UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

PROIECT

INTELIGENȚĂ ARTIFICIALĂ

PROFESORI COORDONATORI:

ALEXE BOGDAN DUMITRU

DIACONU ALEXANDRA

STUDENT:

FLORIAN LUCA-PAUL

BUCUREȘTI

2022

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

IDENTIFICAREA DIALECTULUI SURSĂ

DINTR-UN TEXT TRADUS

PROFESORI COORDONATORI:

ALEXE BOGDAN DUMITRU

DIACONU ALEXANDRA

STUDENT:

FLORIAN LUCA-PAUL

BUCUREȘTI

2022

CUPRINS

INTRODUCERE

DESCRIEREA PROIECTULUI

Proiectul de față prezintă rezolvarea problemei prezicerii dialectului nativ al unor texte, cunoscându-se traducerile acestora în diverse limbi. Problema este prezentată sub forma unui concurs ținut pe platforma online Kaggle.

Concursul constă în antrenarea unui clasificator pentru a prezice dialectul nativ al unui text dat (ce poate fi în engleză, scoțiană sau irlandeză), știind că acesta poate fi tradus într-una din următoarele cinci limbi: daneză, germană, spaniolă, italiană sau olandeză.

Datele de antrenare conțin 41570 de exemple de texte, etichetate cu dialectul original și având drept caracteristică limba în care au fost traduse, stocate sub forma unui fișier *.csv*. Scopul final este aplicarea unui algoritm de învățare automată asupra unor date de testare, 13860 de exemple, cu intenția de a prezice dialectul de proveniență pentru fiecare din acestea.

PROCEDEE UTILIZATE

În rezolvarea acestei probleme au fost utilizate două modele diferite de învățare automată supervizată: modelul mașinilor cu vectori suport (*SVM - Support Vector Machine*) și cel al celor mai apropiați k-vecini (*k-nearest neighbors*).

Înainte de aplicarea celor două modele menționate, textele traduse, atât cele de antrenare, cât și cele de testare, au fost preprocesate, filtrându-se elemente irelevante sau care ar putea influența negativ procesul de învățare. După această parte, conținutul textelor a fost trecut în format numeric prin intermediul modelului *bag-of-words*.

Pentru testarea eficientă a acurateții modelelor, dat fiind faptul că ele au fost aplicate pe un set de date de dimensiuni mici, a fost folosită maniera de împărțire a datelor *k-fold cross-validation*. Metoda presupune împărțirea datelor disponibile în k părți egale, selectarea a k-1 părți drept set de antrenare și folosirea părții rămase ca set de testare, procesul repetându-se de k ori. În final, metrica de evaluare a performanței modelului va fi reprezentată de media metricilor calculate pentru fiecare din cele k cazuri. În cadrul proiectului, datele au fost împărțite în 5 părți iar metrica utilizată este acuratețea.

MEDIUL DE LUCRU

Procedeele prezentate anterior au fost implementate în limbajul de programare Python, versiunea 3.10.7.

Atât pentru partea de citire și preprocesare a datelor, cât și pentru cea de învățare automată, au fost utilizate mai multe librării Python, cu precădere *pandas*, *ntlk* și *sklearn*.

PREPROCESAREA DATELOR

Pentru început, caracteristicile și etichetele, limba în care a fost tradus textul și dialectul de origine, au fost codificate în format numeric, atât pentru ușurința identificării lor pentru programator cât și pentru facilitarea manipulării lor de către algoritmii implementați.

Următorul pas este preprocesarea efectivă a textelor. În cadrul acesteia, au fost scoase toate semnele de punctuație și toate cifrele, urmând ca toate cuvintele să fie rescrise cu litere mici. Mai mult, au fost scoase și toate cuvintele de legătură din limba în care a fost tradus textul (cele conținute în pachetul *stopwords* din librăria *nltk.corpus*), iar fiecare cuvânt a fost trecut printr-un proces de *stemming*, prin care acesta este redus la rădăcina sa lingvistică.

MODELUL BAG-OF-WORDS

Modelul *bag-of-words* presupune construirea unui dicționar de cuvinte unice, extrase din toate cele 41570 de texte de antrenare, și codificarea fiecărui text în parte în format numeric, sub forma unui vector de caracteristici. Această codificare a fost făcută în funcție de frecvența de apariție a cuvintelor prezente în dicționarul creat, în cadrul fiecărui text. Dimensiunea efectivă a dicționarului i.e. numărul de cuvinte cheie extrase din toate textele de antrenare este dat ca hiperparametru modificabil.

Înainte de a se trece la partea de implementare a celor două modele, testarea lor folsind maniera *k-fold cross-validation* și aplicarea pe toate datele de antrenare, toți vectorii numerici rezultați în urma modelului *bag-of-words* au fost normalizați folosind norma Euclidiana L_2.

IMPLEMENTAREA MODELELOR

MAȘINI CU VECTORI SUPORT

Primul model implementat în scopul clasificării textelor este cel al mașinilor cu vectori suport (SVM). Modelul este un clasificator liniar nonprobabilist ce determină un hiperplan de separare între mulțimile de puncte de antrenare.

Din cauza faptului că cele trei mulțimi de texte aferente celor trei tipuri de etichete (dialectele engleză, scoțiană și irlandeză) nu pot fi separate liniar într-o manieră precisă, modelul implementează un SVM „soft-margin" i.e. găsește un hiperplan care separă datele perfect, dar misclasifică unele exemple de antrenare în funcție de un anumit cost (penalitate). Această penalitate este controlată de un hiperparametru C, care în cadrul problemei de optimizare a modelului determină cât de mult poate fi „încălcată" marginea hiperplanului de separare.

În mod normal, SVM este un clasificator binar, însă acesta poate fi folosit și în cazul clasificării multiclasă. Implementarea SVM folosită în cadrul proiectului, provenită din modulul *sklearn*, utilizează un SVM liniar configurat cu schema *one-versus-rest (OVR)* pentru clasificările cu clase multiple. Schema *one-versus-rest* antrenează 3 clasificatori, câte unul pentru fiecare etichetă, luând drept exemple pozitive cele inițiale și exemple negative cele date de reuniunea a celor n-1 clase rămase. În momentul inferenței, se rulează toți cei 3 clasificatori și se alege clasa pe baza clasificatorului cu marginea hiperplanului cea mai mare.

Pentru hiperparametrul C setat la 1, valoare ce maximizează marginea și minimizează penalitatea, iar numărul de cuvinte cheie al dicționarului setat la 5000, procesul de antrenare durează **3 minute și 12.4 secunde**. Metoda de testare *5-fold* returnează 5 metrici de acuratețe diferite, pentru fiecare *batch* de seturi antrenare/testare:

- 0.6861919653596343
- 0.6790955015636276
- 0.6851094539331248
- 0.6828241520327158
- 0.6841472215540053

Acuratețea medie a acestora este calculată ca fiind **0.6834736588886215**. Pe platforma Kaggle, acuratețea calculată pe 40% din datele de testere are o valoare rotunjită de **0.67460**.

Matricea de confuzie medie asociată metodei de testare, calculată pe baza matricelor de confuzie pentru fiecare din cele 5 *batch*-uri, folosind pachetul *confusion_matrix* din modulul *sklearn.metric*, este:

1146	**811**	**150**
388	**3843**	**309**
157	**816**	**693**

Pe coloane se află etichetele prezise, iar pe linii etichetele reale, pentru toate cele trei dialecte.

CEI MAI APROPIAȚI K-VECINI

Al doilea model implementat este cel al celor mai apropiați k-vecini (*k-nearest neighbors*). Acesta este un clasificator care prezice eticheta unui exemplu de test ca fiind eticheta predominantă a celor mai apropiate k exemple de antrenare din spațiul caracteristicilor.

Implementarea modelului, oferită de modulul *sklearn.neighbors*, calculează distanța Euclidiană (în realitate distanța Minkowski cu parametrul p setat la 2) între fiecare exemplu de testare și toate exemplele de antrenare. Pe baza hiperparametrului k, care semnifică numărul de „vecini” ai exemplului de testare, se aleg primele k exemple de antrenare care au distanța cea mai mică în raport cu exemplul de testare. Ultimul pas înainte de etichetare constă într-un proces de selecție ce alege eticheta care apare cel mai des în setul de exemple menționat anterior.

Pentru hiperparametrul k setat la 7 și numărul de cuvinte cheie al dicționarului setat la 5000, procesul de antrenare durează **3 minute și 39.6 secunde**. Metoda de testare *5-fold* returnează, la fel ca în cazul modelului anterior, 5 metrici de acurateț diferite:

- 0.5839547750781814
- 0.5656723598749098
- 0.5840750541255714
- 0.5828722636516719
- 0.6012749579023334

Acuratețea medie este calculată ca fiind **0.5835698821265336**. Pe platforma Kaggle, acuratețea calculată pe 40% din datele de testere are o valoare rotunjită de **0.57665**.

Matricea medie de confuzie asociată metodei de testare este:

981	**1079**	**48**
807	**3648**	**86**
385	**1059**	**223**

SURSE

Implementarea modelului *bag-of-words* provine din cadrul laboratorului 5 de inteligență artificială.

Informațiile legate de cele două clasificatoare au fost preluate din documentația oficială a modulului *scikit-learn*:

- https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html