# *Web İndeksleme Uygulaması*

*Emirhan EKŞİ,Erdem ÖZOĞLU*

180202079,180202094

emiirhaneksi@gmail.com , erdemozoglu@gmail.com

Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü
Yazılım Lab.2 – Proje 1

## 1.Giriş

Web indeksleme uygulamasında hedeflenen; web sitelerinin içeriğinin alınması ve bu siteler arasındaki benzerlik oranını bularak siteler arasındaki ilişkiyi yakalamaktır.

## 2.Özet

Web indeksleme uygulamasını özetlemek gerekirse bu uygulama bir ‘web scraping’ uygulamasıdır.Bir diğer açıdan bakarsak: çeşitli web sitelerinin içeriğini kullanarak elimizdeki verilerle anlamlı bilgi üretmeyi hedeflediğimiz için bu uygulama bir çeşit,başlangıç düzeyinde ‘data science’ uygulaması sayılabilir.

**Web Scraping Nedir?**

Veri kazıma, genel anlamı ile, bir bilgisayar programının bir veri kaynağından anlamlı veri çıkarma işlemini ifade eder.

**Data Science Nedir?**

Veri bilimi, yapılandırılmış ve yapılandırılmamış verilerden bilgi ve öngörü elde etmek için bilimsel yöntemleri, süreçleri, algoritmaları ve sistemleri kullanan çok disiplinli bir alandır. Veri bilimi veri madenciliği ve büyük verilerle ilişkilidir.

## 3.Yöntem

### 3.1 Proje geliştirme ortamı

Projemiz Python Programlama dilinde yazılmıştır.Geliştirme ortamı olarak JupyterLab ve VS Code kullanılmıştır.

#### 3.1.1 Neden Python Kullandık?

Python,Flask ve Beautiful Soup gibi frameworklere sahip olması sebebiyle işimizi kolaylaştıracağını düşündük.BeautifulSoup HTML sayfalarından veri ayıklamak için özelleştirilmiş bir kütüphanedir.Flask ise HTML formlarından alınan bilginin Python programlada kullanılacağı güzel bir kütüphane.Bu iki framework sayesinde Python kullanmaya karar verdik.

### 3.2 Veri Kazıma Yöntemi

Web sitelerinden veri kazımak için kullandığımız yöntem BeautifulSoup kütüphanesinin fonksiyonları oldu.

```
url1 = "https://www.w3schools.com/html/html_intro.asp"

r = requests.get(url1)
source = BeautifulSoup(r.content,"html.parser")
metin = source.find("body").text
```

Yukarıdaki örneği inceleyecek olursak elimizde bir url var.Bu url’i request.get ile alıyoruz.Daha sonra HTML içinde body etiketinin içeriğini alıp text fonksiyonu ile etiketlerden ayıklayıp ‘metin’ isimli str’nin içine atıyoruz.

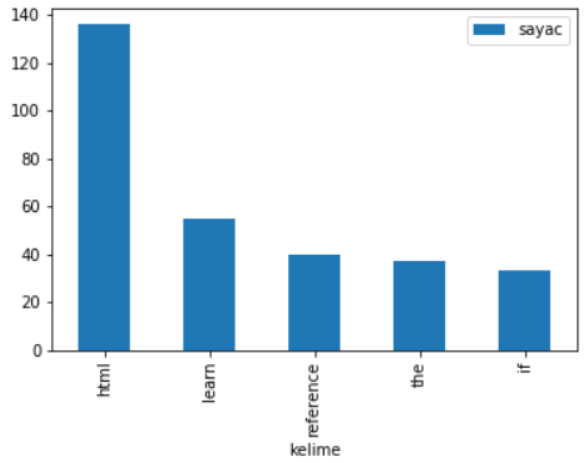
### 3.3 Kelime Frekans Bulma Yöntemi

Web sitesinin HTML etiketlerinden body’nin bütün içeriğini ‘metin’ stringine atadıktan sonra for döngüsü ile hangi kelimeden kaç tane olduğunu saydırıyoruz.Ancak burada dikkat edilmesi gereken şey noktalama işaretlerini ayırmak ve bütün kelimeleri küçük harfe çevirmek.Bu işlemleri de ‘replace’ ve ‘lower’ fonksiyonları sayesinde yaptık.

### 3.4 Anahtar Kelime Bulma Yöntemi

Anahtar kelimeleri bulurken sayfada en çok tekrar eden 5 kelimeyi almayı tercih ettik.Bu aşamada en çok tekrarlanan kelimeler üzerinden gitmenin daha anlamlı sonuçlar vereceğini düşündük.Bu aşamada string'i dataframe'e çevirmek için Pandas kütüphanesinden;veri setini görselleştirmek için Matplotlib kütüphanesinden faydalandık.

```
import matplotlib.pyplot as plt
kw1.plot(x ='kelime', y='sayac', kind = 'bar')
plt.show()
```




### 3.5 Benzerlik Oranı Bulma Yöntemi

```
for x in array2:
    if (list[0]==x) | (list[1]==x) | (list[2]==x) | (list[3]==x) | (list[4]==x) :
        sayac = sayac+1

print(sayac)
```

```
334
```

```
#Formülümüz : Anahtar kelimelerin ikinci sayfada kaç kere geçtiği
#                / ikinci sayfadaki kelime sayısı * 100
```

```
oran= (sayac/len(array2))*100
sayac = 0
```

```
print("Benzerlik oranı : %",oran)
```

```
Benzerlik oranı : % 11.645746164574616
```

Yukarıdaki görselde birinci url'deki anahtar kelimeler,ikinci url'deki bütün kelimelerle tek tek kıyaslanır.Aynı olursa sayaç arttırılır.Bu sayaç "Anahtar kelimelerin ikinci sayfada kaç kere geçtiği" bilgisini tutar.Daha sonra sayac verisi yukarıdaki formülde yerine konulur ve benzerlik oranı bulunur.

## 4.Yalancı Kod

1. Başla.
2. URL al.
3. URL'e GET isteği yap.
4. HTML kodlarını al.
5. Body etiketinin içeriğini text fonksiyonu ile temizle.
6. Text içeriğini noktalama işaretlerinden arındır.
7. Text içeriğinin tamamını küçük harfe çevir.
8. For döngüsü ile kelimeleri saydır.
9. En çok tekrar eden 5 kelimeyi bir listeye gönder.
10. En çok tekrar edenler listesini yazdır.
11. İkinci URL için 1.adıma git.
12. En çok tekrar eden 5 kelime ile ikinci URL içeriğini kıyasla.
13. Aynı olan her kelime için sayacı bir artır.
14. Benzerlik oranını hesapla.
15. Benzerlik oranını yazdır.
16. Bitir.

## 5.Sonuç

Sonuç olarak bu projenin bize kazandırdıkları;

- Python Programlama dili ile proje geliştirmek
- Web Scraping uygulaması geliştirmek
- Temel düzeyde Data Science uygulaması geliştirmek
- Dinamik bir web arayüzü oluşturmak.

# 6.Kaynakça


- https://www.sinanerdinc.com/python-beautifulsoup-modulu
  **erişim tarihi : 30.03.2021**
- https://tr.wikipedia.org/wiki/Veri_bilimi
  **erişim tarihi : 30.03.2021**
- https://ceaksan.com/tr/veri-kazima-data-scraping-nedir
  **erişim tarihi : 30.03.2021**
- https://makdos.blog/python/393/python-flask-ile-ornek-request-sorgulari-form-ve-cerez-incelemesi/
  **erişim tarihi : 30.03.2021**
- https://www.quora.com/How-do-I-get-input-values-from-HTML-page-to-Python-in-a-server
  **erişim tarihi : 30.03.2021**
- https://medium.com/kodlayan-nesil/flask-nedir-9364c1bb5f41
  **erişim tarihi : 30.03.2021**
- https://medium.com/bili%C5%9Fim-hareketi/python-beautifulsoup-k%C3%BCt%C3%BCphanesi-ile-veri-kaz%C4%B1ma-e8076c7212a9
  **erişim tarihi : 30.03.2021**
- https://medium.com/kodluyoruz/python-beautiful-soup-k%C3%BCt%C3%BCphanesi-ile-i%CC%87nternetten-veri-%C3%A7ekme-1d5977c7e677
  **erişim tarihi : 30.03.2021**