Proiect la Inteligenta Artificiala

- Identificarea tweet-urilor misogine scrise in italiana

Pentru aceste 2 submisii folosesc 2 clasificatori :KNN pe cei mai apropiati 3 vecini si SVM SVC cu kernel=linear si C=0.5

Am urmat exemplele predate in laborator si am folosit modelul bag of words pentru a prelucra datele

Mai intai am prelucrat text-ul din train.csv in functia **tokenize** impartind textul in cuvinte eliminand punctuatia, facand toate literele mici si eliminand cuvintele scurte cele mai comune.Dupa am folosit functia **get_corpus_vocabulary** pentru a vedea frecventa de aparitie a fiecarui cuvant din text, dupa am folosit functia **get_representation** pentru a indexa primele 500 cele mai frecvente cuvinte care apar in textul nostru de train. Am folosit functia **corpus_to_bow** pentru a itera prin textele din corpus si pentru fiecare linie de text am aplicat functia **text_to_bow** care ia cuvintele din linia noastra si verifica daca aceste cuvinte se regasesc printre cele mai frecvente 500 de cuvinte iar daca se regasesc, pozitia din vectorul nostru(care acum are toate pozitiile egale cu 0) este incrementata cu 1.Functia **corpus_to_bow** ne face reprezentarea bag of words a textului nostru.Folosesc functia **cross_validate** pentru a imparti datele noastre de train in 10 parti,unde o parte este pentru testare si restul de 9 parti pentru antrenare.

<u>Folosim KNN pe cei mai apropiati 3 vecini:</u>

-Hiperparametrii reprezinta numarul de vecini , in cazul nostru 3.
-Pentru fiecare linie de text din train.csv vedem daca gasim cuvinte care se regasesc printre cele mai frecvente 500 de cuvinte din tot vocabularul nostru.
-Vedem de cate ori apar acele cuvinte in textul nostru
-Acum o sa avem o matrice unde pentru fiecare linie de text,pe coloana corespunzatoare fiecarui cuvant este afisat numarul de aparitii al acelui cuvant in textul nostru
-Facem acelasi lucru si cu liniile de text din test.csv
-Fac distanta dintre o linie de text din test.cvs si prima linie de text din train.csv,dupa a 2a linie de text din train.csv si tot asa(se folosesc numarul de aparitii ale unui cuvant din linia de text din test.csv si nr de aparitii ale aceluias cuvant din prima linie de text din train.csv,dupa nr de aparitii ale aceluias cuvant din linia a 2a de text din train.csv si tot asa)
-Vedem fata de care 3 linii de text din train.csv este distanta cea mai mica.
-Linia de text din test.csv va primi aceeasi eticheta ca eticheta predominanta pe care o gaseste la cei mai apropiati 3 vecini din train.csv
-Timpul de antrenare este: 106 secunde
-Rezultatele in urma antrenarii in maniera 10 fold cross-validation: 0.8538
-Performanta modelului pe Kaggle este: 0.68620

-Matricea de confuzie este:

	1(clasa prezisa)	0(clasa prezisa)
1(clasa actuala)	2109	554
0(clasa actuala)	195	2142

Folosim SVM cu SVC (Support Vector Classifier) cu separabilitate liniara:

-Am ales kernel=linear pentru ca multimea de puncte din plan sa fie separate de o dreapta

-Pentru hiperparamentul C am ales valoare 0.5 reprezentand eroare pe care o poate face modelul nostru pe exemplele de testare

-Se construieste o dreapta de separare intre cele 2 multimi care sa aiba o margine maxima(cea mai mare distanta posibila fata de punctele de antrenare)

-Parametrii w si b sunt folositi pentru a defini hiperplanul de separare

-Timplu de antrenare este 113 secunde

-Performanta modelului pe Kaggle este: 0.70363