Analisi di Decision Tree e Random Forest

Intelligenza Artificiale

Federico Magnolfi

Febbraio 2018

Abstract

Utilizzando delle implementazioni esistenti di Decision Tree e Random Forest, si vogliono analizzare i dati di Stack Overflow Developer Survey, 2017: in particolare si vuole calcolare l'accuratezza ottenuta tramite una 10-Fold Cross Validation nel predire se il salario di un partecipante è sopra o sotto la mediana.

Descrizione del problema

Il dataset in questione contiene i dati ottenuti dal sito Stack Overflow nel 2017 tramite un sondaggio posto ai propri utenti. Ad ogni utente sono state fatte diverse domande, riguardanti principalmente i loro interessi, abitudini, lavoro, istruzione, esperienze nell'ambito informatico. Una delle domande poste riguarda il salario percepito dall'utente.

Si vogliono testare due diversi algoritmi di apprendimento supervisionato per predire se il salario di un utente è sopra o sotto la mediana, conoscendo le varie risposte che sono state date durante il sondaggio. I due algoritmi in questione sono Decision Tree e Random Forest, per ognuno dei quali si misura l'accuratezza ottenuta tramite una 10-Fold Cross Validation.

Descrizione dell'esperimento

L'intero dataset è contenuto in un file csv; si osserva che ogni utente non risponde alla maggior parte delle domande, ovvero il dataset ha molti dati mancanti. Si è quindi reso necessario eliminare ogni riga in cui non è specificato il salario.

Inoltre, per cercare di ridurre il numero di dati mancanti e migliorare quindi l'accuratezza delle previsioni, si è scelto di eliminare quelle colonne per cui i dati mancanti superano una certa percentuale. Sebbene può accadere che una delle colonne eliminate abbia valori ai quali corrisponde sempre lo stesso output desiderato, si ritiene che essi non abbiano una valenza statistica significativa visto che la percentuale di valori nulli è troppo alta in quella colonna.

Si prevede lo svolgimento di più test, per fare la media e ottenere dei risultati più precisi. All'inizio di ogni test viene mischiato casualmente il dataset, in modo da evitare che i risultati siano influenzati da un particolare ordinamento dei dati. Si esegue quindi la 10-Fold Cross Validation prima con un Decision Tree o poi con una Random Forest, misurando in entrambi i casi sia l'accuratezza media sul Train Set che quella sul Test Set.

Si riportano infine i risultati ottenuti e si comparano mediante un istogramma.

Implementazione

Si implementa quanto descritto mediante il linguaggio di programmazione Python, versione 3. Gli algoritmi in analisi fanno parte della libreria scikit-learn, le classi usate per i classificatori sono DecisionTreeClassifier e RandomForestClassifier.

Il programma è composto da tre file:

- **prepare.py**: contiene la funzione che serve per leggere il file csv, filtrare il dataset dai dati non significativi, e creare le strutture dati che contengono le informazioni del dataset. Queste strutture dati vengono poi salvate su dei file tramite la libreria pickle.
- **exp.py**: all'inizio di questo file sono collocati tutti i parametri che regolano il funzionamento, come il numero di test da effettuare o i parametri richiesti dagli algoritmi di apprendimento. Vengono poi caricate le strutture dati dai file, che sono creati se non sono presenti. Esegue 10 test come descritto precedentemente, i risultati vengono mostrati sulla console e tramite un istogramma.
- **utilities.py**: contiene tutte le varie funzioni che vengono usate dagli altri due file, in modo da separare, per quanto possibile, la logica di funzionamento del programma da alcuni dettagli implementativi.

I parametri che regolano il funzionamento del programma sono di due tipologie: alcuni relativi alla lettura del dataset, altri relativi all'esecuzione dei test. Si riportano di seguito i parametri principali con relativa spiegazione, il valore usato nei test viene riportato tra parentesi.

Parametri relativi alla lettura del dataset:

- fileDataset (dataset/survey_results_public.csv): percorso del file csv contenente il dataset;
- separator (,): carattere separatore nel file csv;
- nameColSalary (Salary): nome della colonna nella quale è contenuto lo stipendio;
- nonAvailableValue (NA): stringa che nel file csv rappresenta i dati mancanti;
- acceptedNA (0.05): numero compreso tra 0 e 1 che esprime il numero di valori mancanti ammessi per colonna.

Parametri relativi all'apprendimento:

- K (10): parametro K relativo alla K-Fold cross validation;
- numTests (10): numero di test da eseguire;
- maxDepth (6): massima profondità di ogni albero, sia per Decision Tree che per Random Forest;
- minSamplesLeaf (55): numero di campioni per ogni foglia, sia per Decision Tree che per Random Forest;
- nEstimators (20): numero di alberi nella foresta.

I valori usati nell'esperimento sono stati individuati in modo non rigoroso, quindi molto probabilmente non saranno ottimi, ma sono tali da avere buoni valori di accuratezza ed un basso overfitting, oltre che un buon tempo di esecuzione dei test.

Utilizzo del programma

Per eseguire l'esperimento, i passi sono i seguenti:

1. Procurarsi il dataset, quello usato nell'esperimento è reperibile alla pagina https://www.kaggle.com/stackoverflow/so-survey-2017. In realtà è possibile utilizzare un qualsiasi dataset che sia in formato csv e preveda una colonna per lo stipendio. È necessario inserire, all'inizio del file *exp.py* il corretto percorso al file csv, assegnando l'opportuno parametro.
2. *Opzionale*: modificare a proprio piacimento i parametri di interesse, che sono tutti definiti verso l'inizio del file *exp.py*.
3. Eseguire il file *exp.py*: alla prima esecuzione vengono generati i file intermedi, dalla seconda in poi, e se esistono già, vengono soltanto letti. Se si vuole eseguire un nuovo esperimento, è possibile cambiare i parametri relativi all'apprendimento senza che il file csv venga letto per intero nuovamente. Se si volesse anche cambiare i parametri relativi alla lettura del dataset si può fare, ma è necessario cancellare almeno uno dei file intermedi.

Nota: i tempi di esecuzione variano sicuramente in base alla piattaforma di esecuzione, ma indicativamente sono di una decina di secondi per la preparazione dei file, e di qualche secondo per ogni test. L'avanzamento del programma viene mostrato sulla console.

Risultati sperimentali

Si riportano di seguito i risultati ottenuti nell'esperimento, le accuratezze sono arrotondate a tre cifre significative:

Tabella 1: Accuratezze medie ottenute, in percentuale

	Decision Tree	Random Forest
Train Set	80.7%	80.7%
Test Set	80.1%	80.4%

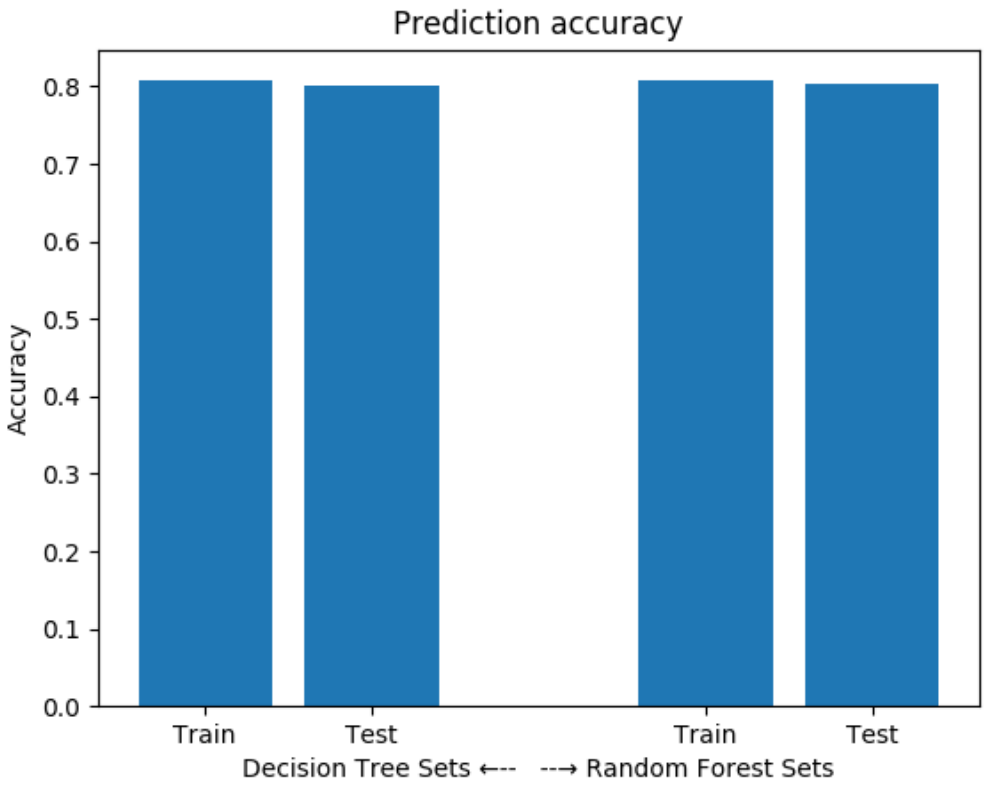

Figura 1: Grafico delle accuratezze, con numeri compresi tra 0 e 1

Conclusione