Documentație Proiect ML

Pentru acest proiect am plecat de la codul făcut in timpul laboratoarelor, adică in prima faza am separat fiecare cuvânt din fiecare text si am aflat numărul de apariții in textul respectiv. Textul pe care îl avem de analizat se afla într-un document de tip .csv , iar pentru a citi acest format folosim librăria “pandas” in modul următor: am creat o variabila “corpus” in care se stochează datele din coloana text, iar apoi cu ajutorul funcției “preprocess” am creat un dicționar cu cele mai folosite cuvinte care apar in .csv .

Funcția “preprocess” este ultima parte din faza de data_cleaning a textului. Pentru data_cleaning am folosit librăriile “string”, “re” , “nltk”, prin care mi am transformat toate cuvintele in cuvinte cu litere mici, apoi am șters din texte datele de care nu aveam nevoie, de exemplu punctuația, userii, URL-eurile, hastag-urile, numerele, cuvintele cele mai des folosite in limba italiana (inițial pentru acest aspect am folosit comanda “ stopword = stopwords.words('italian')) ”, dar am observat ca îmi mai apar niște cuvinte care nu erau in adăugate in librăria de la “nltk”, am încercat sa le adaug manual, dar tot îmi apăreau așa ca am decis sa caut pe net aceste cuvinte si sa le adaug într-un vector), apoi am folosit un tokenizator, pentru a îmi delimita cuvintele intre ele si in ultima faza am făcut ca acele cuvinte scrise greșit sa fie identificate cu cuvintele scrise corect.

După partea de data_cleaning am folosit funcția “Counter” din librăria “counter” cu rolul de a-mi afișa frecventa de apariție a cuvintelor, cu aceste informații creez doua matrici, una care reprezintă cele mai întâlnite cuvinte, iar a doua care reprezintă un bag of words pentru fiecare tweet. Rezultatele obținute care se afișează in matrici le folosesc atât pe textele de antrenare cat si pe textele de testare, astfel voi folosi un clasificator ca sa pot detecta tweeturile misogine.

Pentru partea de clasificatori am folosi clasificatorul Complement Naive Bayes cu care am obținut un scor pe testele finale de 0.73860 si clasificatorul Gaussian Naive Bayes cu care am obținut un scor de 0.72424. Acești doi clasificatori i-am introdus in codul meu cu ajutorul librăriei “sklearn.naive_bayes”.

Complement Naive Bayes:

- în loc să calculăm probabilitatea unui element aparținând unei anumite clase, calculăm probabilitatea ca elementul să aparțină tuturor claselor.

O prezentare step-by-step a algoritmului:

I. Pentru fiecare clasă calculez probabilitatea ca instanța dată să nu îi aparțină.
II. După ce am făcut toate calculele pentru toate clasele, verific toate valorile calculate și selectez cea mai mică valoare.
III. Selectez cea mai mica valoare, deoarece are cea mai mica probabilitate de a nu fi o clasa particulara, ceea ce implica ca are cea mai mare probabilitate de a aparține clasei respective.

Nu selectez cea mai mare valoarea deoarece calculez complementul probabilității, iar cea mai mare valoarea este cel mai probabile sa fie clasa căreia ii aparține elementul.

Gaussian Naive Bayes:

- Valorile continue asociate cu fiecare caracteristica sunt presupuse da a fi distribuite in conformitate cu o distribuție Gaussiana. O distribuție Gaussiana se mai numește distribuție Normala(distribuția de probabilitate continua). Când plotează are forma unei curbe in forma de clopot care este simetrica cu media valorilor caracteristici.