# CENG3528: WEB MINING

## Aşk 101 Sentiment Analysis

# Final Project Report

## 03/06/2020

**Team Members:**

Burak Can Onarım --- 150709032
Sümeyra Özuğur --- 170709046

**Description of Project:**

We made sentiment analysis from EkşiSözlük via web crawling which is created by us for Aşk 101 series on Netflix.

**Our purpose;**

To learn the feelings about the Aşk 101 series, so helping the next season.

Let's begin to explain our project step by step.

**Web Crawling:**

Simply, provides to get data from many pages.

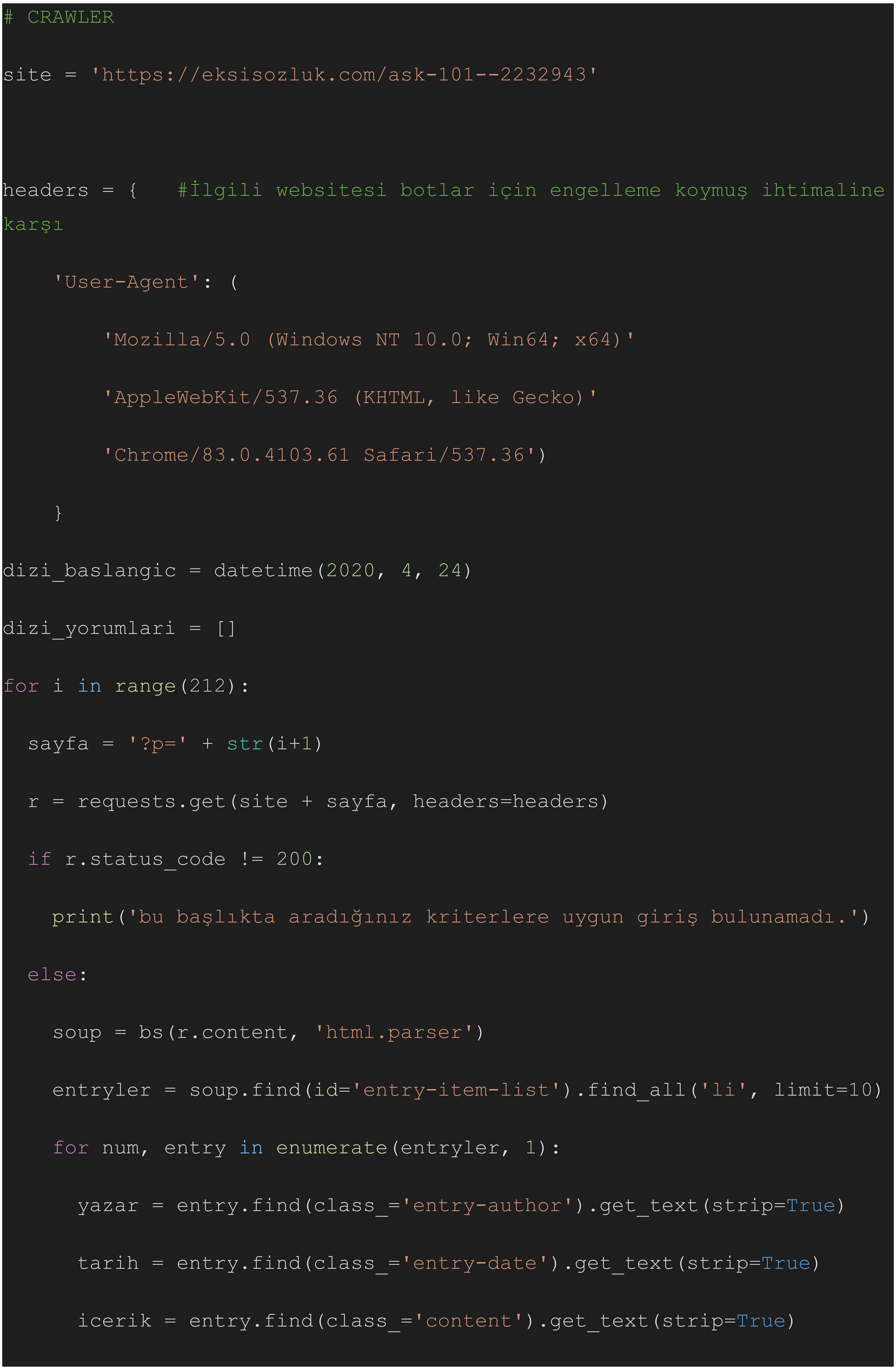
There are multiple libraries and frameworks we can use for crawling. To count some of them;

- LXML
- Selenium
- Requests
- Mechanize
- Beautiful Soup 4
- Scrapy

We used Requests, Beautiful Soup 4 in our project.

Requests model allows us to manage our web requests. We use it when we want to throw requests like “get”, “ post”.

Beautiful Soup is a Python library used to extract data from HTML and XML files. It only takes the content of the URL you provided and then stops. We added a for loop to continue.



```
# CRAWLER
site = 'https://eksisozluk.com/ask-101--2232943'

headers = {   #İlgili websitesi botlar için engelleme koymuş ihtimaline karşı
    'User-Agent': (
        'Mozilla/5.0 (Windows NT 10.0; Win64; x64)'
        'AppleWebKit/537.36 (KHTML, like Gecko)'
        'Chrome/83.0.4103.61 Safari/537.36')
    }
dizi_baslangic = datetime(2020, 4, 24)
dizi_yorumlari = []
for i in range(212):
  sayfa = '?p=' + str(i+1)
  r = requests.get(site + sayfa, headers=headers)
  if r.status_code != 200:
    print('bu başlıkta aradığınız kriterlere uygun giriş bulunamadı.')
  else:
    soup = bs(r.content, 'html.parser')
    entryler = soup.find(id='entry-item-list').find_all('li', limit=10)
    for num, entry in enumerate(entryler, 1):
      yazar = entry.find(class_='entry-author').get_text(strip=True)
      tarih = entry.find(class_='entry-date').get_text(strip=True)
      icerik = entry.find(class_='content').get_text(strip=True)
      if '~' in tarih: # güncellenen yorumları almak için
        zamanlar = tarih.split(' ~ ')
        if '.' in zamanlar[1]:
          tarih_object = datetime.strptime(zamanlar[1], '%d.%m.%Y %H:%M')
          if tarih_object > dizi_baslangic:
            dizi_yorumlari.append(icerik)
          else:
            continue
        else:
          tarih_object = datetime.strptime(zamanlar[0], '%d.%m.%Y %H:%M')
          if tarih_object > dizi_baslangic:
            dizi_yorumlari.append(icerik)
          else:
            continue
      else:
        tarih_object = datetime.strptime(tarih, '%d.%m.%Y %H:%M')
        if tarih_object > dizi_baslangic:
          dizi_yorumlari.append(icerik)
        else:
          continue
```

**Briefly in this code,**

We got the link of Aşk 101 series in the EkşiSözlük . Then, we created the start date of the series since April 4, 2020 there are 212 pages in the EkşiSözlük. Sometimes comments are updated in EkşiSözlük. In this case we got the most recent comments.

**Sentiment Analysis**

After taking comments from EkşiSözlük and then we printed it to a txt file. Then we started the sentiment analysis.

We used artificial neural networks for sentiment analysis.By the way;

Neural Network:

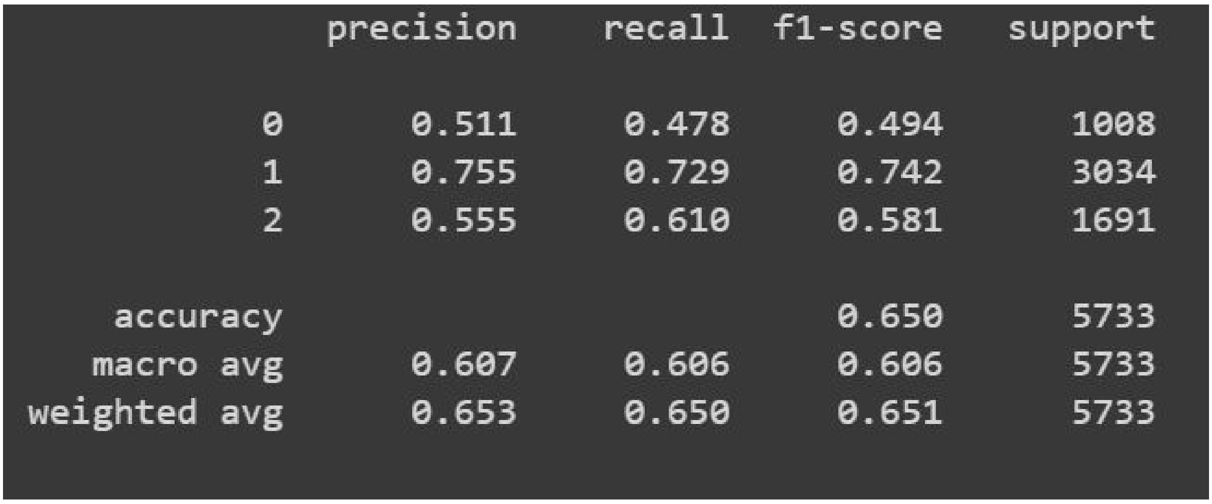
Artificial neural networks are an information technology developed by inspiring the information processing technique of the human brain. Artificial Neural Networks mimic the way the simple biological nervous system works. It is the digital modeling of biological neuron cells and the synaptic bond that these cells establish with each other.

Neural Network is a structure built in layers. The first layer input is called the last layer output. The middle layers are called hidden layers. Each layer contains a certain number of 'Neuron'. These neurons are connected to each other with synapse.

Synapses contain a coefficient. These coefficients say how important the information in the neuron to which they are connected is. The value of a neuron is found by multiplying inputs with coefficients and then adding them to them. This found result is put into an activation function. According to the result of the function, it is decided whether or not that neuron will be fired.

We downloaded test.json, train.json, validation.json from github and trained and tested this data.

These are the results;



```
              precision    recall  f1-score   support

           0      0.511     0.478     0.494      1008
           1      0.755     0.729     0.742      3034
           2      0.555     0.610     0.581      1691

    accuracy                          0.650      5733
   macro avg      0.607     0.606     0.606      5733
weighted avg      0.653     0.650     0.651      5733
```

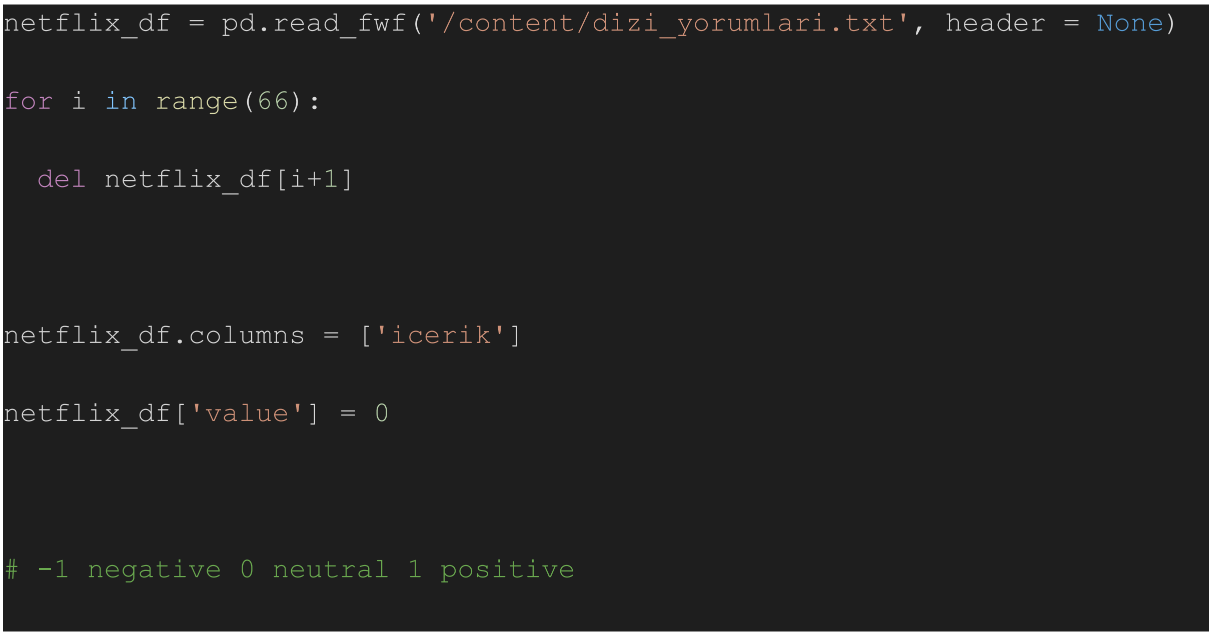
negative comment:0,     accuracy for negative comment  → 0,478

neutral comment :1,     accuracy for neutral comment  → 0,729

positive comment:2     accuracy for positive comment  → 0,610

The average accuracy of  model which is used  is 0,606.

### Now  Test for Aşk 101



```
netflix_df = pd.read_fwf('/content/dizi_yorumlari.txt', header = None)

for i in range(66):

  del netflix_df[i+1]



netflix_df.columns = ['icerik']

netflix_df['value'] = 0



# -1 negative 0 neutral 1 positive
```

```
for idx, row in tqdm(netflix_df.iterrows()):
  X = feature_extraction(row['icerik'])
  netflix_df.at[idx, 'value'] = np.argmax(
      model.predict(np.array(X).reshape(1, -1)))-1
count_row = netflix_df.shape[0]
count_pozitif = 0
count_negatif = 0
count_notr = 0

for i in range(count_row):
  if netflix_df['value'][i] == -1:
    count_negatif += 1
  elif netflix_df['value'][i] == 0:
    count_notr += 1
  elif netflix_df['value'][i] == 1:
    count_pozitif += 1

begeni = count_pozitif - count_negatif
if begeni > 0:
  print("Ekşi sözlüğe göre Aşk101: FEVKALADENİN FEVKİNDE")
elif begeni == 0:
  print("Ekşi sözlüğe göre Aşk101: BANA NE DİZİDEN")
else:
  print("Ekşi sözlüğe göre Aşk101: BUNDAN KÖTÜSÜNÜ ANCAK AYNI KİŞİLER YAPABİLİR")



print(begeni, 'entry farkıyla bu dizinin beğenildiği sonucuna varılmıştır.')

print(count_negatif, 'entry içerisinde dizinin berbat olduğu belirtilmiştir.')

print(count_notr, 'entry içerisinde dizi umursanmamıştır.')

print(count_pozitif, 'entry içerisinde dizinin fevkalade olduğu belirtilmiştir.')
```

## Conclusion:

In conclusion,we used 1959 entry for test;
In 435 entry, the series was reported to be bad.
In 462 entry, the series was reported to be nötr.
In 1062 entry, the series was reported to be perfect.

It is concluded that this series is liked by 627 entry difference.

### Distribution of Tasks:

We did it all together by zoom connection.We researched, wrote code, found errors and finally we wrote the report.