

# B003725 Intelligenza Artificiale (2019/20)

Studente: Niki Dicostanzo (6157106) — <2019-11-12 Tue>

## Elaborato assegnato per l'esame finale

### Istruzioni generali

Il lavoro svolto sarà oggetto di discussione durante l'esame orale e dovrà essere sottomesso per email due giorni prima dell'esame, includendo:

1. Sorgenti e materiale sviluppato in autonomia (non includere eventuali datasets reperibili online, per i quali basta fornire un link);
2. Un file README che spieghi:
  - come usare il codice per riprodurre i risultati sottomessi
  - se vi sono parti del lavoro riprese da altre fonti (che dovranno essere **opportunamente citate**);
3. Una breve relazione (massimo 4 pagine in formato pdf) che descriva il lavoro ed i risultati sperimentali. Non è necessario ripetere in dettaglio i contenuti del libro di testo o di eventuali articoli, è invece necessario che vengano fornite informazioni sufficienti a *riprodurre* i risultati riportati.

La sottomissione va effettuata preferibilmente come link ad un repository **pubblico** su [github](#), [gitlab](#), o [bitbucket](#). In alternativa è accettabile allegare all'email un singolo file zip; in questo caso è **importante evitare di sottomettere files eseguibili** (inclusi files `.jar` o `.class` generati da Java), al fine di evitare il filtraggio automatico da parte del software antispam di ateneo!

---

### Naive Bayes for intrusion detection

In questo esercizio si utilizzano implementazioni disponibili di Naive Bayes (p.es. [scikit-learn](#) in Python o [Weka](#) in Java) al problema dell'intrusion detection come descritto in ([Amor et al. 2004](#)). I dati sono reperibili sulla pagina della [KDD Cup 1999](#). Per semplicità si consideri solo il caso con cinque classi, cercando quindi di riprodurre i risultati delle tabelle 2 e 3 nell'articolo (limitatamente a Naive Bayes).