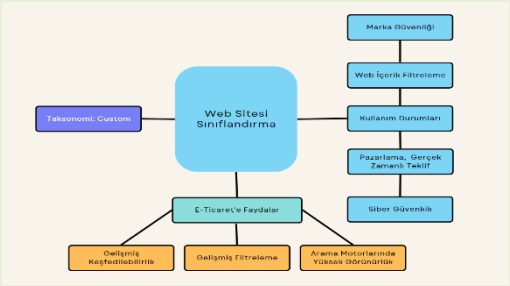
Web Sitelerinin Yapay Zeka Kullanılarak Sınıflandırılması (Ocak 2023)

***Özet* —** Çağımızda bilgiye ulaşmak için milyonlarca web sitesi kullanılmaktadır. Bu web sitelerinin sayısı günden güne çok hızlı bir şekilde artmaktadır. Web sitelerinin kullanımını daha etkin bir hale getirmek için doğru şekilde sınıflandırmak çok önemlidir. Bu çalışmada, doğal dil işleme kullanılarak web sitelerini sınıflandıran 3 farklı model oluşturulmuştur. Bu modeller Custom Model, Logistic Regression Model, Linear Support Vector Machine Modeldir. Modeller eğitilirken açık kaynaklı bir veri setindeki URL'ler ve kategoriler kullanılmıştır. Her bir model için web sayfalarının metinleri URL'lerden çekilerek eğitim veri seti oluşturulmuştur. Oluşturduğumuz modellerden olan Custom model ana modelimiz olarak seçilmiştir. Custom model kelime frekanslarınıiçeren dosyayı kullanarak her kategori için ağırlık değerine göre ana kategoriyi ve alt kategoriyi tahmin eder. Diğer iki model istatistiksel olarak Custom Modelle karşılaştırma yapmak için kullanılmıştır.



***Anahtar Kelimeler* — Doğal dil işleme, Web sitesi sınıflandırma, Custom Model, Logistic Regression Model, Linear Support Vector Machine Model**

I. [1] GIRIŞ

Çağımızda bilgiye ulaşmak için internet ağı üzerinden web siteleri yaygın olarak kullanılmaktadır. Web sitelerinin kullanımını daha etkin ve hızlı bir hale getirmek için doğru bir şekilde sınıflandırmak çok önemlidir. Web sitesi sınıflandırması, genellikle makine öğrenimi çözümleri kullanılarak, web sitesini bir veya daha fazla kategoride sınıflandırmak olarak tanımlanabilmektedir. Web sitesi sınıflandırması, makine öğrenimi ve doğal dil işlemede önemli bir alandır. World Wide Web (WEB), farklı içeriklere sahip web sitelerinin en büyük bilgi kaynaklarından biri haline gelmektedir. Web sitelerinin içeriği kategoriler halinde sınıflandırılabilir ve bu şekilde tüm web siteleri belirli bilgileri arayan kişiler için yapılandırılabilmektedir.

Web sitelerinin içeriklerine göre sınıflandırılması kullanıcılara büyük kolaylık sağlamaktadır. Arama motorlarını kullananlar içeriklerin sınıflandırılması ile istediklerine daha hızlı ve kolay bir şekilde ulaşacaklardır. Ayrıca web sayfasının kategorisinin belirlenmesi de web sitelerine yerleştirilecek reklamların seçiminde önemli bir rol oynamaktadır. Reklamların buna göre yerleştirilmesi kullanıcıları daha çok etkilemesi ve reklamlardan elde edilecek kazancı arttırması beklenmektedir. Bu çalışmada, web sitesi sınıflandırması doğal dil işleme kullanılarak yapılmıştır. Doğal dil işleme kullanılarak custom bir word frequency modeli oluşturulmaktadır. Bu çalışmada, açık kaynaklı bir veri setindeki URL'ler ve 25 farklı kategori kullanılmaktadır. Veri setinde eğitim işlemi için ayrılan URL'lere istek atılarak, web sayfalarından ilgili metin çekilerek word tokenization (sözcük belirleme) işlemi uygulanmaktadır. Bu çalışmada Natural Language ToolKit kütüphanesi kullanılarak her bir kategori için word frequency dizileri oluşturulmaktadır. Custom modelde, Word Frequency dosyası kullanılarak her kategori için verilen URL'in ağırlık değerini hesaplanmaktadır, burada en yüksek ağırlık değerine sahip kategori ana kategoriyi vermektedir. İkinci en yüksek ağırlık değerine sahip kategori ise bize alt kategoriyi vermektedir.

II. ÇALIŞMANIN AMACI VE KAPSAMI

Web sitesi sınıflandırması, bir içeriğine göre belirli bir kategoriye yerleştirilmesidir. Bu işlem, web sayfalarının arama sonuçlarında daha iyi organizasyon ve daha etkin arama yapılmasını sağlar. Web sitesi sınıflandırması ayrıca, bir web sayfasının arama motorlarında daha iyi görünmesini sağlar. Örneğin, bir arama motoru, bir web sayfasının içeriğine uygun bir kategori içinde yer aldığını tespit ettiğinde, o sayfayı daha ilgili arama sonuçlarında gösterebilir. Bu sayede, kullanıcılar daha işe yarar ve daha ilgili sonuçlar elde edebilirler. Eğer bir web sitesi sınıflandırılması yapılmamışsa, arama motorları o web sitesini ilgili arama sonuçlarında gösteremeyebilir. Bu durumda, web sitesine ulaşmak isteyen ziyaretçiler daha az ilgili sonuçlar elde edebilirler ve web sitesine ulaşmak daha zor hale gelebilir. Bu projenin amacı, bahsedilen sorunlara çözüm olabilecek bir model geliştirmektir. Doğal dil işleme kullanılarak oluşturulan bu modelin, URL'i verilen web sitesinin kategorisini yüksek doğrulukta tahmin etmesi istenir.

Bu çalışmanın kapsamı;

- Çalışmaya uygun veri setinin ve taksonominin seçilmesi,
- Doğal dil işleme kullanılarak modelin eğitilmesi,
- Kullanıcı arayüzünün (UI) geliştirilmesi,

* Domantas Meidus , "Website Classification Using Machine Learning Approaches", Bachelour Thesis, 2019

** Mehmet Salih Kurt, Eylem Yücel, "WEB PAGE CLASSIFICATION WITH DEEP LEARNING METHODS" , Article, 2022

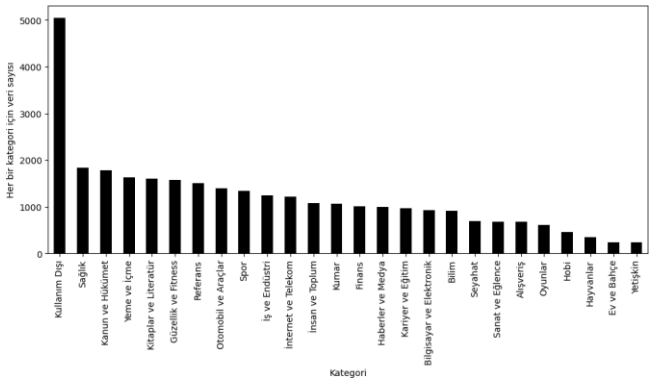
• Sistemin test edilmesi şeklindedir.

III. YÖNTEM

A. Kullanılacak Veri Setinin Belirlenmesi

Web sitesi sınıflandırma işlemi için, URL ve bu URL'ye karşılık gelen kategoriyi içeren yaklaşık 30 bin adet veriden oluşan açık kaynaklı bir veri seti kullanılmaktadır. Bu veri setinde 25 farklı kategori bulunmaktadır. Bu kategoriler:

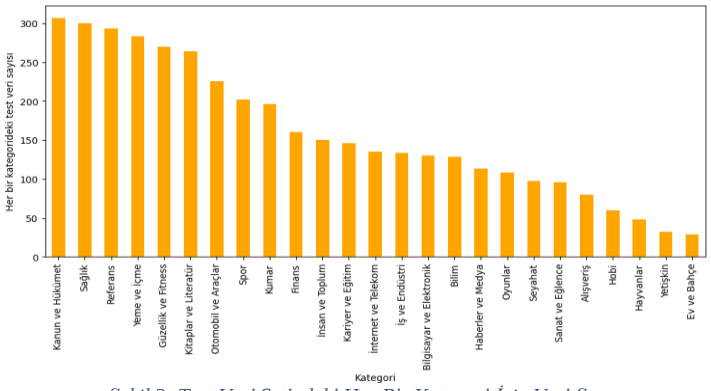
- İnternet ve Telekom
- Kariyer ve Eğitim
- Bilim
- Kumar
- Sağlık
- Spor
- Kitaplar ve Literatür
- Bilgisayar ve Elektronik
- Sanat ve Eğlence
- Güzellik ve Fitness
- Yetişkin
- Alışveriş
- Oyunlar
- Hukuk ve Hükümet
- Referans
- Finans
- Hobi
- Hayvanlar
- Yeme ve İçme
- Haberler ve Medya
- İş ve Endüstri
- İnsan ve Toplum
- Ev ve Bahçe
- Seyahat
- Otomobil ve Araçlar



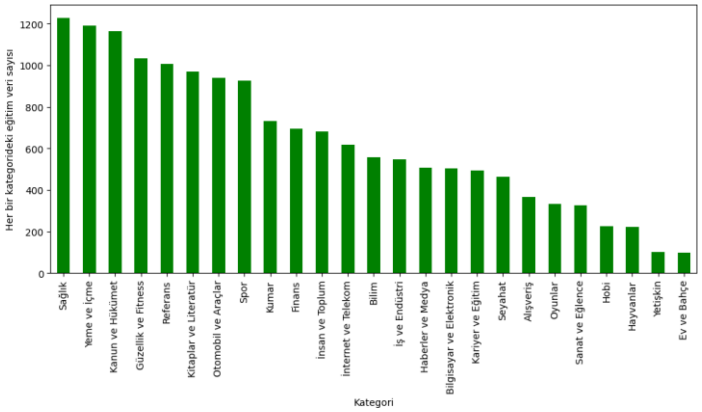
Şekil 1: Her bir kategori için veri sayısı

B. Eğitim ve Test İçin Veri Setinin Belirlenmesi

Kullandığımız veri seti içerisindeki kullanım dışı olan URL'leri çıkaracak bir filtre uygulanmıştır. Sonrasında train (eğitim) ve test verilerini rastgele olarak seçmek için Scikit-learn paketinin train-test-split metodu kullanılmıştır. Train ve test verilerini ayırmak, modelin performansını doğru bir şekilde değerlendirmek için önemlidir. Eğitim sırasında model, train verileri üzerinden öğrenmektedir ve bu verileri kullanarak tahminler yapmaktadır. Ancak modelin gerçek hayatta nasıl performans göstereceği hakkında bir fikir vermemektedir, çünkü model sadece eğitim verileriyle öğrendiği bilgiyi kullanmaktadır. Bu nedenle, modelin performansını doğru bir şekilde değerlendirebilmek için, modelin eğitim verileri dışındaki verilere nasıl tahmin yaptığını görmek gerekmektedir. Bu nedenle, bir test veri kümesi ayrılmaktadır ve modelin bu verilere nasıl tahmin yaptığını görmek için bu verilere tahminler yaptırılmaktadır. Böylece, modelin eğitim verileriyle öğrendiği bilgiyi ne kadar iyi genelleştirebildiğini anlayabilir ve modelin performansını doğru bir şekilde değerlendirebiliriz. 5 Veri seti üzerinde %80 oranında eğitim, %20 oranında test olacak şekilde rastgele bir dağılım yapılmıştır.



Şekil 2: Test Veri Setindeki Her Bir Kategori İçin Veri Sayısı



Şekil 3: Train Veri Setindeki Her Bir Kategori İçin Veri Sayısı

C. Custom Model Kullanılarak Sınıflandırma

1) Veri Ön İşleme (Data Preprocessing)

Web sitesi tarama (website scraping) ve metin ayrıştırma (text parsing) işlemleri, internet üzerindeki bilgiye erişmek ve bu bilgiyi istenen bir biçime dönüştürmek için kullanılan yöntemlerdir. Web sitesi tarama, bir web sitesinden bilgi toplamaya yönelik bir yöntemdir. Bu işlem, bir web sitesinin HTML veya XML kaynak kodunu indirir ve istenen bilgileri elde etmek için bu kaynak kodunu tarar. Çalışmamızda bu işlem için Python BeautifulSoup modülü kullanılır. Metin ayrıştırma, bir metin dosyasından bilgi ayıklamaya yönelik bir yöntemdir. Bu işlem, bir metin dosyasının içeriğini parçalara ayırarak istenen bilgileri elde etmek için kullanılır. Çalışmamızda bu işlem için Python RE (Regular Expression) modülü kullanılır. Web sitesi tarama ve metin ayrıştırma işlemleri, veri madenciliği ve veri analizi gibialanlarda sıklıkla kullanılır. Bu işlemler sayesinde, internet üzerindeki bilgiye daha kolay erişilebilmekte ve bu bilgi daha kolay işlenebilir hale gelmektedir. Internet sitesi tarama ve metin ayrıştırma işlemleri aşağıdaki adımları içermektedir:

a) requests.get() fonksiyonu ile bir URL'ye bir HTTP GET isteği yapılır ve cevabı bir Response nesnesi olarak alınır. Bu istek, https:// ile başlayan bir URL'ler için yapılır.
Bu istek, https:// ile başlayan bir URL'ler için yapılır.
b) Eğer cevap 200 durum kodu değilse, URL'nin başına http:// eklenerek tekrar bir istek yapılır.
c) Eğer cevap 200 durum kodu ise, cevap içeriği bir BeautifulSoup nesnesine dönüştürülür. Bu nesne, HTML içeriğinin bir parse ağacı şeklinde temsil edilmesine yardımcı olur.
d) Soup nesnesi oluşturulur ve bu nesnede script ve style etiketleri kaldırılır. Soup nesnesinden metin çekilir ve metin içindeki tüm karakterler dışında kalanlar (re.sub() ile) silinir.e)
e) Metin içindeki kelimeler (word_tokenize() ile) ayrıştırılır ve anlamı belli olmayan kelimeler (remove_stopwords() ile) kaldırılır.
f) Elde edilen kelime listesi (tokens_lemmatize) ile predict_category() fonksiyonu çağrılır ve bu fonksiyon, kelime listesi kullanarak bir kategori tahmini yapar. Tahmin edilen kategori, fonksiyonun döndürdüğü değer olarak döndürülür.

Metin tokenleştirme (text tokenization), bir metin dosyasındaki kelimeleri veya ifadeleri ayırmak için kullanılan bir yöntemdir. Bu işlem, bir metin dosyasının içeriğini parçalara ayırarak kelime veya ifadelerin bir listesi oluşturmaktadır. Tokenleştirme işlemi, metin dosyasının içeriğini daha kolay işlemeye yönelik bir ön işlemdir. Çalışmamızda bu işlem için NLTK kütüphanesinin word_tokenize fonksiyonu kullanılmaktadır.

Siteden aldığımız metin içinde bulunan, düzenli olarak kullanılan, anlamı genellikle belli olmayan kelimelerin (stopwords) silinmesi gerekmektedir. Örneğin, Türkçe dilinde "ve", "ile", "gibi" gibi kelimeler stopwords olarak sınıflandırılabilmektedir. Çalışmamızda bu işlem için Python'da bulunan remove_stopwords modülü kullanılmaktadır. Bu modül, bir metin içinde bulunan stopwords'leri belirli bir dildeki stopwords listesiyle karşılaştırmakta ve metinden bu kelimeleri çıkarmaya yardımcı olmaktadır. Kullandığımız veri setindeki sitelerde kullanılan orijinal dil İngilizce olduğu için stopword listesi bu dilde tanımlanmaktadır ve 208 tane kelime içermektedir. Bu işlem aşağıdaki Pseudo kod ile ifade edilebilir:

```
WordNetLemmatizer sınıfından wnl nesnesi oluştur
config dosyasından stopWords'ü import et

tanımla
   remove_stopwords(tokens):
   oluştur tokens_list
   tokens listesinin içinde dön:
      word'ü word.lower() ile küçük harfe çevir ve wnl
nesnesinin lemmatize metodunukullanarak lemmasını al
      eğer word config.stopWords listesinde değilse:
         tokens_list listesine word'ü ekle
   tokens_list listesinde len(x) > 1 olan tokenları içeren
bir liste döndür
```

Kelime listesi oluşturulması, dil modelleme uygulamalarında veri kümesinin ön işleme adımı olarak önemlidir. Veri kümesindeki kelimelerin sayısını azaltır ve bu da modelin öğrenme sürecini hızlandırır ve daha etkili hale getirir. Her kategori, kendini ifade eden bazı kelimeler içermektedir. Bu kelimeler o kategoriye ait olan sitelerde en çok tekrar eden kelimelerdir. Çalışmamızda bu işlem için NLTK kütüphanesinin probability modülü içinde yer alan FreqDist sınıfını kullanılır. Bu sınıf, verilen bir veri kümesinde kelime sıklıklarını sayar ve bu sayımları bir sözlük şeklinde saklanır. Her bir kategoriye, o kategori ile en çok ilgili 20.000 adet kelime ataması yapılır. Bu işlem aşağıdaki akış diyagramı ile ifade edilebilir:

2) Custom Model Eğitimi

Custom NLP modelleri, özel bir veri kümesine ve özel bir göreve uygun olarak tasarlandığı için, genellikle daha yüksek performans göstermektedir.

Ancak, custom modelinin oluşturulması daha zor ve zaman alıcıdır, çünkü bu modeller özel veri kümeleri ve görevler için tasarlanmaktadır ve bu veri kümelerinin toplanması ve bu görevler için uygun hale getirilmesi işlemlerini gerektirmektedir. Ayrıca, özel NLP modellerinin eğitimi ve test edilmesi için daha fazla veri gerekmektedir.

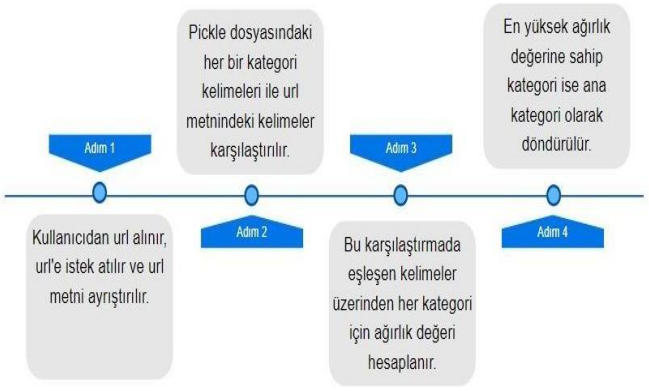
Şekil 5: Custom Model Oluşturma Adımları

Custom model eğitimi aşağıdaki adımları içermektedir:

a) ThreadPoolExecutor, config.threadingWorkers değişkeni tarafından belirtilen sayıda (16) çalışanla başlatılır. Paralel işleme görevlerini gerçekleştirmek için Python'un concurrent.futures modülündeki ThreadPoolExecutor ve ProcessPoolExecutor sınıfları kullanılır.
b) Dataframe'in 'url' sütunundan bir dizin ile scrape işlevi çağrılır ve bunun üzerinde executor.map() yöntemi çağrılır. Kazıma işlevi, belirtilen sayıda iş parçacığını kullanarakparalel olarak her bir URL için uygulanır.
c)config.multiprocessingWorkers değişkeni tarafından belirtilen sayıda (6) çalışanla bir ProcessPoolExecutor başlatılır. parse_request işlevi çağrılır ve bunun üzerinde ex.map() yöntemi çağrılır, parse_request işlevi, belirtilen işlem sayısını kullanarak paralel olarak her bir sonuca uygulanır.
d) ex.map() çağrısından elde edilen her öğe için, ilk öğesi df veri çerçevesindeki bir satırı aramak için kullanılır ve ikinci öğe (tokens) o satır için 'tokens' sütununa atanır.
e) words_frequency adlı bir sözlük oluşturulur ve boş olarak başlatılır. Ardından, benzersiz kategorilerin bir listesini almak için df'nin 'main_category' sütununda unique() yöntemi çağrılır ve kod bu liste üzerinde yinelenir.

f) Her kategori için, yalnızca 'main_category' sütununun geçerli kategoriye eşit olduğu df'den gelen satırları içeren, df_temp adlı yeni bir geçici veri çerçevesi oluşturulur.
g) all_words adlı boş bir liste oluşturur ve ardından df_temp'in "tokens" sütununu yineler.
h)all_words listesini df_temp'in "tokens" sütunundaki kelimeler ile genişletir.
i) all_words üzerinde nltk.FreqDist() işlevi çağrılır ve bu işlev, listedeki her sözcüğün tekrarlanma sayısını sayan bir frekans dağıtım nesnesi oluşturur.
j)Bu nesne üzerinde config.words değerini ileterek most_common() yöntemi çağrılır vebu, all_words içindeki en yaygın sözcüklerin sıklıklarıyla birlikte bir listesini döndürür.
k)Bu listeden yalnızca sözcükler seçilir ve bu listeyi words_frequenc sözlüğündekianahtar kategoriye atar.
l) Böylece words_frequency dosyası hazırlanır.

Custom model çalışma prensibi aşağıdaki adımları içermektedir:
a) Her bir kategori için bir "ağırlık" değeri oluşturulur. Bu ağırlık değeri, tokens listesi içinde yer alan kelimelerin kategori içinde kaçıncı sıradaki kelimeler olduğuna göre hesaplanır (weight += 20000 – index). Örneğin, eğer bir kelime tokens listesi içinde yer alıyor ve aynı zamanda kategori içinde en sık kullanılan kelime ise, bu kelime için ağırlıkdeğeri en yüksek olacaktır.
b) Her bir kategori için ağırlık değerleri bir listeye toplanır.
c) Ağırlık değerleri listesi içinde en yüksek değere sahip olan kategori, ana kategori olarak seçilir.
d) Bu kategori için ağırlık değeri listesinde yer alan değer sıfırlanır.
e) Ağırlık değerleri listesi içinde ikinci en yüksek değere sahip olan kategori, ikinci anakategori olarak seçilir.
f) Son olarak, seçilen ana kategori ve ikinci ana kategori (alt kategori) döndürülür.



Şekil 6: Custom Model Çalışma Prensibi

Çalışma prensibi aşağıda Pseudo kod olarak da ifade edilmektedir:

```
Tanımla predict_category(words_frequency, tokens)
        category_weights = BOŞ
        LİSTE
        words_frequency içinde her kategoriyi döngüyle dolaş
        weight=0
        intersect_words=word_frequency kesişmeleri toplayın
        intersect_words içinde her word'ü döngüyle dolaş.
        eğer word tokens içinde varsa
            Index=word_frequency dizisinde kategorinin içinde
            Yer alan kelimenin indeksi
            Weight+=config.words – index
            category_weights'e weight'i ekle
        category_index = category_weigths içinde en yüksek
        değere sahip index
        main_category = word_frequency listesi  içinde
        category_index
        category_weights viçinde category_index'i sıfırla
        Category_index = category_weigths içinde en yüksek
        ikinci değere sahip index
        Main_category_2= word_frequency listesi  içinde
        category_index
        Main_category ve main_category_2'yi döndür
```

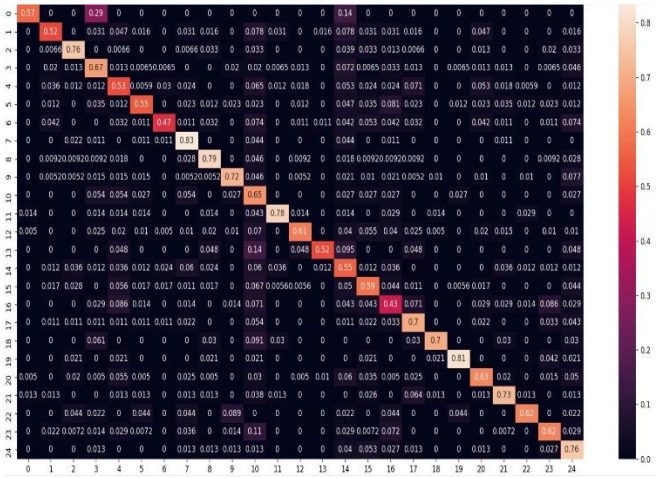
3) Custom Model Performans Değerlendirmesi

Custom model performans değerlendirmesi, bir makine öğrenimi modelinin perfromansını ölçmek için yapılan bir süreçtir. Aşağıdaki görselde precision, recall, f1-score, support ve accuracy değerleri verilmiştir.

	precision	recall	f1-score	support
Hukuk ve Hükümet	0.57	0.57	0.57	7
Referans	0.60	0.51	0.55	63
Finans	0.81	0.76	0.78	148
Hobi	0.77	0.66	0.71	150
Hayvanlar	0.60	0.52	0.56	164
Yeme ve İçme	0.69	0.55	0.61	85
Haberler ve Medya	0.74	0.49	0.59	91
İş ve Endüstri	0.62	0.82	0.70	90
İnsan ve Toplum	0.75	0.81	0.78	106
Ev ve Bahçe	0.90	0.74	0.81	184
Seyahat	0.16	0.68	0.26	34
Otomobil ve Araçlar	0.82	0.79	0.81	68
İnternet ve Telekom	0.91	0.62	0.74	201
Kariyer ve Eğitim	0.71	0.50	0.59	20
Bilim	0.34	0.53	0.42	86
Kumar	0.67	0.59	0.63	181
Sağlık	0.31	0.44	0.36	71
Spor	0.57	0.67	0.61	90
Kitaplar ve Literatür	0.81	0.79	0.80	33
Bilgisayar ve Elektronik	0.85	0.83	0.84	48
Sanat ve Eğlence	0.76	0.66	0.71	195
Güzellik ve Fitness	0.72	0.72	0.72	75
Yetişkin	0.74	0.61	0.67	46
Alışveriş	0.78	0.62	0.69	141
Oyunlar	0.43	0.78	0.56	73
accuracy			0.65	2450
macro avg	0.67	0.65	0.64	2450
weighted avg	0.71	0.65	0.67	2450

Şekil 7: Custom Model Değerlendirme Ölçüt Sonuçları

Aşağıda verilen şekilde Custom modele ait olan confusion matrixi gösterilmektedir.



Şekil 8: Custom Model Confusion Matrix

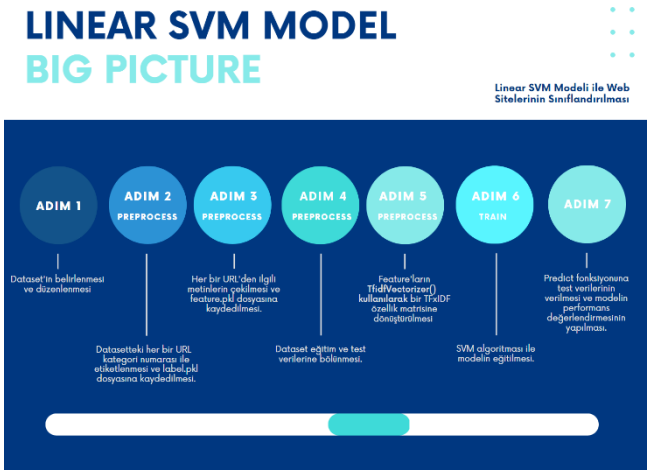
C. Linear Support Vector Machine Model Kullanılarak Sınıflandırma

1) Veri Ön İşleme (Data Preprocessing)

Bu model için veri ön işleme adımları aşağıdaki gibi gerçekleşmektedir.

- Veri setindeki her bir kategoriye, o kategoriyi temsil edecek 1 ile 25 arasında bir numara atanır, böylece kategoriler etiketlenmiş olur. Veri setindeki her bir satıra karşılık gelen kategori numaraları label.pkl dosyasına kaydedilir.
- Veri setindeki her bir URL'e istek atılarak ilgili metinler çekilir ve feature.pkl dosyasına kaydedilir.
- TfidfVectorizer() fonksiyonu kullanılarak feature.pkl dosyasındaki kelimelerden bir kelime matrisi oluşturulur ve SelectPercentile() işlevi kullanılarak matris üzerinden seçim yapılır.

2) LSVM Model Eğitimi



Şekil 9: LSVM Model Oluşturma Adımları

Modeli eğitmek için Sklearn kütüphanesine ait SVM modülünün SVC sınıfı projeye dahil edilmiştir. Doğrusal (linear) bir metodoloji kullanılacağı SVC modeli içindeki kernel seçeneğinde belirtilmiştir. Daha sonra SVC sınıfının fit() fonksiyonuna feature train ve label train verileri parametre verilerek model eğitim işlemi yapılmıştır. Ve eğitim süreci tamamlanmıştır.

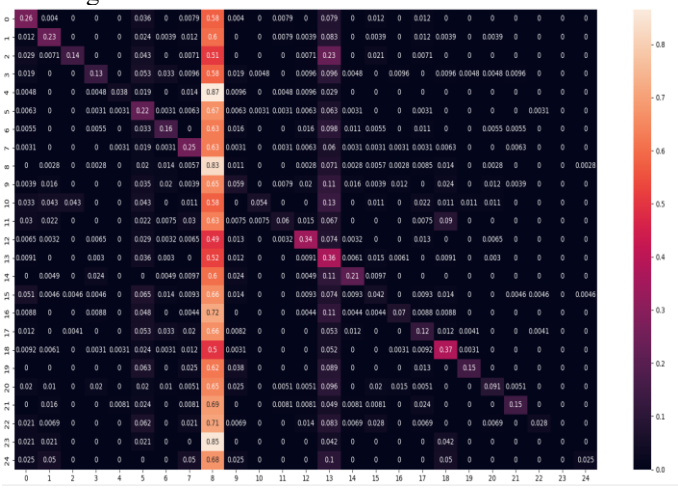
3) LVSM Modeli Performans Değerlendirmesi

Bir LSVM modelinin performansı, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 puanı gibi çeşitli ölçütler kullanılarak değerlendirilebilir. Aşağıdaki şekilde bu modelin doğruluk, kesinlik, geri çağırma ve F1 puanı sonuçları gösterilmektedir.

	precision	recall	f1-score	support
Hukuk ve Hükümet	0.54	0.26	0.35	253
Referans	0.69	0.23	0.35	254
Finans	0.77	0.14	0.24	140
Hobi	0.57	0.13	0.21	209
Hayvanlar	0.67	0.04	0.07	208
Yeme ve İçme	0.30	0.22	0.26	319
Haberler ve Medya	0.43	0.16	0.23	183
İş ve Endüstri	0.63	0.25	0.36	318
İnsan ve Toplum	0.09	0.83	0.16	351
Ev ve Bahçe	0.24	0.06	0.09	254
Seyahat	0.62	0.05	0.10	92
Otomobil ve Araçlar	0.40	0.06	0.10	134
İnternet ve Telekom	0.77	0.34	0.47	309
Kariyer ve Eğitim	0.23	0.36	0.28	329
Bilim	0.67	0.21	0.32	206
Kumar	0.23	0.04	0.07	216
Sağlık	0.55	0.07	0.12	228
Spor	0.46	0.12	0.19	244
Kitaplar ve Literatür	0.73	0.37	0.49	327
Bilgisayar ve Elektronik	0.75	0.15	0.25	79
Sanat ve Eğlence	0.60	0.09	0.16	198
Güzellik ve Fitness	0.69	0.15	0.24	123
Yetişkin	0.57	0.03	0.05	144
Alışveriş	0.00	0.00	0.00	48
Oyunlar	0.33	0.03	0.05	40
accuracy			0.23	5206
macro avg	0.50	0.18	0.21	5206
weighted avg	0.50	0.23	0.24	5206

Şekil 10: LSVM Model Değerlendirme Ölçüt Sonuçları

Aşağıda verilen şekilde LSVM modeline ait olan confusion matrixi gösterilmektedir.



Şekil 11: LSVM Model Confusion Matrix

D. Logistic Regression Modeli Kullanılarak Sınıflandırma

1) Veri Ön İşleme (Data Preprocessing)

Logistic Regression Modeli ön işleme adımları LSVM Modeli ön işleme adımları ile birebir aynıdır, bu sebeple tekrar değinilmeyecektir.

1) Logistic Regression Model Eğitimi

Modeli eğitmek için Sklearn kütüphanesine ait linear_model SVM modülünün LogisticRegression sınıfı projeye dahil edilmiştir. Daha sonra LogisticRegression sınıfının fit() fonksiyonuna feature train ve label train verileri parametre verilerek model eğitim işlemi yapılmıştır. Ve eğitim süreci tamamlanmıştır.

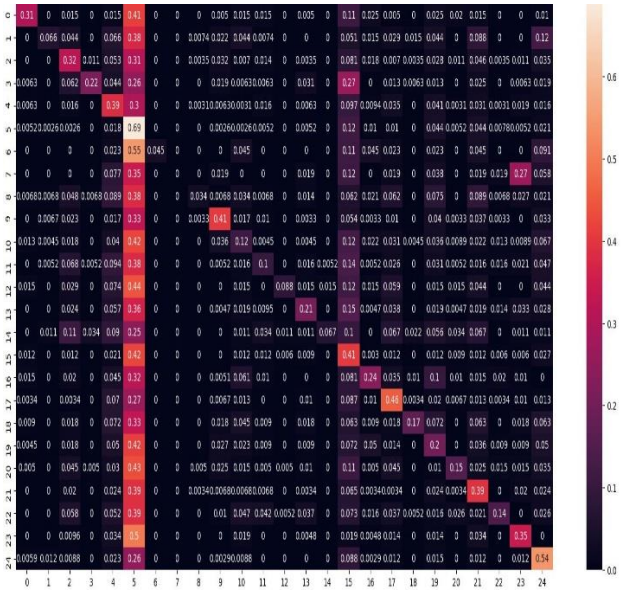
3) Logistic Regression Modeli Performans Değerlendirmesi

Bir Logistic Regression modelinin performansı, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 puanı gibi çeşitli ölçütler kullanılarak değerlendirilebilir. Aşağıdaki şekilde bu modelin doğruluk, kesinlik, geri çağırma ve F1 puanı sonuçları gösterilmektedir.

	precision	recall	f1-score	support
Hukuk ve Hükümet	0.68	0.33	0.45	206
Referans	0.39	0.07	0.12	130
Finans	0.49	0.44	0.46	255
Hobi	0.70	0.22	0.33	146
Hayvanlar	0.34	0.37	0.36	333
Yeme ve İçme	0.25	0.40	0.30	379
Haberler ve Medya	1.00	0.02	0.04	56
İş ve Endüstri	0.00	0.00	0.00	54
İnsan ve Toplum	0.53	0.06	0.10	160
Ev ve Bahçe	0.57	0.41	0.48	268
Seyahat	0.28	0.09	0.14	264
Otomobil ve Araçlar	0.38	0.10	0.15	189
Alışveriş	0.87	0.21	0.33	63
Oyunlar	0.49	0.21	0.29	219
İnternet ve Telekom	0.64	0.08	0.14	86
Kariyer ve Eğitim	0.13	0.74	0.22	349
Bilim	0.49	0.23	0.31	199
Kumar	0.55	0.42	0.48	297
Sağlık	0.67	0.19	0.29	106
Spor	0.31	0.23	0.26	244
Kitaplar ve Literatür	0.51	0.19	0.28	178
Bilgisayar ve Elektronik	0.44	0.39	0.41	305
Sanat ve Eğlence	0.51	0.15	0.24	175
Güzellik ve Fitness	0.55	0.34	0.42	222
Yetişkin	0.49	0.52	0.51	323
accuracy			0.32	5206
macro avg	0.49	0.26	0.28	5206
weighted avg	0.45	0.32	0.32	5206

Şekil 12: Logistic Regression Modeli Değerlendirme Ölçüt Sonuçları

Aşağıda verilen şekilde Logistic Regression modeline ait olan confusion matrixi gösterilmektedir.



Şekil 13 : Logistic Regression Modeli Confusion Matrixi

E. Model Performansını Değerlendirme Ölçütleri

Accuracy, precision, recall, f1-score ve support, model performans değerlendirmesi için kullanılan temel ölçütlerdir.

Precision, modelin doğru pozitif tahminleri (TP) ile yanlış pozitif tahminlerini (FP) arasındaki oranı ifade eder. Formül olarak,

$$\text{Precision} = TP / (TP + FP)$$

Recall, modelin doğru pozitif tahminleri (TP) ile kaçırdığı pozitif tahminleri (FN) arasındaki oranı ifade eder. Formül olarak:

$$\text{Recall} = TP / (TP + FN)$$

F1-score, precision ve recall'un harmonic ortalamasıdır. Formül olarak:

$$\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Support, her sınıf için veri setinde kaç adet örnek olduğunu ifade eder.

Accuracy, modelin tahminlerinin doğru olma oranını ifade eden bir performans değerlendirme ölçütüdür. Formül olarak:

$$\text{Accuracy} = (\text{Doğru Tahminler}) / (\text{Toplam Tahminler})$$

Model performansını değerlendirmek için kullanılan bir diğer yöntem ise Confusion Matrix'dir. Confusion Matrix, bir sınıflandırma problemi için kullanılıyorsa, genellikle NxN boyutunda bir matristir (N, sınıfların sayısıdır). Her bir sütun, gerçek sınıfları temsil ederken, her bir satır ise tahmin edilen sınıfları temsil eder. Matris içinde yer alan her bir hücre, tahmin edilen sınıfın gerçek sınıfın içinde kaç adet olduğunu ifade eder. Confusion matrix, modelin performansının nerede eksik olduğunu veya hangi sınıflar için daha iyi performans sergilediğini anlamak için kullanılabilir.

F. Kullanılan Modellerin Performans Karşılaştırılması

Bu bölümde aynı veri seti kullanılarak geliştirilen 3 farklı modelin accuracy, recall, precision ve f1-score değerleri karşılaştırılacaktır.

Geliştirilen 3 farklı modelin accuracy değerleri :

- Custom model => 0,65
- LSVM Model => 0,23
- Logistic Regression => 0,32

Bu doğruluk değerlerinden yola çıkarak performans sıralaması yapacak olursak;

1. Custom model
2. Logistic Regression
3. LVSM Model

Ancak yalnızca accuracy değeri modeller arasındaki farklı değerlendirmek için yeterli değildir.

Geliştirilen 3 farklı modelin precision değerleri :

- Custom model => 0,67
- LSVM Model => 0,50
- Logistic Regression => 0,49

Precision değer sıralaması sırasıyla Custom, LVSM ve Logistic Regression şeklindedir. Bu custom modelin LSVM modelden, LSVM modelin ise Logistic Regression modelinden daha doğru tahminler yaptığı anlamına gelir. Özellikle, sınıflandırma problemlerinde, precision değeri diğer metriklerden daha önemlidir çünkü sınıflandırma modelinin ne kadar doğru olduğunu ölçer.

Geliştirilen 3 farklı modelin recall değerleri :

- Custom model => 0,65
- LSVM Model => 0,18
- Logistic Regression => 0,26

Custom modelin recall değeri, LSVM ve Logistic Regression modellerinden daha yüksektir. Bu, custom modelin diğer iki modelden daha çok pozitif olarak sınıflandırılmış olan verileri tespit ettiği anlamına gelir. Logistic Regression modelinin de recall değeri LSVM modelin recall değerinden daha yüksektir. Bu da, Logistic Regression modelinin LSVM modelden daha çok pozitif olarak sınıflandırılmış olan verileri tespit ettiği

anlamına gelir. Özellikle, sınıflandırma problemlerinde, recall değeri önemlidir çünkü sınıflandırma modelinin ne kadar çok veriyi doğru sınıflandırdığını ölçer.

Geliştirilen 3 farklı modelin f1- score değerleri :

- Custom model => 0,65
- LSVM Model => 0,23
- Logistic Regression => 0,32

Custom modelin F1-Score değeri, LSVM ve Logistic Regression modellerinden daha yüksektir. Bu, custom modelin diğer iki modelden daha iyi performans gösterdiği anlamına gelir. F1-Score, precision ve recall değerlerinin harmonic ortalamasını verir ve sınıflandırma problemlerinde hem doğru tahmin yapma oranını (precision) hem de pozitif olarak sınıflandırılmış verileri tespit etme oranını (recall) ölçer. Yüksek bir F1-Score değeri, hem doğru tahmin yapmayı hem de pozitif olarak sınıflandırılmış verileri tespit etmeyi iyi yapabilen bir modeli gösterir.

IV. LİTERATÜR ARAŞTIRMASI

Web, neredeyse tüm insan faaliyetleri için ana bilgi kaynağıdır. Web sayfaları katlanarak büyüyor ve daha karmaşık hale geliyor, bu da arama motorları, öneri sistemleri ve Web dizinleri için ciddi bir zorluk oluşturuyor. Bu karmaşık ve geniş içeriği düzenlemek için, genellikle HTML kodunun metinsel analizine dayanan web sayfası sınıflandırması temel bir tekniktir [3].

İnternet, katlanarak büyüyen çok büyük miktarda veri içerir. Bu verilerden yararlanmak için, bir Web bilgi alma sistemi ve web sayfalarının sınıflandırılmasına dayalı olarak internet içeriğinin sınıflandırılması esastır. Web sayfası sınıflandırması, aralarında web dizinlerinin oluşturulması ve odaklanmış tarayıcıların oluşturulması gibi birçok uygulamaya sahiptir [4].

Web sitesi sınıflandırması, makine öğrenimi ve doğal dil işlemede önemli bir alandır. Siber güvenlikten Çevrimiçi Mağaza Sınıflandırmalarına kadar birçok kullanım durumu vardır. Web sitesi sınıflandırmasının önemli bir kısmı, bu amaçla özel makine öğrenimi modellerinin kullanılabileceği web sitelerinden (standart öğeleri kaldırarak) ilgili metnin çıkarılmasıdır. Metin sınıflandırmasının kendisi için, SVM gibi standart olanlardan LSTM veya transformatör modelleri gibi daha karmaşık olanlara kadar çok çeşitli makine öğrenimi modelleri kullanılabilir [5].

Metin sınıflandırması için birçok farklı model vardır. Bunlardan aşağıdabahsedilmektedir.

İstatistiksel öğrenme teorisinden geliştirilen SVM, yüksek genelleme performansı ve yüksek boyutlu sınıflandırmayı işleme toleranslı yeteneği nedeniyle metin sınıflandırması için geniş çapta araştırılmakta ve kullanılmaktadır.[7]. Destek vektör makinesi (SVM), ayırıcı bir hiper düzlem tarafından tanımlanan ayrımcı bir sınıflandırıcıdır. Hiper düzlem, bir düzlemi iki parçaya bölen bir çizgidir. SVM algoritmasındaki ana amaç, veri noktalarını belirgin şekilde sınıflandıran N-boyutlu bir uzayda optimal bir hiper düzlem bulmaktır [1]. SVM, sistemi eğitmek için verilerin bir kısmını kullanır ve eğitim verilerini temsil eden birkaç destek vektörü kullanır [6]. SVM,bir dizi etiketli eğitim verisinden işlevleri öğrenmek için bir yöntemdir. SVM, metin sınıflandırmasında gelecek vadeden sonuçlar göstermiştir. Ayrıca, diğer sınıflandırıcı türleri ile karşılaştırıldığında, SVM hem verimli hem de etkilidir [8].

LSTM ilk olarak 1997 yılında dil modelleri için önerilmiştir (Hochreiter and Schmidhuber 1997). LSTM katmanları, tekrar tekrar bağlanan bellek bloklarından oluşur ve bu bellek bloklarının her biri, üç çarpımsal kapı içerir. Kapılar, geçici bilgilerin belirli bir süre boyunca kullanılmasını sağlamak için sürekli bir tür yazma, okuma ve sıfırlama işlemi gerçekleştirir [2].

Naive Bayes her özellik için sınıflar içinde bulunma olasılıkları ve sınıfların veri üzerinde görülme olasılıklarını hesaplayarak karar veren bir modeldir. Bu sınıflandırma işleminde veri kaynağı üzerinde mutlaka bir sınıflandırma kategori tanımlamasının bulunması gerekir. Test edilecek veri, öğretilmiş veri seti üzerindeki olasılık değerlerine göre hesaplanır. Bu oranlamaya göre, test setinin hangi kategoriye daha yakın olduğu bulunur. Öğretilmiş veri sayısı arttıkça, test verisinin bulunduğu kategoriyi saptamak kolaylaşır [10].

Lojistik regresyon, makine öğreniminin istatistik alanından ödünç aldığı bir sınıflandırma tekniğidir. Lojistik Regresyon, bir sonucu belirleyen bir veya daha fazla bağımsız değişkenin olduğu bir veri kümesini analiz etmek için istatistiksel bir yöntemdir. Lojistik regresyon kullanmanın arkasındaki amaç, bağımlı ve bağımsız değişken arasındaki ilişkiyi açıklamak için en uygun modeli bulmaktır [11].

Lojistik regresyon, basit ama çok etkili bir sınıflandırma algoritmasıdır, bu nedenle sınıflandırma görevi için yaygın olarak kullanılır. Müşteri kaybı, spam e-posta, web sitesi veya reklam tıklama tahminleri, lojistik regresyonun güçlü bir çözüm sunduğu alanlara bazı örneklerdir. Hatta sinir ağı katmanları için bir aktivasyon fonksiyonu olarak kullanılır [12].

Makine Öğrenimi yaklaşımı, web sitesinin kategorisini tahmin etmenin tek yolu değildir , diğer yol, ham istatistikleri kullanmak ve önceki bölümde oluşturulan her kategorinin önceden oluşturulmuş kelime sıklık listesinden bir tahmin modeli oluşturmaktır [1]. Terim sıklığı veya ters belge sıklığı Kısaca tf-idf, bir terimin bir belgede ne kadar önemli olduğunun bir ölçüsüdür [9].

Kategoriler en popüler kelimeler olarak bilindiğinden, custom bir model geliştirmek ve bir URL kategorisini tahmin etmek mümkündür. Custom model oluşturma 2 adım gerektirir:

1. Kategoriler özellikleri oluşturma
2. Kelimelerin ağırlığını hesaplayın

Custom model, her kategorinin ağırlığını belirleyerek web sitesinin kategorisini tahmin eder [1].

- Özellikler tüm kategoriler için hesaplanır.
- Her özelliğin ağırlığı aşağıdaki formülle hesaplanır: ağırlık += 2500 – özellik değerinin 1'e eşit olduğu dizin
- En yüksek ağırlık toplamına sahip kategoriyi belirler

IV. SONUÇ VE BULGULAR

Web sitelerinin sınıflandırılması, internetteki içeriğin daha iyi anlaşılmasını ve kullanıcıların aradıkları bilgiye daha hızlı ve etkili bir şekilde ulaşmasını sağlar. Ayrıca, güvenlik ve güvenlik konularının yanı sıra reklam yönetimi ve pazarlama stratejileri için de önemlidir.

Doğal dil işleme (Natural Language Processing) Custom modeli, standart NLP modelinden farklı olarak, özel bir görev için tasarlandığı anlamına gelmektedir. Örneğin,standart NLP modeli genellikle dil öğrenme, metin sınıflandırma ve özetleme gibi genelgörevler için tasarlandığı için bu modeller genellikle

yüksek performans göstermektedir.Ancak bazen, standart NLP modelinin çözemeyeceği özel bir görev olabilir ve bu durumda, özel bir NLP modeli oluşturulmaktadır.

Custom model, özel olarak tasarlandığı için, özel bir görev için optimize edilebilir ve bu sayede diğer modellerden daha iyi performans gösterebilir. Bu nedenle çalışmamızdaki ana model Custom model olacak şekilde tercih edilmiştir.

Web sitesi sınıflandırma için geliştirilen Custom, LSVM ve Logistic Regression modellerinin accuracy, precision, recall ve f1-score değerleri "Kullanılan Modellerin Performans Karşılaştırması" başlığı altında incelenmiştir. Ve bu modeller karşılaştırıldığında en iyi performans gösteren modelin Custom model olduğu görülmektedir.

REFERANSLAR

[1] Domantas Meidus (2019) "Website Classification Using Machine Learning Approaches", Bachelour Thesis

[2] Mehmet Salih Kurt, Eylem Yücel (2022) , *"WEB PAGE CLASSIFICATION WITH DEEP LEARNING METHODS",* Article

[3] H. González, I. López, P. Bringas, H. Quintián, E.Corchado , Webpage Categorization Using Deep Learning (2021).

[4] S. Lassri, E.Benlahmar, A. Tragha , Machine Learning for Web Page Classification: A Survey ,International Journal of Information Science and Technology (2019)

[5] Senior Quant, medium : Website categorization, URL: https://medium.com/website- categorization

[6] Rung-Ching Chen , Chung-Hsun Hsieh , Web page classification based on a support vector machine using a weighted vote schema (2006)

[7] W. Xue, H. Bao, W. Huang, Y. Lu , Web Page Classification Based on SVM (2006)

[8] A. Sun, E. LIM , Web classification using Support Vector Machine (2002)

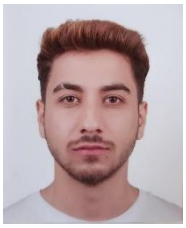
[9] Tobias Eriksson , Automatic web page Categorization using text classification methods 2013

[10] Filiz Erten, Metin Madenciliği Tabanlı Bir Web Sitesi Sınıflandırman Aracı Tasarımı (2015), Yüksek Lisans Tezi

[11] Ashwin Raj, Perfect Recipe for Classification Using Logistic Regression (2020)

[12] Soner Yıldırım, How is Logistic Regression Used as A Classification Algorithm? (2020)

Veri Seti Kaynağı:
https://data.world/crowdflower/url-categorization



Doğancan Karakoç 2001 yılının Nisan ayında Malatya'da doğmuştur. 2019 yılında Sümer Dört Renk Anadolu Lisesinden 94 ortalamayla mezun olmuştur. 2019 yılında Konya Teknik Üniversitesin'de bilgisayar mühendisliği bölümüne başlamıştır. 2020 yılında Bursa Uludağ Üniversitesine yatay geçiş ile geçiş yapmıştır. Şu an bilgisayar mühendisliği 4.sınıf öğrencisidir ve ortalaması 3.16 'dır. 2022 yılında Netssi şirketinde yazılım test otomasyon mühendisi pozisyonunda 2 aylık bir iş deneyimi olmuştur.Şu an da İnterprobe İnformation Tecnology şirketinde aday mühendis pozisyonunda 7 aydır çalışmaktadır.

Dilara Özdemir 2001 yılının Haziran ayında İzmir'de doğmuştur. 2019 yılında Bornova Anadolu Lisesinden 94 ortalama ile mezun olmuştur. 2019 yılında Bursa Uludağ Üniversitesin'de bilgisayar mühendisliği bölümüne başlamıştır. Şu an bilgisayar mühendisliği 4 .sınıf öğrencisidir ve ortalaması 3.37'dir. 2022 yılında Groupe Renault'da proje stajyeri pozisyonunda 2 aylık , Teracity Yazılım Teknolojileri şirketinde ise stajyer pozisyonunda 3 aylık bir deneyimi olmuştur.