

Sistema di raccomandazione agricolo

Stefano Elisio

February 21, 2024

Contents

1	Introduzione	1
2	Materiale e metodi	1
2.1	Librerie usate	1
2.2	Input	2
2.3	Output	2
2.4	Codice	2
3	Esperimenti e risultati	3
4	Conclusioni	4

1 Introduzione

Il programma prevede una parte di machine learning ed una di CSP(constraint satisfaction problem). Il programma ha il compito di assistenza e raccomandazione nell’ambito della coltivazione. inserendo i dati inerenti il nostro terreno e gli spazi a nostra disposizione ci dirà quali sono le piante più adatte a noi e in quanto spazio deve occupare ciascuna coltivazione per massimizzare il guadagno

2 Materiale e metodi

2.1 Librerie usate

csv, random, numpy, scikit-learn, PuLP

2.2 Input

Il programma prende in input i seguenti valori (presenti nel file main come costanti):

- Un seed, per tenere i risultati consistenti (può essere impostato su None per avere l'effetto opposto)
- Un'array di 7 float in quest'ordine [N, P, K, temperature, humidity, ph, rainfall] (può essere impostato su None per fare un test del modello di machine learning)
- Un database, (preso da kaggle.com) in questo caso un file .csv con 2200 righe, già bilanciato (100 campioni per pianta) in cui per ogni settupla di valori associa la pianta più adatta ad essi.
- Lo spazio coltivabile a disposizione e la capacità in chili che puoi gestire

2.3 Output

In caso di test del modello di machine learning il programma restituirà il numero di errori e la percentuale di errore registrata. In caso contrario verrà restituito:

- Un'array contenente le 4 piante più adatte alla tua situazione
- Quanti metri quadrati deve occupare ogniuna di esse
- Il terreno occupato, i chili prodotti e il prezzo stimato di vendita

2.4 Codice

Il codice è diviso in 4 file:

- main, dove vengono inseriti i dati in input, accede a tutti gli altri file e stampa i risultati
- mcEnv, è la parte del programma adibita al machine learning, qui viene letto il Database diviso per dati di input e di output e viene fatto allenare il modello. Come modello viene utilizzato uno importato dalla libreria sklearn.ensemble (RandomForestClassifier). Se il programma vuole fare un test del modello una parte dell'input e output viene tolta dall'allenamento e usata in seguito per verificarne l'accuratezza. Altrimenti il programma si allena su tutto il database e cerca e salva in un array le 4 piante più adatte alla settupla dei dati in input.

- ortaggi, è una classe costituita da una lista dei 23 diversi ortaggi presenti nel database e rispettivamente la produzione in Kg per metro quadro e il prezzo di vendita al Kg. Contiene un metodo che permette dato un ortaggio di calcolare quanto è il rendimento per metro quadro.
- cspEnv, è la parte di programma che si occupa della creazione e risoluzione del CSP, viene usata la libreria pulp che ci permette di creare un problema capace di lavorare con variabili continue. Vengono create 4 variabili positive con il nome delle 4 piante trovate dal mcEnv (che rappresentano lo spazio per ogni pianta) e si stabilisce la funzione obiettivo (massimizzare il guadagno) e i vincoli, che in questo caso sono già inseriti nel codice per comodità e sono i seguenti: lo spazio totale occupato deve essere minore di quello disponibile ma maggiore di $3/4$ dello stesso, ogni pianta deve occupare almeno il $1/20$ del terreno, la produzione di tutti gli ortaggi non deve superare la capienza del magazzino

3 Esperimenti e risultati

Inizialmente il programma doveva essere incentrato sulla finanza, e sulla predizione dei movimenti di uno o più indici azionari, cominciando con un sistema di regressione lineare ho inserito un database con i movimenti degli ultimi 10 anni di un indice. Dopo qualche accorgimento sono riuscito a far funzionare il programma ma non ero soddisfatto dell'accuratezza, con un errore non troppo grande ma simile alla variazione stessa del prezzo (il programma mi dava un errore medio di circa $1/2$ punti percentuale, che era sempre molto in linea con la media di quanto variava giornalmente l'indice). ho provato con tecniche diverse ma la regressione lineare rimase la più accurata. Ho anche provato con un sistema di classificazione per cercare di prevedere se il prezzo sarebbe salito o sceso, e lì ho ottenuto risultati molto più incoraggianti. Non'ostante tutto non sapevo esattamente come implementare quei risultati in un programma che facesse anche altro, e quindi alla fine dopo essermi fatto un pò la mano con i sistemi di machine learning ho optato per il programma che alla fine è stato quello finale. La strategia usata nella versione definitiva ha un accuratezza di media del 99,5 per cento su 440 campioni. Quel minimo errore rimanente viene poi reso ancora più piccolo dal fatto che ogni volta che viene scelta una pianta, il modello viene resettato e fatto riallenare da zero ma senza la pianta scelta precedentemente, fino ad avere 4 piante, così che anche se la pianta "migliore" non

viene scelta la prima volta il programma ha altri 3 tentativi per sceglierla.

4 Conclusioni

Il programma funziona esattamente come me lo ero immaginato, ci sono alcune funzioni che avrei voluto aggiungere (scelta dei vincoli del CSP in fase di esecuzione, inserimento dei propri dati di produzione e vendita) e alcune ovvie limitazioni del programma stesso come il fatto che ci sono solo 23 piante diverse nel database o che i valori della produzione e della vendita di default sono alquanto discutibili e possono variare molto non solo in base alla qualità delle coltivazione ma anche dalla frequenza e dall'attrezzatura a propria disposizione. Sono comunque soddisfatto del risultato e considero queste mancanze comunque secondarie, e sia per un fattore tempo che per un fattore di mancanza di conoscenza del settore agricolo ho preferito lasciare il programma così