TÜRKÇE MAKALELERİN SINIFLANDIRILMASI

MURAT DEMİR
Ö19060012
26.12.2019

K-Nearest Neighbor Algoritması Hakkında Kısa Bilgi

KNN algoritması Machine learning'deki en kolay algoritmalardan birisidir ve bir Supervised Learning algoritmasıdır. Çoğunlukla classification amacıyla kullanılır. "K" harfi komşu sayısını ifade eder yani k tane en yakın komşu algoritması olarak Türkçe'ye çevirebiliriz.

TÜRKÇE MAKALELERİN SINIFLANDIRILMASI PROJESİ

Projemiz 4 Makale Türünü Makine Öğrenmesi Yöntemlerinden K-Nearest Neighbor Algoritması ile Tasarlanmıştır. Makale türlerimiz aşağıda verilmiştir.

- Popüler Kültür
- Otomobil
- Müzik
- Ekonomi

Ide olarak Eclipse kullanılarak gerçeklenmiştir.

Başta Zembereğin Programa Entegresi Aşağıdaki gibi sağlanmıştır.

```
jvmDLLpath = r"C:\Program Files\Java\jdk-13\bin\server\jvm.dll"
jpype.startJVM(jvmDLLpath,"-Djava.class.path=zemberek-tum-2.0.jar", "-ea")

TR = jpype.JClass("net.zemberek.tr.yapi.TurkiyeTurkcesi")
tr = TR()

Z = jpype.JClass("net.zemberek.erisim.Zemberek")
z = Z(tr)
```

Makalelerde geçen kelimelerin köküne inilip stopwords 'lerden kurtarılmıştır.

```
def Zemberek(makale):
    yenidepo = []
    makale = makale.split()
    for line in makale:
        try:
            kok = z.kelimeCozumle(line)[0].kok().icerik()
            if kok:
                yenidepo.append(kok)
        except:
            pass

    return yenidepo

egitim = []
file = open("Makaleler.txt")
for satır in file:
    satır = satır.strip()
    temizle = Zemberek(satır)
    temizle = ' '.join(temizle)
    egitim.append(temizle)
```

Eğitim Makaleler.txt belgesinin okunup ilk 57 satır içerisinde bulunan makalenin 0. İndex değerine ,57-119. Satırlar arasında kalan makalenin 1. İndex değerine, 119-159. Satırlar arasında kalan makalenin 2. İndex değerine ve 159-218. Satırlar arasında kalan makalenin 3. İndex değerine atanması sağlanıp sırasıyla

0.index = Popüler Kültür

1.index = Otomobil

2.index = Müzik

3.index = Ekonomi

Alanlarına atanıp **KNN** algoritması ile eğitilmiştir.

```
vectorizer = TfidfVectorizer(min_df = 0., max_df = 1., use_idf = True)
egitim = vectorizer.fit_transform(egitim)

egitim_y = np.zeros(218)
egitim_y[57:119] = 1
egitim_y[119:159] = 2
egitim_y[159:218] = 3

mkneighbors = KNeighborsClassifier(n_neighbors=15)
mkneighbors.fit(egitim,egitim_y)
DogruEtiketler = ['Popüler Kültür','Otomobil', 'Müzik/Piyano', 'Ekonomi']
```

Ardından TestDokumanlari.txt Dökümanına gireceğimiz Algoritmanın daha önce görmediği yeni makalenin hangi sınıfa dahil edileceği K-Nearest Neighbor algoritması ile hesaplanmış ve programın sağlıklı bir şekilde sonlandırılması sağlanmıştır

```
Dokumani_Temizle = []
for line in testDokumani:
    line = line.strip()
    temizle = Zemberek(line)
    temizle = ' '.join(temizle)
    Dokumani_Temizle.append(temizle)
Test_Dokumani = vectorizer.transform(Dokumani_Temizle)

YuklemEtiketleriKneigbors = mkneighbors.predict(Test_Dokumani)

YuklemEtiketleriKneigbors = YuklemEtiketleriKneigbors.astype(int)
YuklemEtiketleriKneigbors = ''.join(str(v) for v in YuklemEtiketleriKneigbors)
sayac = 0
for i in YuklemEtiketleriKneigbors:
    MevcutSayac = YuklemEtiketleriKneigbors.count(i)
    if(MevcutSayac > sayac):
        sayac = MevcutSayac
        Etiket = i

print("\nDokümanın sinifi: ", DogruEtiketler[np.int(Etiket)], "\n\n\n\n")

jpype.shutdownJVM()
```