UNIVERSITÀ DEGLI STUDI DI TERAMO

Big Data Analytics

Modelli classici: Machine Learning vecchia scuola

Prof. ssa Romina Eramo
Università degli Studi di Teramo
Dipartimento di Scienze della Comunicazione
reramo@unite.it

Outline

Esploriamo tre modelli classici:

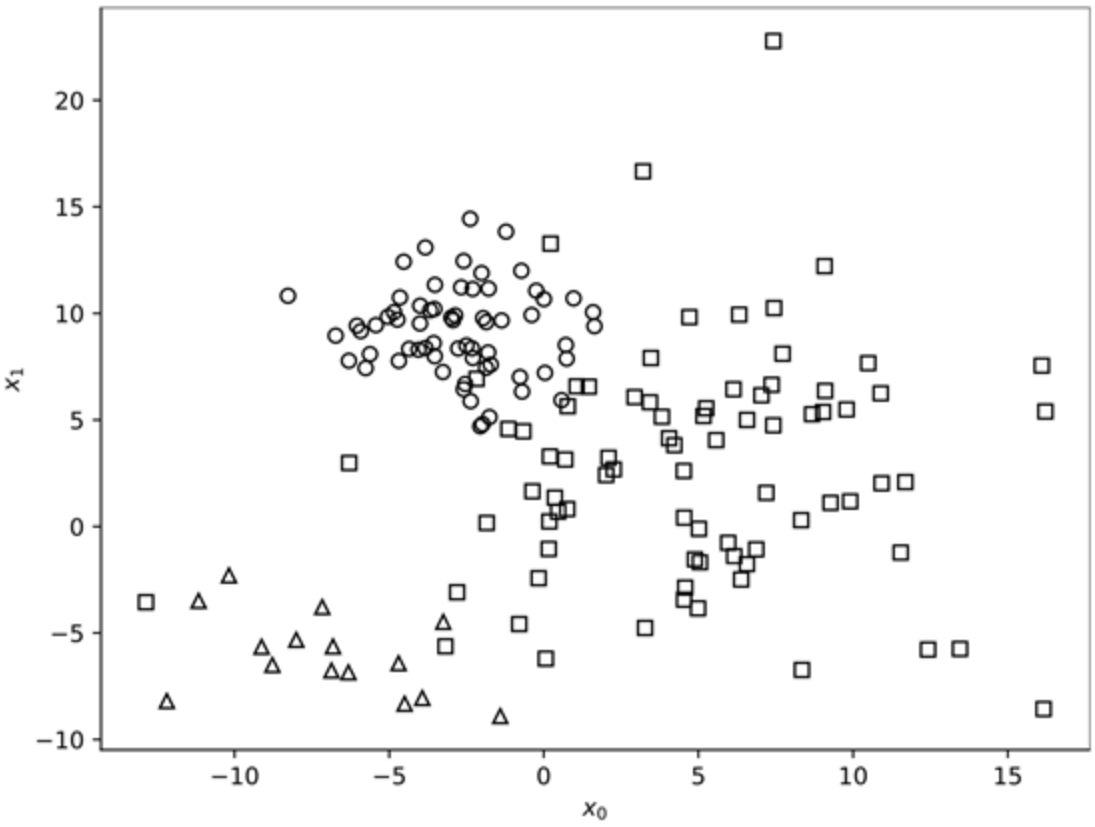
- Nearest Neighbors
- Random Forests
- Support Vector Machines

La comprensione di questi ci preparerà per le Reti Neurali.

Nearest Neighbors

- Campioni di training per un set di dati inventato con due *feature* (X0 e X1 e tre *classi* (cerchi, quadrati e triangoli)
- Ogni forma nella figura rappresenta un campione del set di training
- I dati di addestramento sono il modello!

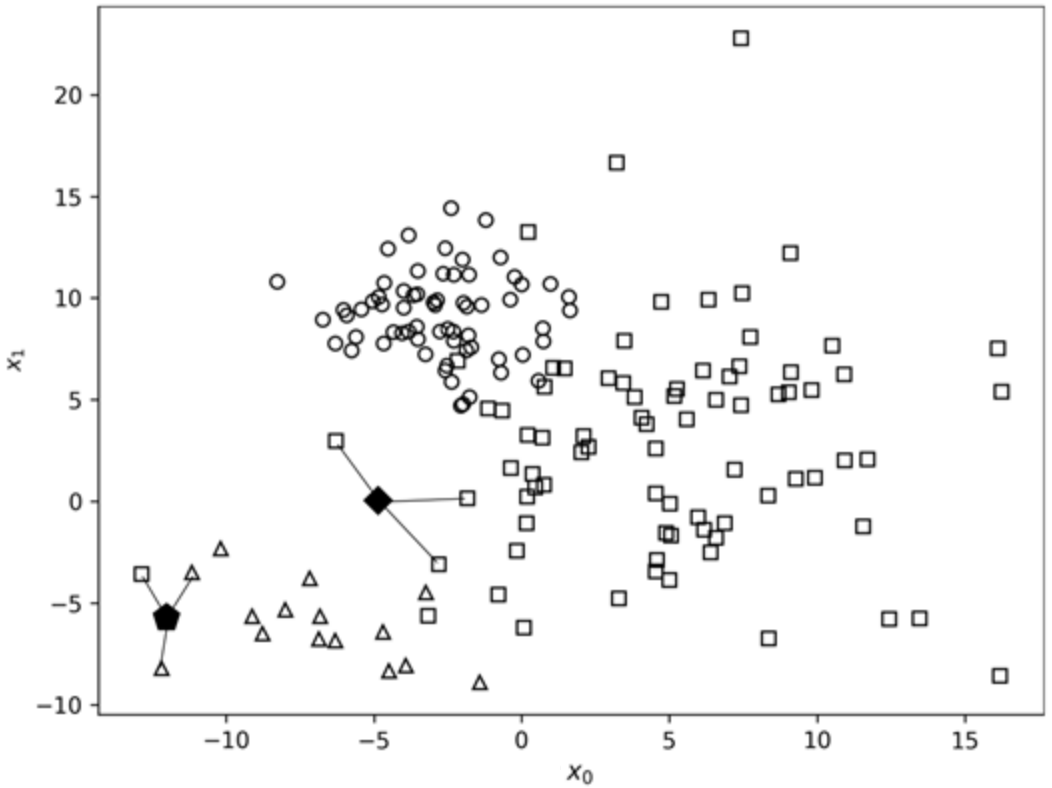
Per assegnare un'etichetta di classe a un nuovo input sconosciuto, trova il campione di addestramento più vicino al campione sconosciuto e restituisci l'etichetta di quel campione



A made-up training set with three classes and two features

Nearest neighbors: Classificare campioni sconosciuti

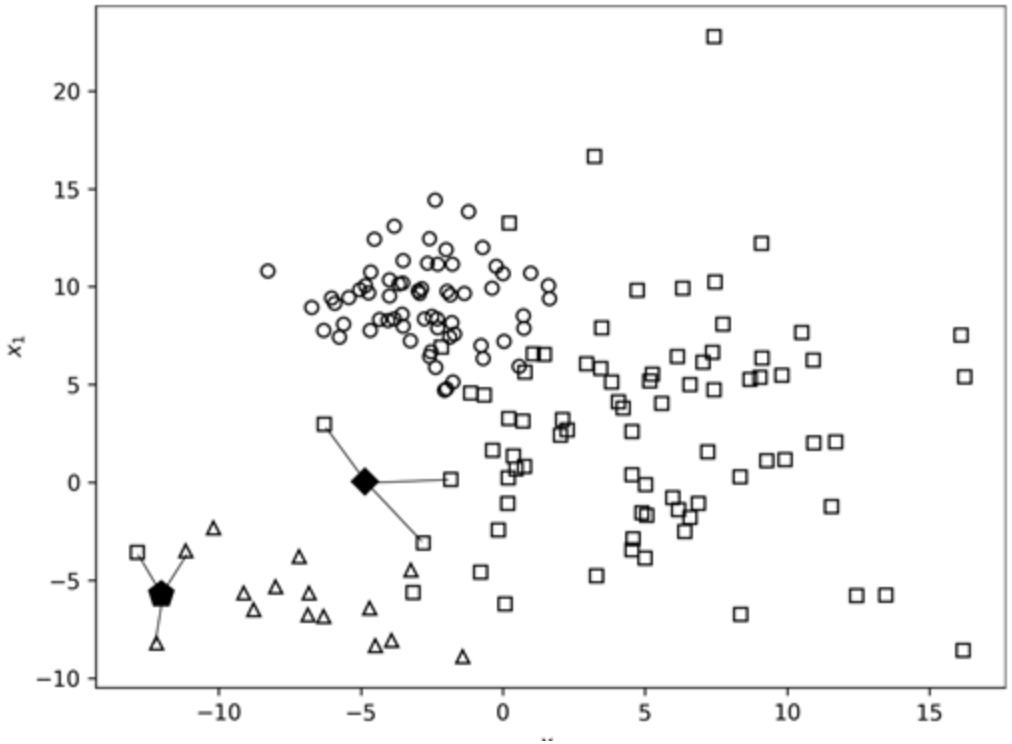
- Figura: campioni di training, insieme a due campioni sconosciuti: il ***diamante*** e il ***pentagono***.
- Vogliamo assegnare questi campioni a una delle tre classi: ***cerchio***, ***quadrato*** o ***triangolo***.
- L'approccio del nearest neighbor dice di localizzare il campione di training più vicino a ciascun campione sconosciuto.
  - per il diamante, è il quadrato in alto a sinistra
  - per il pentagono, sembra essere il triangolo in alto a destra



- Pertanto, un classificatore del Nearest Neighbor assegna la classe quadrato al diamante e la classe triangolo al pentagono.

Nearest neighbors: Classificare campioni sconosciuti

- Il modello individua i k campioni di addestramento più vicini al campione sconosciuto. k è spesso un numero come 3, 5 o 7 (può essere qualsiasi numero)
- Ad esempio, se il modello sta prendendo in considerazione i 5 nearest neighbors a un campione sconosciuto e due sono di classe 0 mentre altri due sono di classe 3, allora assegna l'etichetta scegliendo casualmente tra 0 e 3; in media, la scelta è corretta nel 50% dei casi

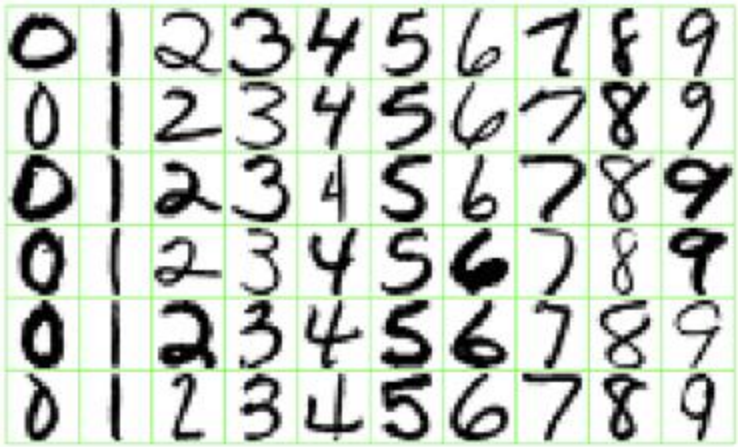


- Nella figura le linee collegano i campioni sconosciuti ai tre campioni di training più vicini. Questi sono i campioni da usare se **k è 3**. In questo caso, il classificatore assegnerebbe di nuovo la classe quadrato al diamante, perché tutti e tre i campioni di training più vicini sono quadrati. Per il pentagono, due dei tre vicini più vicini sono triangoli e uno è un quadrato, quindi assegnerebbe di nuovo la classe triangolo al pentagono.

ESEMPIO

Riconoscimento di cifre scritte a mano (MNIST)

- Input: immagine 28 × 28 in scala di grigi
- Output: la cifra decimale (0–9) rappresentata dall'immagine



- Come misuriamo la distanza tra immagini?
  - Dimensioni 28 × 28 (784 pixel totali)
  - Ogni pixel è in scala di grigi: 0–255
- Un vettore 784-dimensionale per ogni immagine

- Dataset MNIST: 60,000 (training set) + 10,000 (test set) immagini etichettate

ESEMPIO

Riconoscimento di cifre scritte a mano (MNIST)

- Cambio di dimensione del campione di training
- Prestazioni del modello al variare del numero di esempi di addestramento.

Training set size	Accuracy (%)
60,000	97
6,000	94
600	86
60	66

- L'accuratezza (***Accuracy***) è la **percentuale di campioni di test che il modello ha classificato correttamente** assegnando l'etichetta di cifra corretta, da 0 a 9

Nearest neighbor è comunque migliore di di un'ipotesi casuale, possiamo dire che funziona abbastanza bene anche con pochi dati di addestramento…(?)

La maledizione della dimensionalità (curse of dimensionality)

- Si riferisce alla maggiore complessità che sorge man mano che cresce il numero di dimensioni o caratteristiche in un set di dati.
- Man mano che il numero di dimensioni aumenta, aumenta anche, esponenzialmente, il numero di campioni di addestramento necessari per rappresentare lo spazio.
- Molti modelli di ML si basano sul calcolo della distanza, come k-vicini più vicini (k-NN). Negli spazi ad alta dimensione, il concetto di "più vicino" diventa meno chiaro e tutti i punti convergono alla stessa distanza dal punto target.

ESEMPIO

Dataset CFAR-10 (classificare immagini reali)

- 50.000 piccole immagini a colori da 32×32 pixel
- 10 classi diverse, tra cui un mix di veicoli, come aeroplani, auto e camion, e animali, come cani, gatti e uccelli.
- La tabella mostra la classificazione CIFAt-10 con NN. La sua migliore precisione è di poco superiore al 35%, lontanamente vicina al 97% raggiunto con MNIST.

Training set size	Accuracy (%)
50,000	35.4
5,000	27.1
500	23.3
50	17.5

- Le immagini naturali sono molto più complesse di immagini semplici come le cifre MNIST, quindi dovremmo aspettarci che esistano in una varietà di dimensione superiore e di conseguenza siano più difficili da imparare a classificare.

Problemi dei NN

1. Uso lento perché dobbiamo calcolare la distanza tra il campione sconosciuto e ciascuno dei campioni del set di training.
2. Interpretazione dei loro vettori di input come un'unica entità senza parti.

In sintesi, i modelli del NN sono semplici da capire e banali da addestrare, ma lenti da usare e incapaci di comprendere esplicitamente la struttura nei loro input. Cambiamo marcia per contemplare la foresta e gli alberi.

Random Forests

- Gli **alberi decisionali** sono deterministici; gli algoritmi tradizionali degli alberi decisionali restituiscono lo stesso albero decisionale per lo stesso set di addestramento.
  - La **foresta** non è altro che lo stesso albero, ripetuto più e più volte.
  - La **casualità** produce una foresta di alberi unici, ognuno con i suoi punti di forza e di debolezza, ma collettivamente migliori di qualsiasi singolo albero.
- **Una random forest è una raccolta di alberi decisionali, ognuno diverso in modo casuale dagli altri.**
  - **La previsione della foresta è una combinazione delle previsioni dei suoi alberi.**
  - **Le foreste casuali sono una manifestazione della saggezza delle folle.**

Random Forests (2)

- Tre passaggi sono necessari per far crescere una random forest:
  - **Bagging** (chiamato anche bootstrapping)
  - **Random feature selection**
  - **Ensembling** (termine generale per la combinazione di molti classificatori).
- Tutti e tre i passaggi lavorano insieme per far crescere una foresta di alberi decisionali i cui output combinati producono un modello (si spera) con prestazioni migliori.
- L'*explainability* è il prezzo pagato.

Random Forests (3)

- Il bagging si riferisce alla costruzione di un nuovo set di dati dal set di dati corrente tramite campionamento casuale con sostituzione.
  - "con sostituzione" significa che potremmo selezionare un campione di training più di una volta o per niente.
- Esempio: Set di dati di punteggi dei test

  95, 88, 76, 81, 92, 70, 86, 87, 72

- Per valutare **la prestazione di una classe al test**, calcolare il punteggio medio:

  747/9 = 83

Random Forests (4)

» Con il bagging, selezioniamo i valori dalla raccolta di punteggi dei test **a caso**, senza preoccuparci se abbiamo già scelto questo punteggio particolare o se non lo abbiamo mai scelto.

1. 86, 87, 87, 76, 81, 81, 88, 70, 95
2. 87, 92, 76, 87, 87, 76, 87, 92, 92
3. 95, 70, 87, 92, 70, 92, 72, 70, 72
4. 88, 86, 87, 70, 81, 72, 86, 95, 70
5. 86, 86, 92, 86, 87, 86, 70, 81, 87
6. 76, 88, 88, 88, 88, 72, 86, 95, 70

set di dati bootstrapped

» Le rispettive medie di ciascuno sono 83,4, 86,2, 80,0, 81,7, 84,6 e 83,4%. Il valore più basso è 80,0 % e il più alto è 86,2%.

Random Forests (5)

- La parte critica sono i sei nuovi set di dati bootstrapped dal set di dati originale.
- Quando si sviluppa una random forest, ogni volta che abbiamo bisogno di un nuovo albero decisionale, **useremo prima il bagging per produrre un nuovo set di dati, quindi addestreremo l'albero decisionale usando quel set di dati, non l'originale.**
- Nota che molti dei sei set di dati **hanno valori ripetuti.**
  - Ad esempio, il set di dati 1 ha utilizzato sia 81 che 87 due volte, ma mai 72.
  - Questa randomizzazione del set di dati dato aiuta a creare alberi decisionali che si comportano in modo diverso l'uno dall'altro ma sono allineati con ciò che rappresenta il set di dati originale.

Random Forests (5)

- Il secondo trucco usato da una random forest è addestrare l'albero decisionale su un set di feature selezionato casualmente (*random feature selection*).
- Il set di dati è inventato, costituito da 9 vettori di caratteristiche, ciascuno con 6 caratteristiche, da x0 a x5.

A Toy Dataset

#	x_0	x_1	x_2	x_3	x_4	x_5
1	0.52	0.95	0.81	0.78	0.97	0.36
2	0.89	0.37	0.66	0.55	0.75	0.45
3	0.49	0.98	0.49	0.39	0.42	0.24
4	0.43	0.51	0.90	0.78	0.19	0.22
5	0.51	0.16	0.11	0.48	0.34	0.54
6	0.48	0.99	0.62	0.58	0.72	0.42
7	0.80	0.84	0.72	0.26	0.93	0.23
8	0.50	0.70	0.13	0.35	0.96	0.82
9	0.70	0.54	0.62	0.72	0.14	0.53

Random Forests (5)

- Gli alberi decisionali della foresta utilizzano un sottoinsieme selezionato casualmente delle sei caratteristiche.
- Ad esempio, diciamo che manteniamo casualmente le caratteristiche x0, x4 e x5.
- La Tabella mostra il set di dati ora utilizzato per addestrare l'albero decisionale.

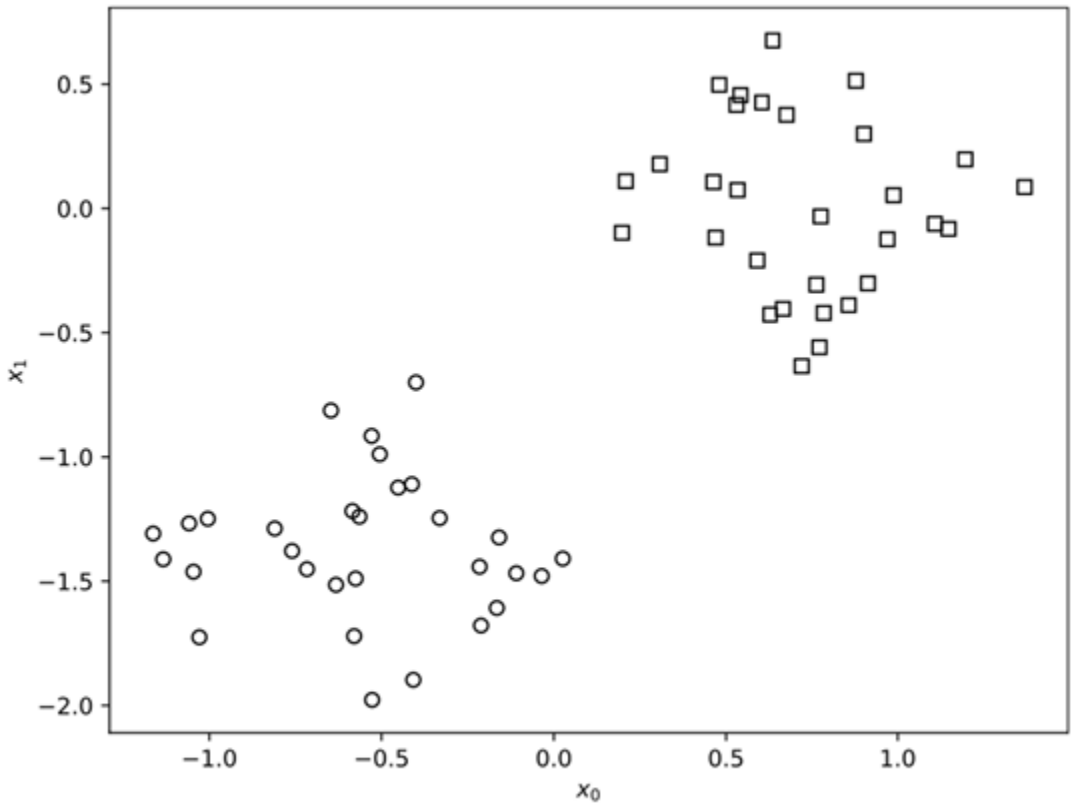
	A Random Collection of Features		
#	x_0	x_4	x_5
1	0.52	0.97	0.36
2	0.89	0.75	0.45
3	0.49	0.42	0.24
4	0.43	0.19	0.22
5	0.51	0.34	0.54
6	0.48	0.72	0.42
7	0.80	0.93	0.23
8	0.50	0.96	0.82
9	0.70	0.14	0.53

Random Forests (6)

- Entra in gioco l'ultimo dei tre pezzi: l'*ensembling*. Musicalmente, un ensemble è una raccolta di musicisti che suonano strumenti diversi.
- Una random forest produce un singolo output, un'etichetta di classe,
  - Combiniamo le etichette prodotte da ogni albero decisionale, votando come un classificatore k-nearest neighbors
  - Assegniamo l'etichetta vincente all'input
- *L'assegnazione casuale di caratteristiche agli alberi, combinata con set di dati bootstrapped e voto di ensemble, conferisce a una random forest la sua potenza.*

Support Vector Machines (SVM)

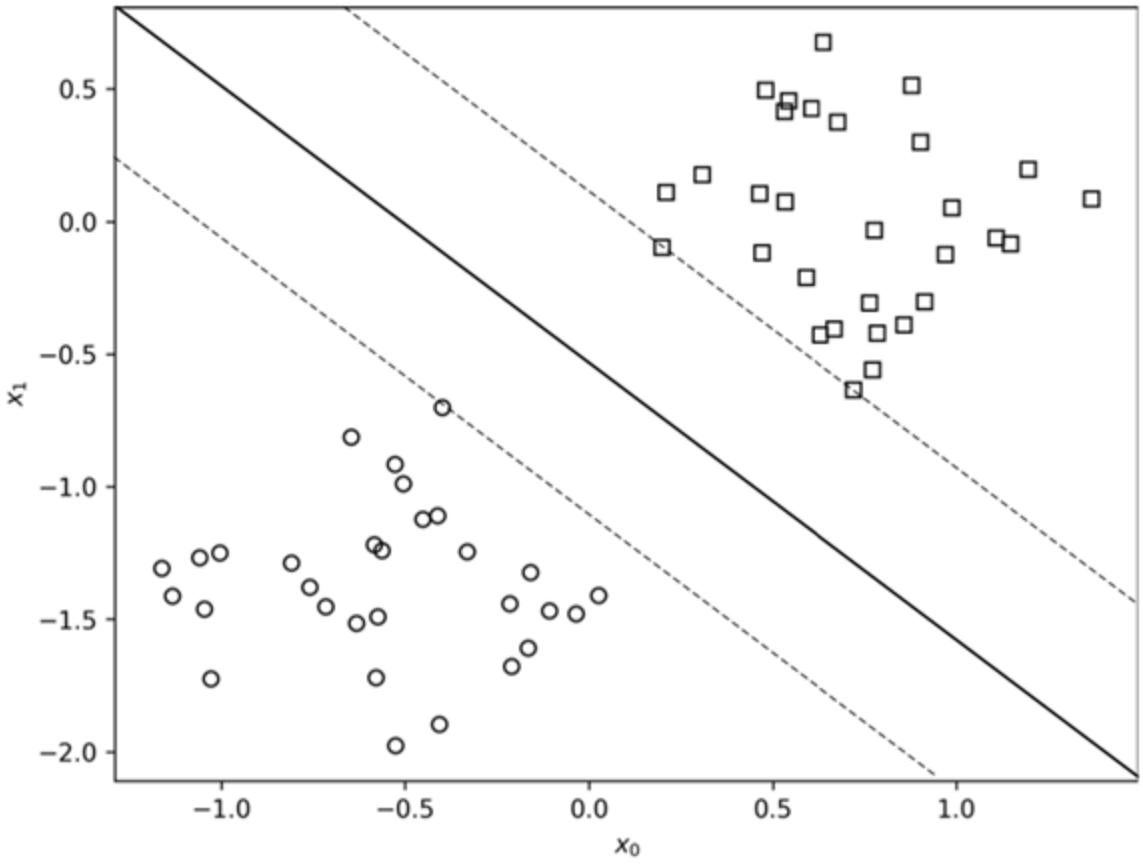
- Concetti da comprendere: *margini*, *vettori di supporto*, *ottimizzazione* e *kernel*.
- Set di dati a due classi (cerchi e quadrati) con vettori di caratteristiche bidimensionali, caratteristiche x0 e x1.



A two-class toy dataset with two features, x_0 and x_1

Support Vector Machines (SVM)

- Le SVM cercano di massimizzare il margine tra i due gruppi.
- Le linee tratteggiate definiscono il *margine* e la linea continua spessa segna il confine posizionato dalla SVM per massimizzare la distanza tra le classi.



The maximal margin separating line (heavy) and maximum margins (dashed)

Support Vector Machines (SVM)

- Le altre tre parti di una SVM (support vector, optimization e kernel) vengono utilizzate per trovare i margini (linee tratteggiate) e la linea di separazione.
- I *support vector* sono membri del training set trovati tramite un algoritmo di ottimizzazione. L'ottimizzazione implica la ricerca del meglio di qualcosa in base ad alcuni criteri.
- L'algoritmo di *ottimizzazione* utilizzato da una SVM individua i support vector che definiscono il margine massimo e, in definitiva, la linea di separazione.
- Il *kernel* trasforma i vettori di caratteristiche in una rappresentazione diversa, semplificando la separazione delle classi.

Impronte di dinosauri

- Dataset open source costituito da contorni di impronte di dinosauri
- Modelli addestrati:
  - Nearest neighbor (k = 1, 3, 7
  - Random forest (300 trees)
  - SVM linear
  - SVM radial basis function
- Risultati (destra) e matrice di

Theropod (top) and ornithischian (bottom) footprints

	Ornithischian	Theropod
Ornithischian	TN	FP
Theropod	FN	TP

Classifying Dinosaur Footprints

Model	ACC	MCC	Train	Test
RF300	83.3	0.65	1.5823	0.0399
RBF SVM	82.4	0.64	0.9296	0.2579
7-NN	80.0	0.58	0.0004	0.0412
3-NN	77.6	0.54	0.0005	0.0437
1-NN	76.1	0.50	0.0004	0.0395
Linear SVM	70.7	0.41	2.8165	0.0007

2022 "A Machine Learning Approach for the Discrimination of Theropod and Ornithischian Dinosaur Tracks" di Jens N. Lallensack, Anthony Romilio e Peter L. Falkingham.

ESEMPIO

Impronte di dinosauri

- Cosa abbiamo imparato:
  - una comprensione generale delle prestazioni dei modelli classici, come base per confrontare una rete neurale
  - anche i modelli classici possono funzionare bene su questo particolare set di dati
- Le loro prestazioni erano alla pari con quelle degli esperti umani (paleontologi), che hanno anche etichettato i contorni delle impronte dei dinosauri.
  - Secondo il documento originale, gli esperti umani avevano ragione solo nel 57% dei casi.

Conclusioni

- I modelli classici sono AI simbolica o connessionismo?
  - Non sono AI simbolica perché non manipolano regole o affermazioni logiche
  - Non sono connessioniste perché non impiegano una rete di unità semplici che apprendono la loro corretta associazione mentre lavorano con i dati
- Questo non significa che l'IA sia fasulla, ma significa che ciò che i professionisti hanno in mente quando parlano di IA è probabilmente diverso da ciò che il pubblico in generale considera "intelligenza artificiale".