Ivan Oana
Grupa 241

DOCUMENTAȚIE PROIECT INTELIGENȚĂ ARTIFICIALĂ
Romanian sub-dialect identification using SVM

Cerința proiectului este de a diferenția între dialectul românesc și cel moldovenesc dintr-un fișier care conține datele de test pe baza unor date deja etichetate folosite pentru antrenare.

Pentru acest proiect am folosit următoarele biblioteci:

- numpy
- pandas pentru salvarea in fisier cvs a rezultatului
- sklearn pentru preprocesarea datelor, utilizarea svm și calcularea f1_score
- re pentru separare test

Conținutul fișierelor se salvează în listele: trainingData ("train_samples.txt"), trainingLabels ("train_labels.txt"), validationData ("validation_samples.txt"), validationLabels ("validation_labels.txt") și testData ("test_samples.txt").

Proiectul a fost abordat folosind metoda Bag of Words.
Se crează lista allWords care o să conțină toate cuvintele distincte care se găsesc în datele de antrenare (trainingData). Se ignoră primul cuvânt de pe fiecare rând pentru că acesta reprezintă eticheta rândului respectiv.

```
allWords = []
for line in trainingData:
  words = line.split()
  for i, w in enumerate(words):
    if i != 0 and w not in allWords:
      allWords.append(w)
```

Pentru fiecare dintre cele 3 seturi de date se creaza o matrice xFeatures (x∈ {train, validation, test) care inițial are toate valorile egale cu zero. Numărul de coloane este egal cu numărul de cuvinte distincte din setul de date de antrenare, iar fiecare linie a matricei corespunde unei linii din respectivul set de date. Matricea ține evidența frecvențelor cuvintelor în fiecare linie din setul de date. Aceasta se obține utilizând funcția următoare:

```
def getFeatures(data):
   features = np.zeros((len(data), noOfWords))
   for i, document in enumerate(data):
     for word in document:
       if word in allWords:
         features[i, np.where(allWords == word)[0][0]] += 1
   return features
```

Datele obținute se normalizează prin utilizare lui preprocessing din biblioteca sklearn.

```
scaler = preprocessing.Normalizer(norm = 'l2')
if scaler is not None:
  scaler.fit(trainFeatures)
  scaledTrainFeatures = scaler.transform(trainFeatures)
  scaledValidationFeatures = scaler.transform(validationFeatures)
  scaledTestFeatures = scaler.transform(testFeatures)
else:
  scaledTrainFeatures = trainFeatures
  scaledValidationFeatures = validationFeatures
  scaledTestFeatures = testFeatures
```

În continuare pentru rezolvarea problemei, a fost folosit modelul vectorilor suport(svm) utilizând biblioteca sklearn. S-au folosit 2 valori pentru C, C = 100 și C = 3.

```
svmModel = svm.SVC(C = 100, kernel = 'linear')
# Antrenarea pe trainingData
svmModel.fit(scaledTrainFeatures, trainingLabels)
```

Pentru verificara modelului, acesta se aplică prima dată pe datele de validare pentru a se verifica acuratețea modelului.

```
# Aplicarea modelului pe datele de validare
predictedLabelsValidation = svmModel.predict(scaledValidationFeatures)
```

Matricea de confuzie rezultată are forma următoare:

$$\begin{bmatrix} 668 & 633 \\ 661 & 694 \end{bmatrix}$$

Nr de cazuri din clasa 0 pe care modelul le – a clasificat ca fiind din clasa 0 668	Nr de cazuri din clasa 0 pe care modelul le – a clasificat ca fiind din clasa 1 633
Nr de cazuri din clasa 1 pe care modelul le – a clasificat ca fiind din clasa 0 661	Nr de cazuri din clasa 1 pe care modelul le – a clasificat ca fiind din clasa 1 694

Iar scorul F1, dat de formula $2 * \frac{precision * recall}{precision + recall}$ și este egal cu 0.5175242356450409, deci aproximativ egal cu 0,52.

Apoi acest model se aplică pe datele de testare.

```
# Aplicarea modelului pe datele de testare
predictedLabelsSvm = svmModel.predict(scaledTestFeatures)
```

Pentru salvarea predicțiilor într-un fișier separat se folosește biblioteca pandas. "testLab" conține id-urile fiecărei predicții, iar "pred" conține predicțiile efective.

```
submission = pd.DataFrame({'id': testLab,'label': pred})
name = 'submission.csv'
submission.to_csv(name, index = False)
```

Pentru realizarea acestui proiect m-am folosit de informațiile și rezolvarea exercițiilor din cadrul laboratorului 5(https://fmi-unibuc-ia.github.io/ia/Laboratoare/solutie_lab5.zip).