Kocaeli İli İçin Hava Durumu Verilerinin Toplanması ve Analizi

Gülsüm Demir  
*Bilişim Sistemleri Mühendisliği*  
Kocaeli Üniversitesiİstanbul, Türkiye  
demirr.gulsumm@gmail.com

Utku Genç  
*Bilişim Sistemleri Mühendisliği*  
Kocaeli Üniversitesiİstanbul, Türkiye

utkugenc2003@gmail.com

*Özet*— Bu çalışmada Kocaeli ili geçmiş hava durumu verileri, web üzerinden otomatik olarak toplanmış ve çeşitli analizlerle görselleştirilmiştir. Hedefimiz sıcaklık, nem, rüzgar hızı, rüzgar yönü gibi temel iklim parametrelerinin zamana göre nasıl değiştiğini gözlemlemektir. Pyhton ve Selenium kütüphanesi ile veri toplama işlemi otomatik hale getirilmiş, ardından veri temizleme, görselleştirme, zaman serisi analizi, mevsimsel değişim analizi, hava durumu trend analizi ve ayrıştırma (decomposition) gerçekleştirilmiştir.

Anahtar kelimeler—web kazıma, veri seti, zaman serisi analizi, zaman serisi ayrıştırma (decomposition), mevsimsel değişim analizi, hava durumu trend analizi

# GİRİŞ

Bu çalışma, Kocaeli ilinin geçmiş yıllardaki günlük hava durumu verilerini inceleyerek iklim değişikliklerini analiz etmeyi amaçlamaktadır. İklim verileri, tarımsal planlama, şehircilik ve çeşitli çevresel analizler için temel oluşturur. Bu projede, Pyhton tabanlı bir veri toplama sistemi geliştirilmiş, zaman serisi ayrıştırması yapılmış ve çeşitli analizler gerçekleştirilmiştir.

# PROJE GEREKSİNİMLERİ

Proje, Pyhton 3.10.11 sürümü ile geliştirilmiştir. Projeyi kurmadan önce gerekli yazılımların ve kütüphanelerin yüklenmiş olduğundan emin olun.

## Gerekli Kütüphaneler ve Kurulum

### Selenium: Veri çekme işlemleri için web tarayıcı otomasyonunu sağlayan kütüphanedir.

### Pandas: Veri dosyalarını okuma, düzenleme ve analiz etme işlemleri için kullanılan kütüphanedir.

### MatPlotLib: Çizgi grafikleri, pasta grafikleri ve çubuk grafikleri gibi farklı grafik türlerini destekleyen veriyi görselleştirmek için kullanılan kütüphanedir.

### Statsmodels: Zaman serisi verilerinde trend, mevsimselli ve rastgele bileşenleri ayrıştırmak için kullanılan kütüphanedir.

# VERİ TOPLAMA VE İLGİLİ KODLARIN AÇIKLANMASI

## Web Sürücüsünün Başlatılması ve Tarayıcı Seçimi

**webdriverChrome()** komutu ile kodda Chrome tarayıcısı kullanılmıştır. Bu, Selenium kütüphanesinin Chrome tarayıcısını açarak dinamik veri çekme işlemlerini yapabilmesini sağlar. Chrome, popüler bir tarayıcı olduğu için genellikle birçok web sitesinde uyumlu çalışır. **chrome\_options.add\_argument("--incognito")**komutu ile Chrome tarayıcısı gizli modda başlatılır. Gizli mod, geçmiş veya önbellek gibi verilerin depolanmamasını sağlar. Bu mod, veri çekimi sırasında oturumun birikmeden devam etmesini ve bazı sayfalarda hız kazandırmasını sağlar.

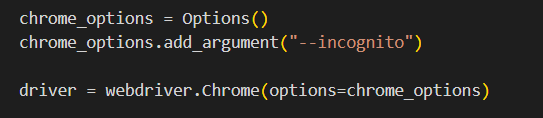
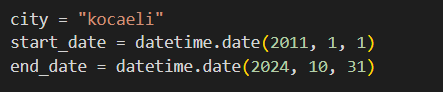
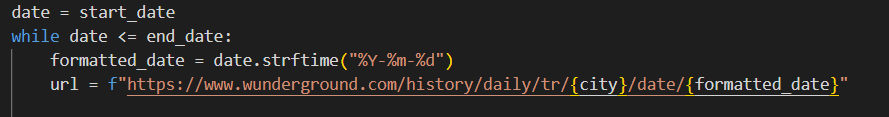


Fig.1 Web sürücüsünün başlatılması ve tarayıcı seçimi yapan kod.

## Tarih Aralığının Belirlenmesi ve Döngü Başlatılması

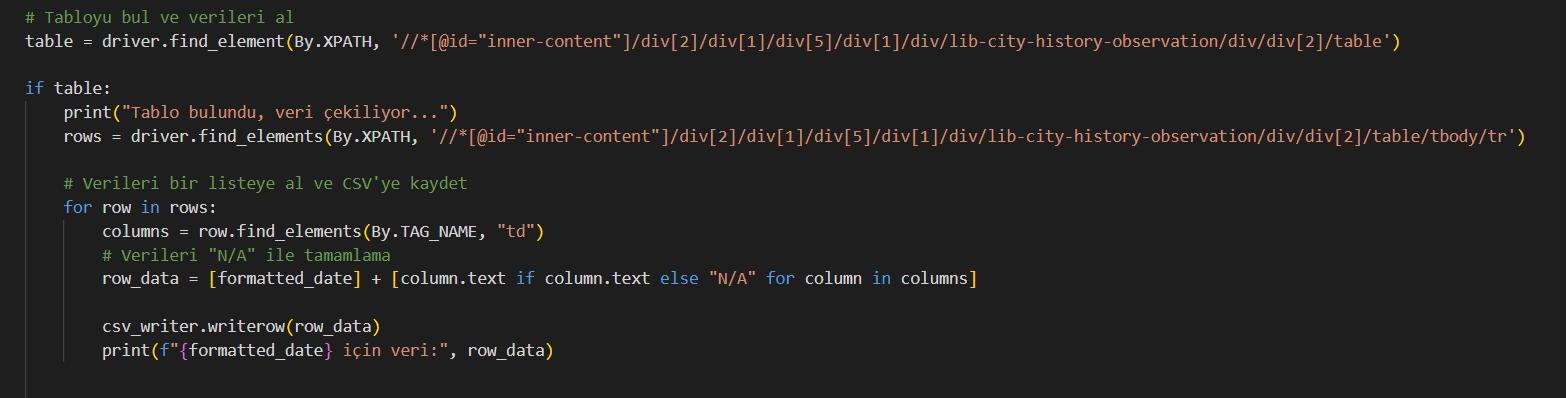
Veri çekimi için tarih aralığı belirlenir. Bu aralıkta döngü başlatılarak her gün için verilerin ayrı ayrı toplanması sağlanır.

Fig.2 Tarih aralığının belirlendiği kod .

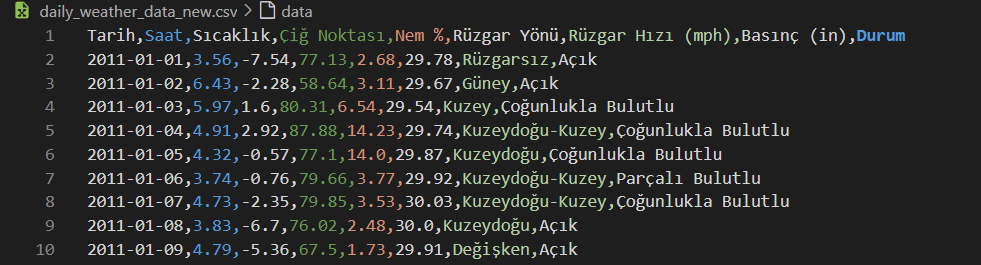
Fig.3 Döngü başlatan ve ilgili siteye erişen kod.

## Verilerin Çekilmesi ve Kaydedilmesi

## İlk adımda, hedef web sayfasındaki verileri içeren tabloyu bulmak için Selenium kütüphanesinin **find\_element** fonksiyonu kullanılmıştır. Tablo, belirli bir XPath ile tanımlanmış ve bu XPath, HTML sayfasındaki öğeye ulaşmamızı sağlamaktadır. Web sayfası yüklendikten sonra, tablonun tüm satırları aynı şekilde **find\_elements** metodu ile alınır. Her bir satırdaki hücreler **(column.text)** toplanarak işlenir. Verilerin eksik olup olmadığı kontrol edilir ve boş hücreler için "N/A" değeri atanır, böylece veri kaybı yaşanmaz. İşlenen her bir satırdaki veriler, **csv\_writer.writerow(row\_data)** komutu ile bir CSV dosyasına kaydedilir. Aynı zamanda, her veri kaydının başarıyla işlendiği ve kaydedildiği, konsola yazdırılarak kullanıcıya bildirilir.

Fig.4 Verilerin çekilmesini ve kaydedilmesini sağlayan kod parçası.

Yukarıdaki kod ana sınıfa eklenip çalıştırıldığında kaydedilen CSV dosyasında çekilen veriler aşağıdaki gibi görünecektir:

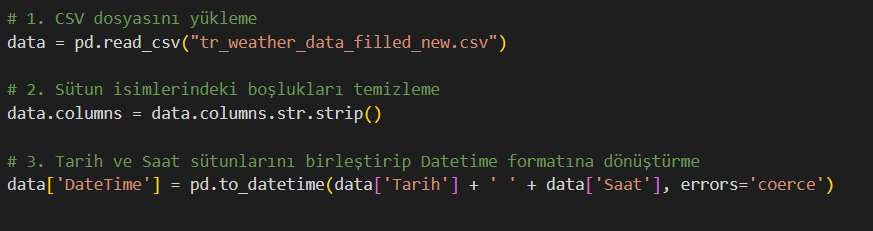
Fig.5 Web kazıma işleminde örnek çıktı.

# Veri Ön İşleme

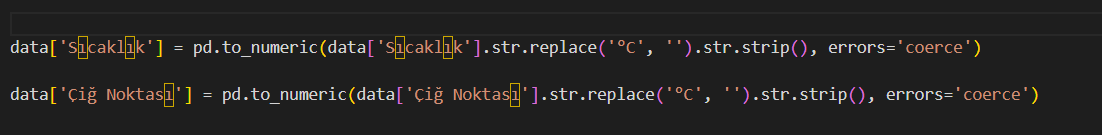
Bu proje kapsamında hava durumu verilerini analiz etmeye başlamadan önce, veri ön işleme süreci uygulandı. Bu süreç, veriyi doğru formatta kullanabilmek ve analiz için hazır hale getirmek amacıyla önem taşımaktadır.

## Verilerin Görselleştirilmesi

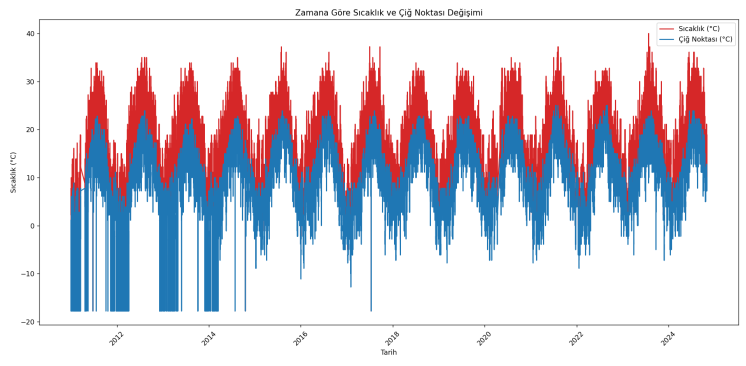
İlk adım olarak, hava durumu verilerini içeren CSV dosyası Pandas kütüphanesi ile yüklendi. Bu işlemle veriler df adlı bir DataFrame'e aktarıldı. Daha sonrasında veri setindeki tarih ve saat sütunları birleştirilerek DateTime formatına dönüştürüldü. Bu işlemle, DateTime sütunu oluşturulmuş oldu ve saat bilgileriyle birlikte tarih bilgisi de doğru formatta alındı.

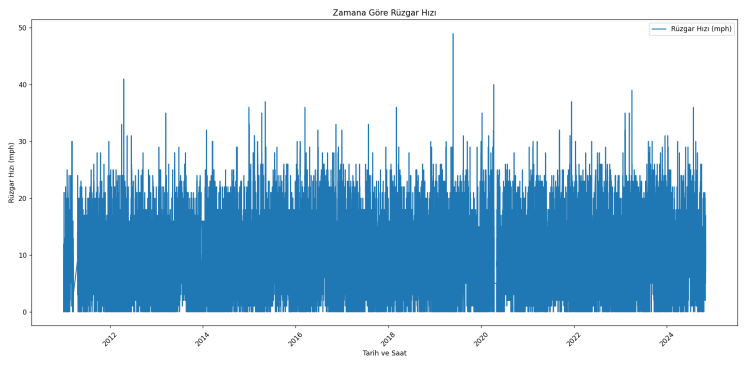
Fig.6 CSV dosyasının yüklenmesi ve sütunların birleştirilmesine ilişkin kod parçası

Daha sonrasında sayısal olmayan verilerin düzenlenmesi amacıyla sıcaklık, çiğ noktası gibi sayısal olmayan sütunlarda birimler temizlendi ve sayısal değerlere dönüştürüldü.

Fig.7 Sayısal olmayan verilerin düzenlenmesini içeren kod parçası.

Verinin görselleştirilmesi amacıyla bu aşamada sıcaklık, rüzgar hızı, nem gibi verilerin zamanla değişimi çizgi grafiği ile, rüzgar yönü polar grafik ie ayrıca hava durumu dağılımları pasta grafiği ile gösterildi.

 Fig.8 Zamana göre sıcaklık ve çiğ noktası değişimi.

 Fig.9 Zamana göre rüzgar hızı değişimi.

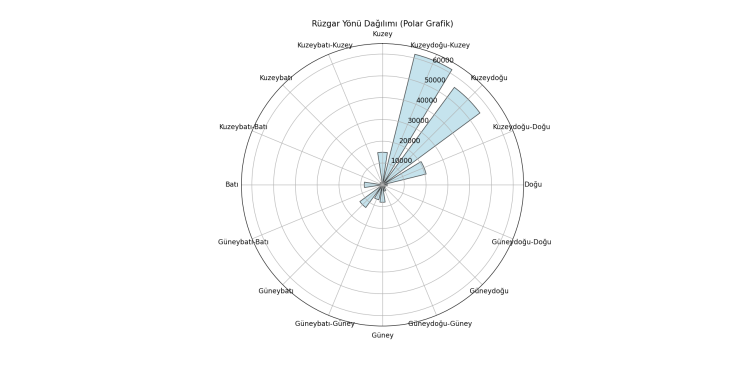
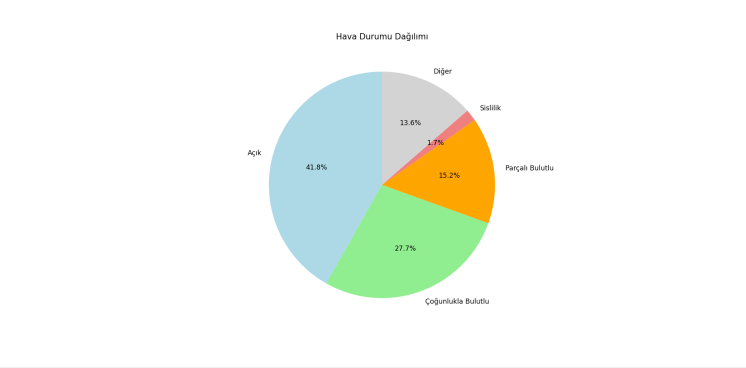
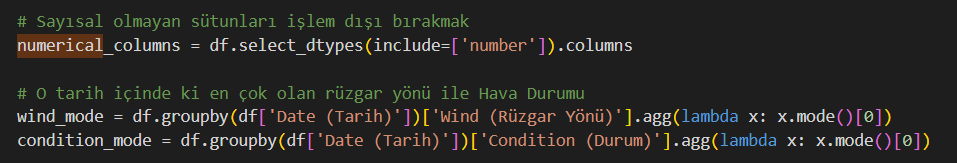


Fig.10 Rüzgar yönü dağılım grafiği.

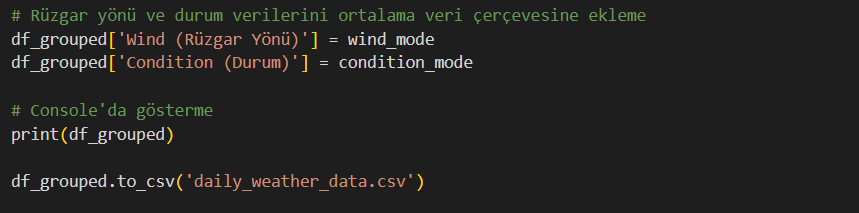
 Fig.11 Hava durumu dağılım grafiği.

## Verilerin Örneklenmesi

Veriler, tarihsel sıralamaya göre gruplanarak her gün için ortalama sayısal veriler ve en sık görülen hava durumu ile rüzgar yönü değerleri hesaplanır. Bu işlem, zaman serisi analizi kapsamında **örneklemleme** ve **özetleme** olarak nitelendirilebilir. Bu amaçla öncelikle analiz sırasında sadece sayısal verilere odaklanabilmesi için, sayısal olmayan sütunlar örneğin**, Rüzgar Yönü** ve  **Durum** işlem dışı bırakılmıştır. Her bir tarih için bu sütunlar üzerinde en sık görülen değeri bulmak amacıyla **.mode()** fonksiyonu kullanılmıştır. Böylelikle her bir gün için en yaygın rüzgar yönünün ve hava durumunun belirlenmesine olanak tanınmıştır.

 Fig.12 Sayısal olmayan sütunların kaldırılması ve sık görülen değerlerin bulunmasını içeren kod parçası.

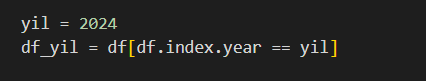
Daha önce hesaplanan en sık görülen rüzgar yönü ve hava durumu değerleri, sayısal sütunların ortalama değerleriyle birlikte oluşturulan yeni veri çerçevesine **(df\_grouped)** eklenmiştir. Sonuçlar, bu veri çerçevesinde bir CSV dosyasına **(daily\_weather\_data.csv)** yazdırılmıştır.

 Fig.13 Sütunların veri çerçevesine eklenmesi ve CSV dosyasına kaydedilmesini gösteren kod parçası.

## Ayrıştırma (Dekompozisyon)

Zaman serisi analizinde, seriyi daha iyi anlayabilmek için veri **trend**, **mevsimsellik**, ve **rastlantısallık** bileşenlerine ayrıldı. Bu işlem, **seasonal\_decompose** fonksiyonu ile 24 saatlik döngü parametresi kullanılarak yapıldı. Burada, tam veri seti ve belirlenen yıla ait veri için ayrı ayrı dekompozisyon yapıldı.

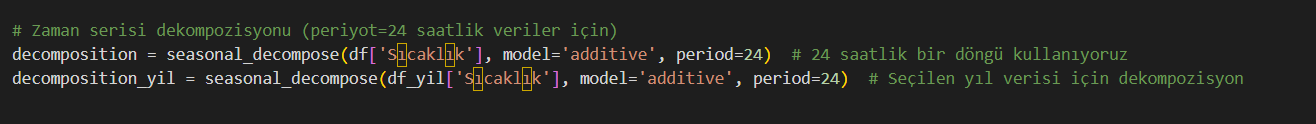
Belirli bir yıl için analiz yapmak amacıyla **df[df.index.year = = yil**]ifadesiyle 2024 yılına ait veriler seçildi ve **df\_yil** olarak saklandı. Bu yıl bazlı ayrım, belirli bir yıldaki sıcaklık değişimlerinin daha detaylı incelenmesini sağlar.

 Fig.14 Yıla göre veri seçme işlemini sağlayan kod.

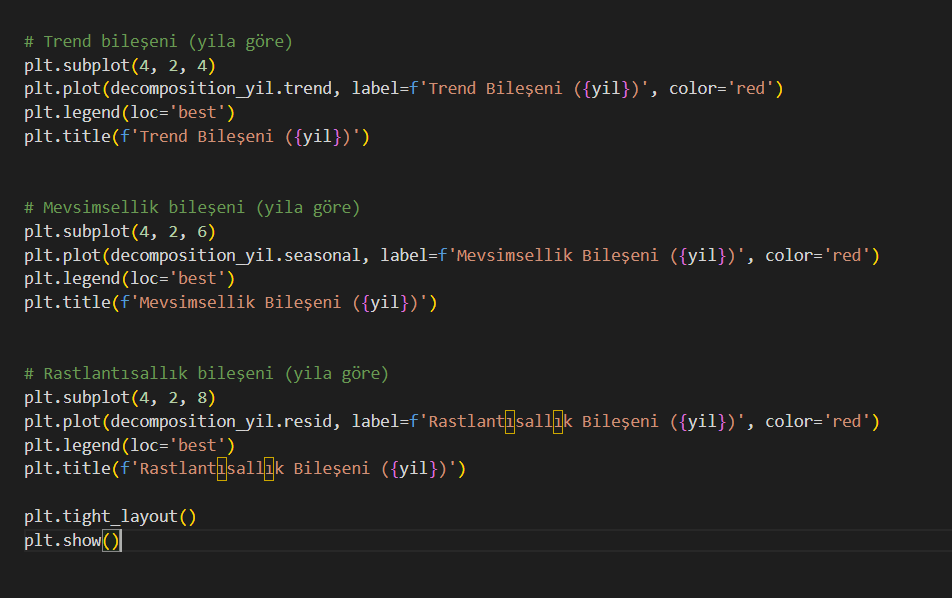
Zaman serisi analizinde, veriyi bileşenlerine ayırmak, trend (uzun vadeli değişimler), mevsimsellik (dönemsel desenler) ve rastlantısallık (gürültü) gibi unsurları ayrı ayrı incelememize olanak tanır.

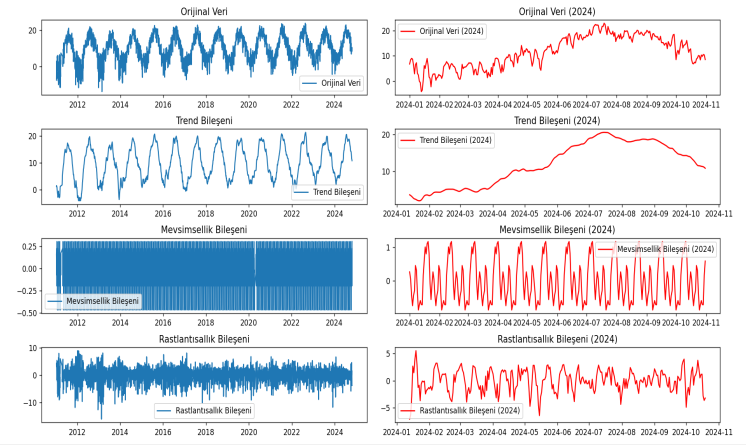
**seasonal\_decompose** fonksiyonu ile tam veri seti verisi için ayrı ayrı dekompozisyon uygulandı ve period=24 kullanılarak her gün için 24 saatlik veriye göre mevsimsel döngü ayarlandı. Additive model kullanılarak, verinin bileşenlerinin toplama dayalı bir yapıya sahip olduğu varsayıldı. Bu durumda, gözlemler (serinin her noktası) bileşenlerin toplamı olarak ifade edilir:

Orijinal = Trend + Mevsimsellik + Rastlantısallık.

 Fig.15 Zaman serisi dekompozisyonunu sağlayan kod parçası.

Her bir bileşen hem tüm veri seti hem de 2024 yılı verisi için görselleştirildi. Böylelikle serinin her bileşeninin detaylıca incelenmesi sağlandı.

 Fig.16 Verinin görselleştirilmesini sağlayan kod parçası.

Fig.17 Verinin setinin grafiği.

##### Kaynaklar

1. *Selenium Documentation, "Selenium WebDriver Documentation," Available:* [*https://www.selenium.dev/documentation/*](https://www.selenium.dev/documentation/)

##### Ekler

Selenium indirme linki: [Selenium](https://www.selenium.dev/downloads/)

Proje Kaynak Kodları : [Proje Github Linki](https://github.com/Utku-Genc/WeatherWebScraping)

Elde Edilen Verilere İlişkin Tablo : [Excel](https://docs.google.com/spreadsheets/d/1Rg4cs4zh44YMo01i1jVTn9LwI-rLiVgC/edit?gid=1587810924#gid=1587810924)

Veri Seti : [OneDrive](https://drive.google.com/drive/u/0/folders/1eSPkRUiUx6AkxZYrtghkakOASbia3wuu)