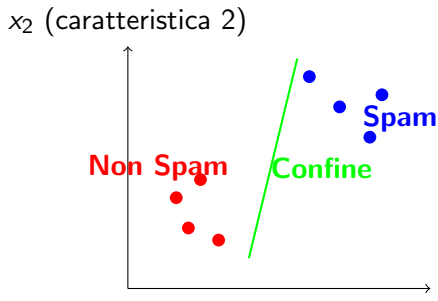
- Spam: (2, 5, 1, 200)
- Non Spam: (0, 1, 0, 150)
- Spam: (4, 8, 1, 80)

Idea Fondamentale

Anche se non possiamo **visualizzare** 4 dimensioni, il computer può **lavorare** in questo spazio!

Il Problema Diventa Geometrico

Classificazione = Separazione geometrica nello spazio delle caratteristiche



Il Problema Diventa Geometrico

Classificazione = Separazione geometrica nello spazio delle caratteristiche

Obiettivo del Machine Learning

Trovare il **confine ottimale** che separa al meglio le due classi

- **Confine lineare:** una retta (2D), un piano (3D), un iperpiano (N-D)
- **Equazione del confine:** $w_1x_1 + w_2x_2 + b = 0$

Come usiamo il confine per classificare nuovi punti?

Equazione del Confine

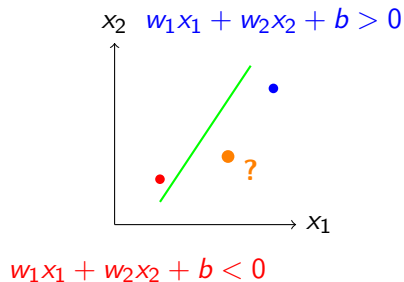
$$w_1x_1 + w_2x_2 + b = 0$$

Regola di Classificazione

Per un nuovo punto (x_1, x_2) :

- Se $w_1x_1 + w_2x_2 + b > 0 \rightarrow$ **Classe A (Spam)**
- Se $w_1x_1 + w_2x_2 + b < 0 \rightarrow$ **Classe B (Non Spam)**

Come usiamo il confine per classificare nuovi punti?



Esempio Numerico Se il confine è:

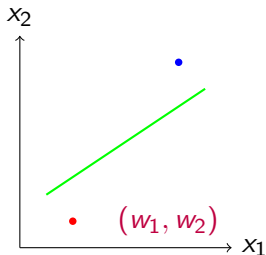
$$2x_1 + 3x_2 - 5 = 0$$

Per il punto arancione (2.5, 1.5):

$$2(2.5) + 3(1.5) - 5 = 4.5 > 0$$

Quindi: **Spam!**

I pesi w_1, w_2, b definiscono completamente il confine di separazione!



Collegamento Cruciale

Il **training** del modello consiste nel trovare i pesi (w_1, w_2, b) che definiscono il confine ottimale!

Processo:

- 1 Prova diversi confini (diversi pesi)
- 2 Misura quanti punti classifica male
- 3 Aggiusta i pesi per ridurre gli errori

Sintesi: Una Nuova Prospettiva

Abbiamo trasformato il problema!

Da: "Come classificare email spam?"

A: "Come trovare il miglior confine geometrico in uno spazio multidimensionale?"

Vantaggi di questa visione geometrica:

- **Intuizione:** Capiamo cosa fa veramente l'algoritmo
- **Generalizzazione:** Funziona per qualsiasi numero di dimensioni
- **Ottimizzazione:** Possiamo usare metodi geometrici potenti

Concetto Universale

Machine Learning = Trovare pattern e confini negli spazi multidimensionali!