Relazione di Intelligenza Artificiale

Alessandro Soci

Gennaio 2018

1 Introduzione

In questo elaborato viene effettuato uno studio tra i classificatori *Naive Bayes* e *Decision Tree* paragonando l'*accuracy* svolta su vari dataset. Inoltre viene messo a confronto il risultato finale con quello descritto nella tabella 1 dell'articolo [1].

2 Strumenti usati

Il linguaggio di programmazione scelto è *Python*(versione 3.6.3). Gli algoritmi classificatori usati sono implementati nella libreria di **Scikit-learn**. Per la lettura dei vari dataset, salvati in *csv*, viene utilizzata la libreria **Pandas**.

2.1 Dataset

Sono stati scaricati 12 Dataset ottenuti dalla UCI repository [2]. Ognuno con valori continui o categorici o un misto di entrambi.

3 Classificatori

3.1 Naive Bayes

I metodi di Naive Bayes sono un insieme di algoritmi di apprendimento basati sul teorema di Bayes con l'assunzione che ogni coppia di features sia indipendenti tra loro. Data una classe y e un vettore di features da x_1 a x_n, il teorema di Bayes permette di arrivare alla seguente regola di classificazione:

$$\hat{y} = arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \tag{1}$$

$P(y)$ è la frequenza della classe y nel training set. I vari metodi di Naive Bayes differiscono principalmente nell'assunzione che fanno rispetto alla distribuzione di $P(x_i|y)$.

Le implementazione utilizzate per il confronto sono *Gaussian Naive Bayes* e *Multinomial Naive Bayes*. Nella prima la probabilità delle features è supposta essere Gaussiana ed il calcolo della $P(x_i|y)$ è dato da:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}} \tag{2}$$

dove σ e μ sono stimati usando il maximum likelihood del training data.

Nel caso del Multinomial, la probabilità delle features è data da:

$$P(x_i|y) = \frac{N_{yi} + \alpha}{N_y + \alpha n} \tag{3}$$

dove $N_{yi} = \sum_{x \in T} x_i$ è il numero di volte che la feature appare in un campione di classe y del training set T, e $N_y = \sum i = 1|T|N_{yi}$ la somma totale di tutte le features per la classe y.

Sono state selezionate queste due implementazioni di Naive Bayes perché, con Multinomial è possibile ricavare un'accuracy più precisa su valori categorici, invece con Gaussian si ottengono risultati migliori per dataset con valori continui e distribuzioni circa Gaussiane.

3.2 Decision Tree

I Decision Tree sono un metodo di apprendimento usato per la classificazione e la regressione. Lo scopo è creare un modello che predice il valore della variabile di target da semplice regole di inferenza date dalle feature di training. Viene utilizzata dalla libreria Scikit-learn la classe *DecisionTreeClassifier*.

4 Implementazione

Per la lettura dei dataset è stato utilizzato pandas, che permette di leggere in maniera intuitiva i formati csv e modificarne la struttura. In ogni dataset è stato aggiunto il nome della feature per ogni colonna con esemplificativi $f1, f2, etc \ldots$ ed invece la colonna della classe (parametro da predire) è stata nominata *label*. In base al dataset e alla relative informazioni vengono applicate diverse funzioni per renderlo più adatto ai classificatori. Alcune funzioni[1] sono:

- *get_dummies* converte valori categorici in variabili indicatrici;
- *factorize*, utilizzato per la colonna della classe, codifica il valore di input in un tipo enumerato;
- *drop* permette di eliminare colonne o righe superflue, come nel caso del dataset 'Echocardiogram'.

Inoltre in alcuni casi vengono normalizzati i valori delle colonne con parametri continui.

In alcuni dataset sono presenti dei valori sconosciuti denotati da "?". Per risolvere tale problema sono state seguite due tipi di strategie:

- drop della riga che contiene tale valore, nel caso in cui i valori mancanti siano relativamenti pochi
- sostituzione con il valore medio della colonna, nel caso di feature continue

Per svolgere la funzione di sostituzione viene usata la classe *Imputer* di Scikit-learn.

Per ottenere risultati validi e indipendenti dal partizionamento in test set e training set si fa uso dello shuffle split come *Cross-Validation*. Nello specifico viene usata una 10-fold cross validation, ovvero viene diviso il dataset in 10 sottoinsiemi, e nel mio caso, 8 dei quali usati come training set e 2 come test set. L'accuracy quindi sarà data da una media di tutte le possibili combinazioni del partizionamento del dataset.

5 Risultati

Di seguito verranno mostrati i risultati Table1 e la reference Table2.

[1] provenienti dalla libreria *Pandas*.

Table 1: Risultati

Dataset	Naive Bayes	Decision Tree
Iris	95.33 ± 3.71	94.33 ± 3.67
Echocardiogram	96.00 ± 4.42	96.00 ± 5.33
Mushroom	99.72 ± 0.20	100.00 ± 0.00
Breats	97.96 ± 1.30	93.43 ± 1.88
Credit	84.05 ± 4.07	80.00 ± 4.33
Pima	75.91 ± 1.94	70.13 ± 3.62
Hepatitis	86.43 ± 6.74	79.29 ± 9.69
Wine	96.67 ± 2.08	90.28 ± 4.85
Voting	95.63 ± 2.23	94.48 ± 2.34
Car	84.60 ± 1.21	97.05 ± 0.93
Dermatology	98.92 ± 3.02	95.14 ± 2.02
Glass	83.02 ± 3.61	97.91 ± 1.63
Average	91.19	90.67

Table 2: Reference

Dataset	Naive Bayes	Decision Tree
Iris	95.3 ± 4.5	95.3 ± 4.5
Echocardiogram	71.9 ± 1.8	73.6 ± 1.8
Mushroom	97.2 ± 0.8	100.00 ± 0.0
Breats	97.5 ± 2.9	92.9 ± 3.0
Credit	85.8 ± 3.0	88.1 ± 2.8
Pima	71.4 ± 5.8	71.9 ± 7.1
Hepatitis	83.0 ± 6.2	81.3 ± 4.4
Wine	98.9 ± 2.4	95.0 ± 4.9
Voting	91.4 ± 5.6	95.7 ± 4.6
Car	86.4 ± 3.7	88.9 ± 4.0
Dermatology	98.4 ± 1.9	94.0 ± 3.5
Glass	71.8 ± 2.4	73.3 ± 3.9
Average	87.41	87.50

Nella prima colonna ci sono i dataset utilizzati, invece nella seconda e nella terza abbiamo l'accuracy media con il relativo gap rispettivamente per Naive Bayes e Decision Tree.

6 Conclusioni

Alla fine di questo eleborato è possibile concludere come l'accuracy media su questi specifici dataset sia leggeremente più alta con implementazione Naive Bayes, rispetto a Decision Tree. Inoltre è possibile vedere come i risultati ottenuti siano paragonabili e migliori rispetto alla reference.

Bibliografia

1 Jin Huang, Jingjing Lu, Charles X. Ling. Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy. Department of Computer Science, The University of Western,Ontario, Canada N6A 5B7

2 C. Blake and C. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998. University of California, Irvine, Dept. of Information and Computer Sciences.