

```
!pip install geoip2
import requests
import json
import pandas as pd
import geoip2.database
import matplotlib.pyplot as plt
import seaborn as sns

# GeoLite2 dosya yolu
geoip_db_path = '/content/GeoLite2-Country.mmdb' # Dosya yolu

# JSON dosyasını oluştur veya temizle
with open('output.json', 'w') as f:
    f.write("")

# İlk 400 veriyi çekmek için 5 sayfayı dolaş
for page in range(1, 6):
    response = requests.get(
        f'https://www.usom.gov.tr/api/address/index?type=ip&page={page}',
        headers={'accept': 'application/json'}
    )
    # Veriyi JSON dosyasına ekle
    with open('output.json', 'a') as f:
        json.dump(response.json(), f)
        f.write('\n') # Sayfalar arasına ayraç ekle

print("Veriler output.json dosyasına kaydedildi.")

# output.json dosyasını oku
with open('output.json', 'r') as f:
    json_lines = f.readlines()

# Boş bir data sözlüğü oluştur
data = {
    "id": [],
    "url": [],
    "type": [],
    "desc": [],
    "source": [],
    "date": [],
    "criticality_level": [],
    "connectiontype": []
}
```

```

# Her sayfadaki veriyi işleme
for line in json_lines:
    page_data = json.loads(line)
    for model in page_data["models"]:
        data["id"].append(model["id"])
        data["url"].append(model["url"])
        data["type"].append(model["type"])
        data["desc"].append(model["desc"])
        data["source"].append(model["source"])
        data["date"].append(model["date"])
        data["criticality_level"].append(model["criticality_level"])
        data["connectiontype"].append(model["connectiontype"])

# Veri çerçevesi oluştur
df = pd.DataFrame(data)

# IP adreslerinden ülke bilgilerini çek
def get_country(ip):
    try:
        with geoip2.database.Reader(geoip_db_path) as reader:
            response = reader.country(ip)
            return response.country.name
    except Exception as e:
        print(f"Hata: {e}")
        return "Unknown"

# Ülke bilgilerini ekle
df['country'] = df['url'].apply(get_country)

# Kritiklik seviyesine göre gruplama (8 ve 10)
df_grouped = df[df['criticality_level'].isin([8, 10])]

# En sık geçen 5 ülkeyi görselleştir
country_counts = df_grouped['country'].value_counts().head(5)
plt.figure(figsize=(10, 6))
sns.barplot(x=country_counts.index, y=country_counts.values, palette="Set2")
plt.title('Kritiklik Seviyesine Göre İlk 5 Ülke')
plt.xlabel('Ülke')
plt.ylabel('Sayı')
plt.show()

```

Adım 1: Top 5 IP Listesi

Verdiğiniz verilere göre, **IP adresleri ABD** kaynağına sahip ve tüm IP adresleri **Operational Technology (OT)** bağlantısı kullanıyor, yalnızca 3 IP adresi **Business**

Communication (BC) bağlantısı kullanıyor. Ayrıca, her IP adresinin **açıklama** kısmında çoğunlukla **MI** (muhtemelen "Machine Interface" veya "Monitoring Interface") ifadesi geçiyor.

Buradaki **Top 5** IP adresini **kritik seviyeye** göre sıralayabiliriz. Ancak **kritik seviyeleri** aynı olduğundan, diğer önemli faktörlere (IP adresi, açıklama, vb.) göre bir sıralama yapabiliriz.

Örnek Görselleştirme:

Aşağıdaki liste, **top 5 IP adresini** gösteriyor:

1. **IP:** 45.142.182.126
Connection Type: OT
Description: MI
Source: US
2. **IP:** 178.211.139.105
Connection Type: OT
Description: MI
Source: US
3. **IP:** 194.50.16.198
Connection Type: OT
Description: MI
Source: US
4. **IP:** 5.181.188.158
Connection Type: OT
Description: MI
Source: US
5. **IP:** 45.148.10.230
Connection Type: OT
Description: MI
Source: US

Adım 2: Şehir ve Ülke Bilgileri

Bu noktada, her IP adresinin kaynağını **şehir ve ülke bilgilerine** çevirerek, daha anlamlı bir görselleştirme oluşturabiliriz. Ancak, **IP adresi coğrafi konumlama** (IP Geolocation) yapılması gerekir. Bu işlem için bir coğrafi veri kaynağı (örneğin, **MaxMind**, **IP2Location** gibi) kullanmak gerekir. Örnek bir geolocation sonucu şu şekilde olabilir:

- **IP: 45.142.182.126 - New York, US**
- **IP: 178.211.139.105 - Los Angeles, US**
- **IP: 194.50.16.198 - Chicago, US**
- **IP: 5.181.188.158 - Houston, US**

- **IP: 45.148.10.230 - Dallas, US**

Adım 3: Veriyi Görselleştirme

Görselleştirme amacıyla **şehir, ülke ve IP adreslerinin** bulunduğu **bar grafiği** veya **coğrafi harita** kullanabilirsiniz. Örneğin:

1. **Bar Grafiği:** Top 5 IP adresini **şehir ve ülke bilgileriyle** sıralayıp, bar grafiğiyle görselleştirebiliriz. X ekseninde şehir, Y ekseninde ise her IP adresinin **kritik seviyesini** veya **bağlantı türünü** gösteririz.
2. **Coğrafi Harita:** IP adreslerinin coğrafi yerlerini harita üzerinde **dot veya pin** kullanarak gösterebiliriz.

Aşağıda, şehir ve ülke bilgileriyle örnek bir **bar grafiği** nasıl görünebilir:

sql

Kodu kopyala

-----	-----
IP Address	**City, Country**
-----	-----
45.142.182.126	New York, US
178.211.139.105	Los Angeles, US
194.50.16.198	Chicago, US
5.181.188.158	Houston, US
45.148.10.230	Dallas, US
-----	-----

Görselleştirme Önerisi:

Top 5 IP Address (by City & Country)

Bir **bar chart** kullanarak her şehri veya ülkeyi **X ekseninde**, IP adreslerini **Y ekseninde** sıralayarak, bu verileri görsel olarak sunabilirsiniz.

Eğer coğrafi bir harita üzerinden görselleştirme yapmanız gerekirse, **IP Geolocation** kullanarak, her IP'nin **yerini** harita üzerinde gösterebilirsiniz.