UNIVERSITATEA DIN BUCURESTI

FACULTATEA DE MATEMATICA SI INFORMATICA

DOCUMENTATIE

PROIECT INTELIGENTA ARTIFICIALA

„Automatic Misogyny Identification”

Coordonatori:

Conf.dr. Bogdan Alexe

Asist.(det.) drd. Sergiu Nisioi

Student:

Damian-Luca Rares

Gr. 352

Pre-procesarea datelor

In vederea obtinerii unor predictii cat mai precise, datele initiale trebuie sa treaca prin mai multe etape de pre-procesare pentru a se ajunge la setul de antrenare. (1)

Pasii urmatori au fost preluati de la utilizatorul Kaggle Rageeni Sah (2):

Initial am incarcat datele din train.csv, test.csv, si am plotat numarul de intrari cu eticheta 0 si 1, pentru a observa daca datele sunt balansate. Au fost preluate coloanele “id” si “text” si incarcate in dataframe-uri pandas. Dupa aceasta am generat trei imagin word cloud, pentru a vedea cele mai comune cuvinte din toate textele, textele cu eticheta 0 si textele cu eticheta 0, pentru a le elimina ulterior.

Urmeaza partea de data cleansing pentru inlaturarea partilor irelevante. (3). Au fost definite functii pentru fiecare pas si apoi aplicate coloanelor de text respective utilizand functii lambda pentru a crea o noua coloana. Aceste functii au rolul de a:

- Inlatura punctuatia- def remove_punct(text):
- Tokenizarea textului- def tokenization(text):
- Inlaturarea cuvintelor de oprire- def remove_stopwords(text):
- Stemming (aducerea la radacina)- def stemming(text):

Corpus-ul, reprezentand textul trecut prin toti pasii de cleansing si avand forma de array de cuvinte , este utilizat pentru a obtine vocabularul. Urmeaza obtinerea dictionarului wd2idx (atribuirea unui index primelor n cele mai comune cuvinte) si formarea modelului bag of words (textul fiind reprezentat ca un multiset alcatuit din propriile cuvinte, pastrandu-se multiplicitatea) (4)

Submisia nr. 1 (MultinomialNB)

Metodele Naive Bayes sunt un set de algoritmi de invatare supervizati bazati pe aplicarea teoremei lui Bayes cu presupunerea de independenta conditionala intre fiecare pereche de caracteristici fiind data valoarea variabilei clasa. (5)

Clasificarea tweet-urilor:

- X_1, X_2, ..., X_{100} sunt aparitiile celor 100 cele mai des intalnite cuvinte
- $c \in \{0, 1\}$ (clasifica daca textul este misogin sau nu)

Teorema lui bayes: $P(c|X) = P(X|c) \times P(c) / P(X)$

$P(c|X)$ - probabilitatea sa avem textul din clasa c dandu-se reprezentarea bag of words X

$P(X|c)$ - probabilitatea sa avem reprezezentarea bag of words X stiind ca textul apartine clasei c

$P(c)$ – probabilitatea a-priori

$P(X)$ – probabilitatea sa observam reprezentarea bag of words X

$P(c = 0|X) = P(X|c = 0) \times P(c = 0) / P(X)$

$P(c = 1|X) = P(X|c = 1) \times P(c = 1) / P(X)$

Intrucat avem numitori egali, putem renunta la acestia

$P(c = 0|X)$ d.p. $P(X|c = 0) \times P(c = 0)$

$P(c = 1|X)$ d.p. $P(X|c = 1) \times P(c = 1)$

Regula de clasificare:

$$c^* = \text{argmax}\, P(X \mid c = i) \times P(c = i)$$

Aceasta alege clasa care maximizeaza numaratorul

Probabilitatea a-priori $P(c = i)$ se calculeaza numarand cate exemple din multimea de antrenare au clasa i.

Pobabilitatea likelihood $P(X|c = i)$ se calculeaza considerand caracteristicile independente din cauza numarlui limitat de date antrenare.

$P(X|c = i) = P(X_1=x_1, X_2=x_2, ..., X_{100}=x_{100} \mid c = i) =$

$= P(X_1 = x_1 \mid c = i) \times P(X_2 = x_2 \mid c = i) \times ... \times P(X_{100} = x_{100} \mid c = i)$

Regula de clasificare pentru naive Bayes:

$$c^* = \text{argmax}(\prod P(X_j = x_j \mid c = i)) \times P(c = i) \quad (6)$$

Am utilizat clasificatorul MultinomialNB din libraria scikit-learn, o varianta des folosita pentru clasificarea textelor. Acesta implementeaza algoritmul naive Bayes pentru date distribuite multinomial. Repartizarea este parametrizata de vectori $\theta_c = (\theta_{c1}, ..., \theta_{cn})$ pentru fiecare clasa c, unde n este dimensiunea vocabularului si θci este probabilitatea $P(x_i|y)$ a caracteristicii i de a aparea intr-o mostra apartinand clasei c. (7)

Pentru antrenare am iterat cu ajutorul unui for valori pentru alfa (utilizat pentru caracteristici absente in datele de invatare). Clasificatorul este initializat la fiecare iteratie cu o noua valoare pentru alfa si este initalizat un vector gol pentru a ajuta la calculul scorului f1 mediu. Apoi au loc 1000 de antrenari, preziceri si calcule de scor f1 10-fold cross-validate. Timpul de antrenare este de 490 secunde, adica 8 min 10 s . Scorul f1 obtinut pe Kaggle pe 40% din date este 0.72351.

Date 10 fold cross-validation si matrice de confuzie:

```
10 fold cv si matrice de confuzie
acc: 84.5 f1:  0.8319407426349429
Matricea de confuzie:
1919   357
 418   2306
```

Submisia nr. 2 (KNN)

Algoritmul de invatare supervizata a vecinilor cei mai apropiati pe date cu etichete discrete rezolva probleme de clasificare. Principiul de functionare este gasirea unui numar predefinit de probe de antrenament cat mai apropiate de punctul nou si prezicerea etichetei pe baza acestora. Distanta poate fi orice metrica, iar in general este folosita distanta Euclidiana. Acest clasificator este bazat pe instanta, adica nu construieste un model intern generalizat, ci retine instantele datelor de antrenare. (8)

Textul a fost tokenizat utilizand nltk.WordPunctTokenizer().tokenize(). Datele de train au fost impartite in antrenare si validare cu un split random de 30%. Pentru antrenare am iterat cu ajutorul unui for valori pentru n_neighbours. Clasificatorul este initializat la fiecare iteratie cu un numar de vecini nou, apoi are loc antrenarea, prezicerea si calcularea acuratetii si scorului f1. Aceste ultime doua valori calculate, impreuna cu numarul vecinilor din for sunt adaugate in vectori pentru a putea fi folosite la realizarea unor grafice. Acestea ajuta la descoperirea valorilor celor mai avantajoase pentru numarul de vecini. Antrenarea dureaza 6.82 secunde. Scorul f1 obtinut pe Kaggle pe 40% din date este 0.70613.

Bibliografie

1. https://en.wikipedia.org/wiki/Data_pre-processing
2. https://www.kaggle.com/ragnisah/text-data-cleaning-tweets-analysis)
3. https://en.wikipedia.org/wiki/Data_cleansing
4. https://en.wikipedia.org/wiki/Bag-of-words_model
5. https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes
6. Cursul 4 de IA - Conf.dr. Bogdan Alexe
7. https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes
8. https://scikit-learn.org/stable/modules/neighbors.html#neighbors