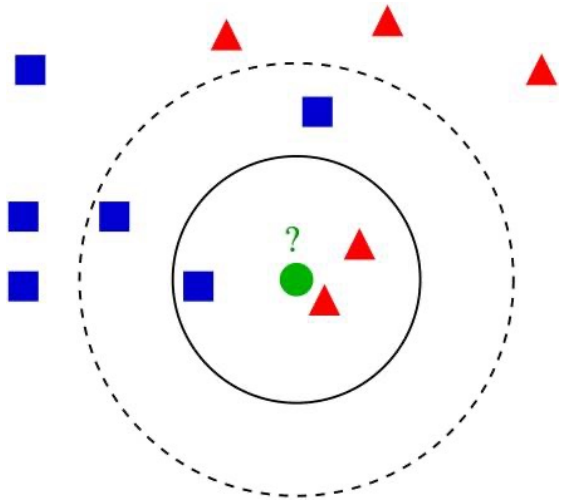
Introduzione agli Algoritmi di ML Supervisionati Non Parametrici

Docente: Tommaso Muraca

Algoritmi ML Non Parametrici: KNN

A **differenza** dei metodi che **abbiamo trattato** finora, **gli algoritmi non parametrici** non hanno una **struttura del modello specificata a priori.** Non speculiamo sulla forma della funzione f che stiamo cercando di apprendere prima di addestrare il modello, come abbiamo fatto in precedenza con la regressione lineare. Invece, la struttura del modello è determinata esclusivamente dai dati.

Questi **modelli sono più flessibili rispetto alla forma dei dati di addestramento**, ma ciò a volte va a scapito dell'interpretabilità.



k-nearest neighbors (k-NN)

k-NN sembra **quasi troppo semplice** per essere un algoritmo di apprendimento automatico. L'idea è di **etichettare un punto dei dati di test x trovando la media (o moda)** delle **k etichette dei punti dati più vicini.**

Diciamo che vogliamo capire se **il misterioso cerchio verde nella grafica** è un triangolo rosso o un quadrato blu. **Cosa facciamo?**

Algoritmi ML Non Parametrici: KNN

Potremmo **provare a inventare un'equazione fantasiosa** che guarda dove si trova il Cerchio Verde sul piano delle coordinate sottostante e faccia una previsione di conseguenza.

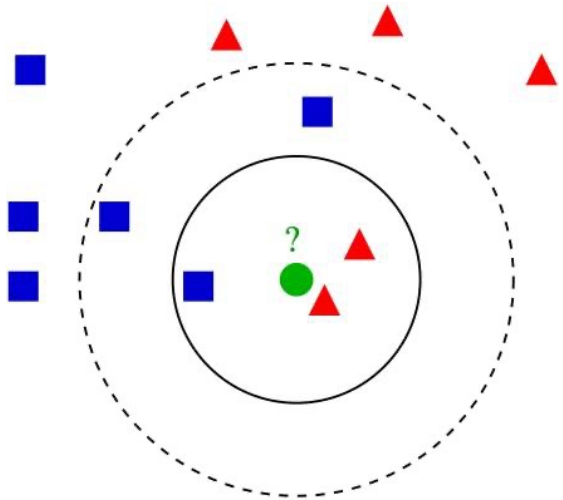
Oppure **potremmo semplicemente guardare i suoi tre vicini** e **indovinare** che il Cerchio Verde è probabilmente un Triangolo Rosso.

Potremmo anche **espandere ulteriormente il cerchio e osservare i cinque più vicini** e fare una previsione in questo modo (3/5 dei suoi cinque vicini sono quadrati blu, quindi immaginiamo che il misterioso cerchio verde sia un quadrato blu quando k= 5).

Guardiamo i k punti dati più vicini e prendiamo la media dei loro valori se le variabili sono continue (come i prezzi delle case) o la moda se sono categoriali (come gatto contro cane).

Se volessimo **indovinare** i **prezzi sconosciuti delle case**, potremmo **semplicemente prendere la media di un certo numero di case geograficamente vicine** e ci troveremmo con delle **ipotesi piuttosto interessanti**.

Queste ipotesi **potrebbero persino superare un modello di regressione parametrica** costruito da qualche economista che stima i coefficienti del modello per il numero di letti/bagni, scuole vicine, distanza dai trasporti pubblici, ecc.



Algoritmi ML Non Parametrici: KNN

Come utilizzare sensatamente k-NN per prevedere i prezzi delle case:

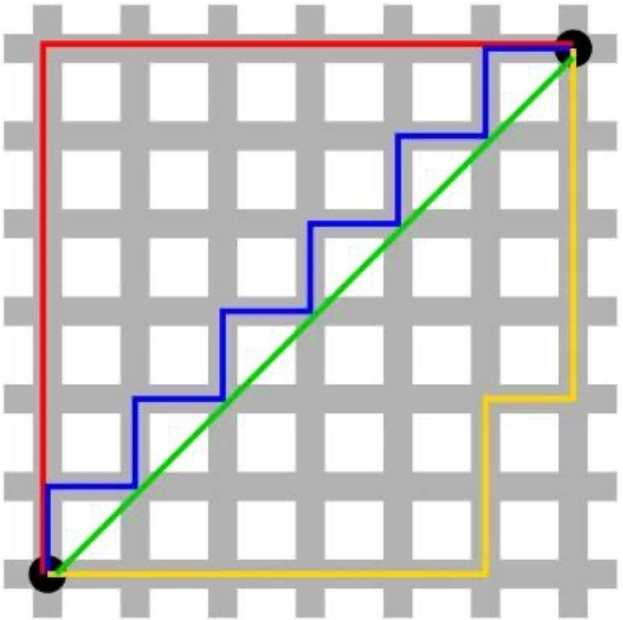
1) Memorizzare i dati di allenamento, una **matrice X di caratteristiche** come codice postale, quartiere, numero di camere da letto, metri quadrati, distanza dai trasporti pubblici, ecc., e una **matrice Y dei corrispondenti prezzi di vendita.**

2) Ordinare le case nel set di dati di addestramento in base alla somiglianza con la casa in questione, in base alle caratteristiche in X.

3) Prendere la media delle k case più vicine.

Questa è la stima del prezzo di vendita (ovvero ŷ).

Il fatto che **k-NN non richieda una funzione parametrica** predefinita f(X) relativa a Y con X lo rende **adatto a situazioni in cui la relazione è troppo complessa per essere espressa** con un semplice modello lineare.



Metriche di distanza: definizione e calcolo della "vicinanza"

Come si calcola la distanza dal punto in questione cioè dove si trovano i "vicini"?

Come determiniamo matematicamente quali dei quadrati blu e dei triangoli rossi nell'esempio sopra sono più vicini al cerchio verde?

La **misura più semplice** è la **distanza euclidea** (una linea retta, "in linea d'aria") (**nel grafico linea verde**).

Un'altra è la **distanza di Manhattan**, come camminare per isolati in una città (**nel grafico linea blu**).

Si potrebbe **immaginare** che la **distanza di Manhattan sia più utile** in un modello che prevede il calcolo delle **tariffe per gli autisti Uber**, ad esempio.

Algoritmi ML Non Parametrici: KNN

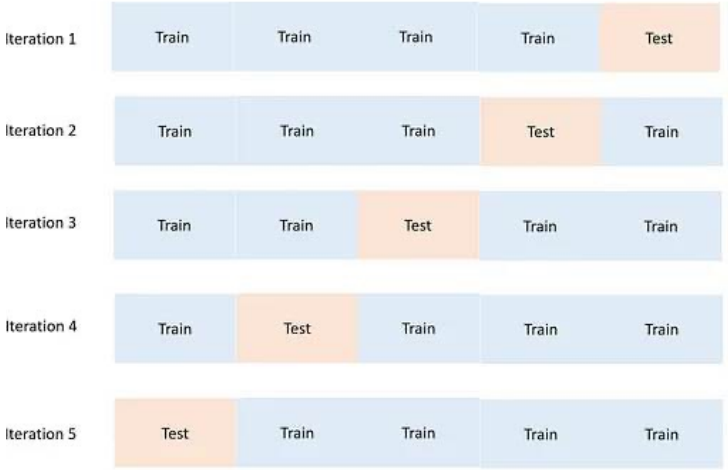
Scelta k: ottimizzazione degli iperparametri con convalida incrociata

Per decidere **quale valore di k utilizzare**, possiamo **testare diversi modelli k-NN** utilizzando diversi valori di k con **convalida incrociata:**

Dividiamo i dati di **addestramento** in **segmenti** e **addestriamo** il tuo **modello su tutti i segmenti tranne uno**, dopo **utilizziamo il segmento trattenuto come dati di "test".**

Abbiamo così le **prestazioni del nostro modello confrontando le sue previsioni (ŷ) con i valori effettivi dei dati di test (y).**

Scegliamo a questo punto **quello che produce l'errore più basso,** in media, in tutte le iterazioni.



Algoritmi ML Non Parametrici: KNN

Un k più alto impedisce il sovraadattamento

Valori più elevati di k **aiutano a risolvere il problema dell'adattamento eccessivo**, ma **se il valore di k è troppo alto il modello sarà molto parziale e poco flessibile** . Per fare un esempio estremo: se k = N (il numero totale di punti dati), il modello classificherebbe semplicemente in modo stupido tutti i dati di test come media o modalità dei dati di addestramento.

Se il singolo animale più comune in un set di dati di animali è un gattino Scottish Fold, k-NN con k impostato su N (il numero di osservazioni di addestramento) predirebbe quindi che anche ogni altro animale al mondo è un gattino Scottish Fold. Il che, secondo Vishal, sarebbe fantastico. Samer non è d'accordo.

Dove utilizzare k-NN nel mondo reale

Alcuni esempi di dove è possibile utilizzare k-NN:

- Classificazione: rilevamento di frodi. Il modello può essere aggiornato praticamente istantaneamente con nuovi esempi di formazione poiché stai semplicemente memorizzando più punti dati, il che consente un rapido adattamento ai nuovi metodi di frode.

- **Regressione:** prevedere i prezzi delle case . Nella previsione dei prezzi delle case, essere letteralmente un "vicino" è in realtà un buon indicatore di un prezzo simile. **k-NN è utile nei domini in cui la vicinanza fisica è importante.**

- **Dati di addestramento mancanti**. Se una delle colonne nel file .csv presenta molti valori mancanti, possiamo imputare i dati prendendo la media o la moda. k-NN potrebbe darci un'ipotesi un po' più precisa per ciascun valore mancante.

Andiamo alla pratica: