

**T.C.**

**İSTANBUL ÜNİVERSİTESİ**

**İŞLETME FAKÜLTESİ İŞLETME İKTİSADI ENSTİTÜSÜ**

**İKİNCİ ÖĞRETİM YÖNETİM BİLİŞİM SİSTEMLERİ MS**

**TEZSİZ YÜKSEK LİSANS PROGRAMI**

**SARI ONLINE PLATFORMDAKİ SUV ARAÇ İLANLARINDAN HAZIRLANAN VERİ SETİNE SINIFLANDIRMA ALGORİTMALARININ UYGULANMASI**

**Proje Danışmanı: Prof. Dr. Emrah ÖNDER**

**Hazırlayan: Alican Alcan**

**DÖNEM PROJESİ**

**Haziran 2023**

**PROJE ONAY**

İşletme Fakültesi İşletme İktisadı Enstitüsü 9501228952 numaralı öğrencilerinden Alican Alcan ’nın bitirme projesi olarak yaptığı “Sarı Online Platformdaki SUV Araç İlanlarından Hazırlanan Veri Setine Sınıflandırma Algoritmalarının Uygulanması” başlıklı çalışması, .…/..../…..... tarihinde, değerlendirilerek başarılı/başarısız bulunmuştur.

**Bitirme Projesi Danışmanı**

**Prof. Dr. Emrah ÖNDER**

**BEYAN**

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez çalışması olarak sunulmadığını beyan ederiz.

## Alican Alcan

## 2023

# 

# **ÖNSÖZ**

Bu tez çalışmasının her aşamasında bilgi birikimi, deneyim ve yönlendirmelerini hiçbir zaman esirgemeyen, her soruma sabır ve içtenlikle cevap veren, saygıdeğer danışmanım Prof. Dr. Emrah ÖNDER hocama teşekkür ederim.

**Alican Alcan**

**İÇİNDEKİLER**

Sayfa

BEYAN.........................................................................................................................................I ÖNSÖZ........................................................................................................................................II İÇİNDEKİLER ..............................................................................................................................III GÖRSELLER LİSTESİ……..………………………………………………………………………………...........................V   
GİRİŞ...........................................................................................................................................1   
**1.BÖLÜM: Veri............................…………………………………………………......................................2**   
1.1. Veri Toplama.................…………………………………………………………………………………….….………2  
1.1.1. Veri Toplama Yöntemleri.............……………………………………………………………………….…….….2  
1.2. Veri Madenciliği................................................……………………………………………….……....…….3  
1.2.1. Veri Madenciliği Çalışma Prensibi.....................................................................................3  
1.2.2. Veri Madenciliği Yöntemleri...……………………………………………………………………….……..………3  
1.2.3.Python İle Veri Madenciliği......……………………………………………………………………….…………….3  
1.3. Sarı Online Platform (sahibinden.com)...………………………………………………………….………..……4  
1.3.1. SUV İlanları............................................………………………………………………………….……..…….4  
**2.BÖLÜM: Veri Ön İşleme..............……………………………………………………………...……….……..……5**  
2.1. Bozuk Veri...........................................................................................................................5  
2.1.1. Kayıp Verileri Tamamlama.............……………………………………………………………..……….……...5  
2.1.2. Gürültülü Veriyle Uğraşma..................................…………………………………..…….…….………..5  
2.2. Boyut İndirgeme..............................………………………………………………………………………………..6  
2.2.1. Özellik Seçimi....................................................................................................................6  
2.2.2. Space Transformations..............................................……………………………………………..…….6  
2.2.3. Instance Reduction..............……………………………………………………………………………….……….7  
2.2.4. Instance Selection..................................……………………………………………………………………….7  
2.2.5. Instance Generation.........................................................................................................7  
2.2.6. Discretization.........................................................................................…………………..…..7  
2.3. Dengesiz (Imbalanced) Öğrenme, Örneklem Azaltma (Undersampling) ve Örneklem Artırma (Oversamling) Yöntemleri.............................................................................................8  
2.4. Veri Madenciliğinin Yeni Alanlarında Veri Ön İşleme.......................………………………………….8  
**3.BÖLÜM: Sınıflandırma Algoritmaları……………………………………………...………………………..…….9**  
3.1. Sınıflandırma Algoritmalarının Temel İlkeleri...........…………………...………………………..…………9  
3.2. Doğrusal Sınıflandırma Algoritmaları.....………………………………………………………………………..10  
3.3. Doğrusal Sınıfladırma Algoritmaları...................................................................................10  
3.4. Derin Öğrenme ve Sınıflandırma.......................................................................................11  
3.5. Sınıflandırma Algoritmalarının Uygulama Alanları.............................................................11  
3.6. Sık Kullanılan Sınıflandırma Algoritmaları..........................................................................12  
3.6.1. Naive Bayes....................................................................................................................12  
3.6.2. Karar Ağaçları.................................................................................................................12  
3.6.3. Rastgele Ormanlar..........................................................................................................13  
3.6.4. Lojistik Regresyon...........................................................................................................13 3.6.5. Destek Vektör Makineleri...............................................................................................14  
3.6.6. K-En Yakın Komşu...........................................................................................................14  
3.7. Sınıflandırma Algoritmalarının Değerlendirmesi...............................................................15  
3.7.1. Değerlendirme Metrikleri...............................................................................................15 3.7.1.1.Doğruluk (Accuracy).....................................................................................................15 3.7.1.2. Hassasiyet (Precision)..................................................................................................15 3.7.1.3. Geri Çağırma(Recall)....................................................................................................15 3.7.1.4. F1 Skoru.......................................................................................................................15 3.7.1.5. ROC Eğrisi....................................................................................................................15 **4.BÖLÜM: Python İle Uygulama.........................................................................................16** 4.1 Anaconda ve Spyder Kurulumu...........................................................................................16 4.2. Spyder’da Gereken Kütüphaneleri İndirmek.....................................................................18 4.2.1. Pandas............................................................................................................................19 4.2.2. Selenium ........................................................................................................................19 4.2.2.1. WebDriver...................................................................................................................19 4.2.3. xlsxwriter........................................................................................................................19 4.2.4. scikit-learn......................................................................................................................19 4.3. sahibinden.com Sitesinden Veri Seti Hazırlama.................................................................20 4.4. Karar Ağacı Algoritması .....................................................................................................24 4.4.1. Karar Ağacı Algoritması Model İyileştirmesi...................................................................25 4.5. Rastgele Orman Algoritması..............................................................................................26 4.5.1. Rastgele Orman Algoritması Modelinin İyileştirmesi......................................................28 5.BÖLÜM: SONUÇ VE DEĞERLENDİRME...................................................................................29 KAYNAKÇA ...............................................................................................................................30

**GÖRSELLER LİSTESİ**

Şekil 1: Veri Toplama (www.masplus.com.tr/veri-toplama-nedir/)

Şekil 2: Python ve Selenium (gainanov.pro/eng-blog/dev/selenium-python/)

Şekil 3: sahibinden.com (egirisim.com/2020/01/03/sahibinden-comun-aylik-kullanici-sayisi-468-milyona-ulasti)

Şekil 4: Veri Ön İşleme (www.veribilimiokulu.com/buyuk-veri-on-isleme-makale-notlari/)

Şekil 5: Özellik Seçimi (miro.medium.com/v2/resize:fit:720/format:webp/1\*Ks4FOGo3-Aj5Kzklqc\_qAw.png)

Şekil 6: Space Transformations (www.statistics4u.com/fundstat\_eng/cc\_transform\_of\_dataspace.html)

Şekil 7: Örnek Seçim Azaltması ve Oluşturması (www.sciencedirect.com/science/article/abs/pii/S0950705112003140)

Şekil 8: Discretization (www.geeksforgeeks.org/ml-binning-or-discretization/)

Şekil 9: Örneklem Azaltma veya Artırma (www.researchgate.net/figure/Differences-between-undersampling-and-oversampling\_fig1\_340978368)

Şekil 10: Makine Öğrenmesi Yöntemleri (miro.medium.com/v2/resize:fit:4800/format:webp/1\*AOh7p7f0MYj-fkHSrrNdGg.png)

Şekil 11: Doğrusal ve Doğrusal Olmayan Sınıflandırma Algoritmaları (devhunteryz.wordpress.com/2018/06/30/derin-ogrenme-perceptron-ogrenme-algoritmasi)

Şekil 12: Derin Öğrenme (www.mouser.com/blog/development-of-deep-learning)

Şekil 13: Naive Bayes (editor.analyticsvidhya.com/uploads/23385Capture6.PNG)

Şekil 14: Karar Ağaçları (medium.com/@sanlian/makine-%C3%B6%C4%9Frenmesi-3-knn-cart-c6d201e7278f)

Şekil 15: Rastgele Ormanlar (serokell.io/blog/random-forest-classification)

Şekil 16: Lojistik Regresyon (www.voxco.com/blog/logistic-regression/)

Şekil 17: Destek Vektör Makineleri (en.wikipedia.org/wiki/Support\_vector\_machine)

Şekil 18: K-En Yakın Komşu (www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning /)

Şekil 19: Anaconda Hoşgeldin

Şekil 20: Anaconda Sözleşme

Şekil 21: Anaconda Profil

Şekil 22: Anaconda Yükleme

Şekil 23: Anaconda Gelişmiş Kurulum Seçenekleri

Şekil 24: Anaconda Navigator İlk Ekran

Şekil 25: Spyder İlk Ekran

Şekil 26: Pandas

Şekil 27: Selenium

Şekil 28: scikit-learn

Şekil 30: Kod Görseli 1

Şekil 31: Kod Görseli 2

Şekil 32: Kod Görseli 3

Şekil 33: Kod Görseli 4

Şekil 34: Kod Görseli 5

Şekil 35: sahibinden.com SUV İlanları

Şekil 35: Kod Görseli 6

Şekil 36: Kod Görseli 7

Şekil 37: Veri Setinin Excel’de Görünümü 1

Şekil 38: Veri Setinin Excel’de Görünümü 2

Şekil 39: Kod Görseli 8

Şekil 40: Kod Görseli 9

Şekil 41: Veri Setinin Spyder’da Görünümü

Şekil 42: Kod Görseli 10

Şekil 43: Spyder’da Bağımsız ve Bağımlı Değişkenler

Şekil 44: Kod Satırı 11

Şekil 45: Kod Satırı 12

Şekil 46: Kod Satırı 13

Şekil 47: Kod Satırı 14

Şekil 48: Karar Ağacı Doğruluk Oranı ve Hata Matrisi

Şekil 49: Kod Satırı 15

Şekil 50: Kod Satırı 16

Şekil 51: Kod Satırı 17

Şekil 52: Kod Satırı 18

Şekil 53: Karar Ağacı İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

Şekil 54: Kod Görseli 19

Şekil 55: Kod Görseli 20

Şekil 56: Veri Setinin Spyder’da Görünümü

Şekil 57: Kod Görseli 21

Şekil 58: Spyder’da Bağımsız ve Bağımlı Değişkenler

Şekil 59: Kod Satırı 22

Şekil 60: Kod Satırı 23

Şekil 61: Kod Satırı 24

Şekil 62: Kod Satırı 25

Şekil 63: Rastgele Orman Doğruluk Oranı ve Hata Matrisi

Şekil 64: Kod Satırı 26

Şekil 65: Kod Satırı 27

Şekil 66: Kod Satırı 28

Şekil 67: Kod Satırı 29

Şekil 68: Rastgele Orman İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

Şekil 69: Karar Ağacı Doğruluk Oranı ve Hata Matrisi

Şekil 70: Rastgele Orman İyileştirilmiş Hata Matrisi

Şekil 71: Karar Ağacı İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

Şekil 72: Rastgele Orman İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

## GİRİŞ

Yöntemler disipline göre değişiklik gösterse de doğru ve dürüst veri toplamaya yapılan vurgu aynı kalır. Bütün veri toplama çalışmalarının amacı, daha sonra zengin veri analizine çevirecek olan kaliteli kanıtları yakalamak ve ortaya atılan sorulara inandırıcı ve güvenilir bir cevap oluşturulmasını sağlamaktır. Bu sebep Python programlama dilinde Selenium kütüphanesi yardımı ile sahibinden.com sitesindeki SUV ilanlarından bir veri seti hazırlayacağız. Amaçlı gözlemlerin yapılması ve ilgilenilen özelliğe yönelik verilerin toplanması, sonraki aşamalardan biridir. Ölçmelerin hassasiyetini ve duyarlılığını artırmak için gözlemlerde sıklıkla bir 'ölçme aracı' ya da 'veri toplama aracı' kullanılır.

İlerleyen bölümlerde ise topladığımız verilerden oluşturduğumuz veri setini sınıflandırma algoritmalarını uygulayabilmek için fiyat bilgilerini, fiyat bilgilerinin ortanca değeri ile karşılaştırdık. Sınıflandırma kavramı, basitçe bir veri kümesi üzerinde tanımlı olan çeşitli sınıflar arasında veriyi dağıtmaktır. Sınıflandırma algoritmaları, verilen eğitim kümesinden bu dağılım şeklini öğrenirler ve daha sonra sınıfının belirli olmadığı test verileri geldiğinde doğru şekilde sınıflandırmaya çalışırlar. En çok kullanılan 5 sınıflandırma algoritması Rastgele Orman, Lojistik Regresyon, Naive Bayes, K En Yakın Komşu, Karar Ağacı ve Destek Vektör Makinesidir. Ben bu çalışmamda Rastgele Orman ve Karar Ağacı algoritmalarını uyguladım.

Çalışmamın son bölümünde ise detaylı olarak anlattığım konuları uygulamasını göstereceğim. Uyguladığımız sınıflandırmaların bulunduğu tahminlerin ne kadar doğru ne kadar yanlış tahmin yaptıklarını göreceğiz. Doğruluk oranlarını karşılaştırıp hangisinin veri seti üzerinde daha sağlıklı tahmin yaptığını göreceğiz.

**1.BÖLÜM: Veri**

Veri ya da yabancı ismiyle data, teknik tanımıyla, bir kişi ya da nesne hakkında **niteliksel ve niceliksel değişkenliklerin sayısal değeridir.**Verileri ham yani işlenmemiş ve işlenmiş veri olarak ayırmak mümkün. Ham veriler, bir uzman tarafından incelenmediği sürece diğer insanlar için pek anlam ifade etmez. İşlenmiş veriler çoğu zaman sonuca yani bilgiye ulaşma yolunda atılmış önemli bir adımdır. Ancak çoğu zaman işlenmiş veriler bile farklı bilgilerle ya da farklı verilerle birleştirilerek **analiz edilmediği takdirde**sonuç yani bilgi vermez.

**1.1. Veri Toplama**

Veri toplama, standart onaylanmış teknikler kullanılarak araştırma için doğru içgörülerin toplanması, ölçülmesi ve analiz edilmesi prosedürüdür. Bir araştırmacı hipotezini toplanan verilere dayanarak değerlendirebilir. Çoğu durumda veri toplama, araştırma alanı ne olursa olsun, araştırma için birincil ve en önemli adımdır. Veri toplama yaklaşımı, gerekli bilgilere bağlı olarak farklı çalışma alanları için farklıdır.

**1.1.1. Veri Toplama Yöntemleri**

Veri toplamak için dört seçenek vardır. Yüz yüze görüşmeler, Posta, telefon ve Online. Bu modların her birinin artıları ve eksileri vardır. Yüz yüze görüşmelerin artı yönleri veriler üzerinde derinlemesine ve yüksek derecede güven sağlamasıdır. Eksileri ise zaman alıcı, pahalı ve anekdotlara dayalı olmasıdır. Posta anketlerinin artı yönleri herkese ve her kesime ulaşılabilirdir. Eksileri ise pahalı, veri toplama hataları ve gecikmesi süresinden bahsedebiliriz. Telefon anketlerinin artı yönleri toplanan verilere yüksek derecede güvenilir ve neredeyse herkese ulaşılabilinir. Eksi yönleri ise pahalı, kendi kendine idare edemiyor ve bir ajan kiralanması gerekir. Son olarak online anketlerin artıları ucuz, kendi kendine uygulanabilir ve çok düşük veri hatası olasılığıdır. Eksi yanları ise tüm müşterilerinizin e-posta adresi olmayabilir veya internette olmayabilir. Diğer bir eksi yanı müşteriler çevrimiçi bilgi ifşa etme konusunda temkinli olabilir.

## veri-toplama-min

Şekil - Veri Toplama

**1.2. Veri Madenciliği**

Veri madenciliği diğer bir adıyla Web Scraping, web sitelerinden veri çıkarma işlemidir. Web üzerinde birçok veri bulunuyor. Veri madenciliği, çeşitli programların yardımıyla internetten yapılandırılmış ya da yapılandırılmamış verilerin çıkarılması ve çeşitli amaçlar için uygun formata dönüştürülme işlemine verilen isimdir. Veri madenciliğinde bu veri toplama işlemini otomatikleştirir. Dağıtık halde olan verileri daha düzgün şekilde sunar. İnternet’ten kazınan verilerle birçok çalışma yapılabilir.

**1.2.1. Veri Madenciliği Çalışma Prensibi**

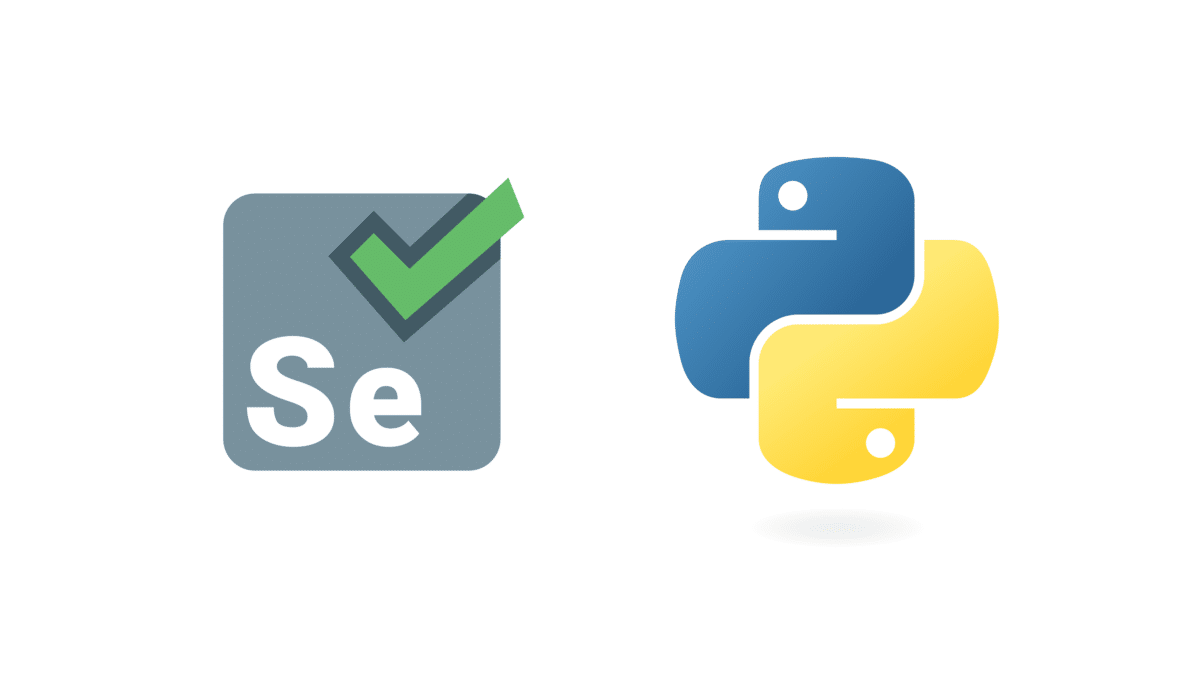
Web Scraping işlemlerini kavramak için, öncelikle web sayfalarının metin tabanlı biçimlendirme dilleri ile oluşturulduğunu anlamak önemlidir. Bir biçimlendirme dili, bir web sitesinin içeriğinin yapısını tanımlar. Evrensel bileşenlerin ve işaretleme dillerinin etiketleri bulunduğundan, web kesicilerinin ihtiyaç duyduğu bilgileri almaları çok daha kolay olur.

**1.2.2. Veri Madenciliği Yöntemleri**

Veri madenciliğinin birçok yöntemi vardır (Request, lxml, Beautiful, Selenium, Scrapy). Bunlardan birincisi, gerçek kullanıcı gibi davranan bir bot oluşturup, web sitesinden verinin kazınması, diğeri ise web sitesinin html içeriğinin indirilip, istenilen kısımların ayrıştırılmasıdır. Bu çalışmada Python programlama dilinde bir bot oluşturup, web sitesine girmek, sayfada tıklamalar yapmak, formları doldurmak, sayfayı kaydırmak gibi kullanıcı işlemleri gerektiren işlemleri otomatikleştirip web sitesinden verinin kazınmasını sağlayacağız.

**1.2.3. Python ile Veri Madenciliği (Selenium)**

Bu çalışmada Python programlama dilinde Selenium adlı kütüphanesinden yararlanacağız. Çünkü diğer bahsettiğim tüm Python kütüphanelerinde bir sınırlama var. Yani Dinamik olarak kodlanmış web sitelerinden kolayca veri çekemiyoruz. Diğer kütüphanelerin JavaScript çalıştıramadığı durumlarda Selenium bu işi yapmakta oldukça iyidir. Ancak her sayfa için JavaScript yükler ve çalıştırır, bu da doğal olarak sayfayı yavaşlatır. Bu sebepten ötürü Selenium, büyük çaplı projeler için çok tercih edilmez. Selenium kütüphanesi aslında web uygulamalarının otomatik testi için yapılmış bir Python kütüphanesidir. İlk başlarda web scraping yapılmasa da son zamanlarda veri bilimcileri tarafından çok sık kullanılan bir kütüphanedir. Selenium WebDriver birçok tarayıcıda çalışmaktadır, bu güzel bir özelliktir çünkü tarayıcının nasıl hareket ettiğini direk görebiliriz. Ek olarak Selenium WebDriver API ile birçok dilde yazabilirsiniz (Python, Java, C#, Ruby, Javascript vb.). Bu yazıda Python programlama dili ile uygulama yapacağız.



Şekil - Python ve Selenium

**1.3. Sarı Online Platform (sahibinden.com)**

sahibinden.com, gayrimenkul, vasıta, alışveriş ürünleri ve hizmetler gibi birçok kategoride ilan ve e-ticaret işlemlerinin yapıldığı online bir platformdur. 1990'lı yılların sonlarında Türkiye'deki İnternet girişimciliği henüz gelişmeye başlarken ilanlar geleneksel medyada özellikle de günlük gazeteler gibi mecralarda yayınlanıyordu. Bu dönemde sahibinden.com’un kurucusu ve Aksoy Grup Yönetim Kurulu Başkan Yardımcısı Taner Aksoy 29 Mart 2020 tarihinde Wayback Machine sitesinde arşivlendi., 1999 yılında İnternet üzerinden satıcılara görsel unsurlar da içeren detaylı ilan verme olanağı sağlayan, diğer yandan alıcılar için ilanların kolay aranmasını sağlayan bir site fikrini oluşturdu. Sonuç olarak sahibinden.com, 2000 yılında 2.700 ilanla yayın hayatına başladı. Aralık 2010 itibarıyla, vasıta, cep telefonu, gayrimenkul ve ayakkabı gibi birçok farklı kategoriden oluşan ilan ağında 2.000.000 ilan bulunmaktadır. 2009 yılında Türkiye'deki 5 bölge müdürlüğü olan Ankara, İzmir, Antalya, Trabzon ve Adana'daki ofisleri açılmıştır. Merkezi İstanbul'da bulunmaktadır. sahibinden.com hâlen Aksoy Group bünyesindedir.

**1.3.1. SUV İlanları**

İhtiyaç duyduğum veri setini hazır üstünde çalışma yapılmış almak yerine kendim hazırlamak istedim. Veri setim sınıflandırma algoritmasına uygun olması gerektiği için bir adet bağımlı değişkene ve bu bağımlı değişkeni etkileyecek bağımsız değişkenlere ihtiyaç duydum. İlanlardaki fiyat değişkeni, bağımlı değişken için uygun gözüküyor. Çünkü fiyat değişkeni aracın üretim yılına, kilometresine vb. bağımsız değişkenlere bağımlı bir değişkendir. Bu yüzden Python programlama dili ile Selenium kütüphanesinin yardımı ile 20 Nisan 2023 tarihine ait sahibinden.com sitesindeki SUV ilanlarından bir veri seti hazırladım.



Şekil - sahibinden.com

**2.BÖLÜM: Veri Ön İşleme**

Veri ön işleme, veri madenciliği modelleri kurulmadan önce veri seti üzerinde yapılan bir takım düzeltme, eksik veriyi tamamlama, tekrarlanan verileri kaldırma, dönüştürme, bütünleştirme, temizleme, normalleştirme, boyut indirgeme vb. işlemlerdir. Bu aşamada ister istemez veri üzerinde bilgi keşfi yapılmış olur.

**2.1. Bozuk Veri**

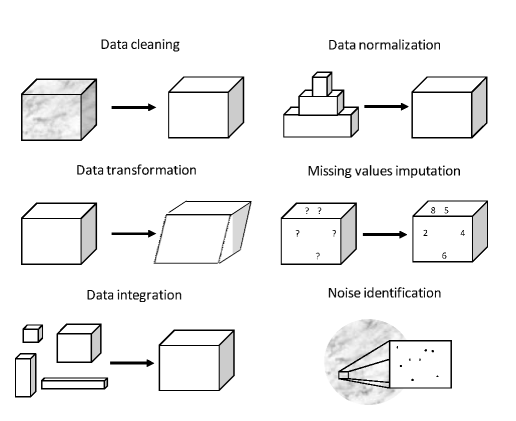
Birçok veri madenciliği algoritması girdi olarak temiz, kaliteli ve gürültüsüz veri aldığı varsayımına dayanarak çalışır. Oysaki gerçekte böyle bir veri yoktur. Yani oluştuğu esnada mükemmel olan ve hiç ön işleme gerektirmeyen anlamında yoktur.

**2.1.1. Kayıp Verileri Tamamlama**

Kayıp veri, veri madencilerinin kaçışı olmayan kabusu gibidir. Hemen hemen her veri setinde karşılarına çıkar. Çünkü veri tabiatı itibariyle kötüdür. Kayıp veriyle uğraşırken dikkatli olmak lazım, yanlış bir hareket felaketle sonuçlanabilir. Eksik verilerin bulunduğu satırları çıkarmak bir yöntem olmakla beraber bazı mahsurları vardır. Veriyi bozabilir, değerli verilerin kaybolmasına sebep olabilir, hele kayıp veriler özellikle bir örüntüye sahip ise ciddi sapmalar oluşturabilir. Kayıp verileri tamamlamak için istatistiksel yöntemleri veya makine öğrenmesi yöntemleri kullanılması çıkarmaktan daha sağlıklıdır.

**2.1.2. Gürültülü Veriyle Uğraşma**

Veri doğası icabı kötüdür dedik. Bazı veri madenciliği teknikleri verinin dağılımı konusunda varsayımları vardır. Örneğin regresyon normal dağılım ister. Aksi halde tip-1 hata olasılığı artar. Gürültülü veriyle uğraşma konusunda iki ana yaklaşım vardır. İlki bozuk veriyi düzeltme yöntemleri (data polishing methods). İkinci yaklaşım ise gürültülü veriyi filtrelemek ve eğitim verisi olarak kullanmamak.

****

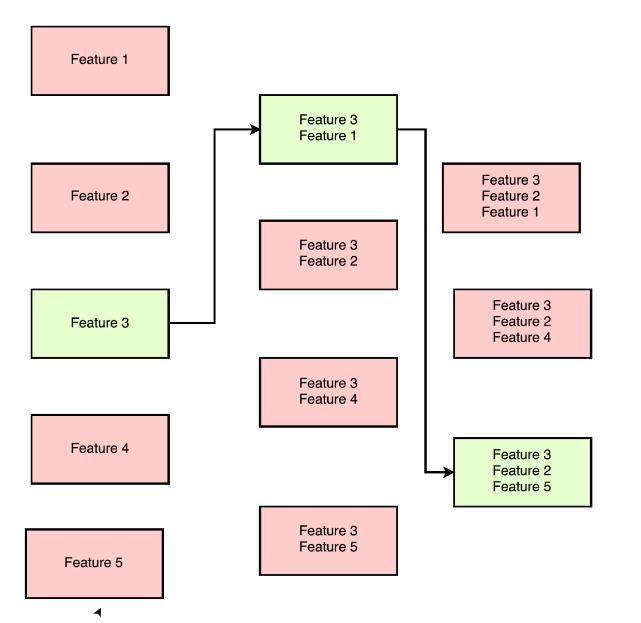
Şekil - Veri Ön İşleme

**2.2. Boyut İndirgeme**

Bağımsız değişken sayısı çok fazla olduğu durumlarda bağımlı değişkene olan etkiler çok zayıflar ve kurulan modellerin yorumlanabilirliği ve gerçek hayata uygulanabilirliği azalır. Bağımsız değişkenin çokluğuna genelde boyutluluğun laneti deniyor. Çok boyutluluk ayrıca hesaplama konusunda da ilave yük getiriyor.

**2.2.1. Özellik Seçimi**

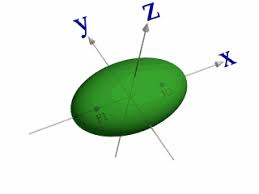
Özellik seçimi, problemi çözmek için gereksiz ve problemin çözümüne etkisi olmayan özellikleri tespit ederek bunları kullanmamaktır. Gereksiz özellikler gereksiz korelasyonlar oluşturur ve modelin genellenebilirliğini zayıflatır. Özellik seçimi ayrıca aşırı öğrenme olasılığını da azaltır, model eğitiminde gereksiz kaynak tüketiminin özellikle ana bellek, önüne geçer. Daha az özellik, daha anlaşılır ve yorumlanır modellerin oluşturulmasını sağlar.



Şekil - Özellik Seçimi

**2.2.2. Space Transformations**

Boyut indirgemenin özellik seçiminden başka yöntemleri de var elbette, örneğin faktör analizi ve ana bileşenler analizi (principal component analysis). Bu ikisi doğrusal yöntemlerdir. Bir de doğrusal olmayanlar var (LLE ve ISOMAP).



Şekil - Space Transformations

**2.2.3. Instance Reduction**

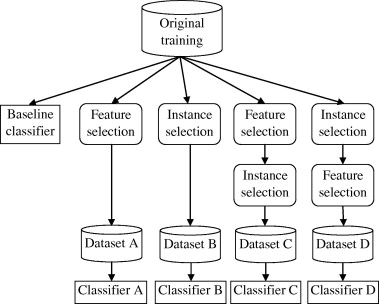
Büyük veri setlerinin veri madenciliği algoritmaları üzerindeki olumsuz etkisini azaltmanın popüler yöntemlerinden birisi IR. Veri boyutunu küçült ama ondan çıkarılacak bilgi kalitesini düşürme felsefesine dayanır.

**2.2.4. Instance Selection**

Örneklem seçmek gibi bir şey. Klasik yolda evrendeki nesnelerin hepsine ulaşıp veri toplayamadığımız için evreni temsil edebilecek bir örneklem seçiyorduk. Ancak burada evrendeki tüm nesnelere ait veri zaten elimizde. Fazla mal göz çıkardığından hepsini değil de hepsini temsil edecek bir örneklem seçiliyor. Yalnız buradaki fark olay tamamen tesadüfi gelişmiyor, temizleme işlemleri de yapılıyor ve algoritmanın verinin önemli kısımlarına odaklanması sağlanıyor.

**2.2.5. Instance Generation**

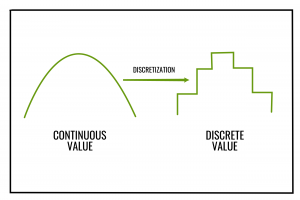
Instance generation bir bakıma instance selection tersi gibi. Burada da yapay bir veri üretimi var. Ama nerede ve niçin? Eksik yerler, bir alanda temsilcinin olmadığı veya yetersiz olduğu yerler. Yanlış etiketlenmiş verilerin düzeltilmesi bir örnek olarak verilebilir.



Şekil - Örnek Seçimi-Azaltması ve Oluşturması

**2.2.6. Discretization**

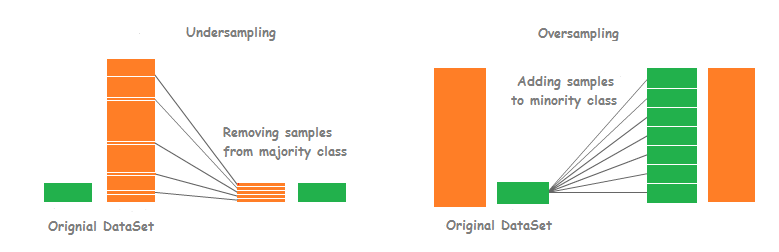
Veriyi kesikli hale getirme olayı. Örneğin karar ağaçları sürekli değişken kullanmaz. Bu sebeple sürekli değişkenler karar ağacı için kesikli hale getirilir. En çok kullanılan veri ön işleme tekniği. Örneğin yaş değişkeninin çocuk, ergen, genç, orta yaş, yaşlı yapılması. Karar ağaçları gibi birçok algoritma kesikli değişken istiyor mesela C4.5, Naive Bayes, Apriori. Kesikleştirme veriyi basitleştirme, daha anlaşılır kılma, hızlı ve yüksek doğrulukla öğrenmeyi gibi faydaları var üstelik verinin okunurluğunu artırıyor. Ancak bilgi kaybı gibi bazı maliyetler var.



Şekil - Discretization

**2.3. Dengesiz (Imbalanced) Öğrenme, Örneklem Azaltma (Undersampling) ve Örneklem Artırma (Oversampling) Yöntemleri**

Birçok denetimli öğrenme uygulamasında sınıflar arası öncül olasılıklar çok farklı ve dengesiz olabiliyor. Buradaki sıkıntı şu dengesiz sınıflar öğrenmede çoğunluğu oluşturan sınıf lehine bir sapma oluşturur. Bu sebeple seyrek sınıf için isabetsiz tahminler artar. Her ne kadar algoritmik olarak dengesiz sınıflara çare mümkünse de burada konumuz veri ön işleme olduğu için bu sorunu veri ön işleme teknikleriyle nasıl çözeriz onu tartışıyor olacağız. Veri ön işlemede dengesizliğin önüne geçmek için denge oluşana kadar yeniden örneklem alma yapılır. Bunun algoritmik düzenlemeye göre avantajı veri madenciliği algoritmasından bağımsız olmasıdır. Örneklem azaltmada baskın sınıftan kırpılarak örneklem alınır, örneklem artırma da zayıf sınıfa ilaveler yapılır. Bunlar gibi sezgisel olmayan teknikler veriden elde edilecek bilgiyi battal edebilir ya da aşırı öğrenme ihtimalini yükseltebilir. Sezgisel yaklaşımlardan Synthetic Minority Oversampling Technique (SMOTE) bu alanda en meşhurudur.



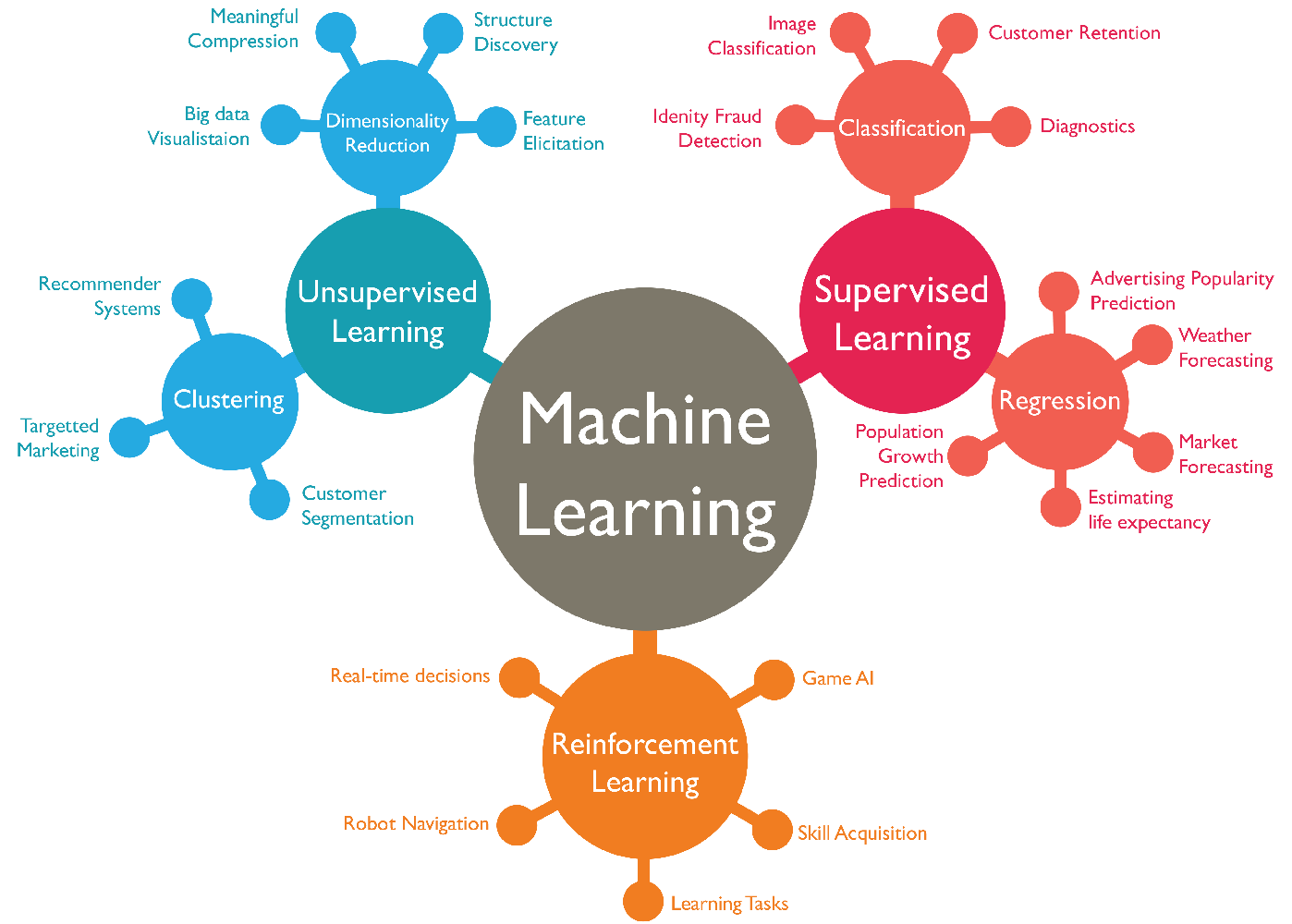
Şekil - Örneklem Azaltma veya Artırma

**2.4. Veri Madenciliğinin Yeni Alanlarında Veri Ön İşleme**

Veri ön işleme genelde denetimli öğrenme için yapılıyor ancak denetimsiz öğrenmenin boynu bükük değil. Onun için de bazı yaklaşımlar var. Feature Selection ve kayıp veri tamamlama bunlardan en popüleri. Hedef değişkenden fazla etikete sahip olan sınıflandırmalar sınıf dengesizliğine çok meyillidir. Bu sebeple tekrar örneklem tekniği önerilir. Akan veri de veri madenciliğinde zor konulardan biri, çünkü bilgi zaman içinde değişiveriyor. Akan veri için de kayıp veri tamamlama, özellik seçimi ve instance reduction öneriliyor.

**3.BÖLÜM: Sınıflandırma Algoritmaları**

Veri analitiği, günümüzde iş dünyasında ve akademik araştırmalarda önemli bir rol oynamaktadır. Büyük veri setleri üzerinde yapılan analizler, bilgiye dayalı kararlar almak ve geleceği tahmin etmek için değerli bir kaynak sunmaktadır. Bu nedenle, verileri etkili bir şekilde anlamak ve işlemek, başarılı sonuçlar elde etmek için hayati öneme sahiptir. Sınıflandırma algoritmaları, verileri belirli kategorilere veya sınıflara ayırmak için kullanılan güçlü araçlardan biridir.



Şekil - Makine Öğrenmesi Yöntemleri

**3.1. Sınıflandırma Algoritmalarının Temel İlkeleri**

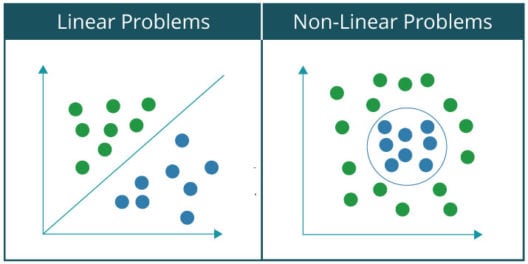
Sınıflandırma algoritmaları, veri setlerini belirli sınıflara ayırmak için kullanılan makine öğrenmesi yöntemleridir. Bu algoritmalar, eğitim verileri üzerinde model oluşturarak, daha sonra yeni örnekleri doğru şekilde sınıflandırmak için bu modele dayanır. Temelde, sınıflandırma algoritmaları, veri özelliklerini analiz eder ve ardından bu özelliklerin temelindeki kalıpları belirler.

**3.2. Doğrusal Sınıflandırma Algoritmaları**

Doğrusal sınıflandırma algoritmaları, girdi verilerini doğrusal bir hiper düzlemle sınıflandırır. En popüler doğrusal sınıflandırma algoritmalarından biri, Lojistik Regresyon'dur. Lojistik Regresyon, bağımlı değişkenin ikili (binary) veya çoklu (multiclass) olduğu durumlarda kullanılır. Diğer bir doğrusal sınıflandırma algoritması ise Lineer Diskriminant Analiz'dir (LDA). LDA, veri boyutunu azaltırken sınıflandırma yapabilen bir yöntemdir.

**3.3. Doğrusal Olmayan Sınıflandırma Algoritmaları**

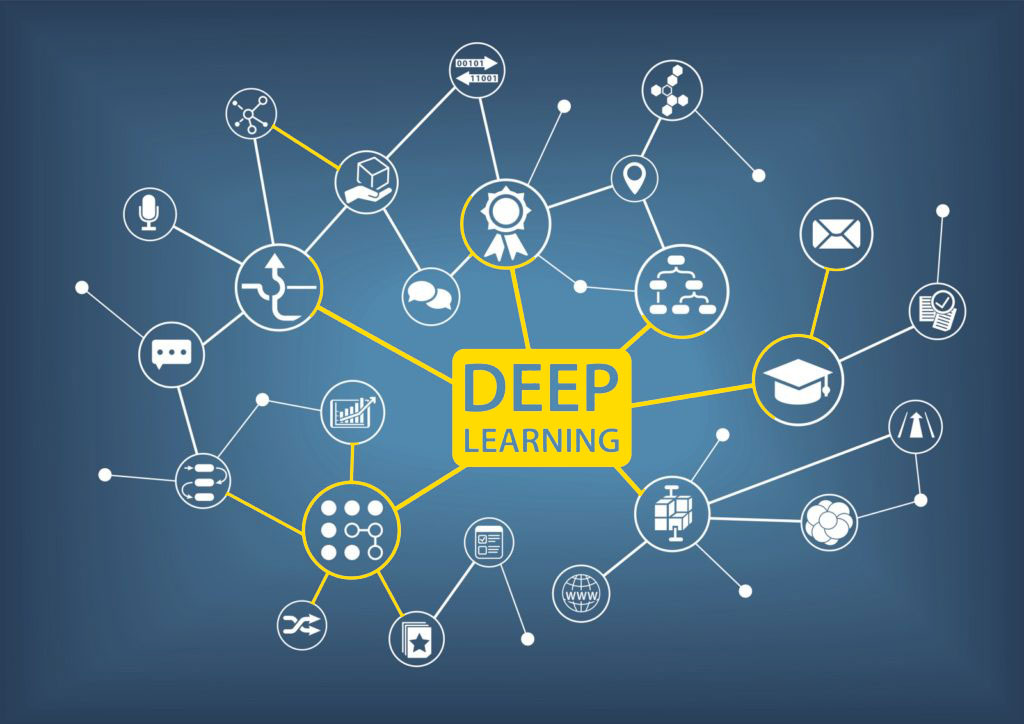
Doğrusal olmayan sınıflandırma algoritmaları, girdi verilerini doğrusal olmayan sınırlarla sınıflandırır. Bunların başlıca örnekleri arasında Destek Vektör Makineleri (SVM), Karar Ağaçları, Rastgele Ormanlar ve K-En Yakın Komşu (KNN) algoritması bulunur. SVM, sınıflandırma problemlerinde yüksek boyutlu veri kümelerini etkin bir şekilde işlemek için kullanılırken, Karar Ağaçları veriye karar ağaçları yapısı uygulayarak sınıflandırma yapar. Rastgele Ormanlar, birden çok karar ağacını birleştirerek daha güçlü bir sınıflandırma modeli oluştururken, K-En Yakın Komşu algoritması, yeni bir veriyi en yakın komşularına dayanarak sınıflandırır.



Şekil - Doğrusal ve Doğrusal Olmayan Sınıflandırma Algoritmaları

**3.4. Derin Öğrenme ve Sınıflandırma**

Derin öğrenme, yapay sinir ağları temelli sınıflandırma algoritmalarını içeren bir alan olarak önemli bir gelişme kaydetmiştir. Derin sinir ağları, çok katmanlı bir yapıya sahip olduğu için daha karmaşık örüntüleri öğrenebilir ve genellikle büyük veri kümeleriyle çalışırken yüksek performans gösterir. Özellikle Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN) görüntü sınıflandırmasında, Uzun-Kısa Dönemli Bellek Ağları (Long Short-Term Memory - LSTM) metin ve zaman serisi verilerinde ve Derin Bölgesel Özellikler (Deep Belief Networks - DBN) genel sınıflandırma problemlerinde yaygın olarak kullanılmaktadır.



Şekil - Derin Öğrenme

**3.5. Sınıflandırma Algoritmalarının Uygulama Alanları**

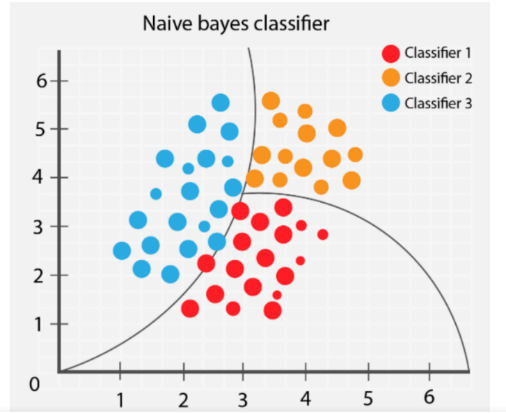
Sınıflandırma algoritmaları, birçok farklı uygulama alanında kullanılır. Örneğin, tıp alanında kanser teşhisi, ilaç keşfi ve hastalık tahmini gibi konularda sınıflandırma algoritmaları önemli bir rol oynar. Finansal analizde kredi riski değerlendirmesi, hisse senedi fiyat tahmini ve dolandırıcılık tespiti gibi görevlerde sınıflandırma algoritmalarının kullanımı yaygındır. Ayrıca, müşteri segmentasyonu, pazarlama hedefleme ve kullanıcı davranışı analizi gibi işletme uygulamalarında da sınıflandırma algoritmaları etkin bir şekilde kullanılır.

**3.6. Sık Kullanılan Sınıflandırma Algoritmaları**

Makine öğrenimi ve veri analitiği alanında daha birçok algoritma bulunmaktadır. Hangi algoritmanın kullanılacağı, veri setinin özelliklerine, boyutuna ve problemin doğasına bağlıdır. Her algoritmanın avantajları, dezavantajları ve özel durumlarda performans farklılıkları olabilir.

**3.6.1. Naive Bayes**

Naive Bayes sınıflandırma algoritması, Bayes teoremini kullanır ve "naive" (basit) varsayımlarda bulunur. Her bir özellik (feature) için sınıf etiketini tahmin etmek için özellikler arasındaki bağımsızlığı varsayar. Sınıflandırma için olasılık temelli bir yaklaşım kullanır.



Şekil - Naive Bayes

**3.6.2. Karar Ağaçları**

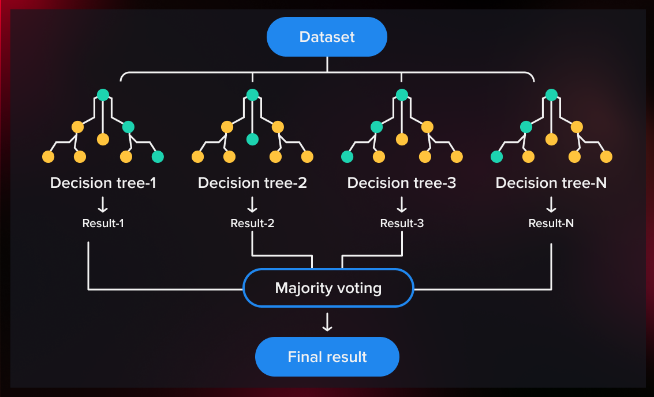
Karar ağaçları, bir veri kümesini sınıflandırmak veya regresyon yapmak için kullanılan popüler bir algoritmadır. Veri kümesindeki özellikleri ve hedef değişkeni kullanarak bir ağaç yapısı oluşturur. Her iç düğüm, bir özellikle ilişkilendirilir ve her yaprak düğümü bir sınıf etiketi veya regresyon değeri temsil eder. Karar ağaçları, veri setindeki özelliklerin önemli özelliklerini belirlemek ve karmaşık kararların basit ve yorumlanabilir bir modelle ifade edilmesini sağlamak için kullanılır.



Şekil - Karar Ağaçları

**3.6.3. Rastgele Ormanlar**

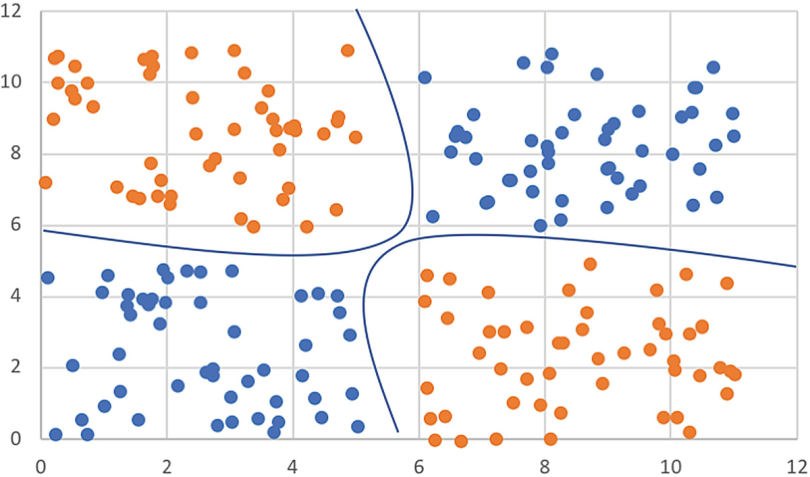
Random Forests (Rastgele Ormanlar), birçok karar ağacının bir araya getirildiği bir ansamble yöntemidir. Rastgele Ormanlar, her bir ağacın bağımsız olarak eğitildiği ve ardından sınıflandırma veya regresyon sonuçlarını birleştiren bir ensemble algoritmasıdır. Bu, overfitting'i azaltabilir ve daha genelleyici bir model sağlayabilir.



Şekil - Rastgele Ormanlar

**3.6.4. Lojistik Regresyon**

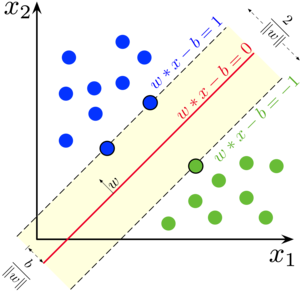
Lojistik regresyon, bir veri kümesindeki değişkenleri kullanarak bir olayın olasılığını tahmin etmek için kullanılan bir istatistiksel modeldir. Sınıflandırma problemlerinde kullanıldığında, sonuç ikili (binary) veya çok sınıflı olabilir. Lojistik regresyon, giriş değişkenleri ile doğrusal bir fonksiyon arasındaki ilişkiyi tahmin etmek için bir logit dönüşümü kullanır.



Şekil - Lojistik Regresyon

**3.6.5. Destek Vektör Makineleri**

Destek Vektör Makineleri, iki sınıf arasında bir karar sınırı belirlemek için kullanılan bir algoritmadır. İki sınıf arasındaki en geniş marjinal sınırlamayı bulmak için bir optimizasyon problemi olarak formüle edilir. Destek vektör makinesi, çok boyutlu verilerde de iyi performans gösterebilir ve lineer olmayan karar sınırlarını keşfetmek için çekirdek (kernel) fonksiyonlarını kullanabilir.



Şekil - Destek Vektör Makineleri

**3.6.6. K-En Yakın Komşu**

K-En Yakın Komşu algoritması, yeni bir örneği sınıflandırmak için en yakın komşularının etiketlerini kullanır. Veri noktaları arasındaki mesafeyi hesaplar ve k-en yakın komşuyu seçer. K-en yakın komşu algoritması, basit bir sınıflandırma algoritması olmasına rağmen etkilidir.



Şekil - K-En Yakın Komşu

**3.7. Sınıflandırma Algoritmalarının Değerlendirilmesi**

Makine öğrenimi ve veri madenciliği alanında, çeşitli sınıflandırma algoritmaları bulunmaktadır. Ancak, her algoritmanın performansı, veri setine ve probleme bağlı olarak değişebilir. Bu nedenle, sınıflandırma algoritmalarının değerlendirilmesi önemli bir adımdır.

**3.7.1. Değerlendirme Metrikleri**

Sınıflandırma algoritmalarının değerlendirilmesi için kullanılan temel metrikler vardır. Bu metrikler, algoritmaların performansını objektif bir şekilde karşılaştırmak için kullanılır.

**3.7.1.1. Doğruluk (Accuracy)**

Doğru sınıflandırılan örneklerin oranını ifade eder. Genellikle çok sınıflı problemler için kullanılır. Ancak, dengesiz veri kümelerinde doğruluk, yanıltıcı olabilir.

**3.7.1.2. Hassasiyet (Precision)**

Pozitif olarak tahmin edilen örneklerin gerçek pozitiflere oranını ifade eder. Yanlış pozitiflerin (yanlış sınıflandırılan negatiflerin) önemli olduğu durumlarda kullanılır.

**3.7.1.3. Geri Çağırma (Recall)**

Gerçek pozitiflerin tespit edilme oranını ifade eder. Yanlış negatiflerin (yanlış sınıflandırılan pozitiflerin) önemli olduğu durumlarda kullanılır.

**3.7.1.4. F1 Skoru**

Hassasiyet ve geri çağırma metriklerini birleştirir ve dengeleyici bir metrik sağlar. F1 skoru, bu metriklerin harmonik ortalamasını ifade eder.

**3.7.1.5. ROC Eğrisi**

Algoritmanın hassasiyetini ve geri çağırmasını farklı eşik değerlerine göre görselleştiren bir eğridir. ROC eğrisi, algoritmanın performansını değerlendirmek için kullanılır ve eğri altında kalan alan (AUC) metriğiyle ölçülür.

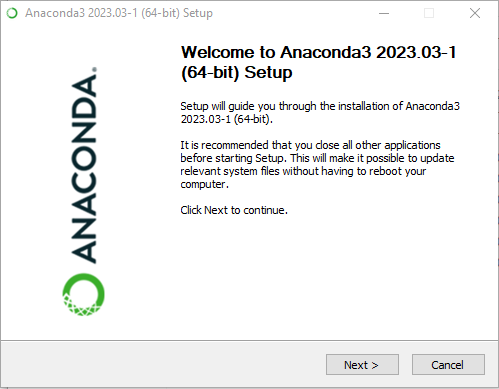
**4.BÖLÜM: Python ile Uygulama**

Bu bölümde Spyder geliştirme ortamında Python programlama dilinde önceki bölümlerde anlattığım konuları uygulayacağız.

**4.1. Anaconda ve Spyder Kurulumu**

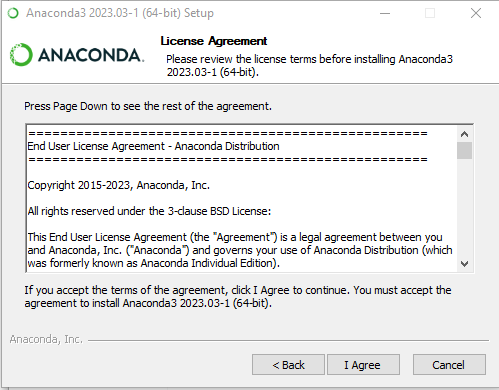
Anaconda, Spyder gibi birbirinden farklı programları barındıran dağıtıcıdır. Spyder, Python tarafından yazılan ve Python’ı geliştirmek için kullanılan açık kaynaklı IDE’dir. Spyder’ın amacı Python kullanırken yaptığımız işleri kolaylaştırmaktır. Spyder, hata ayıklama işlemi, gelişmiş düzenleme yapabilme, etkileşimli test edebilme özelliklerinden dolayı Python için oldukça güçlü bir geliştirme ortamıdır.

Anaconda Kurulumu için ilk adımımızı atmak ve “Anaconda Dağıtım Paketi” yapısını bilgisayarımıza ücretsiz bir şekilde “Individual edition” yapısını indirmek için ilgili indirme sayfasına giriş yapalım. Bahsetmiş olduğum indirme sayfası https://www.anaconda.com/ adresine girerek hızlı bir şekilde ulaşabilirsiniz. Bağlantı yoluna girdikten sonra karşımıza çıkan “Download” yazılı butona tıkladığımızda işletim sistemimize en uygun exe uzantılı kurulum dosyasını otomatik olarak indiriyor. Daha sonra bilgisayarımıza kurulumu gerçekleştirmek için indirmiş olduğumuz exe uzantılı “Setup” dosyasına çift tıklayarak kurulum penceresini açalım.



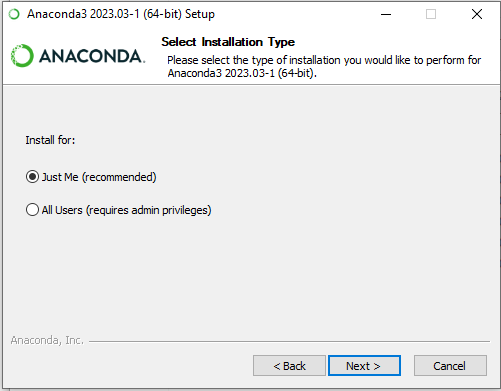
Şekil - Anaconda Hoşgeldin

Önümüze çıkan hoşgeldin ekranını “Next” butonuna basarak geçiyoruz.



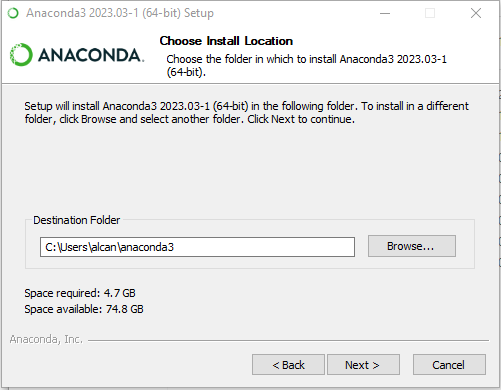
Şekil - Anaconda Sözleşme

Daha sonra karşımıza gelen sözleşmeyi okuyup “I Agree” butonuna basarak kabul ediyoruz.



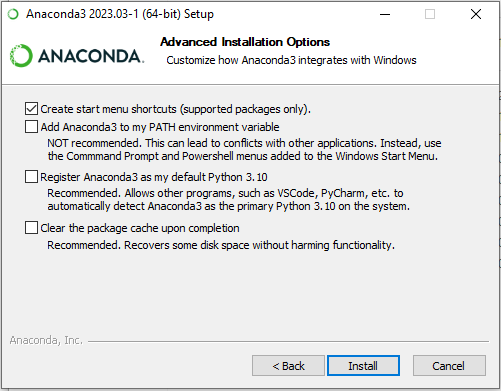
Şekil - Anaconda Profil

Bu aşamada uygulamayı kullanacak profilin seçimini yapıyoruz ve “Next” butonuna basıyoruz.



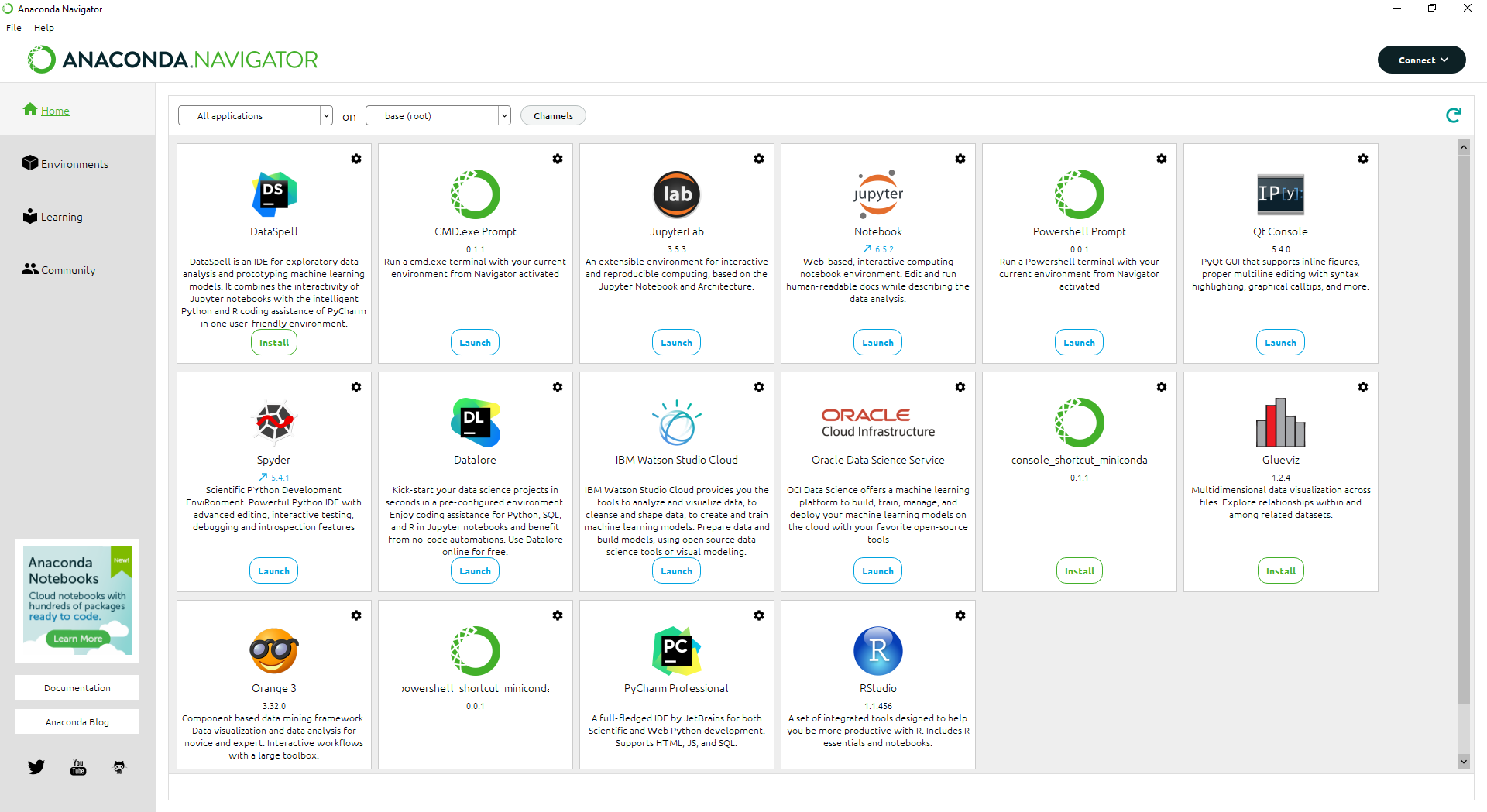
Şekil - Anaconda Yükleme

Kurulumun nereye yapılacağını seçiyoruz ve “Next” butonu ile son aşamaya geçiyoruz.



Şekil - Anaconda Gelişmiş Kurulum Seçenekleri

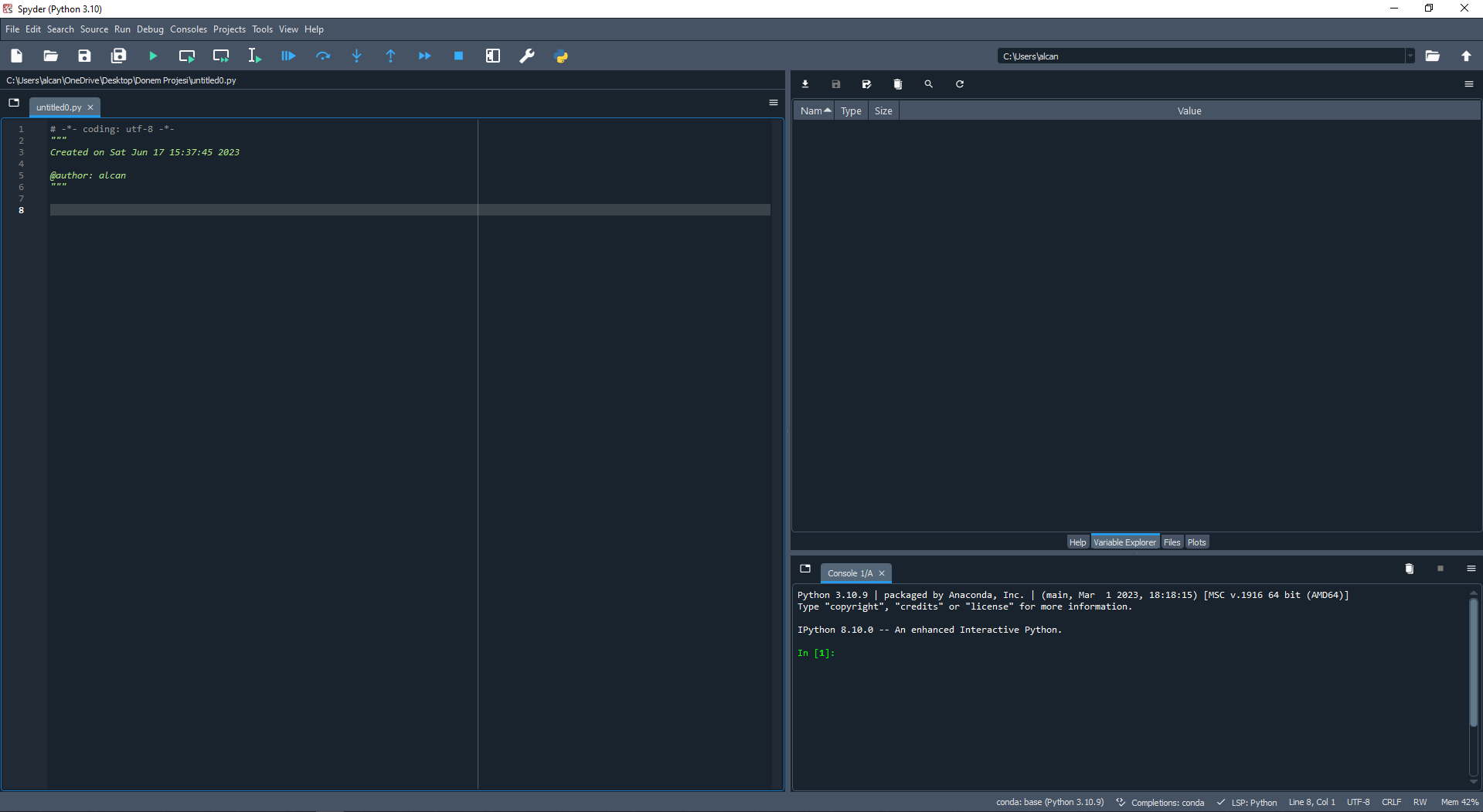
Son aşamada ise gelişmiş kurulum seçeneklerinden dilediğimizi seçip “Install” butonu ile kurulu başlatıyoruz.Kurulum bittikten sonra bilgisayarımıza Anaconda Navigator yüklenmiş oluyor. Programa çift tıklayarak açıyoruz.



Şekil - Anaconda Navigator İlk Ekran

Açılan uygulamada çeşitli geliştirici arayüzlerini görüyoruz. Bu çalışmada kullanılacak geliştirici olan Spyder geliştirici arayüzü kurulu değil ise “Install” butonuna basarak indirebiliriz. Eğer kurulu ise “Launch” butonuna basarak açabiliriz.

**4.2. Spyder’da Gereken Kütüphaneleri Eklemek**



Şekil - Spyder İlk Ekran

Yukardaki görselde Spyder uygulamasının ilk açılıştaki görüntüsünü görüyoruz. Tabi ki indirdiğimiz geliştirici arayüzünde her kütüphane yüklü gelmiyor. Bizim kullanacağımız kütüphaneleri indirip kısaca bahsedeceğim.

**4.2.1. Pandas**

Veriyi topladıktan sonra veriyi analize uygun hale getirmek, veri analizi için oldukça önemlidir. Veri bilimcilerinin en fazla zamanlarını alan aşama veri ön işleme ve veri temizlemedir. Pandas kütüphanesi ile veri organize edilerek analizler daha kolay ve hızlı yapılır. Pandas Python’ın en önemli kütüphanelerinden biridir. Bu kütüphaneyi “pip install pandas” yazarak indirebiliriz.



Şekil - Pandas

**4.2.2. Selenium**

Selenium, test otomasyon dünyasının önemli ürünlerinden biri haline gelen, açık kaynaklı, yazılım test otomasyon kütüphanesidir. Bilinenin aksine tek bir araç değil, paketlerin oluşturduğu bir kütüphanedir. Bu sebeple Selenium Suite olarak adlandırılır. Bu kütüphaneyi “pip install selenium” yazarak indirebiliriz.



Şekil - Selenium

**4.2.2.1. WebDriver**

Selenium WebDriver, test case oluşturmak ve yürütmek için tasarlanmış bir framework’tür. Case’ler WebDriver metotlarındaki element locator’lar aracılığı ile oluşturulur ve yürütülür. Selenium WebDriver bir IDE değil, programlama kütüphanesidir. Bu kütüphaneden “support.iu” içinden “WebDriverWait” , “common.by” içinden “By” ve “support” içinden “expected\_conditions” ekliyoruz.

**4.2.3. xlsxwriter**

Python’ dan Excel’e veri yazdırmak için kullanacağımız modül xlsxwriter ‘dır. Bu modülü “pip install XlsxWriter” şeklinde indirebiliriz.

**4.2.4. scikit-learn**

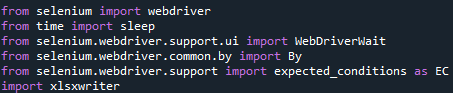
Scikit-learn, veri bilimi ve machine learning için en yaygın kullanılan Python paketlerinden biridir. Birçok işlemi gerçekleştirmenizi sağlar ve çeşitli algoritmalar sağlar. Scikit-learn ayrıca sınıfları, yöntemleri ve işlevleri ile kullanılan algoritmaların arka planıyla ilgili belgeler sunar. Bu kütüphaneyi “pip install scikit-learn” yazarak indirebiliriz.



Şekil - scikit-learn

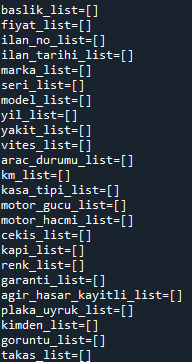
**4.3. sahibinden.com Sitesinden Veri Seti Hazırlama**

İlk olarak gerekli olan kütüphaneleri çalışmamıza ekleyelim.



Şekil - Kod Görseli 1

İlan bilgilerini çektiğimizde ekleyebilmek için tablolarımızı hazırlayalım.



Şekil - Kod Görseli 2

Bu tabloları bir excel’e kaydetmek için boş bir excel ve worksheet oluşturalım.



Şekil - Kod Görseli 3

Webdriver yardımı ile Google Chrome da bulunan otomatik test yazılımını açalım. Sahibinden sitesinde çıkan reklamları kaldırmak için “uBlock Origin” eklentisini “extension\_1\_48\_4\_0.crx” Google Chrome uzantısını ekleyelim.



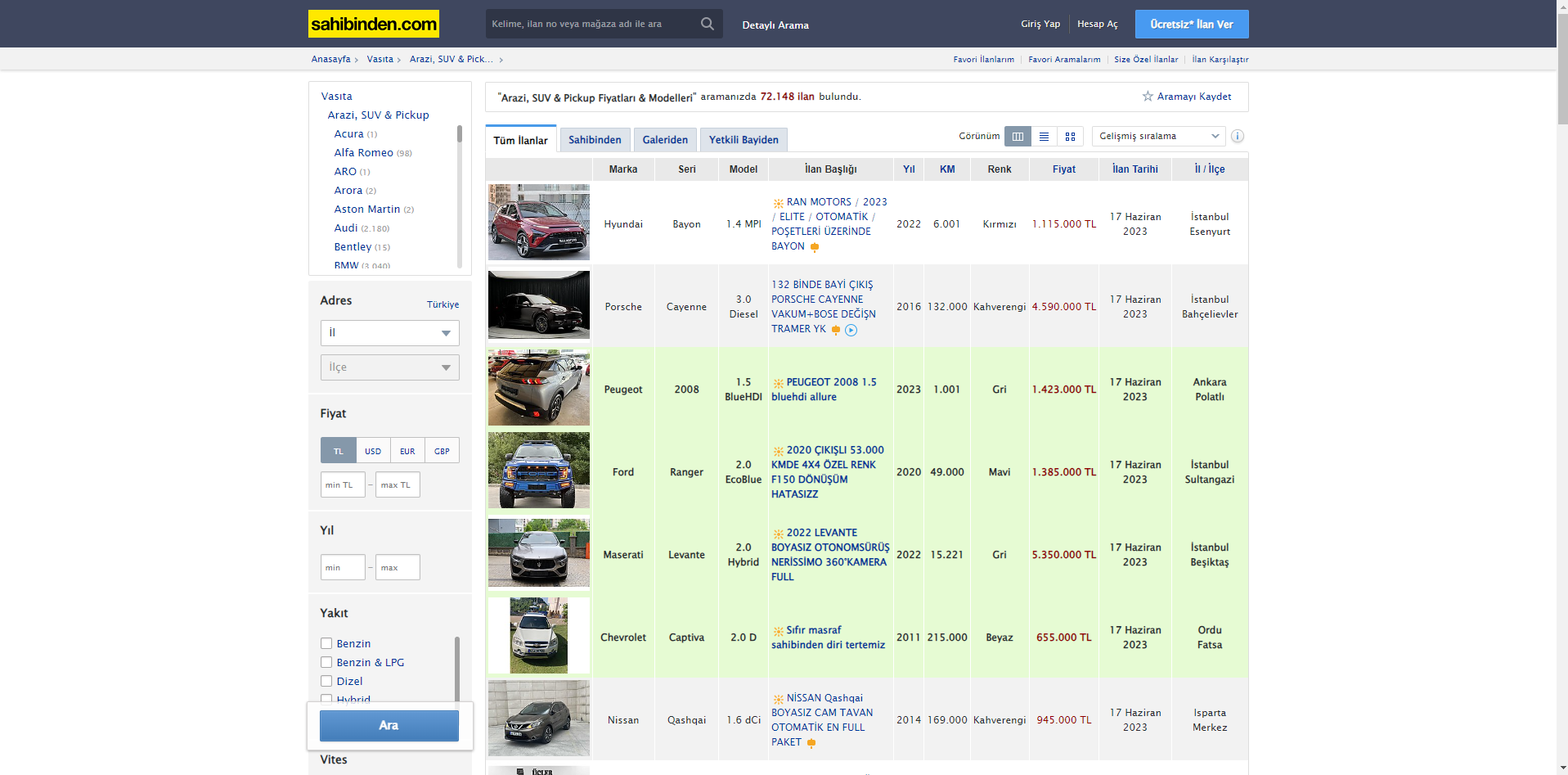
Şekil - Kod Görseli 4

“sayfa” adında bir değişken oluşturuyoruz ve 0 değerini atıyoruz. Açtığımız Google Chrome sayfasında 50 adet SUV ilanlarının bulunduğu linkin içinde bulunan “pagingOffset=” bulunan yere “{}” yazıyoruz. Çünkü bu değerin sayfa değişkeni değiştiğinde değişmesini istiyorum. Son olarak da sahibinden.com da çıkar çerezleri kabul etmesi için butonun XPATH’ini kullanarak tıklatıyoruz.



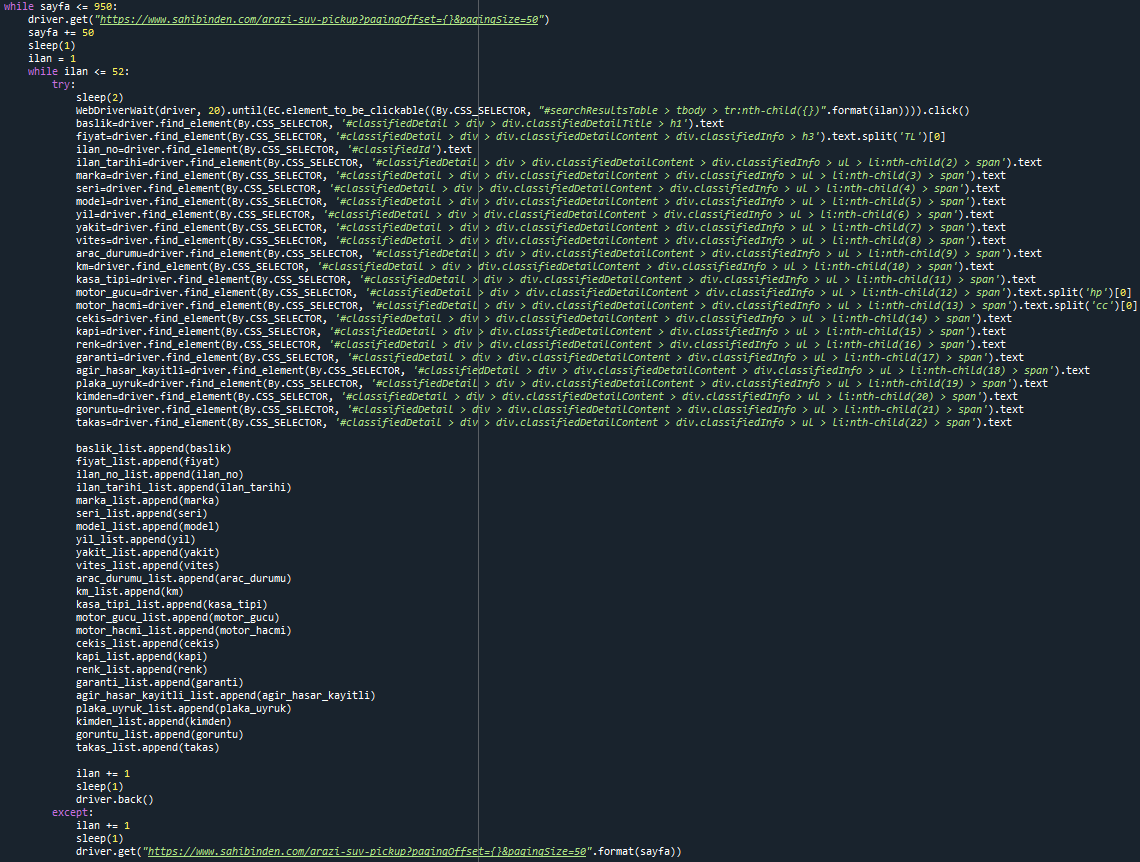
Şekil - Kod Görseli 5

Bu aşamalardan sonra artık ilanların bulunduğu sayfayı görmüş oluyoruz.



Şekil - sahibinden.com SUV İlanları

Sayfa değişkenimizi önceki adımlarda 0 olarak atamıştık. Sayfa değişkenimizin en fazla 950 ye eşit olmasını istiyoruz çünkü bize 950 adet ilan bilgisi yeterli olacaktır. Döngü başa döndüğünde yeni sayfaya geçebilmek için tekrar linkimizi ekledik ve sayı değişkenimizi 50 arttırdık. Aralara 1 saniyelik “sleep” ekliyoruz. Bunun nedeni hem site bot olduğunu algılamasın hem de internette olacak dalgalanmalar ve gecikmelerden etkilenmemesi için. Sonrasında ilan adında yeni bir değişken oluşturuyoruz ve buna da 1 değerini atıyoruz. Bu değişkende bize ilanlar arasında gezmemize yardımcı olacak. Yeni bir while döngüsü açıp ilanların “CSS SELECTOR” adreslerini arttırarak her ilana tek tek girip ilan bilgilerini alacağız.



Şekil - Kod Görseli 6

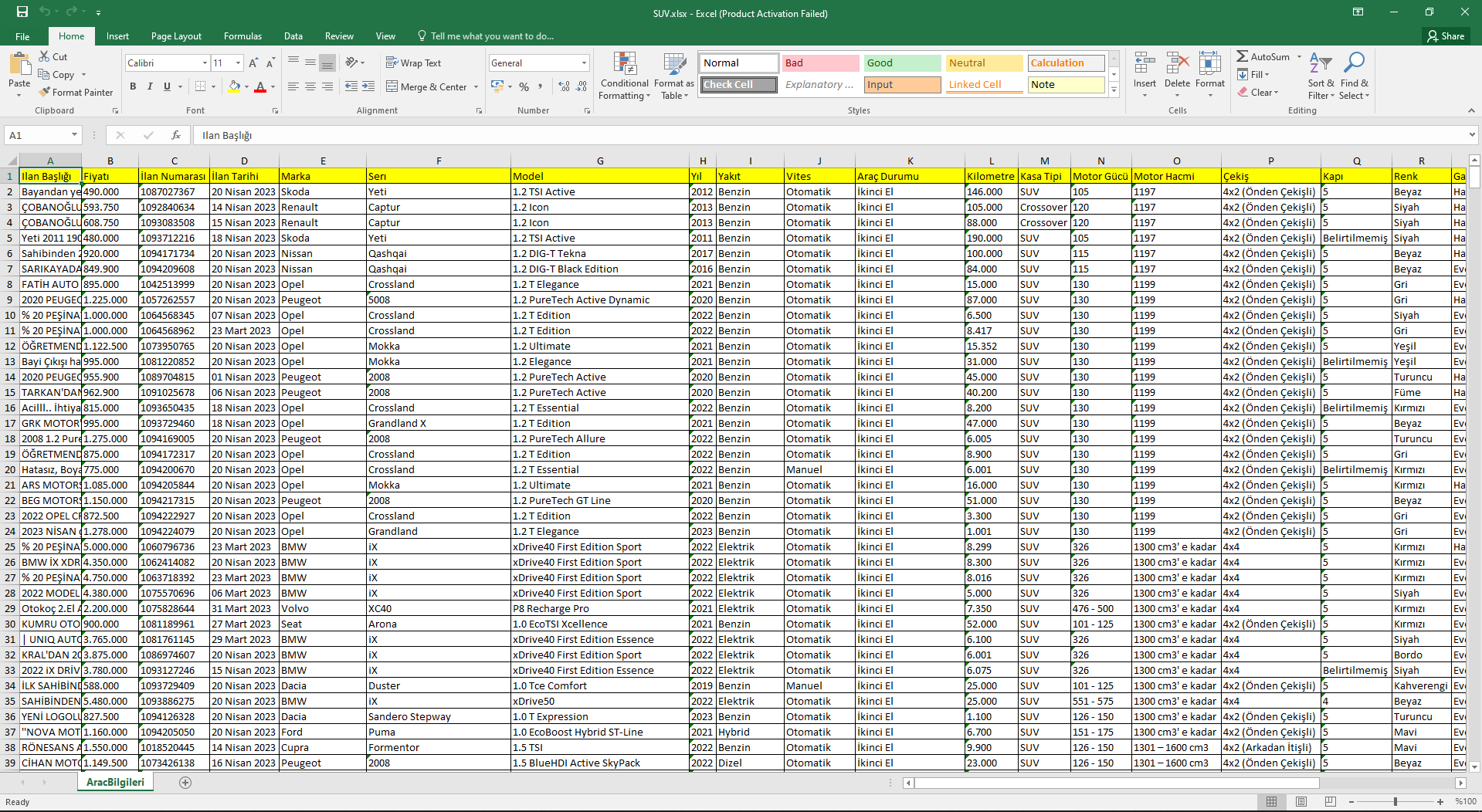
Alacağımız bilgiler başlığı, fiyatı, ilan numarası, ilan tarihi, markası, seri numarası, modeli, yılı, yakıt tipi, vitesi, araç durumu, kilometresi, kasa tipi, motor gücü, motor hacmi, çekişi, kapı sayısı, rengi, garanti, ağır hasar kaydı, plaka/uyruk, kimden, görüntü ve takas verilerini “CSS\_SELECTOR” adresleri yardımı ile değişkenlerine atıyoruz. Bu değişkenleri listelerine atıyoruz.

Bu listeleri for döngüsü ile Excel worksheetimize ekliyoruz. Son olarak excel’imizi ve Google Chrome sayfamızı kapatıyoruz.



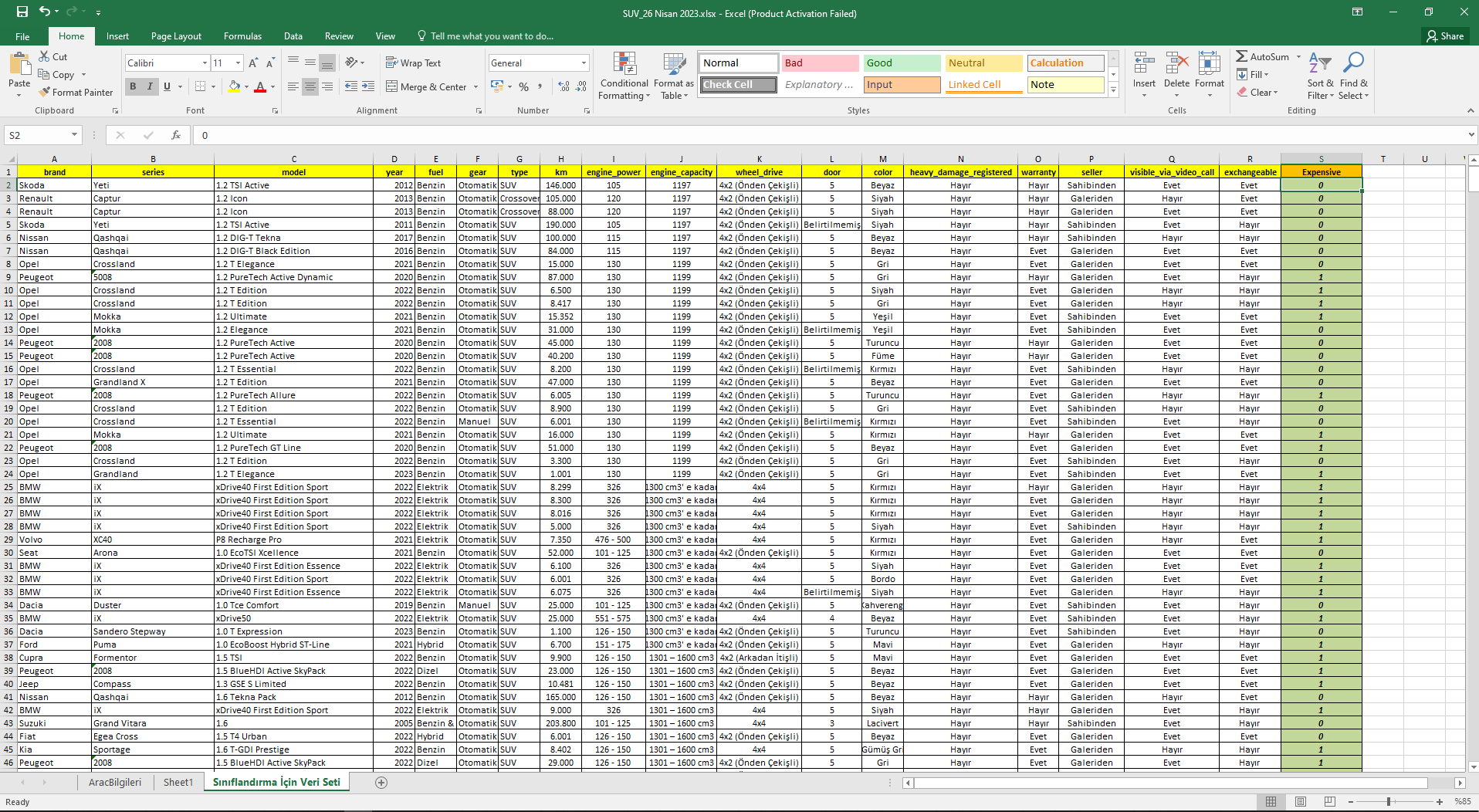
Şekil - Kod Görseli 7

Excel üzerinden tekrar eden ve eksik verileri düzenledikten sonra hazırladığımız veri setinin son görünüşü bu şekildedir.



Şekil - Veri Setinin Excel’de Görünümü 1

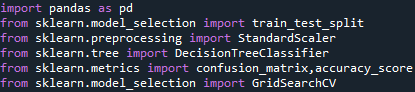
Topladığımız bu verileri sınıflandırma algoritmalarını uygulayabilmek için fiyat verilerinin ortalamasını alıp her bir fiyat verisini bu ortalama değer ile karşılaştırdık. Fiyat değişkeni bu ortalama değerinin üzerinde ise pahalı yani 1 değerini, altında ise ucuz yani 1 değerini aldığı yeni bir sütun açtık. Sınıflandırma algoritmalarını uygularken bu sütunu tahmin yapmak için kullanacağız.



Şekil - Veri Setinin Excel’de Görünümü 2

**4.4. Karar Ağacı Algoritması**

İlk olarak gerekli olan kütüphaneleri çalışmamıza ekleyelim.

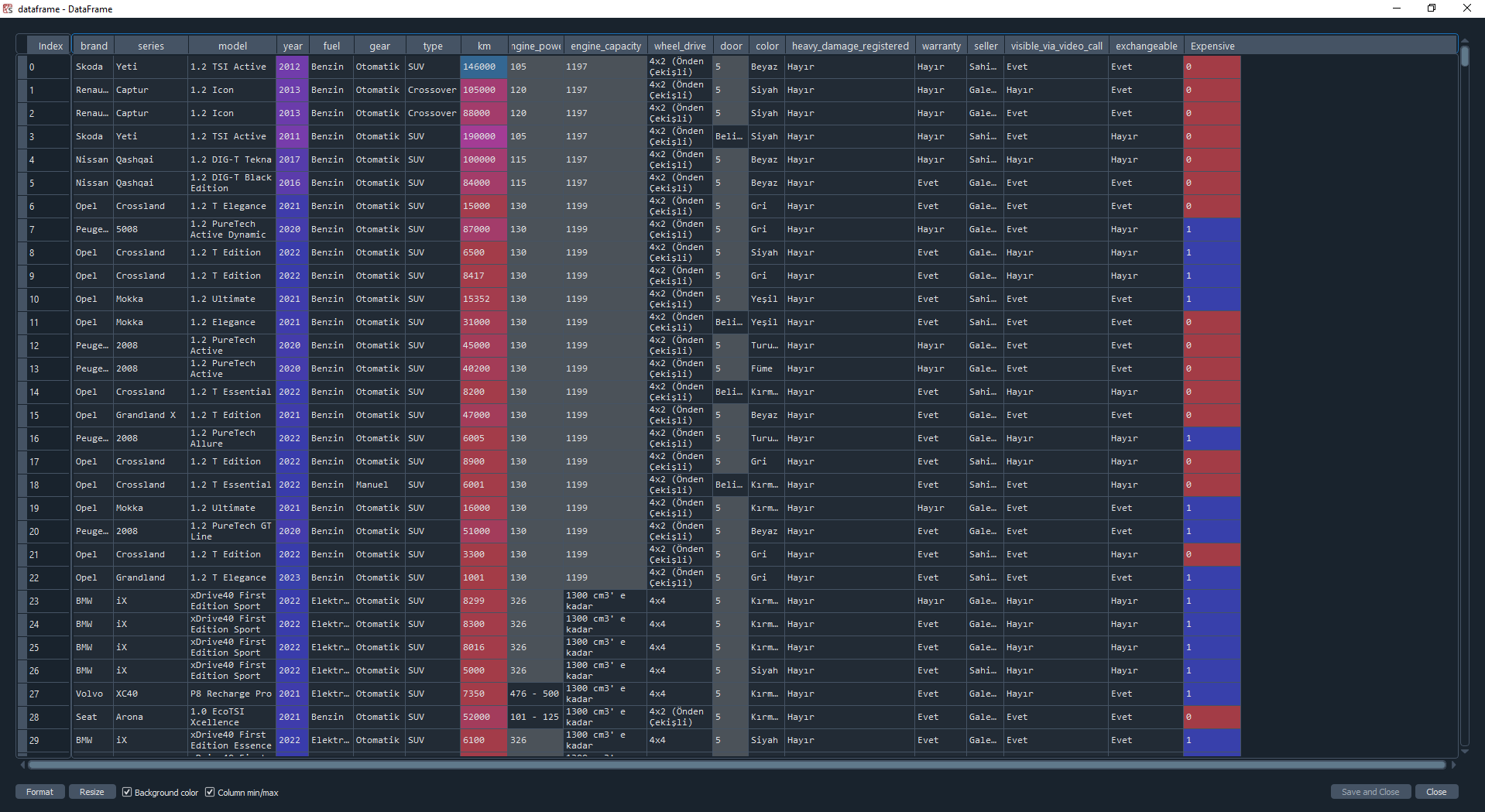


Şekil - Kod Görseli 8

Sınıflandırma algoritmalarını uygulayabilmek için hazırladığım excel sayfamı pandas yardımı ile çalışmamızda bir dataframe içerisine alalım.

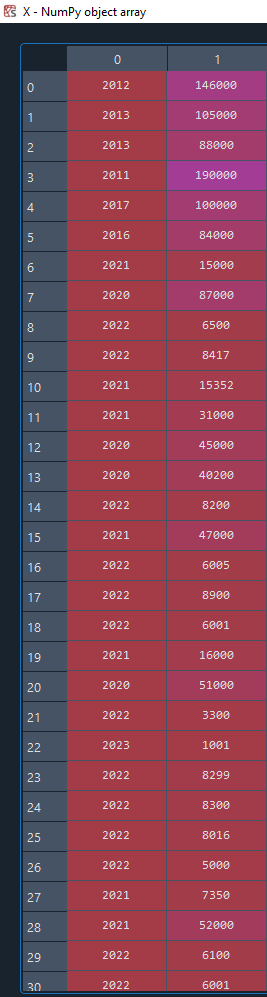
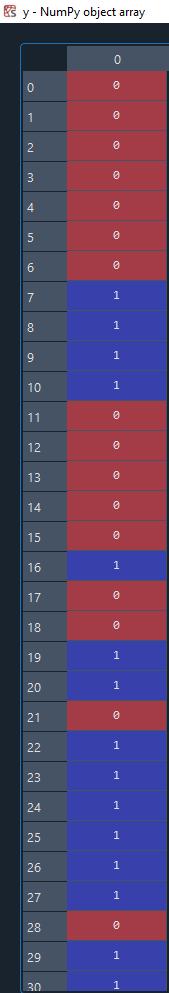


Şekil - Kod Görseli 9



Şekil - Veri Setinin Spyder’da Görünümü

Dataframe içerisine aldığımız yıl(year) ve kilometre(km) bağımsız değişkenlerimizi X adında yeni bir değişkene atıyoruz. 1 ve 0’lardan oluşan fiyat ortalamasına göre pahalı veya ucuz olan (Expensive) bağımlı değişkenini y adında yeni bir değişkene atıyoruz.

Şekil - Kod Görseli 10 Şekil - Spyder’da Bağımsız ve Bağımlı Değişkenler

Ayırdığımız X ve y değişkenlerimizin %24’ünü daha sonra modelimizi test etmek için X\_test ve y\_test adında iki farklı değişkene ayırıyoruz. Geri kalan değişkenleride X\_train ve y\_train adında iki farklı değişkene atıyoruz. Modelimizi bu iki değişken ile eğiteceğiz.



Şekil - Kod Görseli 11

X\_train ve X\_test değişkenlerimize scikit learn kütüphanesinde bulunan StandardScaler ile ortalamaya dayalı bir ölçekleme yaptık.



Şekil - Kod Satırı 12

Artık sınıflandırma modelimizi oluşturmanın zamanı geldi. Kriterimi gini olarak belirttim. X\_train ve y\_train değişkenlerimizi modelimizi eğitmek için kullandık.



Şekil - Kod Satırı 13

Modelimize X\_test değişkenini verdik ve bize bir tahminde bulunmasını istedik. Bu tahminleri y\_pred değişkenimize atadık. Bu tahminler ile y\_test de tuttuğumuz doğru sonuçları karşılaştırdık.



Şekil - Kod Satırı 14

Modelimizin yaptığı tahminleri scikit learn kütüphanesinde bulunan accuracy\_score ile doğruluk oranını 0.63 olarak bulduk ve hata matrisini çıkardığımızda doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 73 fiyat ve ucuz olup ucuz tahmini yaptığı 82 fiyat ile toplamda 155 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 48 fiyat ve ucuz olup pahalı tahmini yaptığı 43 fiyat ile toplamda 91 yanlış tahmin yapmıştır.

Şekil - Karar Ağacı Doğruluk Oranı ve Hata Matrisi

**4.5.1. Karar Ağacı Algoritması Modelinin İyileştirmesi**

Bulduğumuz doğruluk oranı ve hata matrini iyileştirmek için model iyileştirmesi(model tuning) yapalım. İlk olarak rf\_params adlı değişkenimize max\_depth, max\_features ve min\_samples\_split değerlerimizin olabileceği değerleri atadık. Sonra scikit learn kütüphanesinde bulunan GridSearchCV ile tek tek bu değerleri modelimiz ile denemeler yapması için X\_train ve y\_train değişkenlerimizi verdik.



Şekil - Kod Satırı 15

En iyi değerleri görebilmek için yazdırdık.Buna göre en iyi parametreler, max\_depth için 2, max\_features için 2 ve min\_samples\_split için 2’dir.



Şekil - Kod Satırı 16

Bu değerleri modelimize yazıp tekrar eğitelim.



Şekil - Kod Satırı 17

İyileştirilmiş modelimize X\_test değişkenini verdik ve bize bir tahminde bulunmasını istedik. Bu tahminleri y\_pred değişkenimize atadık. Bu tahminler ile y\_test de tuttuğumuz doğru sonuçları karşılaştırdık.



Şekil - Kod Satırı 18

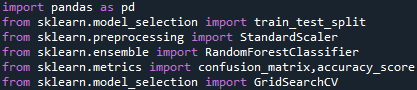
İyileştirilmiş modelimizin yaptığı tahminleri scikit learn kütüphanesinde bulunan accuracy\_score ile doğruluk oranını 0.71 olarak bulduk ve hata matrisini çıkardığımızda doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 95 fiyat ve ucuz olup ucuz tahmini yaptığı 82 fiyat ile toplamda 177 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 26 fiyat ve ucuz olup pahalı tahmini yaptığı 43 fiyat ile toplamda 69 yanlış tahmin yapmıştır.

Şekil - Karar Ağacı İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

**4.5. Rastgele Orman Algoritması**

İlk olarak gerekli olan kütüphaneleri çalışmamıza ekleyelim.

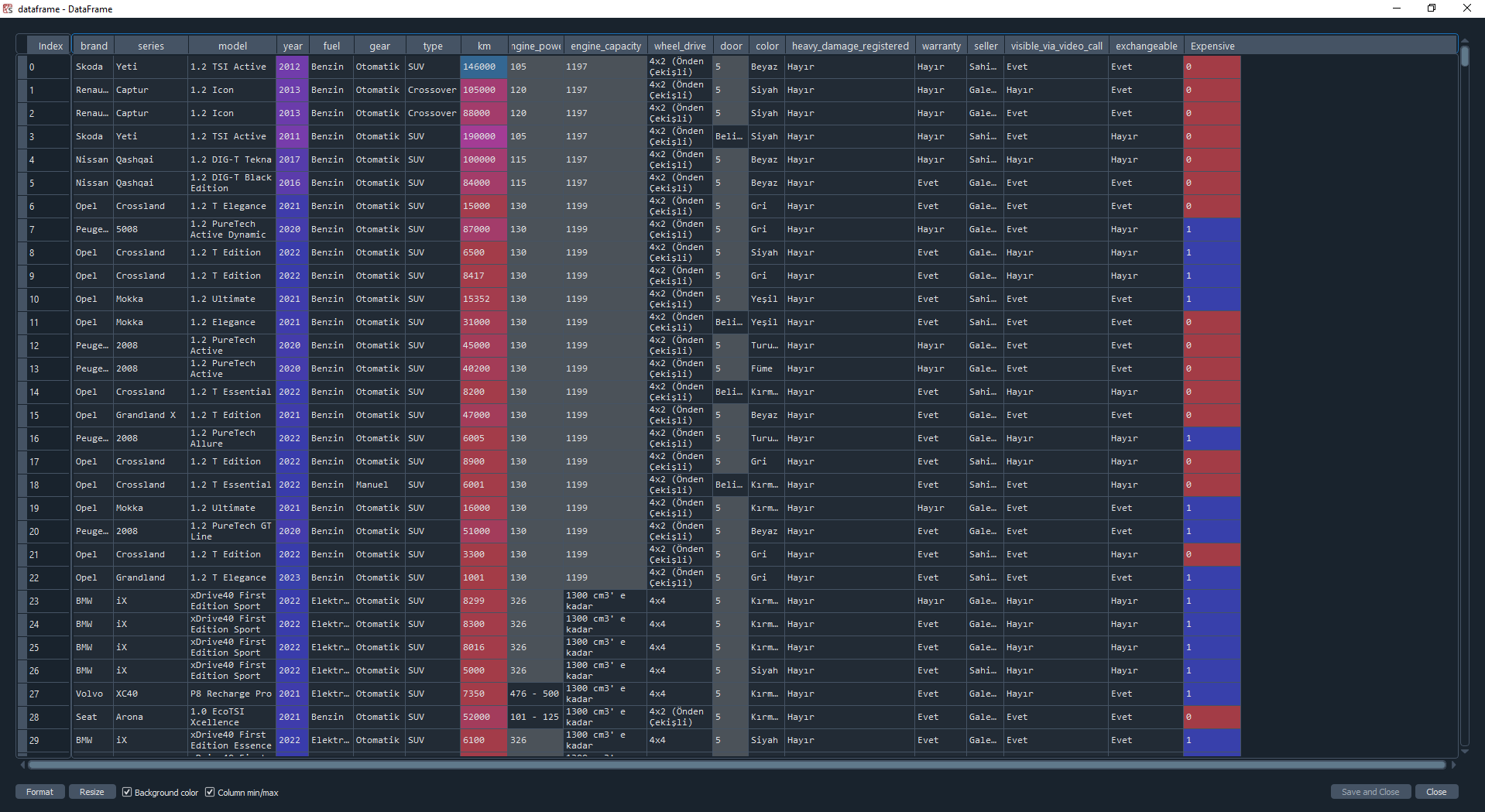


Şekil - Kod Satırı 19

Sınıflandırma algoritmalarını uygulayabilmek için hazırladığım excel sayfamı pandas yardımı ile çalışmamızda bir dataframe içerisine alalım.

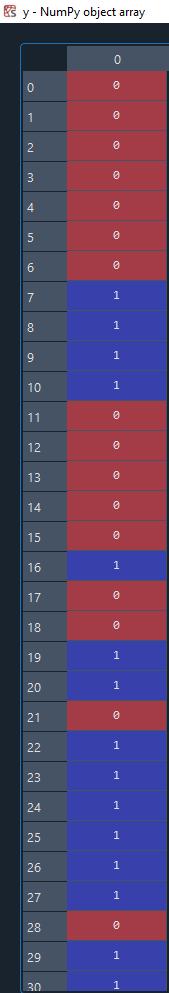
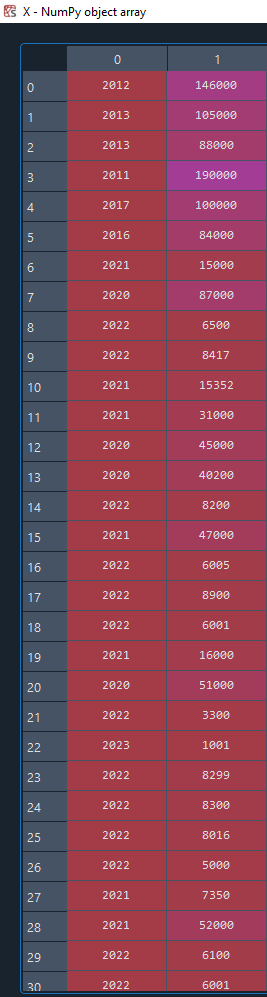


Şekil - Kod Satırı 20



Şekil - Veri Setinin Spyder’da Görünümü

Dataframe içerisine aldığımız yıl(year) ve kilometre(km) bağımsız değişkenlerimizi X adında yeni bir değişkene atıyoruz. 1 ve 0’lardan oluşan fiyat ortalamasına göre pahalı veya ucuz olan (Expensive) bağımlı değişkenini y adında yeni bir değişkene atıyoruz.

Şekil - Kod Görseli 21 Şekil - Spyder’da Bağımsız ve Bağımlı Değişkenler

Ayırdığımız X ve y değişkenlerimizin %24’ünü daha sonra modelimizi test etmek için X\_test ve y\_test adında iki farklı değişkene ayırıyoruz. Geri kalan değişkenleride X\_train ve y\_train adında iki farklı değişkene atıyoruz. Modelimizi bu iki değişken ile eğiteceğiz.



Şekil - Kod Satırı 22

X\_train ve X\_test değişkenlerimize scikit learn kütüphanesinde bulunan StandardScaler ile ortalamaya dayalı bir ölçekleme yaptık.



Şekil - Kod Satırı 23

Artık sınıflandırma modelimizi oluşturmanın zamanı geldi. Kriterimi entropy olarak belirttim. X\_train ve y\_train değişkenlerimizi modelimizi eğitmek için kullandık.



Şekil - Kod Satırı 24

Modelimize X\_test değişkenini verdik ve bize bir tahminde bulunmasını istedik. Bu tahminleri y\_pred değişkenimize atadık. Bu tahminler ile y\_test de tuttuğumuz doğru sonuçları karşılaştırdık.



Şekil - Kod Satırı 25

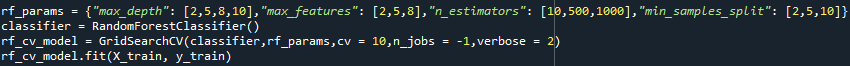
Modelimizin yaptığı tahminleri scikit learn kütüphanesinde bulunan accuracy\_score ile doğruluk oranını 0.63 olarak bulduk ve hata matrisini çıkardığımızda doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 78 fiyat ve ucuz olup ucuz tahmini yaptığı 79 fiyat ile toplamda 157 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 43 fiyat ve ucuz olup pahalı tahmini yaptığı 46 fiyat ile toplamda 89 yanlış tahmin yapmıştır.

Şekil - Rastgele Orman Doğruluk Oranı ve Hata Matrisi

**4.5.1. Rastgele Algoritması Modelinin İyileştirmesi**

Bulduğumuz doğruluk oranı ve hata matrini iyileştirmek için model iyileştirmesi(model tuning) yapalım. İlk olarak rf\_params adlı değişkenimize max\_depth, max\_features,n\_estimators ve min\_samples\_split değerlerimizin olabileceği değerleri atadık. Sonra scikit learn kütüphanesinde bulunan GridSearchCV ile tek tek bu değerleri modelimiz ile denemeler yapması için X\_train ve y\_train değişkenlerimizi verdik.



Şekil - Kod Satırı 26

En iyi değerleri görebilmek için yazdırdık.Buna göre en iyi parametreler, max\_depth için 2, max\_features için 2, n\_estimators için 10 ve min\_samples\_split için 2’dir.



Şekil - Kod Satırı 27

Bu değerleri modelimize yazıp tekrar eğitelim.



Şekil - Kod Satırı 28

İyileştirilmiş modelimize X\_test değişkenini verdik ve bize bir tahminde bulunmasını istedik. Bu tahminleri y\_pred değişkenimize atadık. Bu tahminler ile y\_test de tuttuğumuz doğru sonuçları karşılaştırdık.



Şekil - Kod Satırı 29

İyileştirilmiş modelimizin yaptığı tahminleri scikit learn kütüphanesinde bulunan accuracy\_score ile doğruluk oranını 0.71 olarak bulduk ve hata matrisini çıkardığımızda doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 95 fiyat ve ucuz olup ucuz tahmini yaptığı 82 fiyat ile toplamda 177 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 26 fiyat ve ucuz olup pahalı tahmini yaptığı 43 fiyat ile toplamda 69 yanlış tahmin yapmıştır.

Şekil - Rastgele Orman İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

**5.BÖLÜM SONUÇ VE DEĞERLENDİRME**

Yaptığım çalışmalar sonunda ortaya çıkan sonuçlar gösteriyor ki, eğer model iyileştirmesi yapmayacak ve bir tahmin istiyorsak rastgele orman algoritması bu veri seti için daha uygun bir algoritma olacaktır. Çünkü karar ağacı tahmin sonuçları doğruluk oranı 0.630 iken rastgele orman algoritmasının doğruluk oranı 0.638’dir.

Karar ağacı modelimizin yaptığı tahminler hata matrisinde de göründüğü gibi doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 73 fiyat ve ucuz olup ucuz tahmini yaptığı 82 fiyat ile toplamda 155 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 48 fiyat ve ucuz olup pahalı tahmini yaptığı 43 fiyat ile toplamda 91 yanlış tahmin yapmıştır.

Rastgele orman modelimizin yaptığı tahminler hata matrisinde de göründüğü gibi doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 78 fiyat ve ucuz olup ucuz tahmini yaptığı 79 fiyat ile toplamda 157 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 43 fiyat ve ucuz olup pahalı tahmini yaptığı 46 fiyat ile toplamda 89 yanlış tahmin yapmıştır.

Şekil 69 - Karar Ağacı Doğruluk Oranı ve Hata Matrisi Şekil 70 - Rastgele Orman Doğruluk Oranı ve Hata Matrisi

Eğer model iyileştirmesi yapılarak bir tahmin istiyorsak rastgele orman algoritması bu veri seti için daha uygun bir algoritma olacaktır. Çünkü karar ağacı tahmin sonuçları doğruluk oranı 0.71 iken rastgele orman algoritmasının doğruluk oranı 0.72’dir.

Karar ağacı iyileştirilmiş modelimizin yaptığı tahminler hata matrisinde de göründüğü gibi doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 95 fiyat ve ucuz olup ucuz tahmini yaptığı 82 fiyat ile toplamda 177 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 26 fiyat ve ucuz olup pahalı tahmini yaptığı 43 fiyat ile toplamda 69 yanlış tahmin yapmıştır.

Rasgele orman iyileştirilmiş modelimizin yaptığı tahminler hata matrisinde de göründüğü gibi doğru tahmin yaptığı, pahalı olup pahalı tahmini yaptığı 92 fiyat ve ucuz olup ucuz tahmini yaptığı 86 fiyat ile toplamda 178 doğru tahmin yapmıştır. Yanlış tahmin yaptığı, pahalı olup ucuz tahmini yaptığı 29 fiyat ve ucuz olup pahalı tahmini yaptığı 39 fiyat ile toplamda 68 yanlış tahmin yapmıştır.

Şekil 71 - Karar Ağacı İyileştirilmiş Doğruluk Oranı ve Hata Matrisi Şekil 72 - Rastgele Orman İyileştirilmiş Doğruluk Oranı ve Hata Matrisi

**KAYNAKÇA**

* [**https://acikders.ankara.edu.tr/pluginfile.php/12163/mod\_resource/content/1/veri%20toplama%20y%C3%B6ntemleri.pdf**](https://acikders.ankara.edu.tr/pluginfile.php/12163/mod_resource/content/1/veri%20toplama%20y%C3%B6ntemleri.pdf)
* [**https://www.veribilimiokulu.com/veri-kazima-nedir-neden-yapilir/**](https://www.veribilimiokulu.com/veri-kazima-nedir-neden-yapilir/)
* [**https://medium.com/kaveai/web-scraping-453e96a86195**](https://medium.com/kaveai/web-scraping-453e96a86195)
* [**https://tr.wikipedia.org/wiki/Sahibinden.com**](https://tr.wikipedia.org/wiki/Sahibinden.com)
* [**https://www.veribilimiokulu.com/buyuk-veri-on-isleme-makale-notlari/**](https://www.veribilimiokulu.com/buyuk-veri-on-isleme-makale-notlari/)
* [**https://medium.com/@ilkbaharnaz/data-preprocessing-veri-%C3%B6n-i%CC%87%C5%9Fleme-85236484f913**](https://medium.com/@ilkbaharnaz/data-preprocessing-veri-%C3%B6n-i%CC%87%C5%9Fleme-85236484f913)
* [**https://dergipark.org.tr/en/download/article-file/1873550**](https://dergipark.org.tr/en/download/article-file/1873550)
* [**https://www.turhost.com/blog/makine-ogrenmesi-machine-learning-nedir/**](https://www.turhost.com/blog/makine-ogrenmesi-machine-learning-nedir/)
* [**https://medium.com/@sivri4179/makine-%C3%B6%C4%9Frenmesinde-s%C4%B1n%C4%B1fland%C4%B1rma-algoritmalar%C4%B1-9323a07a4c94**](https://medium.com/@sivri4179/makine-%C3%B6%C4%9Frenmesinde-s%C4%B1n%C4%B1fland%C4%B1rma-algoritmalar%C4%B1-9323a07a4c94)
* [**https://bilgisayarkavramlari.com/2013/03/31/siniflandirma-classification/**](https://bilgisayarkavramlari.com/2013/03/31/siniflandirma-classification/)
* [**https://www.anaconda.com/**](https://www.anaconda.com/)
* [**https://tirendazakademi.medium.com/pandas-k%C3%BCt%C3%BCphanesi%CC%87-nedi%CC%87r-905012e654b8**](https://tirendazakademi.medium.com/pandas-k%C3%BCt%C3%BCphanesi%CC%87-nedi%CC%87r-905012e654b8)
* [**https://medium.com/@ilkebasalak/selenium-nedir-8c7d908c93e6**](https://medium.com/@ilkebasalak/selenium-nedir-8c7d908c93e6)
* [**https://kerteriz.net/python-ile-excel-dosya-islemleri/**](https://kerteriz.net/python-ile-excel-dosya-islemleri/)
* [**https://dergipark.org.tr/tr/download/article-file/787239**](https://dergipark.org.tr/tr/download/article-file/787239)
* [**https://www.karabayyazilim.com/blog/python/scikit-learn-nedir-2020-02-12-062241**](https://www.karabayyazilim.com/blog/python/scikit-learn-nedir-2020-02-12-062241)
* [**https://burcukoculu.medium.com/scikit-learn-yap%C4%B1s%C4%B1-kolay-a%C3%A7%C4%B1klamal%C4%B1-anlat%C4%B1m-%C3%B6rneklerle-afbbbb4593e5**](https://burcukoculu.medium.com/scikit-learn-yap%C4%B1s%C4%B1-kolay-a%C3%A7%C4%B1klamal%C4%B1-anlat%C4%B1m-%C3%B6rneklerle-afbbbb4593e5)
* [**https://dergipark.org.tr/en/download/article-file/1871985**](https://dergipark.org.tr/en/download/article-file/1871985)