ANALISI DEI DATI PER LE IMPRESE INDUSTRIALI (6)
Gli Alberi Decisionali

Nell'ultimo articolo abbiamo affrontato la regressione ed iniziato a trattare la classificazione. Ci occupiamo ora di una famiglia di algoritmi importantissima, quella degli Alberi Decisionali.

Caratteristiche degli Alberi

Gli alberi decisionali sono una serie di algoritmi di classificazione supervisionata molto popolari. Hanno **buone prestazioni,** sono semplici e veloci da addestrare e inoltre sono anche utilizzabili per la regressione (oltre che per la classificazione). Oggi sono anche conosciuti con il nome "CART": Classification and Regression Trees.
Hanno **il grande vantaggio della interpretabilità delle decisioni,** cosa non frequente negli algoritmi.

Funzionamento

Un albero decisionale è una struttura simile a un diagramma di flusso composta da nodi e rami. In ogni nodo viene eseguita una suddivisione dei dati in base a una delle caratteristiche in ingresso, generando due o più rami in uscita. Nei nodi successivi vengono effettuate sempre più suddivisioni e viene generato un numero crescente di rami per suddividere i dati originali. Questo processo continua finché non viene generato un nodo in cui i dati appartengono alla stessa classe e non sono più possibili ulteriori suddivisioni.
Per capire come funziona un albero, prendiamo **l'esempio della concessione di un prestito (Fig.1).**
Innanzitutto, l'algoritmo verifica se il cliente ha una buona storia creditizia. In base a ciò classifica il cliente in due gruppi: clienti affidabili e clienti meno raccomandabili. Si controlla poi il reddito (Income) del cliente e lo si classifica nuovamente in due gruppi. Infine si verifica l'importo del prestito (Loan amount) richiesto. In base ai risultati della verifica di queste tre caratteristiche, **l'albero decisionale decide se il prestito del cliente debba essere approvato o meno** (ciò è quanto accade oggi in banca…).
Un albero decisionale prende una serie di decisioni in base a un insieme

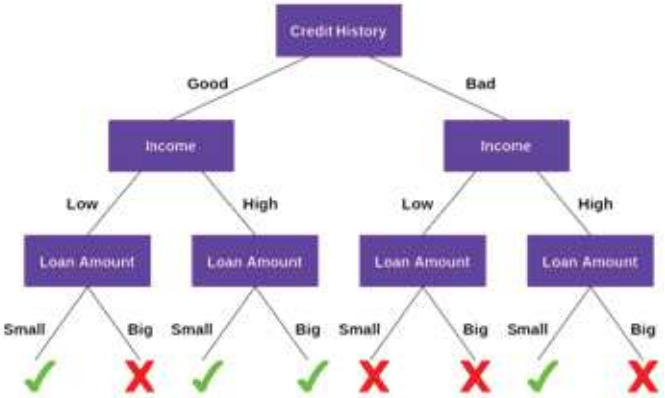
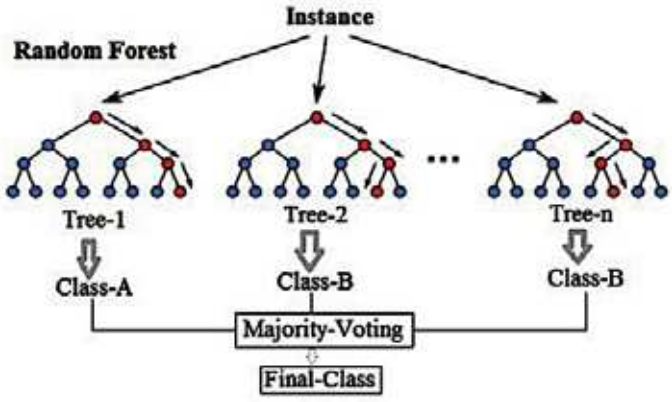


Figura 1

di caratteristiche e attributi presenti nei dati.
L'obiettivo di questi algoritmi è quello di **suddividere ad ogni passo ricorsivamente il training set in sottoinsiemi,** fino a quando ogni partizione è il più "pura" possibile in termini di classe di uscita. Per decidere quale caratteristica porti al sottoinsieme più corretto, occorre essere in grado di **misurare la "purezza" del set di dati.** Le metriche **più utilizzate sono"information gain" e l'indice di Gini.** In fondo all'articolo ho inserito le modalità di calcolo di questi indici.
Ci sono due modi per evitare che un albero vada in **overfitting**, ovvero che diventi troppo specializzato e che non riconosca bene nuovi dati mai visti precedentemente**: la potatura (pruning) e l'arresto anticipato.**
La potatura viene applicata a un albero decisionale dopo la fase di addestramento. In pratica **si lascia che l'albero sia libero di crescere** quanto consentito dalle sue impostazioni, senza applicare alcuna restrizione esplicita. **Alla fine, si procede a tagliare i rami** che non sono popolati in modo adeguato. La loro rimozione dovrebbe favorire la generalizzazione su nuovi dati non visti.
L'altra opzione per evitare l'overfitting è l'arresto anticipato.
Un criterio di arresto comune è il numero minimo di campioni per nodo. Il ramo interromperà la sua crescita quando verrà creato un nodo contenente un numero di campioni di dati



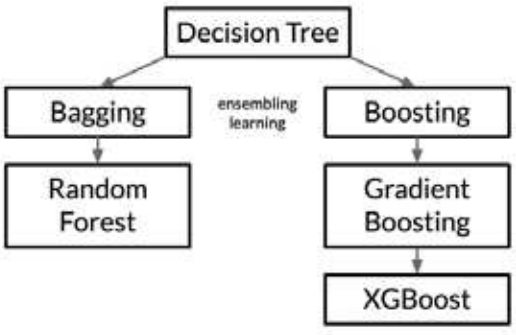
inferiore o uguale al numero minimo impostato. Ciò si fa per mantenere la leggibilità dell'albero.

Apprendimento d'insieme, Random Forest e XGBoost

I Random Forest risolvono il problema dell'overfitting perché combinano il risultato di più alberi decisionali per elaborare una previsione finale. Quando si crea un albero decisionale, una piccola modifica nei dati porta a un'enorme differenza nella previsione del modello. Con Random Forest questo problema non si verifica poiché i dati vengono campionati molte volte prima di generare una previsione. Questo viene fatto usando una **tecnica chiamata bagging.**

Il Bagging

Nel bagging vengono addestrati più modelli dello stesso tipo su dataset diversi, ciascuno ottenuto dal dataset iniziale tramite campionamento casuale con rimpiazzo (**bootstrap**). Il nome bagging deriva dalla combinazione delle parole inglesi bootstrap e



aggregation, in riferimento all'aggregazione di più modelli. Il bootstrap è una tecnica di campionamento e i dati di training sono campionati più volte in modo da generare diversi dataset, costruendo un albero per ognuno e ottenendone le previsioni. La previsione di ciascun albero decisionale verrà combinata per ottenere un unico risultato. Per la **Regressione si calcola il valore medio delle previsioni** mentre per la **Classificazione vince la previsione scelta a maggioranza** fra tutti gli alberi.
Il bagging è una procedura che viene applicata **per ridurre la varianza dei modelli di machine learning.** Nell'algoritmo Random Forest non vengono campionate casualmente solo le righe, ma anche le variabili (colonne). Si fa questo per evitare la similarità degli alberi, producendo potenzialmente lo stesso risultato.

Il boosting (potenziamento)

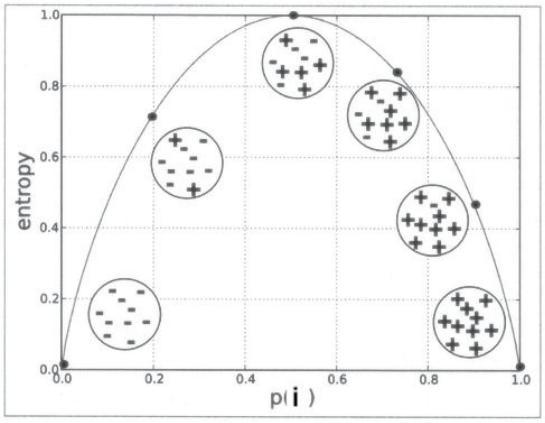
Esiste un altro tipo di apprendimento d'insieme chiamato **Gradient Boosting,** che utilizza la tecnica del "**boosting**". L'idea principale alla base di questo algoritmo è quella di costruire modelli in sequenza, con i modelli successivi che cercano di ridurre gli errori del modello precedente. La differenza fra il Random Forest e gli algoritmi basati sul boosting è che il R.F. fa la media dei risultati ottenuti da ogni singolo albero, quindi in parallelo, mentre il boosting continua a creare nuovi alberi cercando di migliorare i risultati del precedente, quindi in serie. Dalla tecnica del boosting deriva uno dei **più potenti algoritmi attualmente in uso: il XGBoost**, presentato nel 2014. È uno degli algoritmi più popolari al momento perché è vincente in innumerevoli competizioni e **fornisce prestazioni comparabili se non migliori delle Reti Neurali** per database di non enormi dimensioni.

Vantaggi e svantaggi degli alberi

In conclusione, le Random Forest in genere **hanno prestazioni migliori** rispetto agli alberi decisionali singoli, **anche se sono più lente** poiché è necessario più tempo per costruire più alberi. **Gli alberi decisionali sono più facili da interpretare e da visualizzare** ed è facile capire come l'algoritmo abbia raggiunto il suo risultato**. L'algoritmo XGBoost** è estremamente potente, di uso un po' più complesso e, purtroppo, **non permette una spiegazione chiara di come giunge alle sue conclusioni**.

Nota tecnica: calcolo dell'Indice di Gini e dell'Entropia

Con **l'information gain si usa un parametro basato sull'entropia**, un concetto utilizzato per misurare l'informazione o il disordine: **Entropy=$-\Sigma(\hat{p}_i)\log_2(\hat{p}_i)$** (si usa il log base 2), **$\hat{p}_i$** è la probabilità di una classe.
L'entropia è una metrica che **misura l'impurità di una divisione** in un albero decisionale. Determina come l'albero decisionale sceglie di partizionare i dati. **I valori di entropia**



vanno da 0 a 1, come si può vedere dalla figura 4.

L'articolo prosegue sull'edizione digitale della rivista a questo link:
https://dirigentindustria.it/fm-emilia-romagna/analisi-dei-dati-per-le-imprese-industriali-6.-gli-alberi-decisionali.html
Fonti e figure:
https://www.kdnuggets.com/
www.medium.com
www.wikipedia.org