Big Data Analytics

Lista Attività A.A. 2019/2020 – Text Analytics con NLTK

Effettuare a scelta una delle seguenti attività e inviare via mail la consegna al docente (riccardo.martoglia@unimore.it)

1. **Classification - Titanic**
   Scaricare e studiare i dati sui passeggeri del Titanic disponibili su https://www.kaggle.com/c/titanic (scheda "Data"). Quindi, risolvere il problema di "prevedere" la sopravvivenza o meno di ciascuno dei passeggeri in base alle loro caratteristiche, usando un classificatore Naïve Bayes. In particolare, studiare quali feature usare e come modellarle, scrivere il codice Python/NLTK e valutare l'accuratezza dei risultati ottenuti (eventualmente, è possibile anche fare confronti di efficacia tra più "configurazioni", es. diversi set di feature, ecc.)
   Consegna: PDF commentato con discussione e codice Python (includere dati e codice anche in un file .txt per facilitarne il testing)

2. **Text classification - Stack Overflow**
   Estendere l'esercizio di classification su Stack Overflow svolto in classe al riconoscimento di un numero N>=5 di classi (topic). Valutare quindi precision, recall, accuracy e F measure complessive, sia in versione micro-average che macro-average. Effettuare i test su un numero significativo di run (es., almeno 50), scegliendo ogni volta in maniera casuale la composizione di test-set e training-set a partire dall'insieme di post estratti. Valutare anche l'inclusione di altre feature estratte dal graph database, con l'obiettivo di aumentare l'efficacia.
   Consegna: PDF commentato con discussione e codice Python (includere dati e codice anche in un file .txt per facilitarne il testing)

3. **Statistical significance testing – confronto prestazioni machine learning**
   Studiare l'approfondimento su "Statistical significance testing" (capitolo "Speech and Language processing – Chapter 4", sezione 4.9, disponibile sul sito dell'insegnamento). Riprendere quindi l'esercizio di "gender classification" visto a lezione, estenderlo con codice Python opportuno per implementare la tecnica studiata e applicarla al confronto di prestazioni di due o più diverse configurazioni (es. più versioni della funzione di estrazione delle feature, ecc.).
   Consegna: PDF commentato con discussione e codice Python (includere dati e codice anche in un file .txt per facilitarne il testing)

4. **Sentiment analysis – Amazon reviews**
   Modificare l'esercizio di sentiment analysis sulle review Amazon svolto in classe e verificare l'efficacia del metodo effettuando queste varianti: (a) Utilizzare come tokenizer il "sentiment tokenizer" di Christopher Potts (link disponibile nelle slide del corso); (b) Modificare il dataset recuperando anche recensioni a 2, 3 e 4 stelle ed effettuare una classificazione a più classi (es. 5 classi di sentiment corrispondenti al numero di stelle delle recensioni). Effettuare quindi un confronto di efficacia tra queste varianti e la versione originale vista in classe. Svolgere i test su un numero significativo di run (es., almeno 50), scegliendo ogni volta in maniera casuale la composizione di test-set e training-set a partire dall'insieme di review estratte.
   Valutare anche l'inclusione di altre feature estratte dai dati, con l'obiettivo di aumentare l'efficacia.
   Consegna: PDF commentato con discussione e codice Python (includere dati e codice anche in un file .txt per facilitarne il testing)

NOTE

Per quanto riguarda il codice Python, è possibile (e gradito) produrre e consegnare un notebook jupyter .ipynb (https://jupyter.org/) invece di codice .py e relativi commenti su PDF.

Scadenza per premio partecipazione: **15 gennaio 2020**