Naive Bayes in Intrusion Detection Problem

Nicole Barbuzzi

10 gennaio 2022

1 Introduzione

L'esercizio proposto prevede l'utilizzo delle implementazioni disponibili di Naive Bayes per il problema di rilevamento delle intrusioni in una rete, come descritto nell'articolo "Naive Bayes vs Decision Tree in Intrusion Detection Systems" del 2004, i cui autori sono: Nahla Ben Amor, Salem Benferhat e Zied Elouedi.

2 Dataset

Il dataset utilizzato é il KDD Cup 1999 [2], in particolar modo come set di:

- *training* il *kddcup.data_10_percent.gz* composto da 494021 connessioni;
- *testing* il *corrected.gz* composto da 311029 connessioni.

All'interno sono presenti connessioni TCP/IP descritte da 41 caratteristiche (discrete e continue) ed etichettate come normali o come attacchi, questi sono divisi in quattro categorie principali:

- DOS: denial of service attack
- U2R: user to root attacks
- R2L: remote to user attacks
- Probing: sorveglianza

Il dataset di train contiene un totale di 24 tipi di attacco, mentre il dataset di test ne contiene 14 in piú.
Rispetto al dataset usato nell'articolo, quello utilizzato in questo esercizio é leggermente differente, specificato dalla presenza di una nota sul sito ufficiale che avverte della modifica del dataset, senza peró la specifica di quale essa sia. Osservando i dati di train utilizzati si puó infatti notare che la loro distribuzione é differente rispetto a quella di riferimento:

Training	Dati osservati	Articolo
Normal	19.69%	19.65%
DOS	79.24%	79.07%
R2L	0.23%	0.23%
U2R	0.01%	0.22%
Probing	0.83%	0.83%

Table 1: Distribuzione dati nel dataset di addestramento.

Testing	Dati osservati = Articolo
Normal	19.48%
DOS	73.90%
R2L	5.21%
U2R	0.07%
Probing	1.34%

Table 2: Distribuzione dati nel dataset di test.

3 Procedimento

Il codice sviluppato si concentra sul caso five-classes, cioé la classificazione di una connessione in 5 categorie: 4 di attacco (DOS, R2L, U2R, Probing) e una di non attacco (Normal). Le strategie di raccolta dei risultati sono due:

- **Raccolta prima della classificazione (before classification)**: il dataset viene leggermente modificato raggruppando gli attacchi nelle varie categorie a cui appartengono prima di addestrare i classificatori.
- **Raccolta dopo la classificazione (after classification)**: ogni connessione viene classificata in uno dei 38 attacchi (o nella categoria nessun attacco) e successivamente assegnata ad una delle categorie a cui appartiene.

La categoria di attacco di appartenenza di una connessione é elencata al seguente link:

`https://kdd.ics.uci.edu/databases/kddcup99/training_attack_types`

3.1 Preparazione dei dati

Per estrarre i dati dal dataset é stata utilizzata la funzione *read_csv()* della libreria *pandas*, creando i dataset con i dati di train e di test. Inizialmente sono stati uniti in modo da:

- trasformare i dati categorici in numerici attraverso la funzione $LabelEncoder()$ fornita dalla libreria sklearn;
- discretizzare i dati continui attraverso la funzione $KBinsDiscretizer()$ fornita sempre dalla libreria sklearn, scegliendo come codifica *onehot-dense* e 24 come numero di intervalli di discretizzazione, in quanto portano ad un'accuratezza migliore.

Fatto questo il dataset é utilizzabile per l'addestramento.

3.2 Classificazione

Il classificatore Naive Bayes appartiene alla famiglia di algoritmi classificatori basati sul teorema di Bayes con l'assunzione *naive*, cioé che si ha indipendenza condizionale tra le caratteristiche. L'analisi effettuata si basa sul classificatore bayesiano gaussiano, fornito dalla libreria sklearn, che presuppone che i dati di ciascuna etichetta siano tratti da una distribuzione gaussiana.

4 Risultati

L'obiettivo dell'esperimento é sapere a quale categoria appartiene una data connessione. Per valutare l'efficienza della classificazione si calcola la *percentuale di classificazione corretta* **PCC** delle istanze appartenenti al training e testing set. Parlando di distribuzione gaussiana di fatto andiamo a calcolare l'accuratezza che misura la percentuale di previsioni esatte sul totale delle istanze, data dalla formula:

$$PCC = \frac{TP + TN}{TP + TN + FP + FN}$$

La seguente tabella mostra l'accuratezza calcolata sulle due strategie di raccolta dei risultati, tra parentesi sono indicati i risultati relativi all'after classification:

Training set	Testing set
94.51% (96.89%)	87.97% (79.36%)

Table 3: Distribuzione dati nel dataset di test.

Le successive figure sotto mostrano le matrici di confusione, la prima relativa al raggruppamento prima dell'addestramento e la seconda dopo l'addestramento:

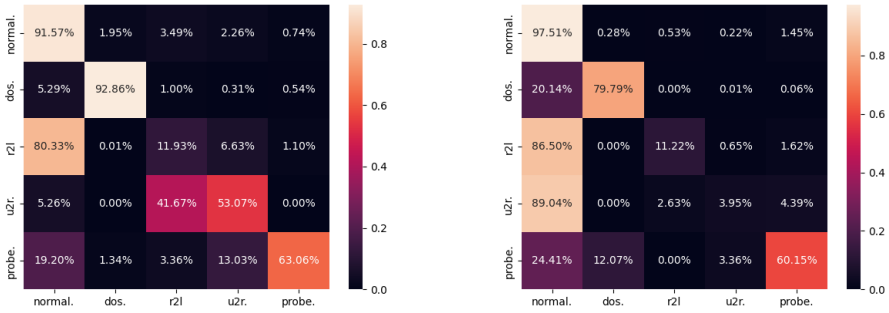

Figure 1: Before classification / After classification

Come si puó notare dalle tabelle si ha una buona classificazione delle categorie Normal, Dos e Probing con entrambe le tecniche ed un'errata classificazione delle categorie R2L e U2R, che potrebbe essere dovuto dal fatto che il numero di esempi del training set, sopratutto per la categoria U2R é molto basso (0.01%)

5 Conclusioni

I risultati ottenuti sono simili a quelli ottenuti nell'articolo, che peró vengono dalla stima della densitá del kernel che li porta ad aumentare, e quindi ad essere migliori come mostrato nel seguente articolo: "John, G.: Enhancements to the Data Mining Process. PhD thesis, Stanford University, 1997".

6 Riferimenti

1. Articolo di riferimento: `https://dl.acm.org/doi/10.1145/967900.967989`
2. Dataset KDD Cup 1999: `https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
3. Migliori risultati con la densitá del kernel: `https://www.proquest.com/openview/8a040a25364c70bfe5f2ff9a2d35bede/1?pq-origsite=gscholar&cbl=18750&diss=y`