Documentație proiect ML Fundamentals
Task 2 - Stackoverflow Data Analysis

Tița Andrei-Petruț

Februarie 2021

Partea 1

Pentru prima parte, unde trebuia să selectez top 3 topicuri dintr-un anumit post, am selectat din baza de date, doar coloanele „title”, „body”, „tags” pentru că toate 3 pot conține un topic. Cred totuși că cele mai multe topicuri se află în coloana „tags” (cel puțin numele limbajului de programare).

În acest exemplu datele nu vin însoțite de label-uri. Deci voi face o verificare luând la întamplare câteva postări.

În prima parte o să filtrez datele cu aceleași procedee de la task-ul 1 (Twitter Analysis). În plus am șters tagurile de html si pipe-urile de la tags.

Când am vizualizat datele mi s-a părut că elementele din tag-uri au mai mare relevanță și pare ca ar putea fi topicuri așa că la construirea stringului cu title+body+tags m-am gândit să adaug de două ori tags pentru a crește apariției cuvintelor din tags și a le acorda o șansă mai mare de a deveni topicuri.

Inițial am încercat să folosesc tf-idf pentru găsirea topicurilor însă apoi mi-am dat seama, din scoruri, că multe topicuri au foarte multe cuvinte în comun pentru că datele provin de pe un site cu întrebări de programare. Deci nu cred că este o variantă bună să decidem topicul unei întrebări în funcție de frecvența inversă a cuvintelor din alte întrebări. Așa că mă voi baza doar pe frecvența cuvintelor din întrebarea curentă.

Pentru asta am folosit metoda get_top_n_words care calculeaza frecventa cuvintelor dintr-un string (fiecare linie din setul meu de date) si returneaza un vector cu primele n elemente (si cu frecventa lor).

Am pus si un ngram_range la CountVectorizer pentur ca observasem ca la un text pusese cele mai frecvente cuvinte „visual” si „studio” care ar trebui sa fie un singur topic.

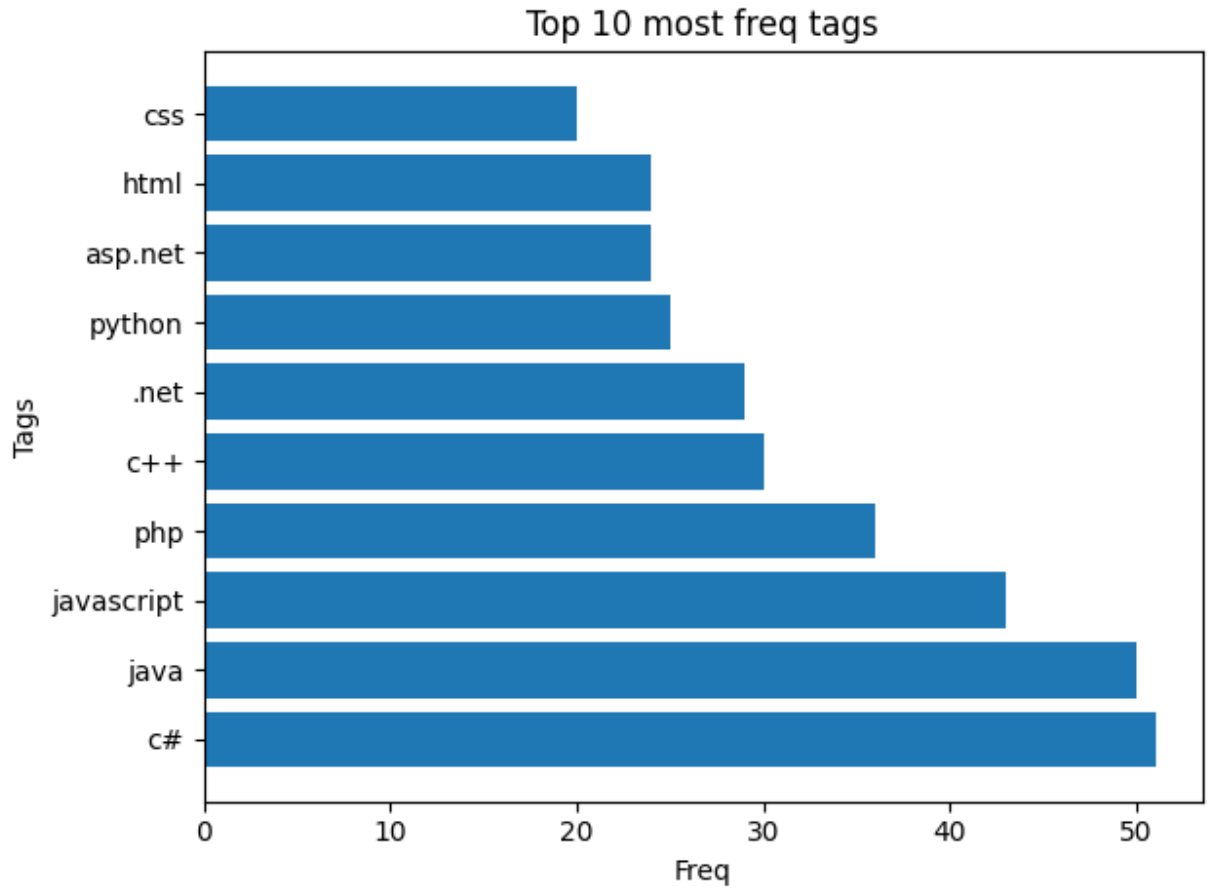
Am adaugat toate aceste rezultate intr-un vector pe care l-am pus ca o noua coloana in dataframe.

Partea 2

Ma gandisem sa incerc sa fac un model de ML care sa prezica in functie de mai multe criterii (precum scor, vizualizari si altele) ce badge ar trebui sa primeasca un utilizator dar dupa ce am verificat pe site ce conditii sunt la badges mi-am dat seama ca ar fi prea complicat asa ca am ales sa fac niste taskuri mici de data analysis. Asa ca am vrut sa aflu care sunt cele mai folosite 10 taguri. Observ ca cel mai folosit tag este „C#” urmat de Java si JavaScript.

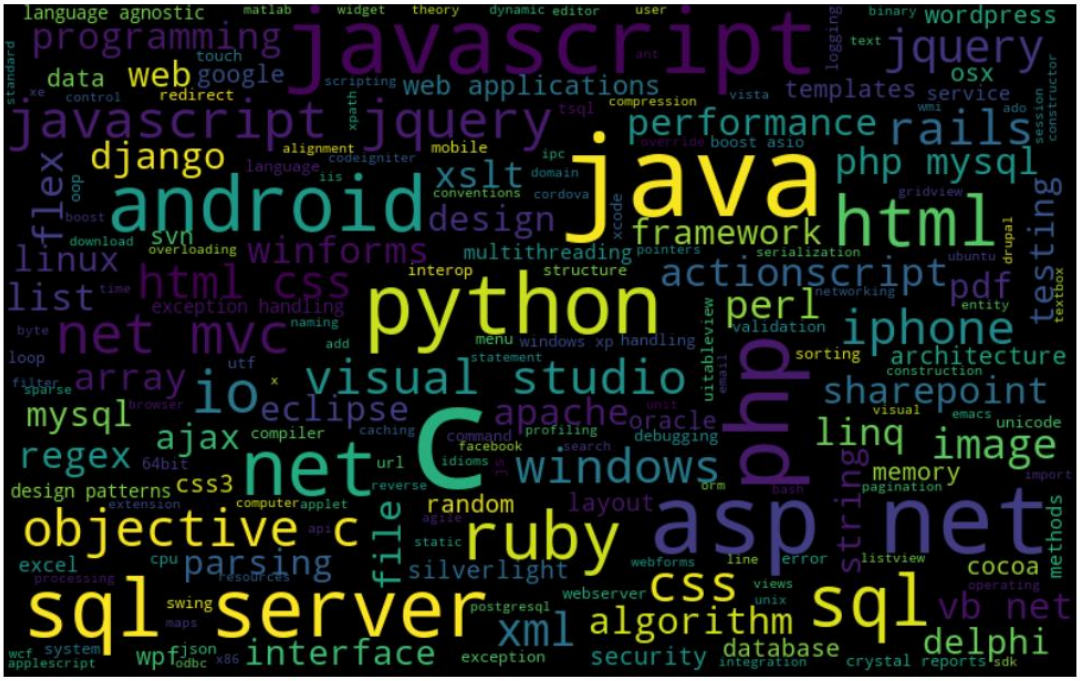
Am folosit acelasi fisier .csv pentru ca am avut nevoie doar de aceleasi campuri de date.

Nu am explicat fiecare parte din cod in documentatie pentru ca am comentat codul si mi se pare destul de self explanatory.

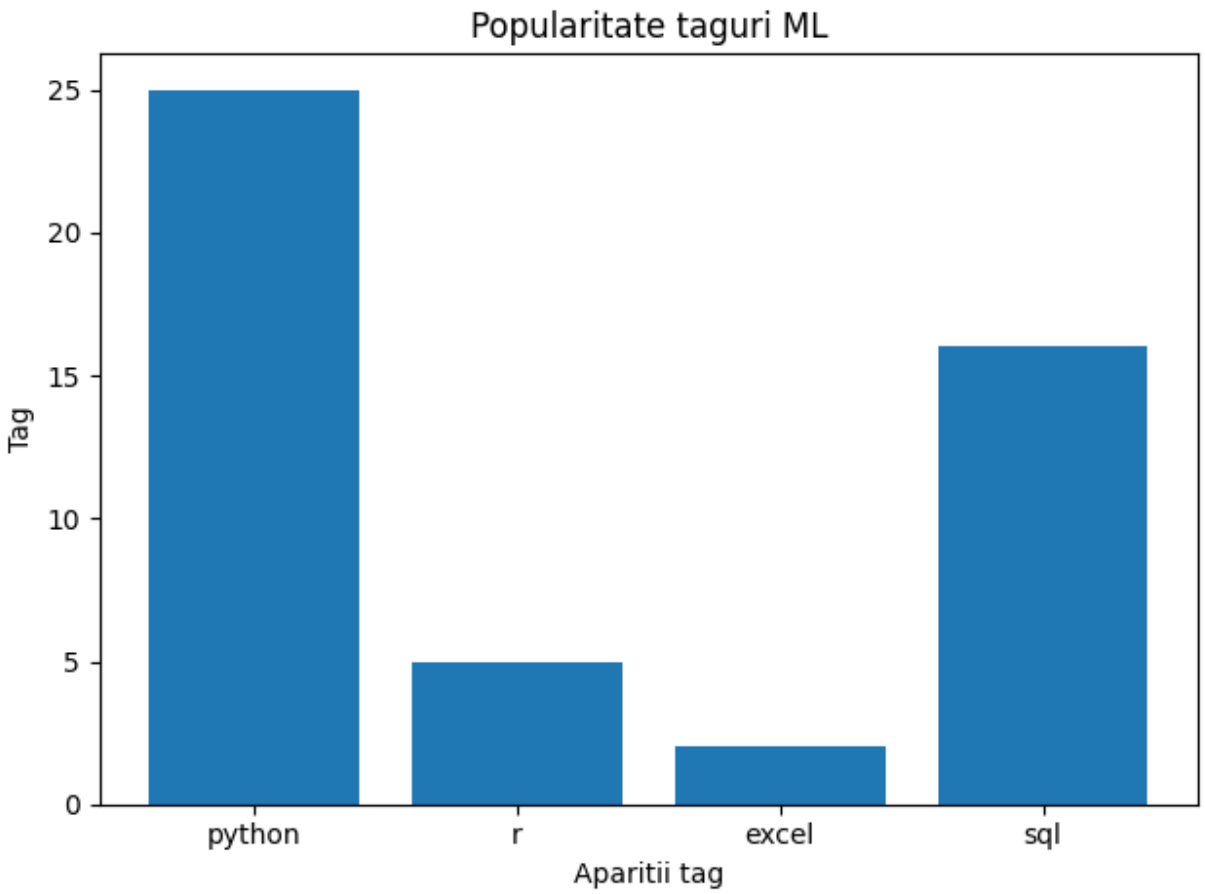


Pentru o vizualizare a mai multor taguri importante am combinat vectorul cu toate tagurile intr-un singur string si am folosit wordcount astfel am obtinut:

Observ ca C (fara #, nu stiu de ce, probabil a fost eliminat), java si javascript apar destul de mari in imagine deci se confirma graficul de mai sus.



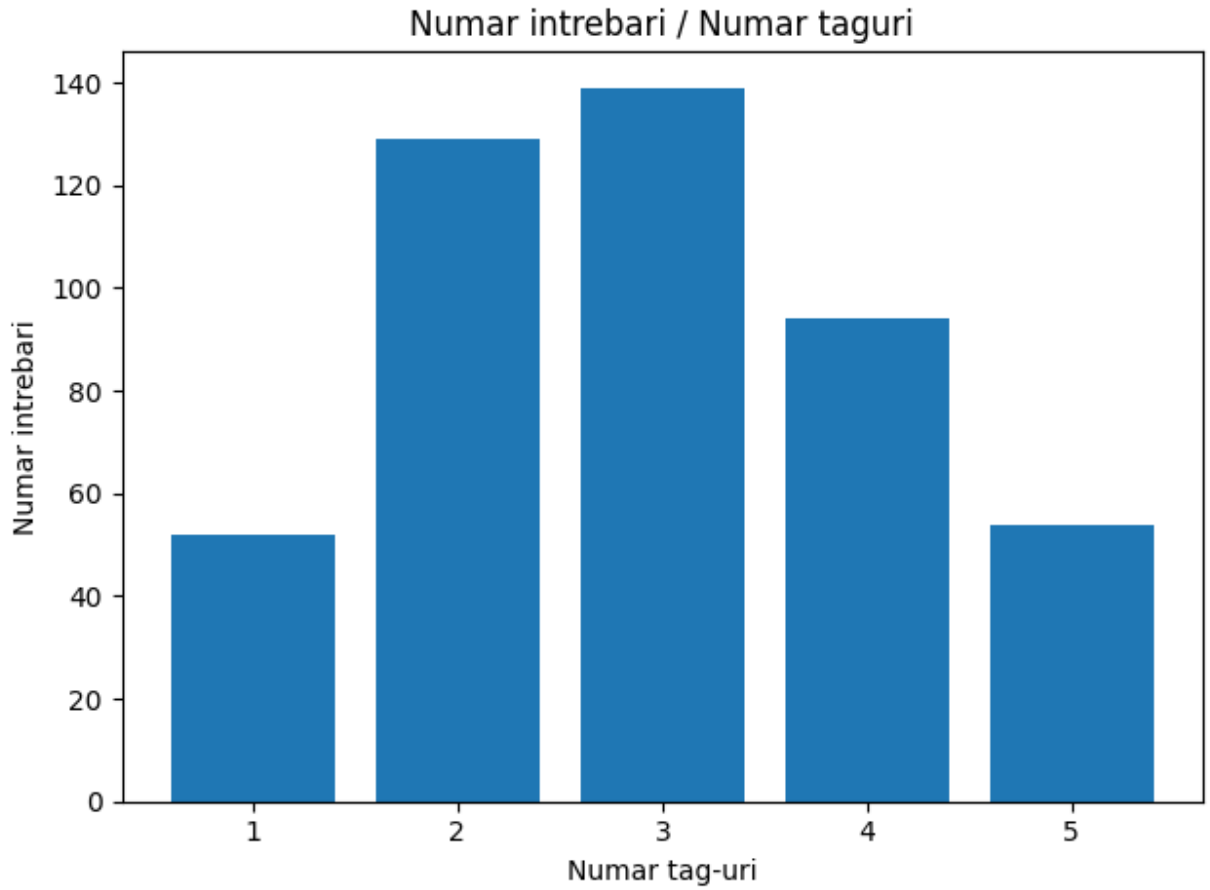
In continuare m-am gandit sa vad cat de populare sunt tagurile de ML in baza de date. Astfel am definit un vector cu taguri care au legatura cu machine learningul: ml_tags = [‚python’, ‚r’, ‚excel’, ‚sql’]. Dupa definirea valorilor de pe axa x si axa y si plotare am obtinut urmatorul grafic:



Observam ca python si sql sunt destul de intalnite, python chiar aparea pe locul 7 la frecventa tuturor cuvintelor.

Mai departe am încercat să aflu câte taguri ar trebui să folosesc ca să am cea mai mare șansă să mi se răspundă la o întrebare, deci am făcut un grafic cu numarul de întrebari în funcție de numărul de taguri.

După afișarea graficului observ că cele mai multe întrebări au 3 taguri deci probabil șansele de a primi un răspuns sunt cele mai mari când folosim 3 taguri.



Am aflat astfel, prin tehnicile de mai sus câteva inside-uri despre baza de date folosind numai 3 coloane.