Clasificarea cuvintelor in functie de complexitatea lor lexicala

Proiect Inteligenta Artificiala

Nicoi Alexandru
Grupa 353
Facultatea de Matematica-Informatica
Universitatea din Bucuresti

Cuprins

1 K-Nearest Neighbors

Metoda celor mai apropiati k-vecini reprezinta un model de invatare supervizata care prezice eticheta unui exemplu test ca fiind eticheta predominanta ale celor mai apropiate k exemple de antrenare din feature space.

Modelul k-NN nu invata in mod explicit un model insa, in realizarea de predictii, memoreaza si foloseste datele de antrenare

1.1 Caracteristici folositi

Pentru incercarea cu modelul k-NN, am pornit de la caracteristicile prezentate la laborator, la care am adaugat cateva care mi s-au parut relevante pentru cerinta.

- Numarul de silabe al cuvantului
- Verificare daca este in setul de date DALE CHALL
- Lungimea cuvantului
- Numarul de vocale
- Daca este titlu
- Numarul de sinonime al cuvantului
- Formula gunning fog pe fraza - $0.4[(\frac{\text{words}}{\text{sentences}}) + 100(\frac{\text{complex words}}{\text{words}})]$
- tipul de corpus din care face parte cuvantul

1.2 Hiperparametrii si antrenarea acestora

Hiperparametrii pe care i-am utilizat sunt

- n_neighbors - 5,7,9,11,13,15
- weights - uniform, distance
- metric - minkowski, euclidean, manhattan

Folosind metoda GridSearchCV din libraria sklearn.model_selection, am testat cum ar functiona fiecare parametru in parte.

```
grid_params = { 'n_neighbors' : [5,7,9,11,13,15],
                'weights' : ['uniform','distance'],
                'metric' : ['minkowski','euclidean','manhattan']}
gs = GridSearchCV(KNeighborsClassifier(), grid_params, verbose = 1, cv=3, n_jobs = -1)
gs.fit(X_train, y_train_compr)
print(gs.best_params_)
```

Cele mai bune rezultate obtinute au fost:

- n_neighbors - 13
- weights - uniform
- metric - manhattan

1.3 Durata antrenare

Durata de antrenare pentru k-NN este relativ scurta, avand un timp de executie de 6.71 secunde

1.4 Performanta pe kaggle

In aceasta etapa a proiectului, performanta pe k-NN pe setul publid de 40% a fost una destul de slaba, 0.59175, insa m-am acomodat mai bine cu acest domeniu.

2 Gaussian Naive Bayes

Clasificatorii Naive Bayes au la baza Teorema lui Bayes. Formula aplicata pentru cazul nostru am considerat-o astfel:

$$P(\text{Cuvant complex}|\text{Cuvant}) = \frac{P(\text{Cuvant}|\text{Complex}) * P(\text{Complex})}{P(\text{Cuvant})}$$

In incercarea unui model cu acuratete cat mai mare, am incercat fiecare tip de Naive Bayes:

- Multinominal Naive Bayes
- Bernoulli Naive Bayes
- Gaussian Naive Bayes

Cu toate trei am reusit sa imbunatatesc scorul obtinut, avand un progres de aproximativ 0.12 acuratete, ceea ce este o reusita depasind baseline-ul. Dintre toate cele trei, cel mai bine a performat cel Gaussian, care se diferentiaza de toate cele trei prin faptul ca lucreaza pe un set de date continuu de distributie gaussiana. In cadrul testelor, se putea observa fluctuatia acuratetei, uneori avand rezultate intre 0.70 si 0.78, astfel ca am luat in considerare sa incerc si alti algoritmi precum SVC (Support Vector Classification), sau MLP (MultiLayer Perceptron) insa nu am obtinut un scor mai performant.

2.1 Caracteristici folositi

Dupa experimentul de la k-NN, a fost nevoie de adaugarea a mai multor caracteristici pentru a spori sansele de reusita in predictia datelor. Pentru a obtine scorul cat mai mare, am decis adaugarea urmatoarelor feature-uri pe langa cele enuntate mai sus:

- Scor IDF dupa transformarea IDF.

  Avand in vedere faptul ca trebuie sa complexitatea lingvistica a cuvantului intr-un anumit context, este de la sine inteles ca trebuie facute caracteristici asupra frazei. In prima faza am incercat sa aplic modelul Bag-Of-Words insa nu mi-a oferit o schimbare in ceea ce priveste acuratetea. De aceea am inceput sa ma documentez si am gasit diferite tehnici de preprocesare a frazei, dand astfel de CountVectorizer si TFIDFTransformer. Dupa ce le-am incercat pe amandoua, am decis sa raman pe Tf-idf Transformer deoarece performa mai bine. Termenul TF provine de la numarul de aparintii al cuvantului in textul respectiv impartit la numarul de cuvinte din text, iar IDF reprezeinta logaritmul raportului dintre numarul total de texte si numarul de texte care contin acel cuvant. Folosind acest algoritm, am reusit sa obtin un scor care arata frecventa cuvantului in intreg corpusul, considerand cuvantul mai complex in functie de frecventa lui in text. Dupa realizarea transformarii, din cauza discrepantei majore intre date care ar fi daunat procesului de fitting, am normalizat datele.

- Numarul de consoane al cuvantului
- Valoarea morfologica (ca urmare a preprocesarii frazei, am eliminat stopwords-urile, caracterele numerice si semnele de punctuatie, ramanand cu patru valori morfologice - substantiv, adjectiv, verb si adverb)

- Frecventa cuvantului in limba engleza folosind libraria wordfreq
- Numarul de hypernyms al sinonimelor cuvantului
- Dale Chall Readibility Score pentru fraza folosind libraria textstat
- Dificultatea cuvantului folosind libraria textstat

2.2 Hiperparametrii si antrenarea acestora

Deoarece prin schimbarea parametrilor nu am facut decat sa scad acuratetea, am decis sa las functia cu parametri standard.

2.3 Durata antrenare

Durata de antrenare este de 6.45 secunde, putin mai rapid decat k-NN

2.4 Rezultatele pe 10 fold cross-validation si matricea de confuzie asociata

```
#   Numarul 1: 0.6853468208092486
#   Numarul 2: 0.757048561999057
#   Numarul 3: 0.7205911507614671
#   Numarul 4: 0.7401955089373142
#   Numarul 5: 0.6912238704691535
#   Numarul 6: 0.7471829981457709
#   Numarul 7: 0.7287382828347533
#   Numarul 8: 0.7547673531655226
#   Numarul 9: 0.7751680220148529
#   Numarul 10: 0.7047686496694996

#   SECUNDE: 36.21242880821228

#   SCOR MEDIU: 0.730503121880664

#   MATRICEA DE CONFUZIE ASOCIATA: [[5333 1579]
#                                   [ 232  518]]
```

2.5 Performanta pe kaggle

Cel mai bun rezultat pe care am reusit sa-l obtin pe datele publice (40%) a fost 0.77876), un progres foarte bun fata de k-NN.

3 Concluzie

Prin intermediul acestui proiect, am cunoscut ce inseamna sa te confrunti cu sine, unde atentia la detalii si la fiecare caracteristica trebuia sa fie la un nivel foarte ridicat pentru a nu strica acuratetea. In plus, consider ca am aprofundat si cunostintele despre procesarea limbajului natural, unde orice semnificatie lingvistica conteaza.