

Guida alla scelta di un modello di Machine Learning

Prof. Fedeli Massimo - Tutti i diritti riservati

1 Introduzione

Nel Machine Learning una delle domande più importanti è: *quale modello devo usare per il mio problema?*

Non esiste un algoritmo perfetto per ogni situazione. La scelta dipende da:

- il tipo di problema
- la quantità di dati
- la complessità delle relazioni nei dati
- la necessità di spiegare le decisioni del modello

In questa guida vediamo **quali modelli usare**, perché usarli, quali librerie Python permettono di implementarli e un **esempio concreto** per ciascuno.

2 Problemi di classificazione

Il modello deve assegnare un elemento a una categoria.

2.1 Regressione Logistica

Perché usarla: è semplice, veloce e funziona bene quando le classi sono separabili in modo abbastanza regolare. **Quando usarla:** problemi con due classi e relazioni non troppo complesse. **Libreria Python:** `sklearn.linear_model.LogisticRegression`

Esempio concreto: prevedere se uno studente sarà *promosso* o *non promosso* in base a ore di studio, assenze e media dei voti.

2.2 Albero Decisionale

Perché usarlo: prende decisioni tramite regole semplici del tipo “se... allora...”, quindi è facile da interpretare. **Quando usarlo:** quando vuoi un modello comprensibile e i dati hanno relazioni non lineari. **Libreria Python:** `sklearn.tree.DecisionTreeClassifier`

Esempio concreto: stabilire se una richiesta di prestito va accettata o rifiutata in base a reddito, età e storico dei pagamenti.

2.3 Random Forest

Perché usarla: combina molti alberi decisionali, riducendo errori e migliorando l'accuratezza. **Quando usarla:** problemi con relazioni complesse e un buon numero di dati. **Libreria Python:** `sklearn.ensemble.RandomForestClassifier`

Esempio concreto: riconoscere il tipo di pianta (specie) a partire da misure di foglie e fiori.

2.4 Reti Neurali

Perché usarle: riescono a modellare relazioni molto complesse. **Quando usarle:** immagini, audio, testo o dataset molto grandi. **Librerie Python:** TensorFlow/Keras, PyTorch

Esempio concreto: riconoscere se in una foto è presente un cane o un gatto.

3 Problemi di regressione

Il modello deve prevedere un valore numerico continuo.

3.1 Regressione Lineare

Perché usarla: è il modello più semplice e spesso sorprendentemente efficace. **Quando usarla:** relazioni quasi lineari tra variabili. **Libreria Python:** `sklearn.linear_model.LinearRegression`

Esempio concreto: prevedere il prezzo di una casa in base a metri quadrati, numero di stanze e zona.

3.2 Decision Tree Regressor

Perché usarlo: cattura relazioni non lineari senza trasformazioni complicate. **Quando usarlo:** dati con soglie e comportamenti “a scalini”. **Libreria Python:** `sklearn.tree.DecisionTreeRegressor`

Esempio concreto: stimare il tempo di consegna di un pacco in base a distanza, traffico e tipo di spedizione.

3.3 Random Forest Regressor

Perché usarlo: più stabile e preciso di un singolo albero. **Libreria Python:** `sklearn.ensemble.RandomForestRegressor`

Esempio concreto: prevedere il consumo elettrico di un edificio in base a temperatura esterna, numero di persone presenti e orario.

3.4 Gradient Boosting Regressor

Perché usarlo: spesso tra i modelli più accurati per dati tabellari. **Librerie Python:** `xgboost.XGBRegressor`, `lightgbm.LGBMRegressor`

Esempio concreto: prevedere il punteggio finale di uno studente usando molte variabili (voti, assenze, partecipazione, studio a casa).

3.5 Reti Neurali per regressione

Perché usarle: utili con grandi quantità di dati e relazioni molto complesse. **Librerie Python:** `TensorFlow/Keras`, `PyTorch`

Esempio concreto: prevedere l’andamento futuro della temperatura usando dati meteorologici storici molto dettagliati.

4 Problemi di clustering

4.1 k-Means

Perché usarlo: semplice e veloce. **Quando usarlo:** gruppi abbastanza compatti e simili tra loro. **Libreria Python:** `sklearn.cluster.KMeans`

Esempio concreto: suddividere clienti di un negozio in gruppi in base a quanto spendono e quanto spesso acquistano.

4.2 DBSCAN

Perché usarlo: trova gruppi di forma irregolare e individua dati anomali. **Quando usarlo:** dati con densità variabile. **Libreria Python:** `sklearn.cluster.DBSCAN`

Esempio concreto: individuare gruppi di sensori che registrano valori simili e isolare quelli che funzionano in modo anomalo.

5 Strategia pratica consigliata

1. Parti sempre da un modello semplice con `scikit-learn`.
2. Valuta le prestazioni.
3. Passa a Random Forest se serve maggiore accuratezza.
4. Usa reti neurali solo quando il problema è davvero complesso o i dati sono moltissimi.