Universitatea din Bucureşti
Facultatea de Matematică şi Informatică
Calculatoare şi Tehnologia Informaţiei

INTELIGENȚĂ ARTIFICIALĂ

COORDONATOR ȘTIINȚIFIC:
Prof. Dumitru Bogdan Alexe
Prof. Alexandra Diaconu

STUDENT:
Andrei-Laurențiu Cojocaru
Grupa 361

BUCUREȘTI

2022

Universitatea din Bucuresti
Facultatea de Matematică şi Informatică
Calculatoare şi Tehnologia Informatiei

Translation Source

Dialect Identification

COORDONATOR ŞTIINŢIFIC:
Prof. Dumitru Bogdan Alexe
Prof. Alexandra Diaconu

STUDENT:
Andrei-Laurenţiu Cojocaru
Grupa 361

BUCUREŞTI

2022

1. KNN - K-Nearest-Neighbors (Modelul celor mai apropiați k-vecini)

1.1. Caracteristici folosite în proiect:

- pentru preprocesarea datelor am definit funcția proceseaza(text) care are rolul de a returna textul prelucrat urmărind pașii:
  - elimină caracterele de tip „\n” cu funcția replace();
  - transformă literele mari în litere mici cu funcția lower();
  - elimină toate caracterele de tip cifre cu funcția isdigit();
  - elimină semnele de punctuatie cu functia translate();
  - împarte textul rezultat după spații;
  - în final returnează textul prelucrat;
- în continuare am realizat implementarea de la laborator de tip „Bag of Words” pentru evidențierea caracteristicilor.

1.2. Hiperparametrii modelului:

- hiperparametrul pentru acest model este k = 3301 (număr impar și nedivizibil cu 3 pentru a asigura un vor majoritar).

1.3. Cât durează antrenarea modelului

- antrenarea modelului durează aproximativ 5 - 6 minute.

1.4. Performanța obținută pe cele 40% de date din setul de date de test public de pe Kaggle este de 0.50649.

2. SVM – Support vector machines (Mașini cu vectori suport)

2.1. Caracteristici folosite în proiect

- funcția proceseaza(text) are rolul de a returna textul prelucrat urmărind pașii:
  - elimină caracterele de tip „\n” cu functia replace();
  - transformă literele mari în litere mici cu funcția lower();
  - elimină toate caracterele de tip cifre cu funcția isdigit();;
  - elimină toate cuvintele care se afla în lista de „stopwords” formată din cele 5 în limbi în care s-a tradus;
  - elimină semnele de punctuatie cu functia translate();
  - în final returnează textul prelucrat;
- functia „map” aplicată pe textul din datele de antrenare mapează textele și filtrează cuvintele din acestea cu funcția procesează;
- în continuare vom lematiza cuvintele cu ajutorul WordNetLemmatizer;
- după aceea vom contrui un pipeline care va executa următorii pași: CountVectorize(), TfidfTransformer() și svm.LinearSVC();
- primii 2 pași au rolul de a construi un vocabular (similar cu „Bag of Words”) și a reprezenta și balansa datele (cuvintele care se repeta excesiv nu vor influența în mod negativ predicțiile modelului) pentru a putea antrena eficient modelul.

2.2. Hiperparametrii modelului:

- hiperparametrul pentru acest model este C = 0.5.
- de asemenea, am folosit ca parametru și max_iter=5000 pentru a asigura convergența modelului

2.3. Cât durează antrenarea modelului

- antrenarea modelului durează 1 – 2 minute.

2.4. Performanța obținută pe cele 40% de date din setul de date de test public de pe Kaggle obținută este de 0.69967.

2.5. Rezultatele în urma antrenării în maniera 5 fold cross-validation

Rezultatele obținute pe modelul SVM în urma antrenării în maniera 5 fold cross-validation sunt: [0.6920134712533077, 0.6433004570603801, 0.7502285301900409, 0.7665864806350734, 0.7058936733221073].
Media acestora este 0,711604522
Matricea de confuzie este :
[[4114 745 787]
 [230 1230 125]
 [212 112 759]]

Etichetele în matrice au ordinea England, Ireland, Scotland.