**Yazılım Geliştirme Laboratuvarı – 1**

**Grup 13**

**Proje Linki :**

[*https://github.com/x-x00/yazlab-1*](https://github.com/x-x00/yazlab-1)

***Veri Linki:***

<https://drive.google.com/file/d/1OOgX5hOHI6_ZeTWsouNzOCKsyssPX3aQ/view?usp=sharing>

*Kaan Albayrak ( 211307020 )*

Bilişim Sistemleri Mühendisliği / Teknoloji Fakültesi

Kocaeli Üniversitesi

[kaannalbayrak@gmail.com](mailto:kaannalbayrak@gmail.com)

# Özet

*İngilizce aksan tespiti için oluşturulmuş crawler’ la ilgili verilerin toplanması ve daha sonra bu verilerin pre-processing işleminden geçirilip işlenmeye hazır duruma getirilmesi.*

# 

# ****1. Web Crawler****

Web tarayıcısı, crawler veya web örümceği, internet üzerinden web sitesi içeriğini ve diğer bilgileri aramak ve otomatik olarak index' lemek için kullanılan bir bilgisayar programıdır. Bu programlar veya botlar, en yaygın olarak bir arama motoru index' leme için girdi oluşturmada kullanılır.

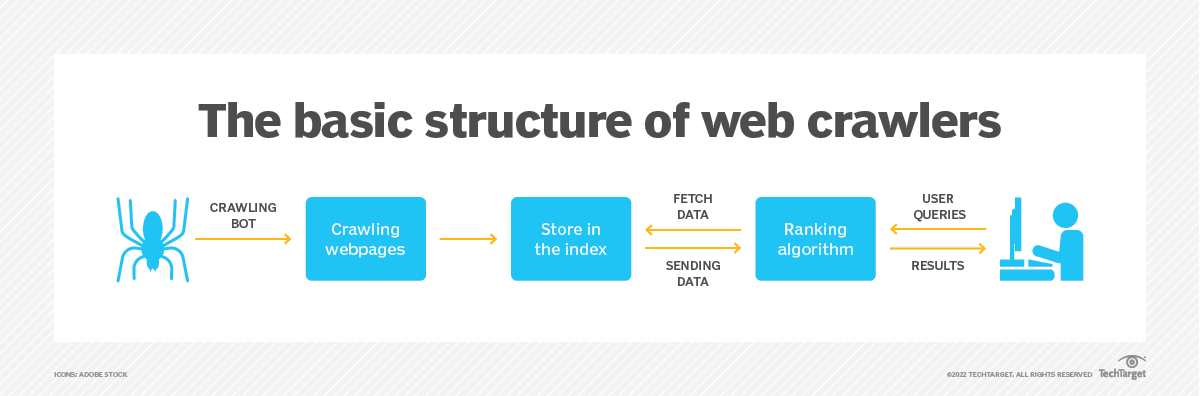
Web tarayıcıları, web sitesindeki her sayfanın ne içerdiğini öğrenmek için web sayfalarını sistematik olarak tarar, böylece bu bilgiler indekslenebilir, eklenebilir, güncellenebilir ve bir kullanıcı arama sorgusu yaptığında alınabilir. Diğer web siteleri kendi web içeriklerini güncellerken web crawling botlarını kullanır.

Google veya Bing gibi arama motorlari, kullanıcı aramalarına yanıt olarak ilgili bilgileri ve web sitelerini görüntülemek için web tarayıcıları tarafından toplanan verilere bir arama algoritması uygular.

Bir kuruluş veya web sitesi sahibi, web sitesinin bir arama motorunda listelenmesini istiyorsa, önce indekslenmesi gerekir.

Web sayfaları taranmaz ve indekslenmezse, arama motoru bunları kolay kolay bulamaz. Web tarayıcıları bilinen sayfalardan oluşan belirli bir kümeyi taramaya başlar, ardından bu sayfalardan yeni sayfalara giden bağlantıları takip eder. Arama motorları tarafından taranmak veya bulunmak istemeyen web siteleri, robots.txt dosyası gibi araçlar kullanarak botların bir web sitesini indekslememesini veya yalnızca bazı bölümlerini indekslemesini isteyebilir.

Bir tarama aracıyla site denetimleri gerçekleştirmek, web sitesi sahiplerinin kırık bağlantıları, yinelenen içeriği ve yinelenen, eksik veya çok uzun ya da kısa başlıkları belirlemelerine yardımcı olabilir.



# ****a. Web Crawler Nasıl Çalışır ?****

Web crawler, bir seed'den veya bilinen URL'lerin listesinden başlayarak, web sayfalarını inceleyerek ve ardından kategorize ederek çalışır. Her sayfa incelenmeden önce, web crawler web sayfasının robots.txt dosyasına bakar, bu dosya web sitesine erişen botlar için kuralları belirtir. Bu kurallar hangi sayfaların taranabileceğini ve hangi bağlantıların takip edilebileceğini tanımlar.

Crawler, bir sonraki web sayfasına ulaşmak için görünen bağlantıları bulur ve takip eder. Crawler’ in hangi bağlantıyı takip edeceği, tarayıcının hangi sırayı takip etmesi gerektiği konusunda daha seçici olmasını sağlayan tanımlanmış kurallara bağlıdır. Örneğin, tanımlanmış kurallar aşağıdakileri içerebilir:

* o sayfaya kaç sayfanın bağlantı verdiği;
* sayfa görüntüleme sayısı; ve
* marka otoritesi.

Bu faktörler, bir sayfanın indeksleme için daha önemli bilgilere sahip olabileceğini gösterir.

Crawler, bir web sayfasındayken meta etiketler olarak adlandırılan kopyayı ve açıklayıcı verileri depolar ve ardından arama motorunun anahtar kelimeleri taraması için indeksler. Bu işlem daha sonra sayfanın bir sorgu için arama sonuçlarında görünüp görünmeyeceğine karar verir ve eğer öyleyse, önem sırasına göre indeklenmis web sayfalarının bir listesini döndürür.

Bir web sitesi sahibinin, arama motorlarının siteyi taraması için site haritasını göndermemesi durumunda, bir web crawler, siteye bağlı olan indeklenmis sitelerden bağlantıları izleyerek web sitesini yine de bulabilir.

# ****b. Web Crawler Örnekleri****

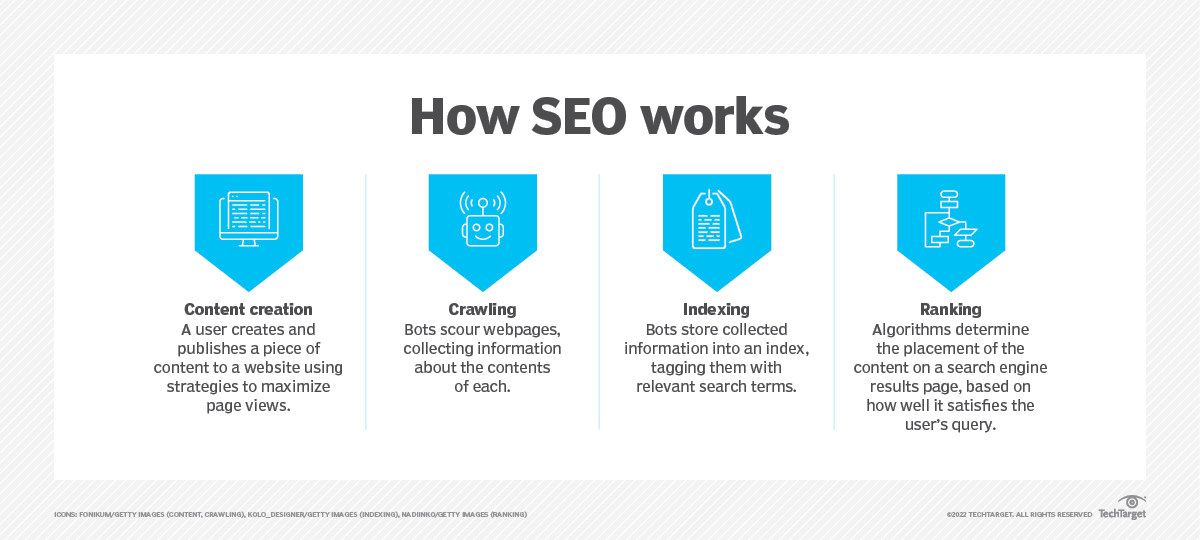
Çoğu popüler arama motorunun, web sayfaları hakkında bilgi toplamak için belirli bir algoritma kullanan kendi web crawler’ lari vardır. Web crawler araçları masaüstü veya bulut tabanlı olabilir. Arama motoru indeksleme için kullanılan bazı web tarayıcı örnekleri aşağıdakileri içerir:

* Amazonbot, Amazon web crawler’ idir.
* Bingbot, Microsoft'un Bing için arama motoru crawler’ idir.
* DuckDuckBot, DuckDuckGo arama motorunun crawler’ idir.
* Googlebot, Google'ın arama motorunun crawler’ idir.
* Yahoo Slurp, Yahoo'nun arama motorunu crawler’ idir.
* Yandex Bot, Yandex arama motorunun crawler’ idir.

# ****c. Web Crawler' ları SEO için neden önemlidir?****

Arama motoru optimizasyonu (SEO), insanlar ürün veya hizmet aradığında görünürlüğünü artırmak için bir web sitesini geliştirme sürecidir. Bir web sitesinde taranmasını zorlaştıran hatalar varsa veya taranamıyorsa, arama motoru sonuç sayfası (SERP) sıralaması daha düşük olur veya organik arama sonuçlarında görünmez. Bu nedenle web sayfalarında kırık bağlantılar veya başka hatalar olmadığından emin olmak ve web crawler botlarının web sitelerine erişmesine izin vermek ve onları engellememek önemlidir.

Aynı şekilde, düzenli olarak crawl' lanmayan sayfalar, SEO'yu artırabilecek herhangi bir değişikliği yansıtmayacaktır. Sayfaların düzenli olarak taranması ve güncellenmesinin sağlanması, özellikle zamana hassas içerikler için SEO'nun iyileştirilmesine yardımcı olabilir.



# ****d. Web Crawling vs. Web Scraping****

Web crawling ve web scraping kolayca karıştırılabilen iki benzer kavramdır. İkisi arasındaki temel fark, web crawling web sayfalarını bulmak ve indekslemekle ilgiliyken, web scraping bir veya daha fazla web sayfasında bulunan verileri extract etmekle ilgilidir.

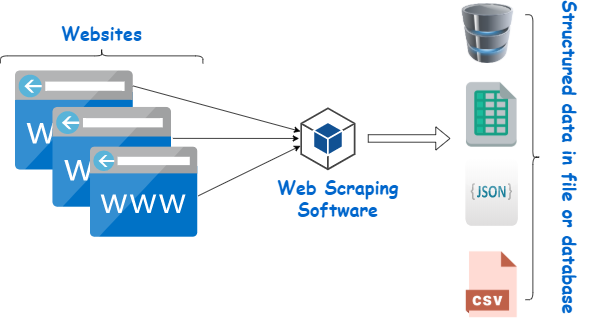
Web scraping, çeşitli web sayfalarından izin almadan otomatik olarak veri toplayabilen bir bot oluşturmayı içerir. Web crawler'ları bağlantıları sürekli olarak hyperlink'lere göre takip ederken, web scraping genellikle çok daha hedefli bir süreçtir -- ve sadece belirli sayfaların peşinde olabilir.

Web crawler'lar robots.txt dosyasını takip ederek web sunucularını aşırı zorlamamak için istekleri sınırlandırırken, web scraper'lar neden olabilecekleri herhangi bir yükü göz ardı eder.

Web scraping, daha hedefli veri setleri oluşturmak için analitik amaçlarla - veri toplamak, depolamak ve ardından analiz etmek - kullanılabilir.

Web scraping işleminde basit botlar kullanılabilir, ancak daha sofistike botlar bir sayfadaki uygun verileri bulmak ve bir analitik uygulaması tarafından işlenmek üzere doğru veri alanına kopyalamak için yapay zeka kullanır. Yapay zeka web scrap tabanlı kullanım alanları arasında e-ticaret, işgücü araştırması, tedarik zinciri analitiği, kurumsal veri yakalama ve pazar araştırması yer almaktadır.

Ticari uygulamalar, yeni ürün lansmanlarında duygu analizi yapmak, şirketler ve ürünler hakkında yapılandırılmış veri setlerini düzenlemek, iş süreci entegrasyonunu basitleştirmek ve tahmini olarak veri toplamak için web scraping'i kullanır.

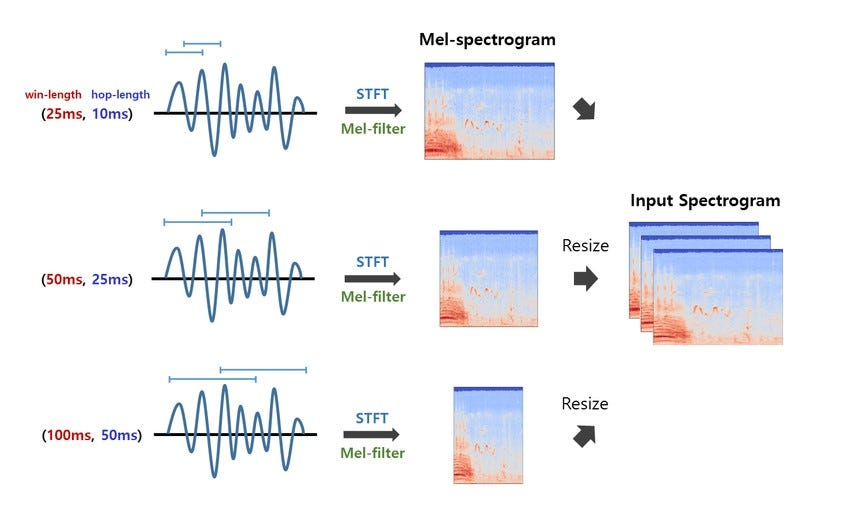


# ****2. Audio Pre-processing****

****Ses ön işleme****, ses verisi analizi ve makine öğrenimi uygulamalarınında kritik bir adımdır. Sesin kalitesini artırmak, anlamlı özellikler çıkarmak ve daha fazla analiz veya makine öğrenimi modellerine giriş için hazırlamak amacıyla ham ses verilerine uygulanan bir dizi tekniği içerir. Etkili ön işleme, ses verileri üzerinde eğitilen modellerin performansını ve doğruluğunu önemli ölçüde etkileyebilir ve bu da onu ses sinyali işlemenin önemli bir yönü haline getirir.

# ****a. Ses Veri Kümeleri Için Neden Ön İşleme Yapılmalıdır?****

1. ****Gürültü Azaltma:**** Gerçek dünya ortamlarından toplanan ses verileri genellikle arka plan gürültüsü, parazit veya artefaktlar içerir. Filtreleme ve denoising gibi ön işleme yöntemleri istenmeyen gürültünün giderilmesine yardımcı olarak modelin ilgili sinyale odaklanmasını sağlayabilir.
2. ****Formatların Standartlaştırılması:**** Ses veri kümeleri çeşitli formatlarda, örnekleme oranlarında veya çözünürlüklerde olabilir. Ön işleme, standardizasyonu sağlayarak farklı veri kümeleriyle çalışmayı kolaylaştırır ve model performansını etkileyebilecek tutarsızlıkları önler.
3. ****Özellik Çıkarma:**** Ses sinyalleri karmaşık ve yüksek boyutludur. Ön işleme, spektral özellikler, Mel frekanslı cepstral katsayılar (MFCC'ler) veya kroma özellikleri gibi ham verilerden ilgili özellikleri çıkarır. Bu özellikler, analiz için gerekli bilgileri koruyarak sesin daha kompakt bir temsilini sağlar.
4. ****Yeniden örnekleme:**** Ön işleme sırasında ses sinyallerinin örnekleme oranının yeniden örnekleme yoluyla standartlaştırılması yaygındır. Bu adım, verileri hesaplama açısından daha verimli ve belirli bir örnekleme hızı gerektiren modellerle uyumlu hale getirebilir.
5. ****Normalleştirme:**** Ses sinyallerinin genliğinin ölçeklendirilmesi, modelin daha yüksek veya daha düşük enerji seviyelerine sahip sinyallere karşı esnek olmasını sağlar. Normalleştirme, veri kümesi genelinde tutarlı sinyal büyüklüklerinin korunmasına yardımcı olur.
6. ****Değişken Uzunlukların İşlenmesi:**** Bir veri kümesindeki ses klipleri farklı uzunluklara sahip olabilir. Ön işleme genellikle tek tip bir uzunluk elde etmek için sinyalleri bölümlere ayırmayı veya doldurmayı içerir ve sabit uzunlukta girdiler gerektiren modellerle uyumluluğu sağlar.
7. ****Model Verimliliği:**** İyi işlenmiş veriler daha verimli eğitim ve çıkarım süreçlerine yol açabilir. Hesaplama yükünü azaltır, eğitim sırasında yakınsamayı hızlandırır ve modelin yeni, görülmemiş verilere genelleme yeteneğini geliştirir.
8. ****Geliştirilmiş Model Performansı:**** Ön işleme, ilgili özellikleri vurgulayan ve modele yüksek kaliteli girdiler sağlanmasını sağlayan sinyal-gürültü oranını artırır. Bu da model performansının, doğruluğunun ve güvenilirliğinin artmasını sağlar.



# ****3. Projede Kullanılan Teknolojiler****

* **Python,** web uygulamaları, yazılım geliştirme, veri bilimi ve makine öğreniminde (ML) yaygın olarak kullanılan bir programlama dilidir.
* **Selenium,** web otomasyon kütüphanesidir.
* **BeautifulSoup4,** python tabanlı web scraping kütüphanelerinden birisidir.
* **PyTubeFix,** YouTube videolarını indirmek için oluşturulmuş bir kütüphane.
* **Pydub,** ses dosyalarını okumak, yazmak ve manipüle etmek için kullanılır.
* **Librosa,** müzik ve ses analizi için bir python paketidir.
* **Soundfile,** pydub gibi ses dosyalarını okumak ve yazmak için kullanılan bir modül.
* **Noisereduce,** ses dosyasındaki gürültüyü azaltmak için oluşturulmuş modül.
* **Pandas,** Python programlama dilinde geliştirilen güçlü bir veri analizi ve manipülasyon kütüphanesidir.

# ****4. Proje Açıklaması ve Geliştirme Aşamaları ( Ilk Kısım )****

**Proje Açıklaması:** İngilizce aksan tespiti için oluşturulmuş crawler’ la ilgili verilerin toplanması ve daha sonra bu verilerin pre-processing işleminden geçirilip işlenmeye hazır duruma getirilmesi.

**Toplam sınıf sayısı:** 5 ( American – British – Australian – Indian – Scottish )

**Her sınıftakı toplam verisi sayısı:** En az 5000.

**Geliştirme Aşamaları:**

1. Konuyla ilgili bir web crawler geliştirildi ve ilgili veriler YouTube üzerinden toplandı.
2. Toplanan veriler işlenip 5 saniyelik parçalara bölündü.
3. Yapılan Pre-processing işlemleri:

* Ses dosyasi mp3 formatindan wav formatina cevrildi.
* 44.1kHz’ den 16kHz’ e resample edildi.
* Sessiz olan kisimlar filtrelendi.
* Gurultu azaltildi.
* Ses normalizasyonu yapildi.
* 5 saniyelik parcalara bolundu.
* Her bir ses parcasi icin MFCC cikarilip csv olarak kaydedildi.

1. Veriler işlenmeye hazır duruma getirildi.

**Kaynaklar:**

1. https://www.elastic.co/what-is/web-crawler
2. https://www.techtarget.com/whatis/definition/crawler
3. https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/
4. https://www.akamai.com/glossary/what-is-a-web-crawler
5. https://www.seo.com/basics/how-search-engines-work/crawling/
6. https://hikeseo.co/learn/onsite/technical/crawling/
7. https://nextjs.org/learn-pages-router/seo/introduction-to-seo/webcrawlers
8. https://www.woorank.com/en/blog/how-a-crawler-works-back-to-the-basics
9. https://www.geeksforgeeks.org/preprocessing-the-audio-dataset/
10. https://huggingface.co/learn/audio-course/chapter1/preprocessing
11. https://medium.com/huawei-developers/basics-of-audio-processing-e69efce7765f
12. https://www.tensorflow.org/tutorials/audio/simple\_audio
13. https://www.youtube.com/watch?v=WJI-17MNpdE
14. https://cloud.google.com/speech-to-text/docs/best-practices-provide-speech-data
15. https://stackoverflow.com/questions/42492246/how-to-normalize-the-volume-of-an-audio-file-in-python
16. https://stackoverflow.com/questions/71695438/why-do-i-need-no-sandbox-to-run-selenium-chromedriver-even-with-admin-privil
17. https://www.kaggle.com/code/super13579/mfcc-feature-extraction
18. https://github.com/rctatman/getMFCCs/blob/master/getMFCCs.py
19. <https://www.youtube.com/watch?v=WJI-17MNpdE>