

Il Machine Learning Spiegato Semplice

Come i computer imparano dai dati

Prof. Massimo

Tutti i diritti riservati

14 gennaio 2026

Indice

1	Cos'è il Machine Learning?	3
1.1	Una definizione semplice	3
1.2	Le tre famiglie del Machine Learning	3
2	Regressione Lineare: Prevedere il Futuro con una Retta	4
2.1	L'idea di base	4
2.2	Come funziona?	4
2.3	Esempio visivo	4
2.4	Quanto è precisa la previsione?	5
3	Regressione Logistica: Sì o No?	6
3.1	Quando serve?	6
3.2	La curva a S	6
3.3	Come si usa?	6
4	K-Nearest Neighbors: Chiedi ai Vicini!	7
4.1	Il principio	7
4.2	Come funziona in 3 passi	7
4.3	Esempio visivo	7
4.4	Scegliere K	7
5	Alberi Decisionali: Un Gioco di Domande	9
5.1	Come funziona?	9
5.2	Visualizzazione	9
5.3	Quando usare gli alberi?	9
5.4	Vantaggi e svantaggi	9
6	Support Vector Machine: Trova il Confine Migliore	11
6.1	L'idea principale	11
6.2	Cosa vuol dire "migliore"?	11
6.3	Quando serve?	11

7	K-Means: Fare Gruppi	13
7.1	Il problema	13
7.2	Come funziona?	13
7.3	Esempio passo-passo	13
7.4	Quanti gruppi?	13
8	Reti Neurali: Il Cervello del Computer	15
8.1	Cos'è una rete neurale?	15
8.2	Un neurone artificiale	15
8.3	Come impara?	15
8.4	Reti a più livelli	15
8.5	Dove si usano?	16
9	Random Forest: La Saggezza della Folla	17
9.1	Il principio	17
9.2	Come funziona?	17
9.3	Perché è meglio di un solo albero?	17
10	Confronto e Scelta dell'Algoritmo	19
10.1	Quale algoritmo usare?	19
10.2	Domande per scegliere	19
10.3	Schema decisionale	20
11	Concetti Importanti	21
11.1	Overfitting e Underfitting	21
11.2	Training e Test	21
11.3	Accuratezza	22
12	Il Futuro del Machine Learning	23
12.1	Dove stiamo andando?	23
12.2	Sfide e questioni etiche	23
12.3	Come prepararsi?	23
13	Riepilogo Finale	24
13.1	Cosa abbiamo imparato	24
13.2	Tabella riassuntiva	24
13.3	Prossimi passi	24

1 Cos'è il Machine Learning?

1.1 Una definizione semplice

Immagina di insegnare a un bambino a riconoscere i frutti. Gli mostri tante mele, arance e banane. Dopo un po', il bambino impara a riconoscerle da solo. Il **Machine Learning** (ML) funziona in modo simile: invece di programmare un computer per ogni singola situazione, gli mostriamo tanti esempi e lui impara da solo!

Nota Importante

Machine Learning significa "apprendimento automatico". È come dare al computer la capacità di imparare dall'esperienza, proprio come facciamo noi!

1.2 Le tre famiglie del Machine Learning

Esistono tre modi principali in cui un computer può imparare:

- **Apprendimento Supervisionato** (con un "maestro"): come quando studi con un libro di testo che ha tutte le risposte. Il computer impara da esempi già classificati. *Esempi*: riconoscere se una email è *spam* o *non spam* partendo da messaggi già etichettati; prevedere il prezzo di una casa conoscendo i prezzi di case simili; riconoscere cifre scritte a mano avendo a disposizione immagini già associate al numero corretto.
- **Apprendimento Non Supervisionato** (da solo): come quando organizzi i tuoi vestiti per colore senza che nessuno ti dica come fare. Il computer trova pattern o somiglianze nei dati senza conoscere in anticipo le risposte corrette. *Esempi*: raggruppare clienti con comportamenti di acquisto simili; individuare gruppi di studenti con risultati affini; analizzare dati di navigazione per scoprire profili di utenti senza etichette predefinite.
- **Apprendimento per Rinforzo** (per tentativi): come quando impari un videogioco provando e riprovando. Il computer impara dalle conseguenze delle sue azioni, ricevendo ricompense o penalità. *Esempi*: un agente che impara a giocare a scacchi o a un videogioco; un robot che impara a muoversi evitando ostacoli; un sistema che ottimizza i tempi di un semaforo in base al traffico.

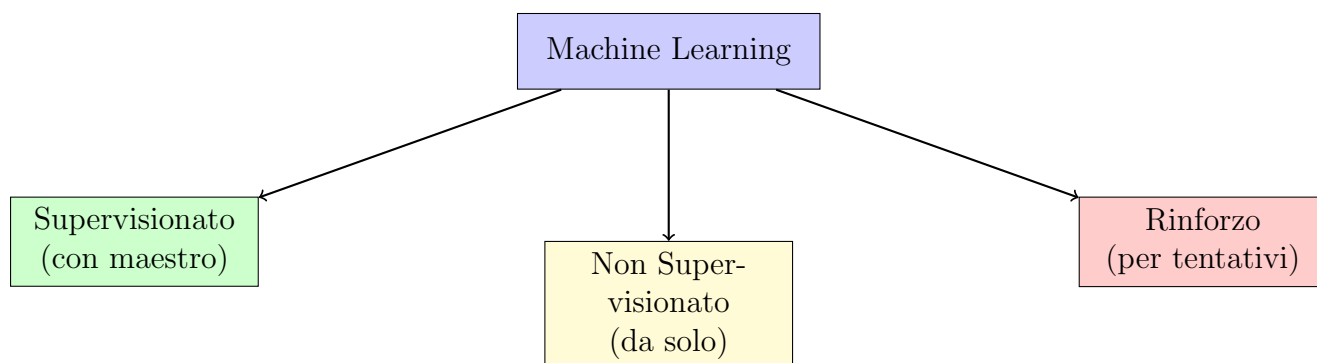


Figura 1: I tre modi di imparare del Machine Learning

2 Regressione Lineare: Prevedere il Futuro con una Retta

2.1 L'idea di base

Se studi prendi voti migliori (di solito..), più ti alleni, più corri velocemente. La **regressione lineare** trova proprio queste relazioni tra causa ed effetto.

Esempio Pratico

Immagina di voler prevedere il prezzo di una casa in base alla sua grandezza. Se raccogli dati di case vendute, noterai che case più grandi costano di più. La regressione lineare traccia una retta che rappresenta questa relazione!

2.2 Come funziona?

L'obiettivo è trovare la "retta migliore" che passa vicino a tutti i punti. Matematicamente si scrive così:

$$\text{Prezzo} = a \times \text{Grandezza} + b \quad (1)$$

Dove:

- **a** indica quanto aumenta il prezzo per ogni metro quadro in più
- **b** è il prezzo base (quando la casa ha zero metri quadri - ovviamente è teorico!)

2.3 Esempio visivo

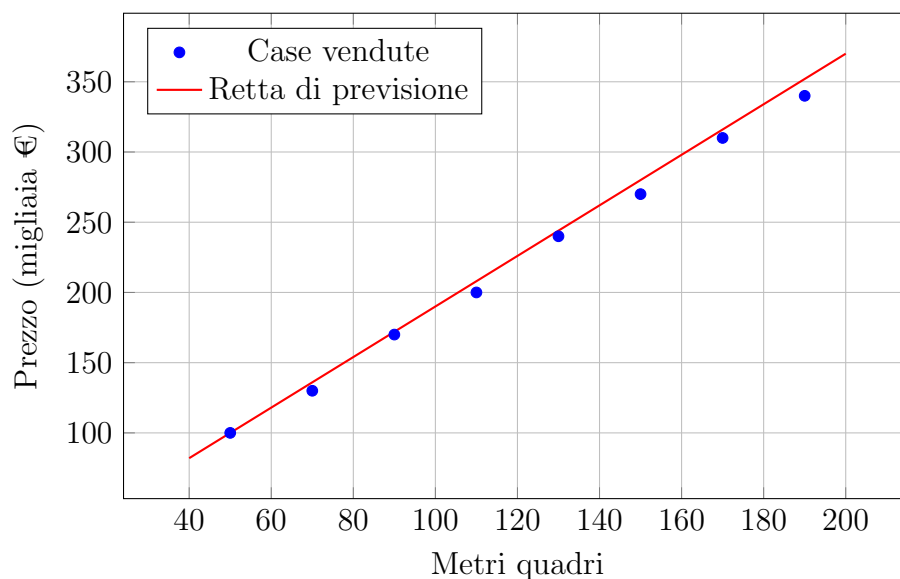


Figura 2: Previsione del prezzo delle case

Nota Importante

La regressione lineare è utile quando c'è una relazione "proporzionale": più di questo, più di quello!

2.4 Quanto è precisa la previsione?

Per capire se la nostra retta è buona, calcoliamo quanto sbagliamo in media. Si chiama **errore quadratico medio**:

$$\text{Errore Medio} = \frac{\text{Somma di tutti gli errori}^2}{\text{Numero di case}} \quad (2)$$

L'obiettivo è avere l'errore più piccolo possibile!

3 Regressione Logistica: Sì o No?

3.1 Quando serve?

A volte non vogliamo un numero come risultato, ma una risposta tipo SÌ/NO:

- Questa email è spam? SÌ o NO
- Il paziente ha la malattia? SÌ o NO
- Lo studente passerà l'esame? SÌ o NO

Esempio Pratico

Vogliamo capire se uno studente passerà l'esame in base alle ore di studio che dedica alla preparazione. Con 10 ore probabilmente sì, con 1 ora probabilmente no. Ma quanto è "probabile"? La regressione logistica ci dice la probabilità!

3.2 La curva a S

Per ottenere un valore binario come risposta (quindi sì o no), invece di una retta, usiamo una curva a forma di S chiamata **sigmoide**. Questa curva va sempre da 0 (0% di probabilità) a 1 (100% di probabilità).

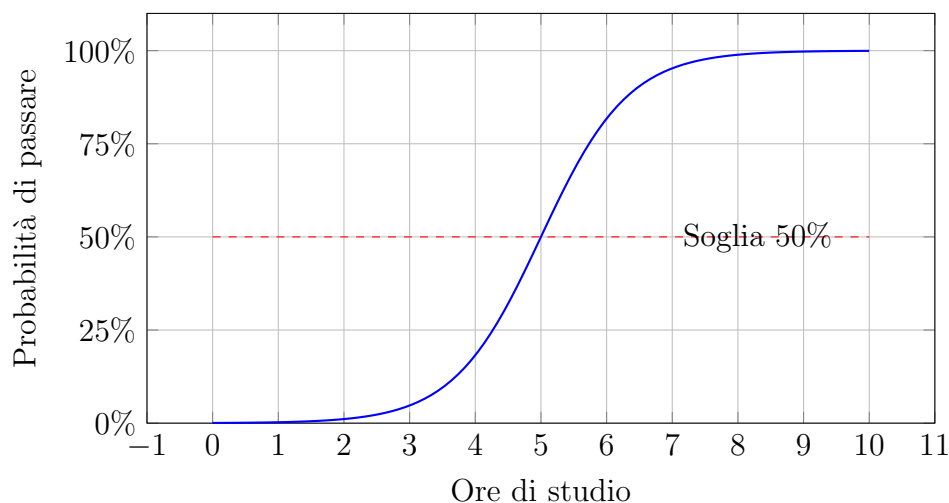


Figura 3: Probabilità di passare l'esame in base alle ore di studio

3.3 Come si usa?

Se la probabilità è maggiore del 50%, diciamo SÌ. Altrimenti diciamo NO. Semplice!

Nota Importante

La regressione logistica è perfetta per decisioni binarie: spam/non spam, malato/sano, passa/non passa!

4 K-Nearest Neighbors: Chiedi ai Vicini!

4.1 Il principio

Immagina di essere nuovo in città e vuoi sapere qual è il miglior ristorante. Cosa fai? Chiedi alle persone che abitano vicino a te! L'algoritmo **KNN** fa la stessa cosa con i dati.

Esempio Pratico

Vogliamo classificare un frutto. Guardiamo i 3 frutti più simili (più "vicini") che già conosciamo. Se 2 su 3 sono mele, probabilmente anche il nostro frutto è una mela!

4.2 Come funziona in 3 passi

1. **Misura le distanze:** calcola quanto è lontano il nuovo punto da tutti gli altri
2. **Trova i K più vicini:** seleziona i K punti più vicini (K è un numero che scegliamo noi, tipo 3 o 5)
3. **Vota:** guarda che categoria è più frequente tra i vicini

4.3 Esempio visivo

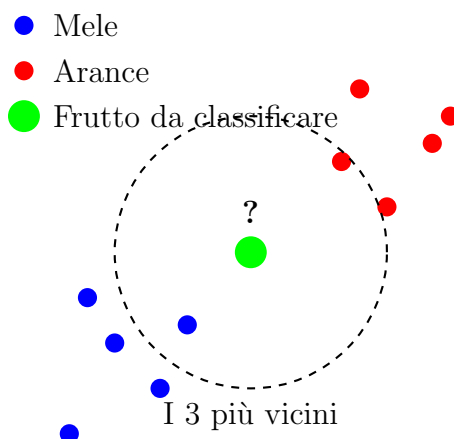


Figura 4: Nel cerchio ci sono 2 mele e 1 arancia, quindi il frutto è probabilmente una mela!

4.4 Scegliere K

- **K piccolo (tipo 1 o 3):** molto sensibile, può sbagliare se c'è un dato "strano"
- **K grande (tipo 10 o 20):** più stabile, ma potrebbe confondere i gruppi
- **Il valore giusto:** si trova provando! Di solito $K=5$ funziona bene

Nota Importante

KNN è come chiedere consiglio: se chiedi a una sola persona ($K=1$) potresti essere sfortunato, ma se chiedi a troppe persone (K grande) rischi di avere pareri troppo diversi!

5 Alberi Decisionali: Un Gioco di Domande

5.1 Come funziona?

Pensiamo al gioco delle venti domande: un giocatore pensa a qualcosa (una cosa, un concetto), e l'altro giocatore deve indovinarla, e per farlo può fare delle domande, alla quale il primo giocatore risponde soltanto con dei sì o dei no. Gli **alberi decisionali** funzionano più o meno così.

Esempio Pratico

Immagina di dover decidere se portare l'ombrello:

- Piove? → SÌ → PORTA L'OMBRELLO
- Piove? → NO → È nuvoloso? → SÌ → PORTA L'OMBRELLO
- Piove? → NO → È nuvoloso? → NO → NON SERVE

5.2 Visualizzazione

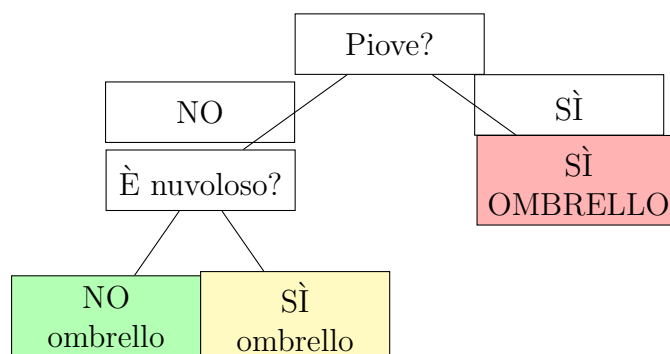


Figura 5: Albero decisionale per decidere se portare l'ombrello

5.3 Quando usare gli alberi?

Gli alberi decisionali sono perfetti quando:

- Si vuol capire PERCHÉ il computer ha fatto una determinata scelta
- Habbiano dati con categorie (tipo: piove/non piove, grande/piccolo)
- Vogliamo poter spiegare la decisione ad altri

5.4 Vantaggi e svantaggi

Vantaggi:

- Facili da capire.
- Si possono disegnare
- Funzionano con qualsiasi tipo di dato

Svantaggi:

- Possono diventare troppo complicati
- Se cambiano un po' i dati, l'albero può cambiare molto

6 Support Vector Machine: Trova il Confine Migliore

6.1 L'idea principale

Immaginiamo di avere caramelle rosse e blu sparse su un tavolo. Vogliamo dividere il tavolo con un nastro in modo che da una parte ci siano solo le rosse e dall'altra solo le blu. La **SVM** trova il nastro "migliore" per effettuare la suddivisione delle palline in due insiemi.

Le Support Vector Machines, abbreviate SVM, sono algoritmi che identificano l'iperpiano ottimale di separazione tra due classi. L'iperpiano ottimale è quello che massimizza il margine, definito come la distanza minima tra l'iperpiano e i punti più vicini di ciascuna classe. Questi punti critici sono chiamati vettori di supporto.

Esempio Pratico

In una scuola, gli studenti del primo anno stanno da una parte del cortile e quelli del quinto dall'altra. SVM trova la linea che li divide meglio, lasciando il massimo spazio possibile tra i due gruppi.

6.2 Cosa vuol dire "migliore"?

Il confine migliore è quello che:

- Separa perfettamente i due gruppi
- È il più lontano possibile da entrambi i gruppi (massimo margine)

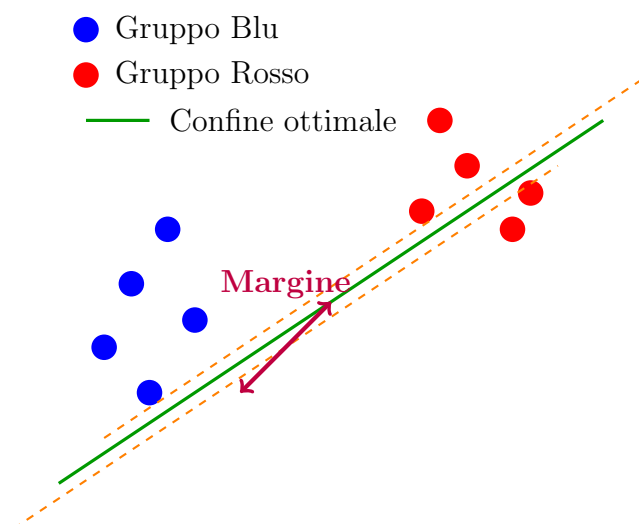


Figura 6: SVM trova il confine con il margine più grande

6.3 Quando serve?

SVM è ottimo per:

- Riconoscere volti nelle foto

- Classificare testi
- Diagnosticare malattie
- Qualsiasi problema con due categorie ben separate

Nota Importante

Si può pensare a SVM come a un arbitro che cerca di stare il più lontano possibile da entrambe le squadre per essere imparziale.

7 K-Means: Fare Gruppi

7.1 Il problema

Immagina di avere 100 studenti e vuoi dividerli in 5 gruppi per attività, in modo che ogni gruppo abbia studenti con interessi simili. Come fai? Usa **K-Means**!

Esempio Pratico

Un negozio online vuole raggruppare i clienti:

- Gruppo 1: giovani che comprano vestiti
- Gruppo 2: adulti che comprano per la casa
- Gruppo 3: anziani che comprano libri

K-Means trova questi gruppi automaticamente!

7.2 Come funziona?

È come un gioco in 3 mosse, che si ripete finché i gruppi non cambiano più:

1. **Scegli K centri** a caso (K è il numero di gruppi che vuoi)
2. **Assegna ogni persona** al centro più vicino
3. **Ricalcola i centri** come punto medio di ogni gruppo
4. **Ripeti** i passi 2 e 3 fino a quando i centri non si spostano più

7.3 Esempio passo-passo

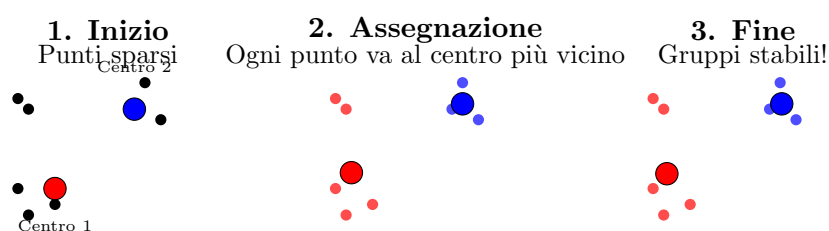


Figura 7: Il processo di K-Means in 3 passi

7.4 Quanti gruppi?

Come scegliere K (il numero di gruppi)? Usa il **metodo del gomito**:

- Prova con $K = 1, 2, 3, 4, 5...$
- Per ogni K, misura quanto sono "compatti" i gruppi
- Scegli il K dove migliora poco aumentare ancora

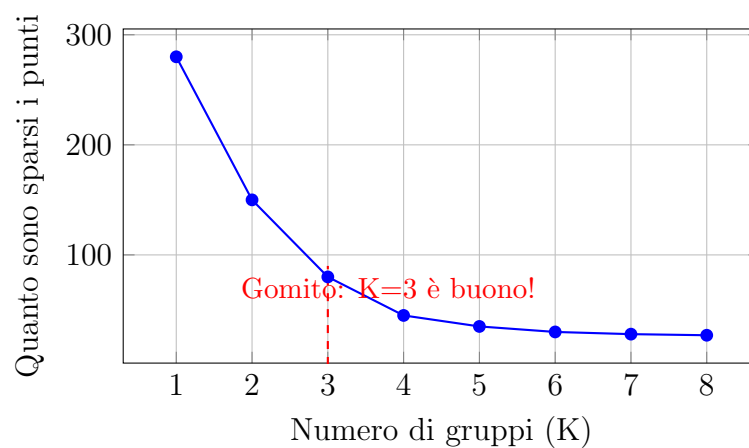


Figura 8: Dopo $K=3$, migliorare diventa difficile

Nota Importante

K-Means è come organizzare i tuoi amici in gruppi per interessi: all'inizio sono sparsi, ma poi trovi i gruppi naturali!

8 Reti Neurali: Il Cervello del Computer

8.1 Cos'è una rete neurale?

Le **reti neurali** sono ispirate al nostro cervello! Come i neuroni nel cervello si passano segnali elettrici, i neuroni artificiali si passano numeri.

Esempio Pratico

Riconoscere un gatto in una foto:

- **Input:** i pixel della foto
- **Neuroni nascosti:** cercano orecchie, baffi, occhi
- **Output:** "È un gatto!" oppure "Non è un gatto!"

8.2 Un neurone artificiale

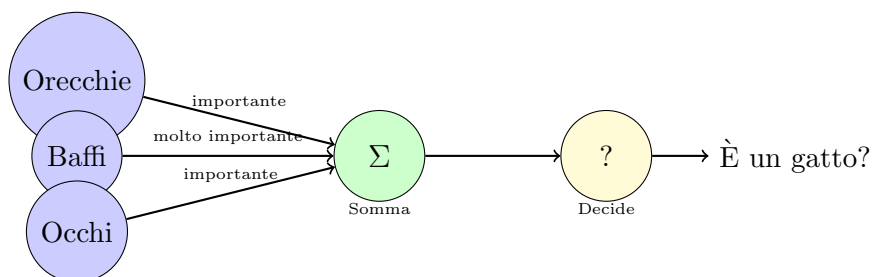


Figura 9: Un neurone che decide se c'è un gatto

8.3 Come impara?

La rete neurale impara per tentativi:

1. **Prova:** fa una previsione
2. **Sbaglia:** confronta con la risposta giusta
3. **Corregge:** modifica i pesi per sbagliare meno
4. **Ripete:** migliaia di volte!

Nota Importante

È come imparare ad andare in bici: all'inizio cadi, ma dopo tanti tentativi diventi bravissimo!

8.4 Reti a più livelli

Le reti vere hanno tanti livelli (layer):

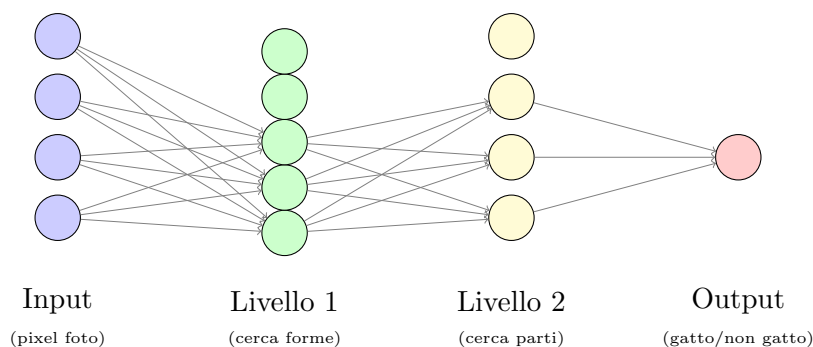


Figura 10: Una rete neurale con più livelli

- **Livello 1:** cerca linee e curve semplici
- **Livello 2:** combina linee per trovare orecchie, occhi
- **Output:** decide se è un gatto!

8.5 Dove si usano?

Le reti neurali sono ovunque:

- Riconoscimento vocale (Siri, Alexa)
- Traduzione automatica
- Riconoscimento facciale
- Guida autonoma
- Filtri Instagram

9 Random Forest: La Saggezza della Folla

9.1 Il principio

Se devi prendere una decisione importante, cosa fai? Chiedi il parere di tanti amici e poi scegli quello più votato! **Random Forest** fa esattamente questo con tanti alberi decisionali.

Esempio Pratico

Vuoi sapere se domani piovgerà. Chiedi a 100 esperti meteo (100 alberi decisionali). Se 70 dicono SÌ e 30 dicono NO, probabilmente piovgerà!

9.2 Come funziona?

1. **Crea tanti dataset:** prendi campioni casuali dai dati originali
2. **Costruisci tanti alberi:** uno per ogni dataset (tipo 100 alberi)
3. **Chiedi a tutti:** ogni albero dà la sua risposta
4. **Vota:** la risposta più frequente vince!

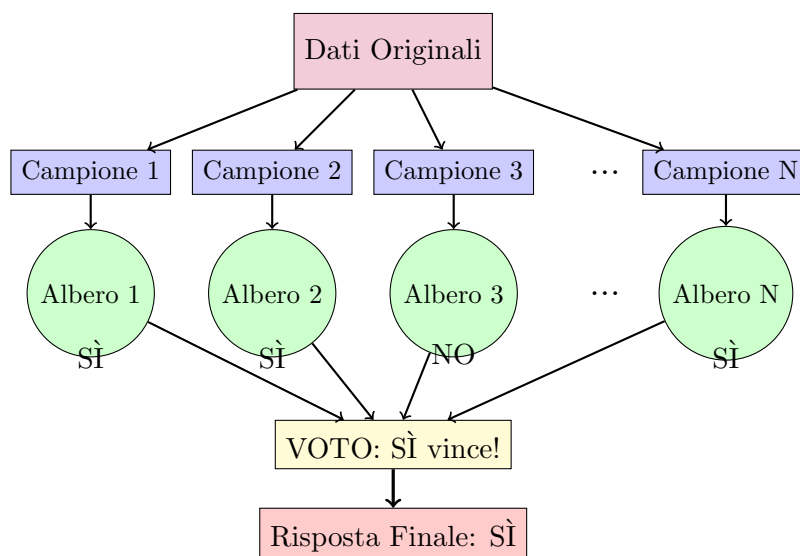


Figura 11: Random Forest: tanti alberi votano insieme

9.3 Perché è meglio di un solo albero?

Un solo albero può:

- Essere troppo sicuro di sé e sbagliare
- Imparare troppo bene i dati di training (overfitting)
- Essere influenzato da dati strani

Con tanti alberi:

- Gli errori si compensano
- La decisione è più affidabile
- Funziona meglio su nuovi dati

Nota Importante

È come un consiglio di classe: meglio decidere insieme che da soli!

10 Confronto e Scelta dell'Algoritmo

10.1 Quale algoritmo usare?

Algoritmo	Quando usarlo	Esempi pratici
Regressione Lineare	Prevedere numeri con relazioni lineari	Prezzo case, vendite future, temperature
Regressione Logistica	Rispondere SÌ/NO	Email spam, diagnosi malattie, pass/fail esami
KNN	Classificare cose simili	Raccomandare film, riconoscere scrittura
Alberi Decisionali	Decisioni spiegabili	Approvare prestiti, diagnostica medica
SVM	Separare gruppi netti	Riconoscere volti, classificare immagini
K-Means	Trovare gruppi naturali	Segmentare clienti, organizzare foto
Reti Neurali	Problemi complessi	Riconoscimento vocale, traduzione, immagini
Random Forest	Previsioni affidabili	Prevedere crediti, diagnosi complesse

Tabella 1: Guida rapida alla scelta dell'algoritmo

10.2 Domande per scegliere

Poniti queste domande:

1. Che risposta voglio?

- Un numero? → Regressione Lineare
- SÌ o NO? → Regressione Logistica, Alberi, SVM
- Trovare gruppi? → K-Means

2. Ho bisogno di spiegare perché?

- Sì → Alberi Decisionali, Regressione Lineare
- No → Reti Neurali, SVM

3. Quanti dati ho?

- Pochi (centinaia) → KNN, Alberi semplici
- Tanti (migliaia) → Random Forest, Reti Neurali

4. Il problema è complesso?

- Semplice → Regressione, KNN
- Complesso → Reti Neurali, Random Forest

10.3 Schema decisionale

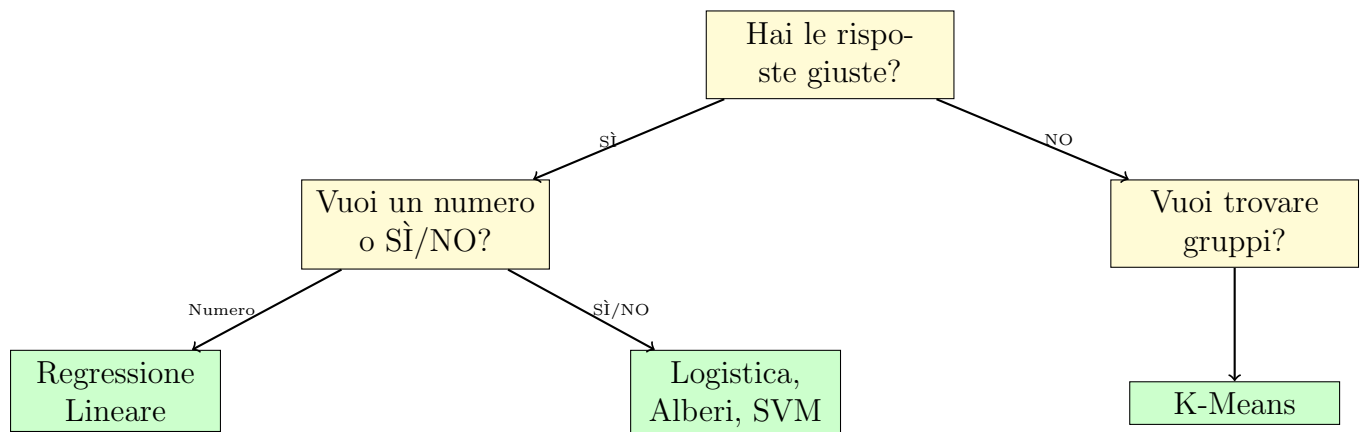


Figura 12: Come scegliere l'algoritmo giusto

11 Concetti Importanti

11.1 Overfitting e Underfitting

Overfitting (impara troppo bene):

- È come studiare SOLO i compiti degli anni scorsi
- Il modello funziona perfettamente sui dati di training
- Ma sbaglia su dati nuovi!

Underfitting (impara troppo poco):

- È come studiare troppo poco
- Il modello non funziona nemmeno sui dati di training
- Troppo semplice per il problema

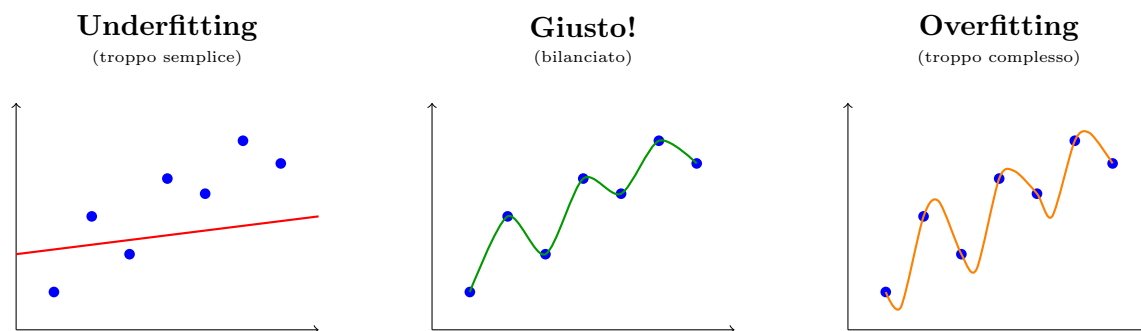


Figura 13: I tre livelli di apprendimento

11.2 Training e Test

Per evitare overfitting, dividiamo i dati:

- **Dati di Training (70-80%)**: per imparare
- **Dati di Test (20-30%)**: per verificare se ha imparato davvero

Esempio Pratico

È come preparare un esame:

- Training = studiare sui libri
- Test = fare l'esame vero

Se riesci bene solo sui libri ma non all'esame, hai fatto overfitting!

11.3 Accuratezza

Come misuriamo se il modello è bravo?

$$\text{Accuratezza} = \frac{\text{Previsioni corrette}}{\text{Totale previsioni}} \times 100\% \quad (3)$$

Esempio Pratico

Su 100 email:

- 90 classificate correttamente
- 10 sbagliate
- Accuratezza = 90%

Nota Importante

100% di accuratezza è quasi sempre sospetto - probabilmente c'è overfitting!

12 Il Futuro del Machine Learning

12.1 Dove stiamo andando?

Il Machine Learning sta cambiando il mondo! Ecco alcune direzioni:

- **Auto che si guidano da sole:** usando reti neurali e sensori
- **Diagnosi mediche:** aiutare i dottori a trovare malattie prima
- **Assistenti vocali:** sempre più intelligenti (Siri, Alexa, Google)
- **Arte e creatività:** IA che dipinge, compone musica, scrive
- **Traduzione in tempo reale:** capire e parlare tutte le lingue
- **Cambiamento climatico:** prevedere e combattere i problemi ambientali

12.2 Sfide e questioni etiche

Con grandi poteri vengono grandi responsabilità:

- **Privacy:** come proteggere i dati personali?
- **Bias:** l'IA può avere pregiudizi se i dati lo hanno
- **Lavoro:** alcuni lavori potrebbero sparire, altri nascere
- **Trasparenza:** capire PERCHÉ l'IA ha deciso qualcosa
- **Sicurezza:** prevenire usi dannosi della tecnologia

12.3 Come prepararsi?

Se vuoi lavorare con il Machine Learning:

1. **Matematica:** algebra, statistiche, probabilità
2. **Programmazione:** Python è il linguaggio più usato
3. **Curiosità:** sperimenta con progetti personali
4. **Etica:** pensa sempre all'impatto delle tue creazioni
5. **Continua a imparare:** il campo cambia rapidamente!

Nota Importante

Il Machine Learning non è magia - è matematica, logica e tanto lavoro. Ma i risultati possono sembrare magici!

13 Riepilogo Finale

13.1 Cosa abbiamo imparato

Punti Chiave

1. Il Machine Learning permette ai computer di **imparare dai dati**
2. Esistono **diversi algoritmi** per problemi diversi
3. La **scelta dell'algoritmo** dipende dal tipo di problema
4. Bisogna evitare **overfitting** e **underfitting**
5. Il ML sta **cambiando il mondo** in tanti modi
6. Serve **responsabilità** nell'usare queste tecnologie

13.2 Tabella riassuntiva

Algoritmo	Difficoltà	Pro	Contro
Regressione Lineare	*	Semplice, veloce	Solo per relazioni lineari
Regressione Logistica	*	Buono per SÌ/NO	Solo 2 classi
KNN	**	Intuitivo	Lento con tanti dati
Alberi Decisionali	**	Spiegabile	Tende all'overfitting
SVM	***	Preciso	Difficile da capire
K-Means	**	Trova gruppi	Serve scegliere K
Reti Neurali	***	Potentissime	Serve tanti dati
Random Forest	**	Robusto	Più lento

Tabella 2: * = Bassa, ** = Media, *** = Alta difficoltà

13.3 Prossimi passi

Se vuoi approfondire:

- Prova a programmare in Python con librerie come scikit-learn
- Segui tutorial online (YouTube, Coursera, Khan Academy)
- Partecipa a competizioni su Kaggle
- Costruisci un tuo progetto personale
- Resta curioso e continua a sperimentare!

Il Machine Learning è il futuro - e il futuro inizia adesso!

Risorse Utili

Siti web e piattaforme:

- Khan Academy - Matematica e Informatica
- Coursera - Corsi gratuiti di ML
- YouTube - 3Blue1Brown (ottime animazioni su ML)
- Kaggle - Competizioni e dataset
- Scikit-learn - Documentazione Python

Libri consigliati (per iniziare):

- "Python Crash Course" - Eric Matthes
- "Make Your Own Neural Network" - Tariq Rashid
- "Machine Learning for Kids" - Dale Lane