Manisa Celal Bayar Üniversitesi

Hasan Ferdi Turgutlu Teknoloji Fakültesi

Yazılım Mühendisliği

YZM_3226 Makine Öğrenmesi - Doç.Dr.Akın ÖZÇİFT

Vize Ödevi

Mücahit Toktaş 172803036

İÇİNDEKİLER:

1- PROBLEM

“Makine öğrenmesi, analitik model oluşturmayı otomatikleştiren bir veri analizi yöntemidir. Sistemlerin verilerden öğrenebileceği, kalıpları belirleyebileceği ve minimum insan müdahalesi ile kararlar alabileceği fikrine dayanan yapay zekanın bir dalıdır.”

Bu ödevimizde bir makine öğrenmesi modeli oluşturmadan önce gerekli olan VERİ TOPLAMA ve VERİ ÖN İŞLEME aşamalarını gerçekleştirmiş bulunmaktayız.

Bu projede çok bilindik bir web sitesinden araba verileri çekerek bu verileri işliyoruz, amacımız bu veriler ışığında araç markasını tahmin etmek veya daha farklı tahminlerde bulunması için bir makine öğrenmesi modeli oluşturmak…

2- BEAUTİFULSOAP

BeautifulSoup, HTML veya XML dosyalarını işlemek için oluşturulmuş güçlü ve hızlı bir kütüphanedir. Bu modül ile bir kaynak içerisindeki HTML kodlarını ayrıştırıp sadece istediğimiz alanlardaki bilgileri çekebiliriz.

Bu ödevde bende BeautifulSoup yöntemi ile veri toplamayı tercih etmiş bulunmaktayım.

Projede BeautifulSoap ile veri toplama kısmını incelersek;

```
import pandas as pd
from bs4 import BeautifulSoup
import requests
```

```
head_param ={ "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198
```

```
r=requests.get("https://www.sahibinden.com/otomobil/bolu?pagingOffset=20&pagingSize=50",headers=head_param)
soup = BeautifulSoup(r.content,"html.parser")
```

```
araclar=soup.find_all("tbody",attrs={"class":"searchResultsRowClass"})
```

```
for arac in araclar:
    arac2 = arac.find_all("tr", attrs={"class":"searchResultsItem"})
    for arac3 in arac2:
        try:
            arac_link = arac3.find("a").get("href")
        except:
            pass

        link_basi = "https://www.sahibinden.com"
        link = link_basi+arac_link
        print(link)


        r_arac=requests.get(link, headers=head_param)
        arac_soup = BeautifulSoup(r_arac.content, "lxml")

        detaylar = arac_soup.find_all("ul", attrs={"class":"classifiedInfoList"})

        for i in detaylar:
            li=i.find_all("li")
            liste = []
            for a in li:
                try:
                    ozellikler = a.find("span").text
                    liste.append(ozellikler)
                except:
                    pass

            df_liste=pd.DataFrame(liste).T
            df_liste.to_csv(r"sahibinden_arabaverileri.csv", encoding="utf-8",index=False,mode="a")
```

```
https://www.sahibinden.com/ilan/vasita-otomobil-tofas-dosta-gider-temiz-aile-araci-takasa-acik-881751358/detay
https://www.sahibinden.com/ilan/vasita-otomobil-skoda-2018-model-48.000-km-de-skoda-octavi-1.6-tdi-otomatik-boyasiz-882016491/detay
https://www.sahibinden.com/ilan/vasita-otomobil-volkswagen-1.4-tsi-hiqhiline-dsg-u-ledkusursuz-882013324/detay
https://www.sahibinden.com/ilan/vasita-otomobil-volkswagen-1.4-tsi-hiqhiline-dsg-u-ledkusursuz-882013324/detay
https://www.sahibinden.com/ilan/vasita-otomobil-peugeot-degisensiz-301-882001617/detay
```

BeautifulSoup Modülünü ekleyerek başladığımız projemizde head_param değişkenimize User-Agent imizi ekleyerek devam ediyoruz. User-Agent imiz ile birlikte verilerini çekmek istediğimiz adrese istekte bulunuyoruz. Cevap sonucu BeatifulSoup “Html.Parser” ile parçalama işlemini gerçekleştirip soup değişkenine atıyoruz. Verisini çekmek istediğimiz sayfayı incelersek o sayfada ilanları barındıran (ki o ilanlar bizim hedefimiz olan ilanlar) bir <tbody> bulunmakta. <tbody> nin adı ile içindeki ilan linklerini çekiyoruz. Çektiğimiz linklere gidip o adreste ilan detaylarının saklandığı <ul> nin içinden, <li> lerin, larını yani hedefimiz olan bilgileri topluyoruz.

*(User agent kullanma sebebimiz tarayıcı taklidi yapıp bağlanmak istediğimiz sitenin cevap vermesini sağlamaktır)

3- VERİ SETİ HAKKINDA

Veri setimizde araç bilgileri ile ilgili başlangıçta ;

İlan id – ilan tarihi – marka – seri – model – yıl – yakıt türü – vites tipi – kilometre – kasa tipi – motor gücü – motor hacmi – çekiş tipi – rengi – garantisi – plaka uyruğu – kimden satılık olduğu – görüntülü arama ile görme – takas seçeneği – sıfır/ikinci el durumu

Sütunları vardı.

“” df_2 . drop (' ilan_id ' , axis = 1 , inplace = True) “”

Yukarıdaki kod parçası ile

İlan id – ilan tarihi – seri – model – görüntülü arama

Sütunlarını kaldırdım.

Çünkü çok fazla seçenekte seri ve ondan kat kat fazlası model seçeneği vardı. 500 satır küsurluk bir veri setinde bunun gereksiz olduğu düşündüm ayrıca çok detay özellikler…

Ayrıca tarih ve ilan id ‘leri ile de bir işimiz yoktu.

Görüntülü arama seçeneğinin çektiğimiz veri özelinde yani oluşturacağımız model hedefinde bir işe yarayacağınıda düşünmedim.

Belki daha silinmesi gerekli sütunlarda vardır fakat veri setinin zenginliğini kaybetmesini istemedim.

4 – VERİ ÖN İŞLEME AŞAMALARI HAKKINDA

Veri ön işlemeye öncelikle sütunlardaki \n\t , \n , cc, hp leri temizleyerek başladım,

\n \tDizel	\n \tYarı Otomatik	220.00	\n \tSedan	\n \t110 hp	\n \t1461 cc	\n \tÖnden Çekiş	\n \tBeyaz	\n \tHayır	\n \tTürkiye (TR) Plakalı	\n \tGaleriden	\n Evet \n	\n İkinci El \n
\n \tDizel	\n \tManuel	78.32	\n \tHatchback 5 kapı	\n \t90 hp	\n \t1461 cc	\n \tÖnden Çekiş	\n \tGri	\n \tHayır	\n \tTürkiye (TR) Plakalı	\n \tSahibinden	\n Hayır \n	\n İkinci El \n
\n \tBenzin & LPG	\n \tYarı Otomatik	2.70	\n \tSedan	\n \t125 hp	\n \t1597 cc	\n \tÖnden Çekiş	\n \tBeyaz	\n \tEvet	\n \tTürkiye (TR) Plakalı	\n \tGaleriden	\n Hayır \n	\n İkinci El \n
\n \tBenzin	\n \tOtomatik	235.30	\n \tSedan	\n \t333 hp	\n \t4398 cc	\n \tArkadan İtiş	\n \tMavi	\n \tHayır	\n \tTürkiye (TR) Plakalı	\n \tGaleriden	\n Hayır \n	\n İkinci El \n
\n \tDizel	\n \tManuel	103.00	\n \tHatchback 5 kapı	\n \t75 hp	\n \t1461 cc	\n \tÖnden Çekiş	\n \tBeyaz	\n \tEvet	\n \tTürkiye (TR) Plakalı	\n \tGaleriden	\n Hayır \n	\n İkinci El \n

Görüldüğü üzere sütunlardaki ifadeler çok kötü gözükmekte

“” df_2 [" ilgili_sütun "] = df_2 . ilgili_sütun . str [x : y] “”

Yukardaki kod parçasında bulunanan x’e verdiğim pozitif değerler metnin başından itibaren x kadar boşluk-harf silmekte iken y de aynı işi verilen negatif değerler ile sondan başlayarak gerçekleştirmektedir.

Dizel	Yarı Otomatik	220.00	Sedan	110	1461	Önden Çekiş	Beyaz	Hayır	Türkiye (TF Plaka
Dizel	Manuel	78.32	Hatchback 5 kapı	90	1461	Önden Çekiş	Gri	Hayır	Türkiye (TF Plaka
Benzin & LPG	Yarı Otomatik	2.70	Sedan	125	1597	Önden Çekiş	Beyaz	Evet	Türkiye (TF Plaka
Benzin	Otomatik	235.30	Sedan	333	4398	Arkadan İtiş	Mavi	Hayır	Türkiye (TF Plaka
Dizel	Manuel	103.00	Hatchback 5 kapı	75	1461	Önden Çekiş	Beyaz	Evet	Türkiye (TF Plaka

12 sütunda bu temizleme işlemini gerçekleştirdikten sonra verilerim daha anlaşılır ve güzel gözükmekte. Ardından kategorik ifadeleri etiketlemeye başladım….

Etiketleme işlemi için sklearn kütüphanesi içinden LabelEncoder nesnesini ve OneHotEncoder nesnesini kullanıcam.

LabelEncoder elimizdeki verileri direk sayısal temsillere dönüştürür ve kategorik her veriye sayısal bir değer atar.

OneHotEncoding yaklaşımında ise kategorik türde özniteliğe ait tüm değerler yeni birer öznitelik haline getirilir.

Bu verisetinde bana LabelEncoder yaklaşımı yeterli idi , her sutunda veriler sayısal hale getirilecekti, OneHotEncoder yaklaşımı araç markaları ve renkleri sütunları sebebi ile çok çok fazla sütun açılmasına sebep olur ve verisetimizi gereksiz yere çok büyütürdü.

Öte yandan yakıt tipi gibi seçenekleride OneHotEncoder ile kategorilere bölmek Dizel / Benzin / Benzin-LPG olmak üzere 3 sütunda ele almakta LabelEncoder'i gereksiz kılmakta.

Bende her iki yaklaşımıda kullanmayı seçtim(makine öğrenmesi modeli oluşturma kısmında bir problem yaratır mı bilmemekteyim!!!) .

Araç marka ve renk sütunlarında LabelEncoder kullandım.

“” df_3 [" marka "] = le . fit_transform (df_2 . marka) “”

	marka	yıl	yakıt	vites	KM	kasatipi	motor_gücü	motor_hacmi	çekiş	renk	g
0	Renault	2012	Dizel	Yarı Otomatik	220.00	Sedan	110	1461	Önden Çekiş	Beyaz	
1	Dacia	2014	Dizel	Manuel	78.32	Hatchback 5 kapı	90	1461	Önden Çekiş	Gri	
2	Honda	2020	Benzin & LPG	Yarı Otomatik	2.70	Sedan	125	1597	Önden Çekiş	Beyaz	
3	BMW	2003	Benzin	Otomatik	235.30	Sedan	333	4398	Arkadan İtiş	Mavi	
4	Renault	2017	Dizel	Manuel	103.00	Hatchback 5 kapı	75	1461	Önden Çekiş	Beyaz	

...

	marka	yıl	KM	motor_gücü	motor_hacmi	renk	g
)	19	2012	220.00	110	1461	1	
1	5	2014	78.32	90	1461	4	
2	9	2020	2.70	125	1597	1	
3	2	2003	235.30	333	4398	9	
4	19	2017	103.00	75	1461	1	
	...	...	...	...	...	...	

le.classes_ komutu ile etiketlediğimiz sutunun içindeki verileride görebilmekteyiz.

```
: df_3["marka"] = le.fit_transform(df_2.marka)
  le.classes_
```

```
: array(['Alfa Romeo\xa0', 'Audi\xa0', 'BMW\xa0', 'Chevrolet\xa0',
         'Citroën\xa0', 'Dacia\xa0', 'Daewoo\xa0', 'Fiat\xa0', 'Ford\xa0',
         'Honda\xa0', 'Hyundai\xa0', 'Kia\xa0', 'Mazda\xa0',
         'Mercedes - Benz\xa0', 'Mini\xa0', 'Mitsubishi\xa0', 'Nissan\xa0',
         'Opel\xa0', 'Peugeot\xa0', 'Renault\xa0', 'Rover\xa0', 'Seat\xa0',
         'Skoda\xa0', 'Suzuki\xa0', 'Tofaş\xa0', 'Toyota\xa0',
         'Volkswagen\xa0', 'Volvo\xa0'], dtype=object)
```

```
: df_3["renk"] = le.fit_transform(df_2.renk)
  le.classes_
```

```
: array(['Bej', 'Beyaz', 'Bordo', 'Füme', 'Gri', 'Gümüş Gri', 'Kahverengi',
         'Kırmızı', 'Lacivert', 'Mavi', 'Mor', 'Sarı', 'Siyah', 'Turkuaz',
         'Yeşil', 'Şampanya'], dtype=object)
```

Araç yakıt, çekiş_tipi, kasa_tipi, vites_tipi, durumu, kimden, plaka_uyruk sütunlarında OneHotEncoder kullandım.

“” df_2 [' vites '] = pd . Categorical (df [' vites '])

dfDummies3 = pd . get_dummies (df_2 [' vites '] , prefix = ' ') “”

Etiketleme işlemlerinden de sonra sadece evet ve hayır seçenekleri olan “takas”,”garanti” sütunlarınıda “loc” fonksiyonu ile 0-1 e çevirdim.

“” df_3 . loc [df_2 . takas = = ' Evet ' , " takas "] = 1 “”

_Coupe	_Hatchback 3 kapı	_Hatchback 5 kapı	_MPV	_Sedan	_Station Wagon	_Galeriden	_Sahibinden	_Sıfır	_İkinci El
0	0	0	0	1	0	1	0	0	1
0	0	1	0	0	0	0	1	0	1
0	0	0	0	1	0	1	0	0	1
0	0	0	0	1	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...
0	0	0	0	1	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1
0	0	0	0	1	0	1	0	0	1
0	0	0	0	1	0	1	0	0	1

Veri toplama ve Ön işleme kısımlarım bu kadardı.

Ön işlemeyi etiketleme öncesi ve sonrası olarak iki farklı aşamada ele aldım,

Excel’de de aynı şekildedir. Toplam 3 excel sayfası bulunmaktadır.