Project Fuzzy Logic

-Ceausu Dan Andrei-

Consideram datele ce nu au asignate coloane date corupte.

Date corupte:

facebook - 834 linii

google - 9595 linii

Importam cele 3 csv-uri in dataframe-uri, verificand ce encode au, in cazul de fata fiind ENCODE UTF-8. Facem un clean initial pe dataset-uri, astfel eliminam coloanele unnamed din website, transformam phone din facebook dataset din float in string in format .0f. Facem clean de altfel si in phone(website dataset) unde observam ca numerele de telefon sunt sub forma X.XXe+XX si verificand rezultatul cu numere de telefon de pe domain, observam ca au fost rotunjite numerele de telefon, astfel le consideram data corupte reprezentand un numar de 5337 de linii, pentru restul numerelor de telefon vom sterge orice nu reprezinta cifra. Pentu phone_raw(google dataset) se va sterge orice caracter ce nu este reprezentat de cifra.
Se face join intre cele 3 tabele considerand chei unice domain-ul, resprectiv numarul de telefon. In noul data frame initializam orice nan/none cu blank si avem grija de datele ce contin emoji-uri (ex: *Soft volume* 🎬).

Tratare nume tara:

Cautam in coloanele comune de country_name daca gasim aceasi tara pe linie in celulele care nu sunt blank, atunci o asignam ca si tara buna. Daca gasim tari diferite, atunci facem count pentru fiecare tara si cautam sa vedem de cate ori o regasim in coloanele comune de adresa, respectiv country_name, o asignam pe cea gasit de cele mai multe ori, daca vom avea acelasi numar de count, atunci ne uitam in celulele comune de phone_country_code, daca si dupa aceasta noua verificare count-urile noastre sunt egale, atunci punem blank pentru linia respectiva. Daca toate celulele din country_name sunt blank, ne ducem si verificam in country_code tara, daca sunt la fel in celulele ce nu sunt blank, atunci asignam tara corespunzatoare codului ISO 3166. Daca regasim date diferite atunci facem count in a tarii in adresa, asignam valoare care a fost regasita cu un count mai mare, daca se regasesc egale, atunci ne uitam in phone_country_code, daca si dupa aceasta noua verificare count-urile sunt egale atunci linia corespunzatoare primeste blank. In caz ca regasim country_code blank in coloanele comune, atunci se verifica in phone_country_code, daca celulele care nu sunt blank sunt la fel, asignam linii respective tara corespunzatoare codului ISO 3166, daca gasim coloane diferite ne uitam in adrese si facem count pentru fiecare tara. Linia primeste count-ul mai mare, daca regasim count-uri egale, in acest caz punem blank pentru linia respective. Se creaza o coloana noua Country_Name cu lista in care se gasesc numele tarilor corespunzatoare fiecarei linii

Tratare cod tara

In acest moment avem deja numele de tara, deci tot ce trebuie facut este sa transformam numele de tara in cod ISO 3166, folosim datele din libraria countries si corectam datele conform acelora. (“Vietnam” ramane la fel deoarce in acel dataset este regasit “Viet Nam” si consider ca nu este correct). Se creaza o coloana noua Country_Code ce primeste o lista in care se gasesc codurile de tara corespunzatoare.

Tratare cod tara numar de telefon

Verificam daca phone_country_code este comun pe toate celulele ce nu sunt blan, in caz contrar se verifica cu country_code, daca nu se gaseste nici un match vom pune blank.

Tratare nume companie

Selectarea numelui corect din cele 3 posibilitati este realizata cu ajutorul distantei levenshtein. Astfel vom verifica fiecare nume cu fiecare in parte dupa care rezultatele intre ele si facem append unei liste a cuvintelor ce au o similaritate mai mare de 85%.
Vom compara name1 cu name2 => lista12; name1 cu name3 => lista13; name2 cu name3 => lista23
Vom extrage din lista12 si lista13 cuvinte ce au o similaritate mai mare de 85% in lista123.
Vom adauga in lista_finala similaritatea dintre lista123 si lista23.
In aceasta noua lista vom elimina duplicatele, dupa care cu ajutorul logici Fuzzy string-ul rezultat din lista_finala se compara cu cele 3 nume de company.
Acum in caz ca se gaseste o similaritate de acelasi procent intre ele vom extrage numele care are cea mai mica lungime.
Lista company_name primeste acest nume returnat, urmand sa fie create o noua coloanal Company_Name cu lista company_name.

Tratare nume oras, nume regiune, cod regiune si zip code

Pentru nume oras, regiune, cod de regiune si zip code, verificam daca se gasesc in coloanele comune cu celule care nu sunt blank acelasi oras, regiune, cod de regiune, respectiv zip code si asignam numele lini respective, in caz contrar se cauta in adrese de cate ori se regaseste fiecare si vom extrage numele care s-a regasit de cele mai multe ori. Daca count-urile sunt egale, in acest caz vom pune blank.
Aceasta procesare de data rezulta 4 liste cu ajutorul carora se creaz 4 noi coloane “City_Name”, “Region_Name”, “Zip_Code”, “Region_Code”

Tratare Adresa

Daca adresele sunt aceleasi atunci asignam valuarea liniei respective, in caz contrar delimitam fiecare adresa dupa virgula, iar cu ajutorul logici Fuzzy vom compara string-urile ce au o similaritate mai mare de 85 , le punem intr-o noua lista, eliminam duplicatele si duplicatele similare.
Vom luat lungimea maxima dintre cele 3 si asignam string-ul unei liste ce creaza o noua coloanal Address.

Tratare Categorie

Daca categorile sunt aceleasi asignam valuarea lini respective, in caz contrar fiecare categorie se delimiteaza dupa spatiu formand noi liste. Vom face acelasi lucru pentru fiecare linie cu coloanele text si description si vom face count pentru fiecare lista generata de categorie din care se compara fiecare element cu toate elementele din listele text, respective description, iar in cazul in care avem o similaritate mai mare de 85% crestem contorul.
Comparam contorul fiecarei categori si asignam categoria cu cel mai mare contor, daca contorul este egal cu altul, extragem una din categori deoarece ambele variante sunt viabile.

Oriunde nu se intra pe o conditie se pune blank pentru linia respective in coloanal corespunzatoare.

More Clean

Se elimina coloanele comune cu exceptia coloanei “Domain” si se pastreaza doar cele noi. Coloanele {“email”, “link”, “domain_suffix”, “tld”, “page_type”, “site_name”, “language”} au fost eliminate deoarece coloanele de interes sunt reprezentate prin Company name, Phone, Address, Category.
Se creaza un nou dataframe ce-l copiaza pe actualul, deoarece in caz ca nu avem nevoie spre exepmlu de coloanele “Region_Name”, “Region_Code”, “Zip_code”, putem obtine un data set mai mare.

In noul dataframe se inlocuieste orice blank sau “ “ cu NaN/None. Se face drop la linile duplicate dupa numele companiei si numarul de telefon si drop liniilor ce contin NaN/None

Se exporta noul dataframe intr-un csv “Company.csv”.

Probleme intampinate

1. Alegerea intre a face un dataset nou accurate sau coverage. Pentru a doua varianta aplicam aceleasi filtre doar ca se facea join-ul intre cele 3 seturi de data doar pe baza domain.
2. Nu am gasit nici un dataset care sa contina numele de regiune cu codul corespunzator pentru a putea aplica aceasi logica ce am folosit-o pentru tara si cod tara.
3. O lista cu toate modele de zip code din lume cautata si pe baza acesteia create functi regex in cazul in care nu avem nici un zip code sa incercam sa-l cautam in adrese.
4. Un data set cu numele corect al companiilor pentru a putea delimita numele de companie de numele legal spre exemplu.
5. Tratarea emoji-lor in text.

Bibliografie

https://www.datacamp.com/community/tutorials/fuzzy-string-python

https://stackoverflow.com/questions/57514169/how-can-i-remove-emojis-from-a-dataframe

https://pypi.org/project/pycountry/