DEÜ Fen Fakültesi
Bilgisayar Bilimleri Bölümü

BİL 3013 Veri Madenciliğine Giriş

Ödev 2. Kümeleme

Kodların ve Veri setlerinin Kaggle linki:
https://www.kaggle.com/code/ozgurd5/bil3013-data-mining-assignment-2-code
https://www.kaggle.com/datasets/ozgurd5/bil3013-data-mining-assignment-2-data

Grup 2
Öğrenci: Özgür Dalbeler - 2022280084

Öğretim Üyesi: Prof. Dr. Efendi NASİBOĞLU

İzmir 2024

1- Ödevin Tanımı

Herkes bir önceki ödevinde topladığı verilerdeki Fiyat (veya Kira bedeli) atributunu kullanarak, piyasayı “Ucuz”, “Orta” ve “Pahalı” gibi 3 grupa (kümeye) ayırması gerekiyor. Kümeleme için K-ortalamalar kümeleme algoritması ve Öklit uzaklığı kullanılmalıdır.

Sakai sistemine aşağıdaki dosyaların yüklenmesi gerekiyor:

a) Topladığınız verilerin temizlenmemiş halini içeren ham veriseti,

b) Temizlenmiş ve ön hazırlık işlemlerinden geçmiş veriseti,

c) Kümeleme sonuçlarının yansıtılması (kümeleme sonrası her verinin ait olduğu kümeyi belirten numarası 0,1 veya 2 olarak temizlenmiş veri setine ayrı bir sütun şeklinde eklensin)

d) Her bir kümedeki nesne sayısı, min, max, ortalama değeri, standart sapması, histogramı gibi gösterge ve grafiklerini oluşturarak hazırlanmış ve piyasa analizini içeren rapor dosyası,

e) Program kodlarını içeren dosya.

2- Kullanılan Yöntemler ve Teknolojiler

Pandas: Pandas, Python'da veri analizi ve manipülasyonu için kullanılan bir kütüphanedir. DataFrame, tablo yapısında verileri saklar ve CSV, XLSX, JSON gibi formatlardan veri okuma ve yazma imkanı sunar.

Kurulumu:
```
pip install pandas
```

Kullanımı:
```
import pandas as pd
df = pd.DataFrame(ilanlar)
```

Openpyxl: Pandas kütüphanesinin Excel dosyasına veri yazabilmesi için openpyxl kütüphanesinin yüklenmesi gerekir.

Kurulumu:
```
pip install openpyxl
```

Kullanımı:
```
df.to_excel('ilanlar.xlsx', index=False)
```

Matplotlib: Matplotlib, Python'da veri görselleştirme için kullanılan bir kütüphanedir. Grafikler, histogramlar, çubuk grafikler ve diğer görseller oluşturmak için sıkça kullanılır.

Kurulumu:
```
pip install matplotlib
```

Kullanımı:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 35]

plt.plot(x, y, marker='o')
plt.xlabel('X Ekseni')
plt.ylabel('Y Ekseni')
plt.title('Örnek Grafik')
plt.show()
```

3- Uygulama

Bu projede **hepsiemlak.com** sitesindeki Buca ilçesindeki kiralık dairelerin ilanlarının fiyatları k-means kümeleme yöntemi kullanılarak “Pahalı”, “Orta” ve “Ucuz” olmak üzere üç gruba kümelenmiştir.

Uygulamanın önemli adımları aşağıdaki gibidir:

3.1- Modüller

```
# Pandas modülü
import pandas as pd

# Matplotlib modülü
import matplotlib.pyplot as plt
```

3.2- Ham veriyi okumak

```
# Excel dosyasını oku
df = pd.read_excel("ham_veri.xlsx")
```

3.3- Veriyi temizlemek

Veri temizleme aşamasında duplikeler ve boş veriler kaldırılıp fiyat formatlanmış ve ardından veriler küçükten büyüğe sıralanmıştır.

```
# Duplikeleri Link sütununa göre tespit et ve yazdır
duplikeler = df[df.duplicated(subset="Link", keep="first")]
print("Duplikeler (aynı linke sahip satırlar): ", len(duplikeler))
print(duplikeler[["Başlık", "Fiyat", "Link"]])

# Boş fiyatları tespit et ve yazdır
bos_fiyatlar = df[df["Fiyat"].isna()]
print("\nBoş Fiyatı Olan Satırlar:", len(bos_fiyatlar))
print(bos_fiyatlar[["Başlık", "Fiyat", "Link"]])

# Duplikeleri kaldır
df = df.drop_duplicates(subset="Link", keep="first")

# Fiyatı olmayan satırları kaldır
df = df.dropna(subset=["Fiyat"])
```

```
# Fiyatları düz sayı formatına çevir
# Değerler numpy değeri olarka dönüyor, bunu string'e çevir, ardından regex
kullanarak sadece sayıları al ve integer'a çevir
# \D sayı olmayan karakterleri temsil eder
df["Fiyat"] = df["Fiyat"].astype(str).str.replace(r"\D", "",
regex=True).astype(int)

# Fiyatları küçükten büyüğe sırala
df = df.sort_values("Fiyat")
```

3.4- Outlier Tespiti

Outlier tespiti için IQR yöntemi kullanılmıştır. Alt değer için katsayı 1.5 üst değer için ise 3. seçilmiştir.

```
# IQR yöntemi

# Çeyrek değerleri hesapla
Q1 = df["Fiyat"].quantile(0.25)
Q3 = df["Fiyat"].quantile(0.75)
IQR = Q3 - Q1

# Outlier sınırlarını belirle
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 3 * IQR

# Outlierları seç
outliers = df[(df["Fiyat"] < lower_bound) | (df["Fiyat"] > upper_bound)]
print("\nOutlierlar (IQR yöntemi):")
print(outliers[["Fiyat"]])

# Outlierları kaldır
df = df[(df["Fiyat"] >= lower_bound) & (df["Fiyat"] <= upper_bound)]
```

3.5- K-Means Kümeleme

Hazır kütüphaneler aşırı kuvvetli ve çok parametreli olduğu için manuel k-means algoritması yazılmıştır.

```
k = 3

# Veri kümesindeki fiyat sütununu al.
# Değerler numpy değeri olarak dönüyor, bunu integer'a cast et, ardından
listeye çevir
fiyatlar = df["Fiyat"].values.astype(int).tolist()

# K değeri kadar rastgele merkez seç
# Merkezlerin tipini float yap ve listeye çevir
merkezler = df.sample(n=k)["Fiyat"].values.astype(float).tolist()

# Önceki merkezleri tut, başlangıçta tüm değerleri 0 yap
önceki_merkezler = [0] * k

iterasyon_sayacı = 0
```

```
# Merkezler değişene kadar döngüyü devam ettir
while True:
    iterasyon_sayacı += 1
    print("iterasyon:", iteration_counter)

    # K değeri kadar küme oluştur
    kümeler = []
    for i in range(k):
        kümeler.append([])

    # Veri kümesindeki her fiyat için en yakın merkezi bul
    for fiyat in fiyatlar:
        en_yakin_merkez_index = 0
        en_kucuk_fark = abs(fiyat - merkezler[0]) # Öklit mesafesi

        for i in range(1, k):
            fark = abs(fiyat - merkezler[i]) # Öklit mesafesi
            if fark < en_kucuk_fark:
                en_yakin_merkez_index = i
                en_kucuk_fark = fark

        kümeler[en_yakin_merkez_index].append(fiyat)

    # Her kümenin ortalamasını alarak yeni merkezleri hesapla
    for i in range(k):
        önceki_merkezler[i] = merkezler[i]
        merkezler[i] = sum(kümeler[i]) / len(kümeler[i])

    # Merkezler değişmediyse döngüyü sonlandır
    if önceki_merkezler == merkezler:
        break
```

3.6- Grafikler

Verileri görselleştirmek için sayıların değerlerinin gösterildiği sade bir grafik, scatter grafik ve box grafik kullanılmıştır.

```
# Grafik için plot oluştur
plt.figure(figsize=(10, 6), dpi=300)

# Grafik plot başlığını belirle
plt.title(f"Grafik")

# Index sütununu göster
plt.xlabel("Index")

# Fiyat sütununu göster
plt.ylabel("Fiyat")

# Grafik oluştur
plt.plot(range(len(fiyatlar)), fiyatlar, marker='o')

# Grafiği kaydet
```

```
plt.savefig("grafik.png", dpi=300, bbox_inches='tight')

# Histogram için bir plot oluştur
plt.figure(figsize=(10, 6), dpi=300)

# Başlığı belirle
plt.title(f"Histogram")

# Fiyat sütununu göster
plt.xlabel("Fiyat")

# Histogram oluştur
plt.hist(fiyatlar, bins=100)

# Histogram'ı kaydet
plt.savefig("histogram.png", dpi=300, bbox_inches='tight')

# Scatter için plot oluştur
plt.figure(figsize=(10, 6), dpi=300)

# Plot başlığını belirle
plt.title(f"K-Means Scatter Plot")

# Index sütununu göster
plt.xlabel("Index")

# Fiyat sütununu göster
plt.ylabel("Fiyat")

# Grid çiz
plt.grid(True)

# Veri kümesindeki index değerlerini x ekseninde göstermek için al
x_values = df.index

# Veri kümesindeki fiyat değerlerini y ekseninde göstermek için al
y_values = df["Fiyat"]

# Her küme için farklı renkler belirle
colors = ["red", "green", "blue"]
color_values = []
for i in range(k):
    color_values.extend([colors[i]] * len(kümeler[i]))

# Veri noktalarını çiz
plt.scatter(x_values, y_values, color=color_values, s=2)

# Merkezleri çiz
plt.scatter(range(k), merkezler, color="black", s=100, marker="x")

# Scatter Plot'u kaydet
plt.savefig("scatter_plot.png", dpi=300, bbox_inches='tight')
```

```
# Box için bir plot oluştur
plt.figure(figsize=(10, 6), dpi=300)

# Başlığı belirle
plt.title(f"K-Means Box Plot")

# Index sütununu göster
plt.xlabel("Küme")

# Fiyat sütununu göster
plt.ylabel("Fiyat")

# Grid çiz
plt.grid(True)

# Kümeleri çiz
plt.boxplot(kümeler, patch_artist=True, showmeans=True, showfliers=False)

# Box plot'u kaydet
plt.savefig("box_plot.png", dpi=300, bbox_inches='tight')

# Her bir küme için histogram oluştur
df["Küme"] = -1
for i in range(k):
    df.loc[df["Fiyat"].isin(kümeler[i]), "Küme"] = i + 1

for i in range(k):
    # Histogram için bir plot oluştur
    plt.figure(figsize=(10, 6), dpi=300)

    # Başlığı belirle
    plt.title(f"Küme {i + 1} Histogram")

    # Fiyat sütununu göster
    plt.xlabel("Fiyat")

    # Histogram oluştur
    plt.hist(kümeler[i], bins=100)

    # Histogram'ı kaydet
    plt.savefig(f"küme_{i + 1}_histogram.png", dpi=300, bbox_inches='tight')
```

3.7- Veri Sayısı, Ortalama, Mod, Medyan, Standart Sapma, Min, Max

```
# Veri sayısı, ortalama, mod, medyan, standart sapma, minimum ve maksimum değerleri hesapla
veri_sayisi = len(df)
print("\nVeri Sayısı:", veri_sayisi)

ortalama = df["Fiyat"].mean()
print("Ortalama:", ortalama)
```

```python
mod = df["Fiyat"].mode().values[0]
print("Mod:", mod)

medyan = df["Fiyat"].median()
print("Medyan:", medyan)

standart_sapma = df["Fiyat"].std()
print("Standart Sapma:", standart_sapma)

minimum = df["Fiyat"].min()
print("Minimum:", minimum)

maksimum = df["Fiyat"].max()
print("Maksimum:", maksimum)

# Her bir küme için bunları hesapla
for i in range(k):
    print(f"\nKüme {i + 1} ({len(kümeler[i])} eleman):")
    print("Ortalama:", sum(kümeler[i]) / len(kümeler[i]))
    print("Mod:", pd.Series(kümeler[i]).mode().values[0])
    print("Medyan:", pd.Series(kümeler[i]).median())
    print("Standart Sapma:", pd.Series(kümeler[i]).std())
    print("Minimum:", min(kümeler[i]))
    print("Maksimum:", max(kümeler[i]))
```

3.8- Temizlenmiş Veriyi Kaydetmek

```python
# Fiyatları küçükten büyüğe sırala, kümeleriyle birlikte yaz
df = df.sort_values("Fiyat")

df["Küme"] = -1
for i in range(k):
    df.loc[df["Fiyat"].isin(kümeler[i]), "Küme"] = i + 1

# Excel dosyasına yaz
print("\nExcel dosyası 'temizlenmiş_veri.xlsx' olarak oluşturuldu.")
df.reset_index()[["Fiyat","Küme"]].to_excel("temizlenmiş_veri.xlsx",
index=False)
```

3.9- Ekran Görüntüleri

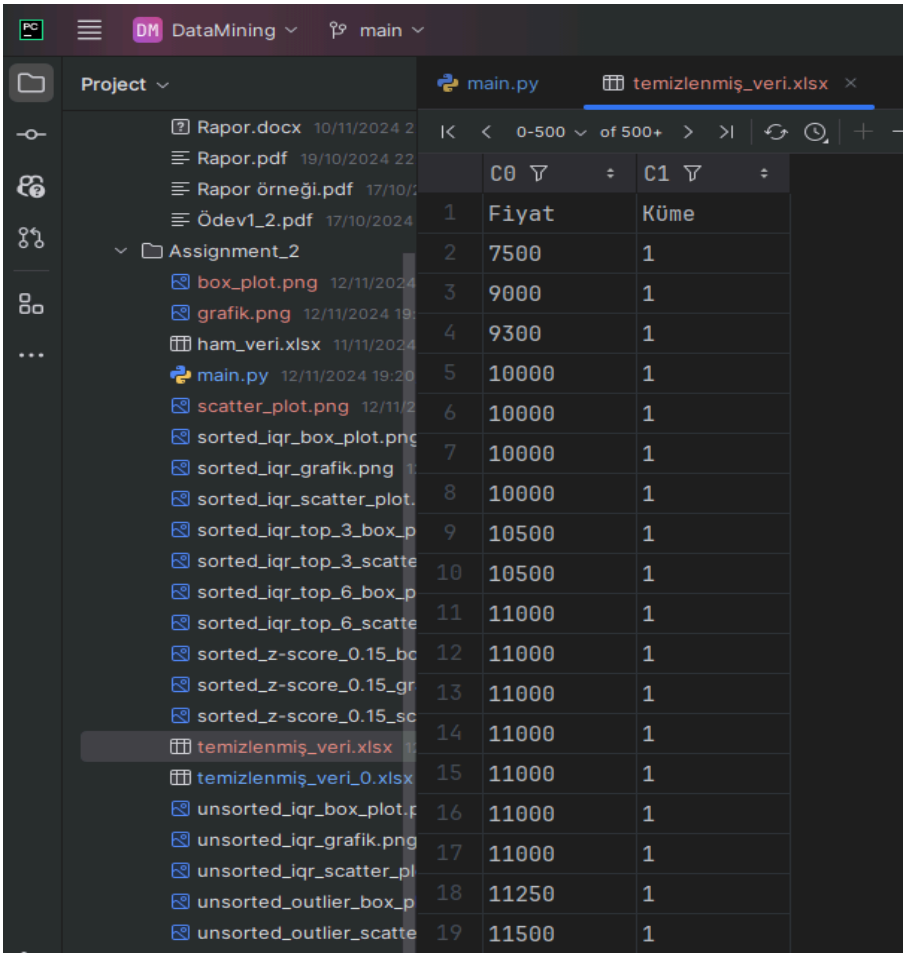
```
Duplikeler (aynı linke sahip satırlar):  0
Empty DataFrame
Columns: [Başlık, Fiyat, Link]
Index: []

Boş Fiyatı Olan Satırlar: 0
Empty DataFrame
Columns: [Başlık, Fiyat, Link]
Index: []
```

```
Outlierlar (IQR yöntemi):
       Fiyat
36       200
133      200
153      200
431      300
293      300
101      350
447      400
473      400
91       400
213     5000
418    45000
240    45000
350    45000
70     50000
111    50000
570   300000
323  3100000
```

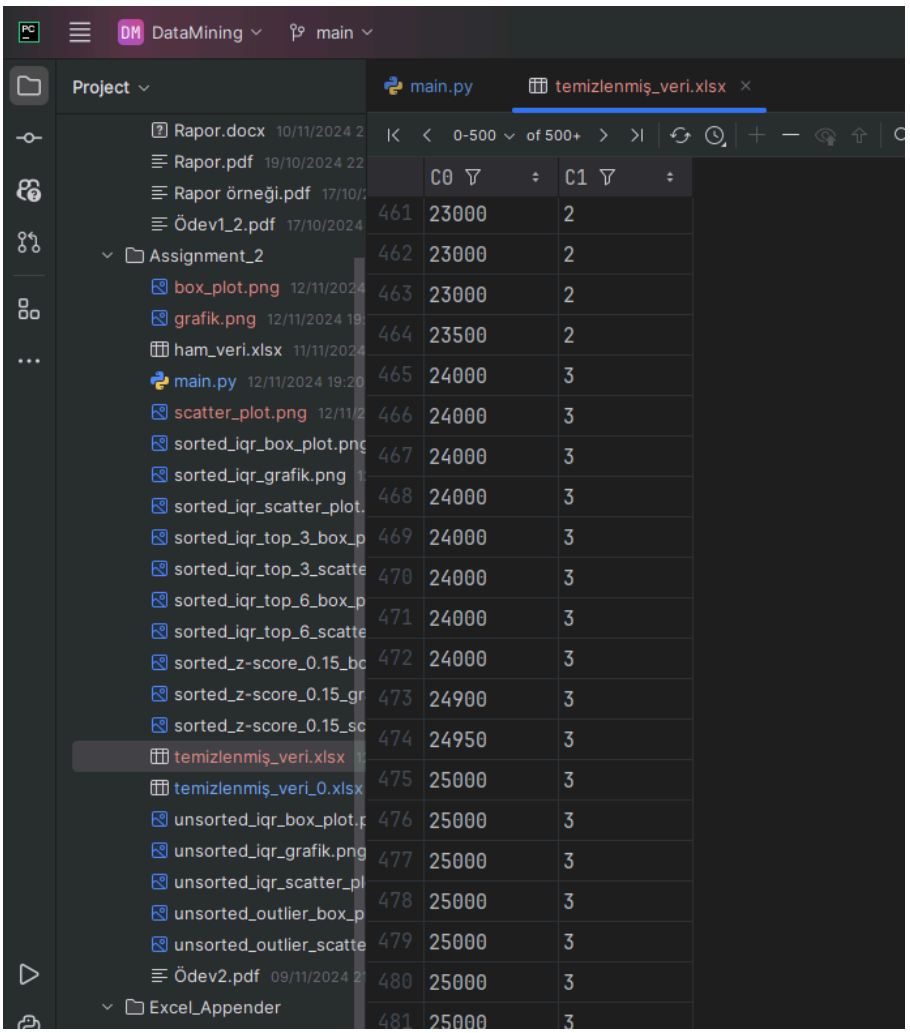
```
iterasyon: 1
iterasyon: 2
iterasyon: 3
iterasyon: 4
iterasyon: 5
iterasyon: 6

Veri Sayısı: 555
Ortalama: 18642.98018018018
Mod: 15000
Medyan: 18000.0
Standart Sapma: 5164.77726003899
Minimum: 7500
Maksimum: 40000
```

```
Küme 1 (229 eleman):
Ortalama: 14171.61135371179
Mod: 15000
Medyan: 15000.0
Standart Sapma: 1652.8521951443013
Minimum: 7500
Maksimum: 16500

Küme 2 (233 eleman):
Ortalama: 19522.339055793993
Mod: 20000
Medyan: 20000.0
Standart Sapma: 1853.159290070134
Minimum: 16950
Maksimum: 23000
```

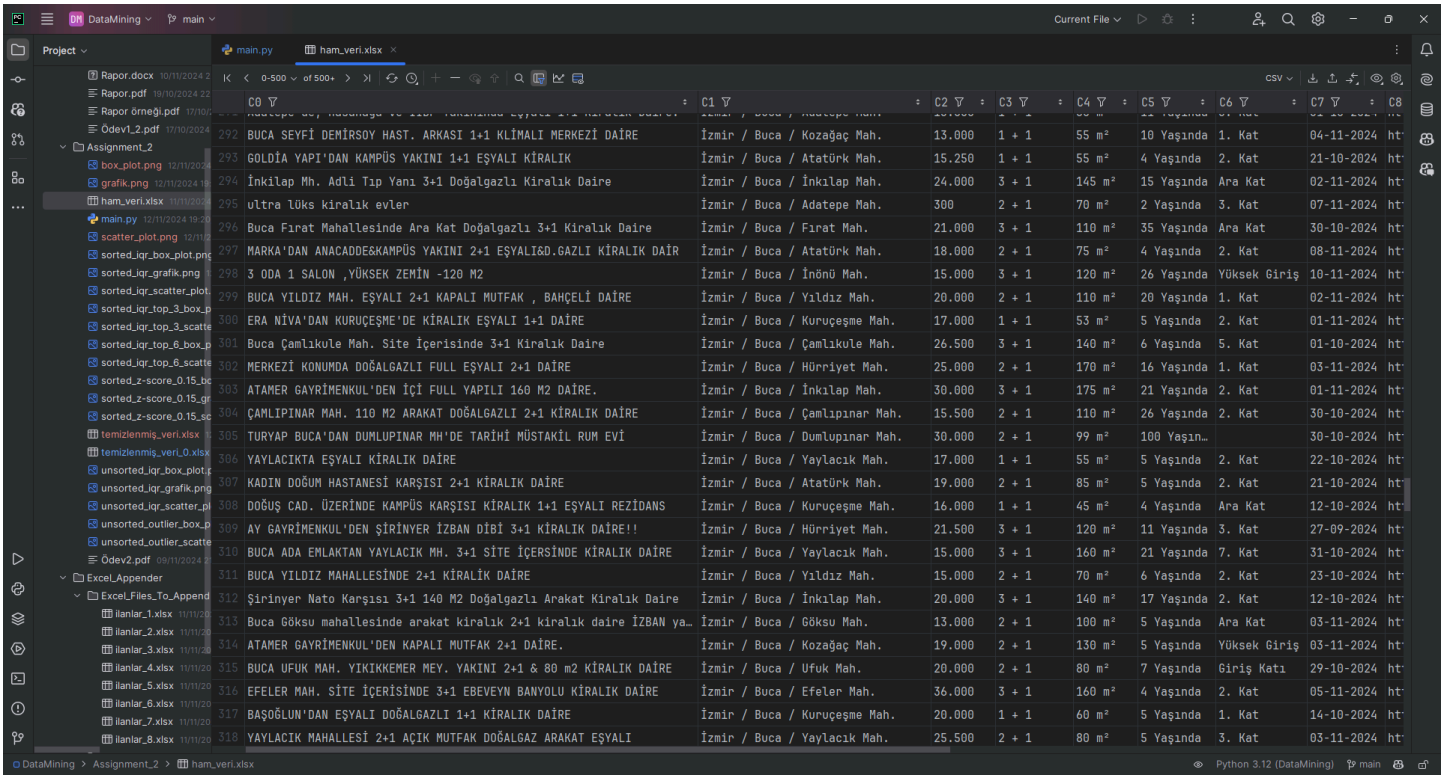
```
Küme 3 (93 eleman):
Ortalama: 27450.0
Mod: 25000
Medyan: 26000.0
Standart Sapma: 3881.785783575031
Minimum: 23500
Maksimum: 40000

Excel dosyası 'temizlenmiş_veri.xlsx' olarak oluşturuldu.

Process finished with exit code 0
```



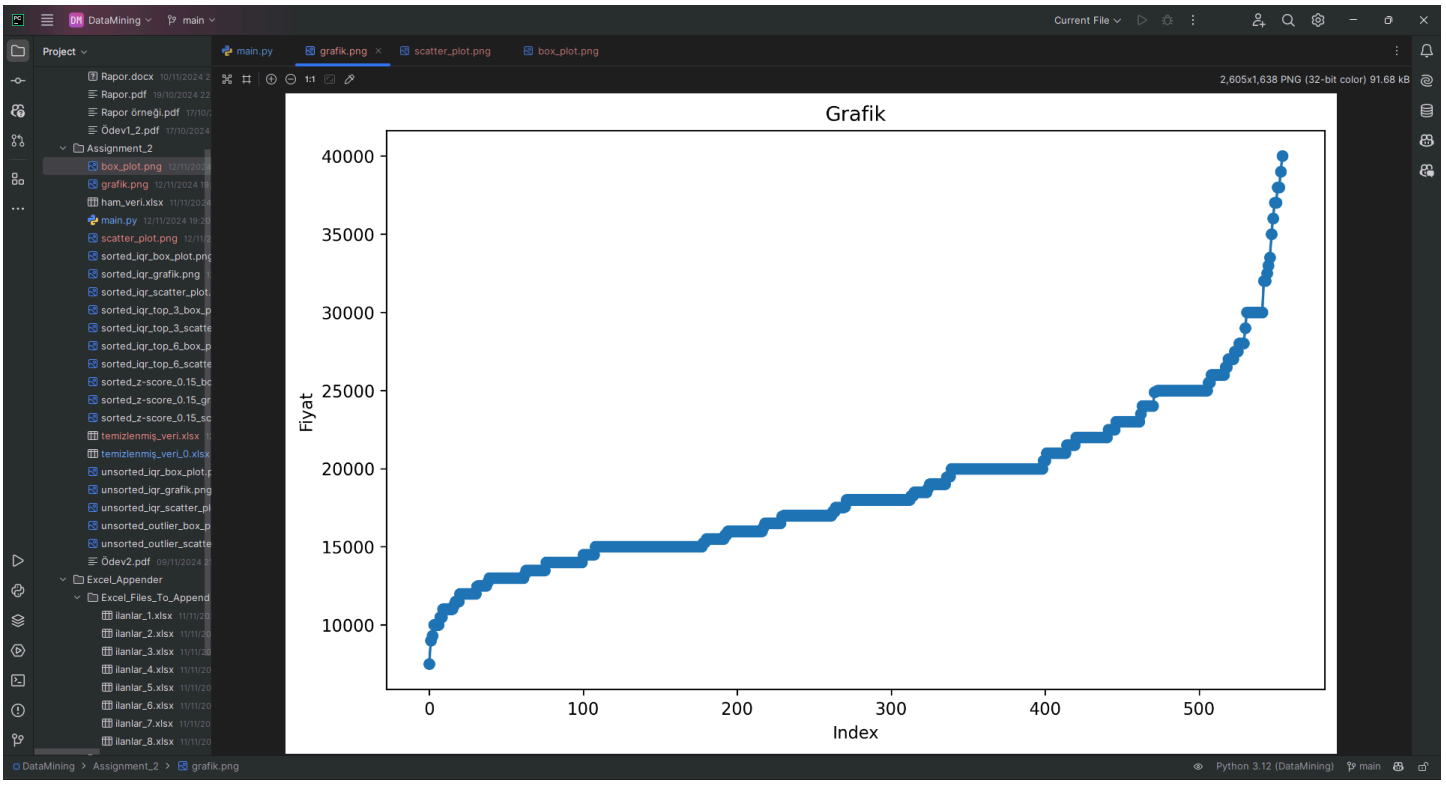
	C0	C1
1	Fiyat	Küme
2	7500	1
3	9000	1
4	9300	1
5	10000	1
6	10000	1
7	10000	1
8	10000	1
9	10500	1
10	10500	1
11	11000	1
12	11000	1
13	11000	1
14	11000	1
15	11000	1
16	11000	1
17	11000	1
18	11250	1
19	11500	1

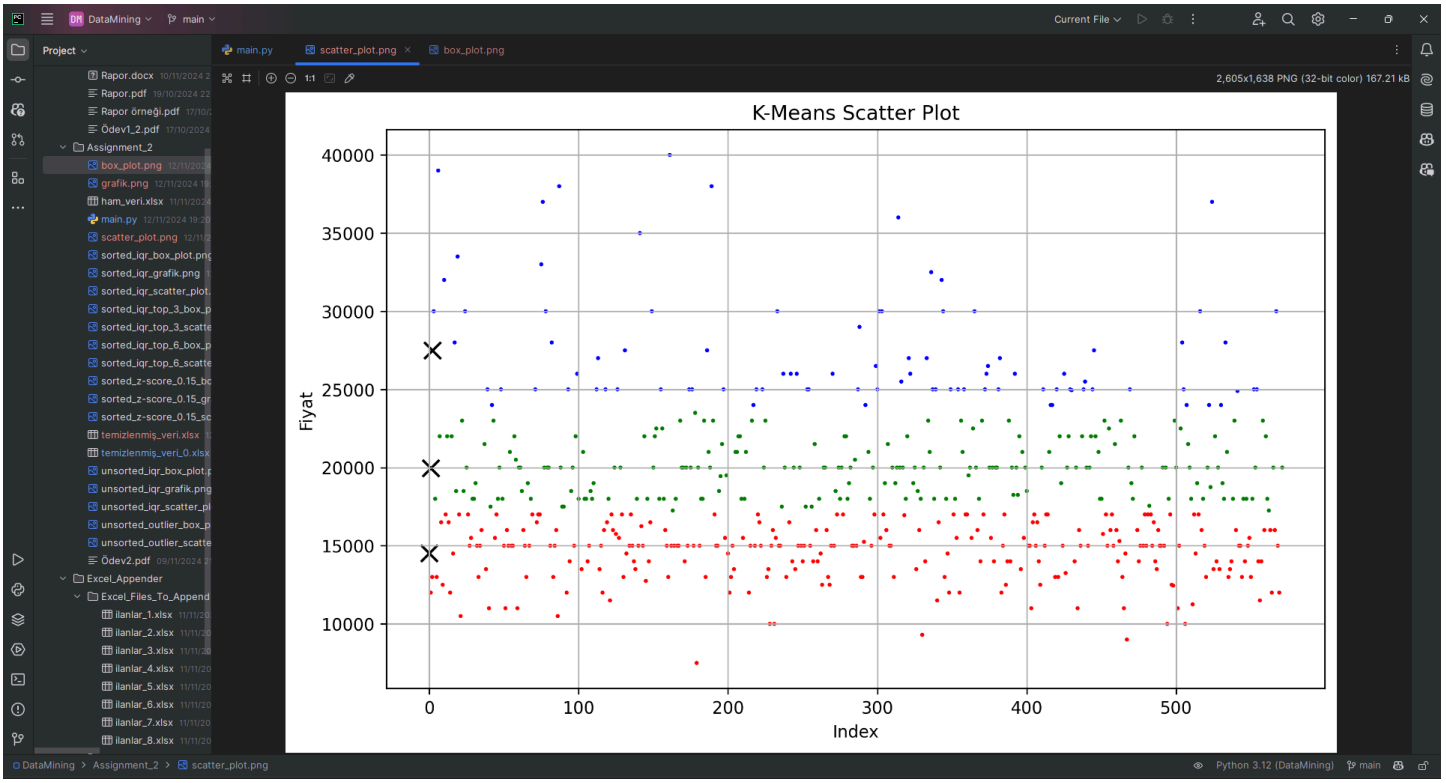


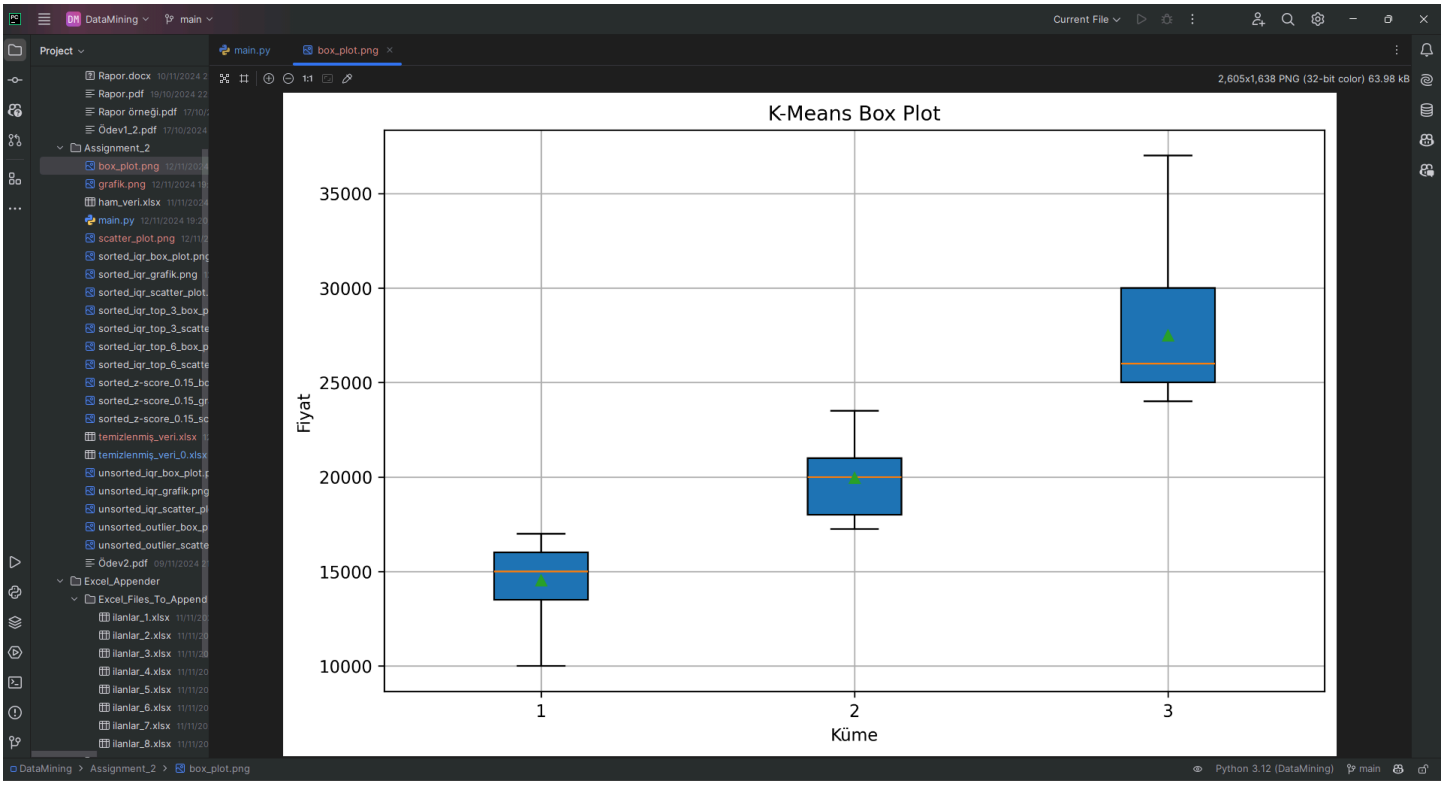
	C0	C1
461	23000	2
462	23000	2
463	23000	2
464	23500	2
465	24000	3
466	24000	3
467	24000	3
468	24000	3
469	24000	3
470	24000	3
471	24000	3
472	24000	3
473	24900	3
474	24950	3
475	25000	3
476	25000	3
477	25000	3
478	25000	3
479	25000	3
480	25000	3
481	25000	3

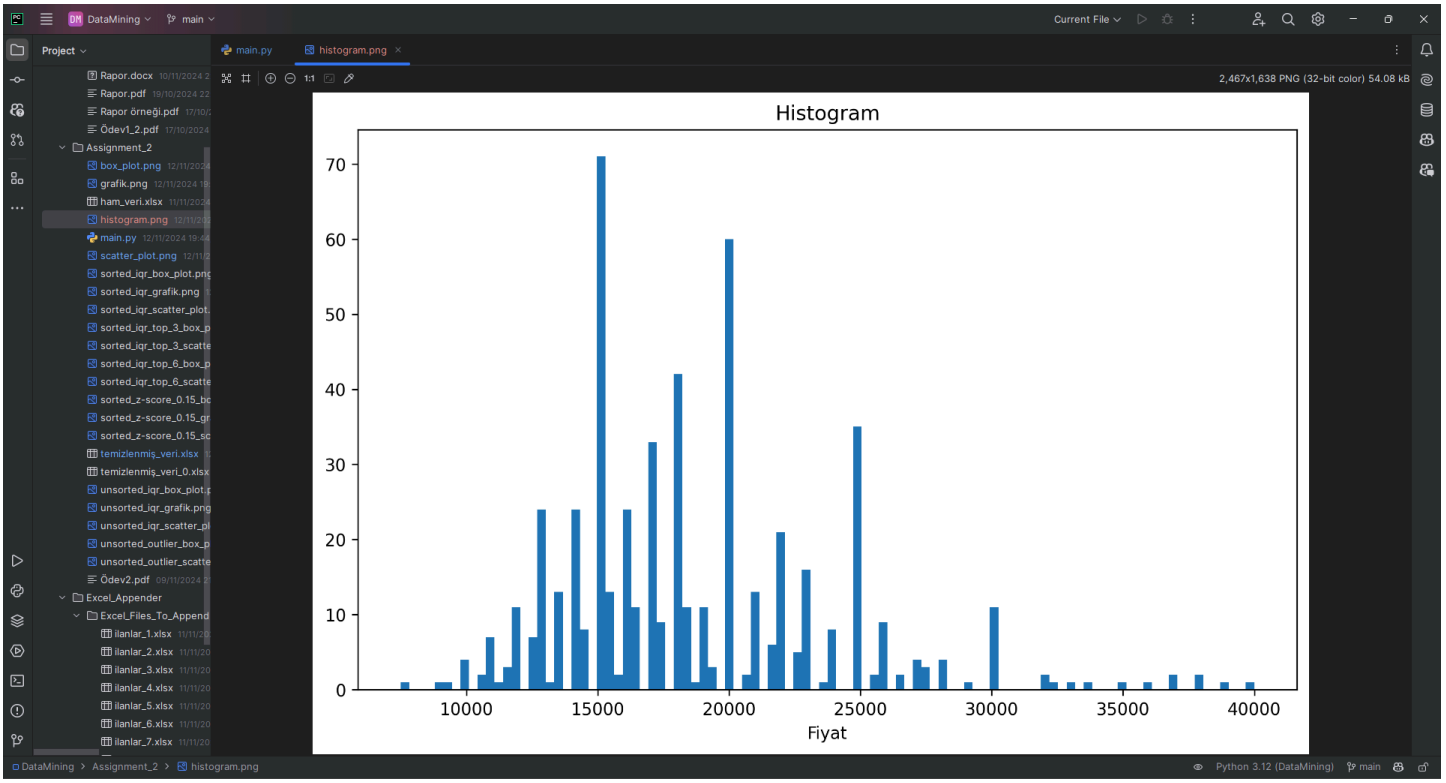


C0	C1	C2	C3	C4	C5	C6	C7	C8
BUCA SEYFİ DEMİRSOY HAST. ARKASI 1+1 KLİMALI MERKEZİ DAİRE	İzmir / Buca / Kozağaç Mah.	13.000	1 + 1	55 m²	10 Yaşında	1. Kat	04-11-2024	ht
GOLDİA YAPI'DAN KAMPÜS YAKINI 1+1 EŞYALI KİRALIK	İzmir / Buca / Atatürk Mah.	15.250	1 + 1	55 m²	4 Yaşında	2. Kat	21-10-2024	ht
İnkilap Mh. Adli Tıp Yanı 3+1 Doğalgazlı Kiralık Daire	İzmir / Buca / İnkılap Mah.	24.000	3 + 1	145 m²	15 Yaşında	Ara Kat	02-11-2024	ht
ultra lüks kiralık evler	İzmir / Buca / Adatepe Mah.	300	2 + 1	70 m²	2 Yaşında	3. Kat	07-11-2024	ht
Buca Fırat Mahallesinde Ara Kat Doğalgazlı 3+1 Kiralık Daire	İzmir / Buca / Fırat Mah.	21.000	3 + 1	110 m²	35 Yaşında	Ara Kat	30-10-2024	ht
MARKA'DAN ANACADDE&KAMPÜS YAKINI 2+1 EŞYALI&D.GAZLI KİRALIK DAİR	İzmir / Buca / Atatürk Mah.	18.000	2 + 1	75 m²	4 Yaşında	2. Kat	08-11-2024	ht
3 ODA 1 SALON ,YÜKSEK ZEMİN -120 M2	İzmir / Buca / İnönü Mah.	15.000	3 + 1	120 m²	26 Yaşında	Yüksek Giriş	10-11-2024	ht
BUCA YILDIZ MAH. EŞYALI 2+1 KAPALI MUTFAK , BAHÇELİ DAİRE	İzmir / Buca / Yıldız Mah.	20.000	2 + 1	110 m²	20 Yaşında	1. Kat	02-11-2024	ht
ERA NİVA'DAN KURUÇEŞME'DE KİRALIK EŞYALI 1+1 DAİRE	İzmir / Buca / Kuruçeşme Mah.	17.000	1 + 1	53 m²	5 Yaşında	2. Kat	01-11-2024	ht
Buca Çamlıkule Mah. Site İçerisinde 3+1 Kiralık Daire	İzmir / Buca / Çamlıkule Mah.	26.500	3 + 1	140 m²	6 Yaşında	5. Kat	01-10-2024	ht
MERKEZİ KONUMDA DOĞALGAZLI FULL EŞYALI 2+1 DAİRE	İzmir / Buca / Hürriyet Mah.	25.000	2 + 1	170 m²	16 Yaşında	1. Kat	03-11-2024	ht
ATAMER GAYRİMENKUL'DEN İÇİ FULL YAPILI 160 M2 DAİRE.	İzmir / Buca / İnkılap Mah.	30.000	3 + 1	175 m²	21 Yaşında	2. Kat	01-11-2024	ht
ÇAMLIPINAR MAH. 110 M2 ARAKAT DOĞALGAZLI 2+1 KİRALIK DAİRE	İzmir / Buca / Çamlıpınar Mah.	15.500	2 + 1	110 m²	26 Yaşında	2. Kat	30-10-2024	ht
TURYAP BUCA'DAN DUMLUPINAR MH'DE TARİHİ MÜSTAKİL RUM EVİ	İzmir / Buca / Dumlupınar Mah.	30.000	2 + 1	99 m²	100 Yaşın…		30-10-2024	ht
YAYLACIKTA EŞYALI KİRALIK DAİRE	İzmir / Buca / Yaylacık Mah.	17.000	1 + 1	55 m²	5 Yaşında	2. Kat	22-10-2024	ht
KADIN DOĞUM HASTANESİ KARŞISI 2+1 KİRALIK DAİRE	İzmir / Buca / Atatürk Mah.	19.000	2 + 1	85 m²	5 Yaşında	2. Kat	21-10-2024	ht
DOĞUŞ CAD. ÜZERİNDE KAMPÜS KARŞISI KİRALIK 1+1 EŞYALI REZİDANS	İzmir / Buca / Kuruçeşme Mah.	16.000	1 + 1	45 m²	4 Yaşında	Ara Kat	12-10-2024	ht
AY GAYRİMENKUL'DEN ŞİRİNYER İZBAN DİBİ 3+1 KİRALIK DAİRE!!	İzmir / Buca / Hürriyet Mah.	21.500	3 + 1	120 m²	11 Yaşında	3. Kat	27-09-2024	ht
BUCA ADA EMLAKTAN YAYLACIK MH. 3+1 SİTE İÇERSİNDE KİRALIK DAİRE	İzmir / Buca / Yaylacık Mah.	15.000	3 + 1	160 m²	21 Yaşında	7. Kat	31-10-2024	ht
BUCA YILDIZ MAHALLESİNDE 2+1 KİRALİK DAİRE	İzmir / Buca / Yıldız Mah.	15.000	2 + 1	70 m²	6 Yaşında	2. Kat	23-10-2024	ht
Şirinyer Nato Karşısı 3+1 140 M2 Doğalgazlı Arakat Kiralık Daire	İzmir / Buca / İnkılap Mah.	20.000	3 + 1	140 m²	17 Yaşında	2. Kat	12-10-2024	ht
Buca Göksu mahallesinde arakat kiralık 2+1 kiralık daire İZBAN ya…	İzmir / Buca / Göksu Mah.	13.000	2 + 1	100 m²	5 Yaşında	Ara Kat	03-11-2024	ht
ATAMER GAYRİMENKUL'DEN KAPALI MUTFAK 2+1 DAİRE.	İzmir / Buca / Kozağaç Mah.	19.000	2 + 1	130 m²	5 Yaşında	Yüksek Giriş	03-11-2024	ht
BUCA UFUK MAH. YIKIKKEMER MEY. YAKINI 2+1 & 80 m2 KİRALIK DAİRE	İzmir / Buca / Ufuk Mah.	20.000	2 + 1	80 m²	7 Yaşında	Giriş Katı	29-10-2024	ht
EFELER MAH. SİTE İÇERİSİNDE 3+1 EBEVEYN BANYOLU KİRALIK DAİRE	İzmir / Buca / Efeler Mah.	36.000	3 + 1	160 m²	4 Yaşında	2. Kat	05-11-2024	ht
BAŞOĞLUN'DAN EŞYALI DOĞALGAZLI 1+1 KİRALIK DAİRE	İzmir / Buca / Kuruçeşme Mah.	20.000	1 + 1	60 m²	5 Yaşında	1. Kat	14-10-2024	ht
YAYLACIK MAHALLESİ 2+1 AÇIK MUTFAK DOĞALGAZ ARAKAT EŞYALI	İzmir / Buca / Yaylacık Mah.	25.500	2 + 1	80 m²	5 Yaşında	3. Kat	03-11-2024	ht




K-Means Scatter Plot
Fiyat
Index
40000
35000
30000
25000
20000
15000
10000
0
100
200
300
400
500


K-Means Box Plot
Fiyat
Küme
35000
30000
25000
20000
15000
10000
1
2
3



4- Değerlendirme

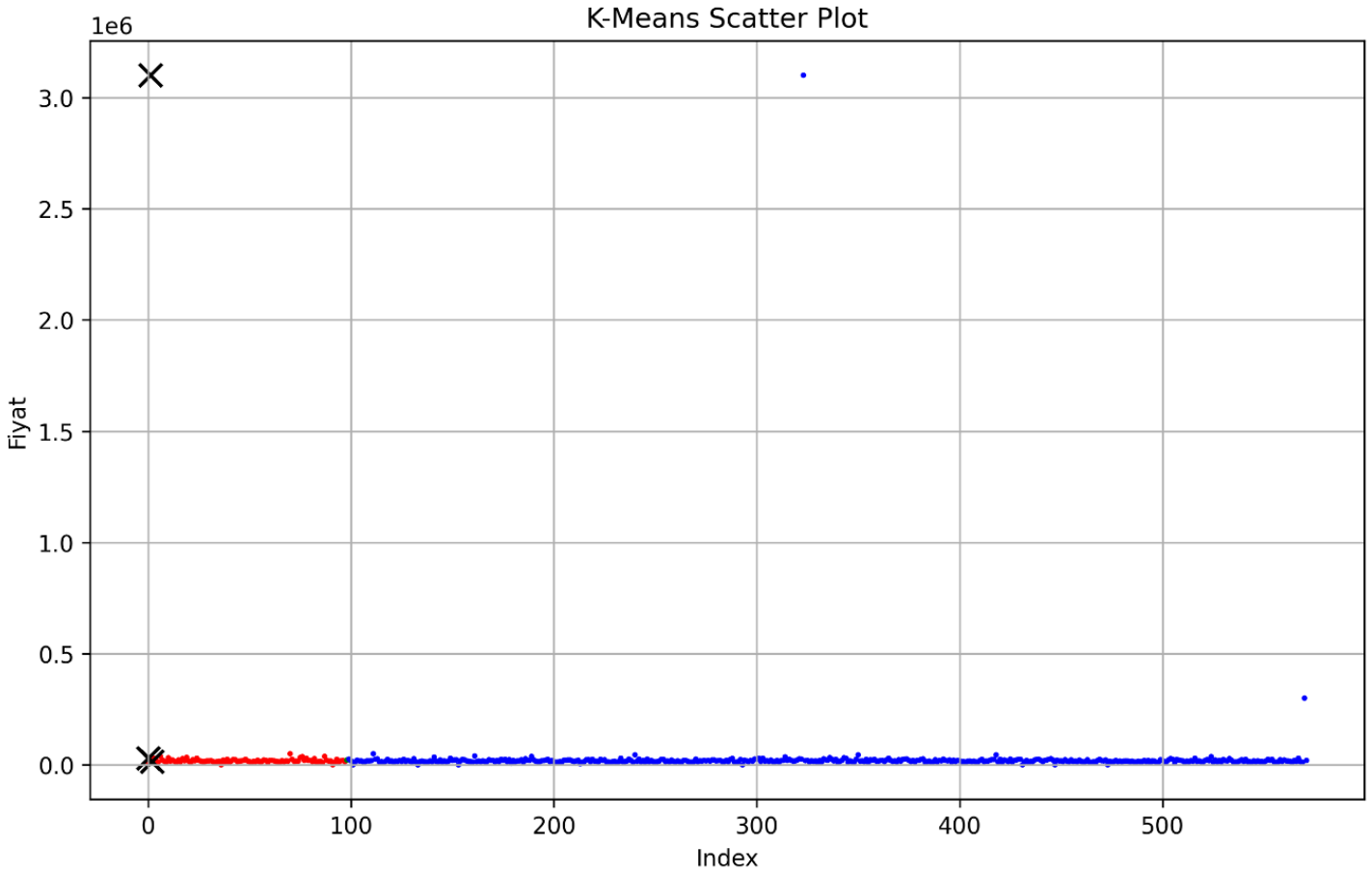
Projeyi yapmak için öncelikle elimde kayda değer miktarda veri olması gerekiyordu. Önceki ödevi az veri ile göndermiştim. Bu yüzden daha fazla veri çektim. Cloudflare doğrulaması her çıktığında modemimi açıp kapattım ve yeni bir ip adresine geçtim. Bu işlemi 8-9 defa yaptım. Elimde 573 ilan verisi mevcut olunca bıraktım.

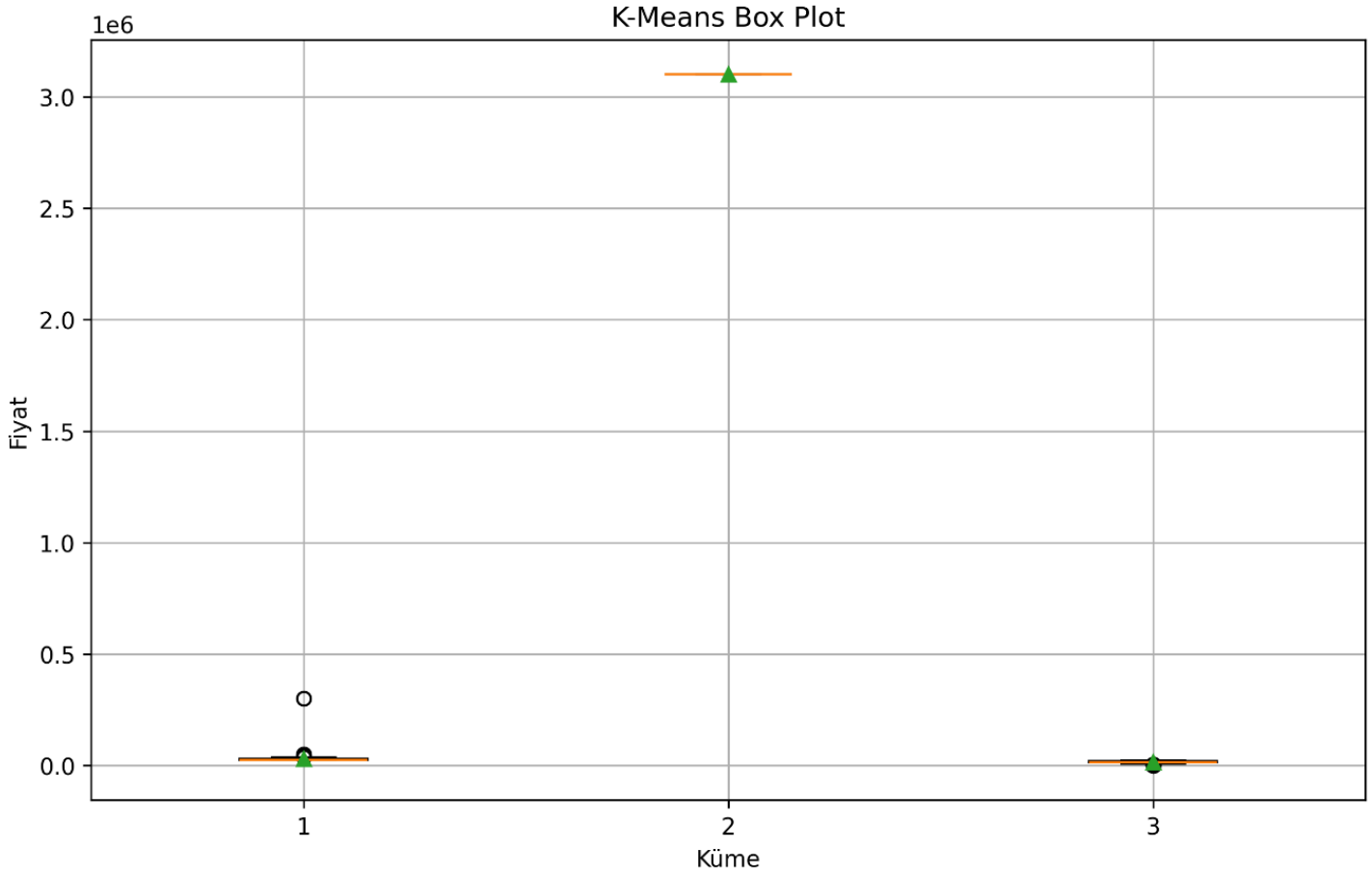
Ödevi yapmak için ilk önce yapay zekaya başvurdum ancak bu konuda fazlasıyla yetersizdi. Yeterli olduğu sohbetlerde ise ilerledikçe halüsinasyon görmeye başlayıp olmayan methodları ve attributeları kullanmaya başlıyordu. Kullandığım ide'deki copilot pek çok şeyi kolaylaştırmış olsa da o da çok iyi değildi. Bu yüzden kütüphane öğrenme sürecimi dökümantasyon, eğitim/tutorial siteleri ve forumlardan halletmeye çalıştım.

İlk olarak scikit-learn kütüphanesini kullandım ancak methodlar, parametreler ve genel olarak kullanımı hiç alışık olmadığım bir yoldaydı. Kendimi kaybolmuş ve seviyemin çok üstünde araçlar kullanıyormuş gibi hissettim. Bunun en büyük sebebi okul dışı zamanımın neredeyse tamamını oyun geliştirme, nesneye yönelik programlama ve c# ile geçirmem. Farklı kodlama tarzı, farklı programlama dili ve farklı konseptler ile uğraşıyordum. Bu yüzden slaytlarda yer alan k-means algoritmasını kendim yaptım ve dış kütüphanelere olabildiğince az bağlı olmaya çalıştım.

Pandas her ne kadar kolay kullanımı hedefleyen bir kütüphane olsa da veri tiplerinin belirsizliğinden doğan cast hataları, parametre hataları, operatörlerin geçersizliği gibi şeyler bana çok engel oldu. Python compile edilen bir dil olmadığı için bunların hepsi runtime'da belli oldu ve bu durum beni daha da yavaşlattı. Yine aynı sebepten ötürü kullandığım ide'nin yardımı çok az dokundu.

İlk çalışan verisyonumda verileri düzgün temizleyemediğimi fark ettim. Emlakçılar bazı ilanları satılık yerine kiralık olarak eklemiş ve bu ilanlar scrape yaparken veri setime girmiş. Kira bedeli 3 milyon olan bir ilan algortitmayı inanılmaz şekilde bozdu ve bu da böyle bir grafik ortaya çıkardı:

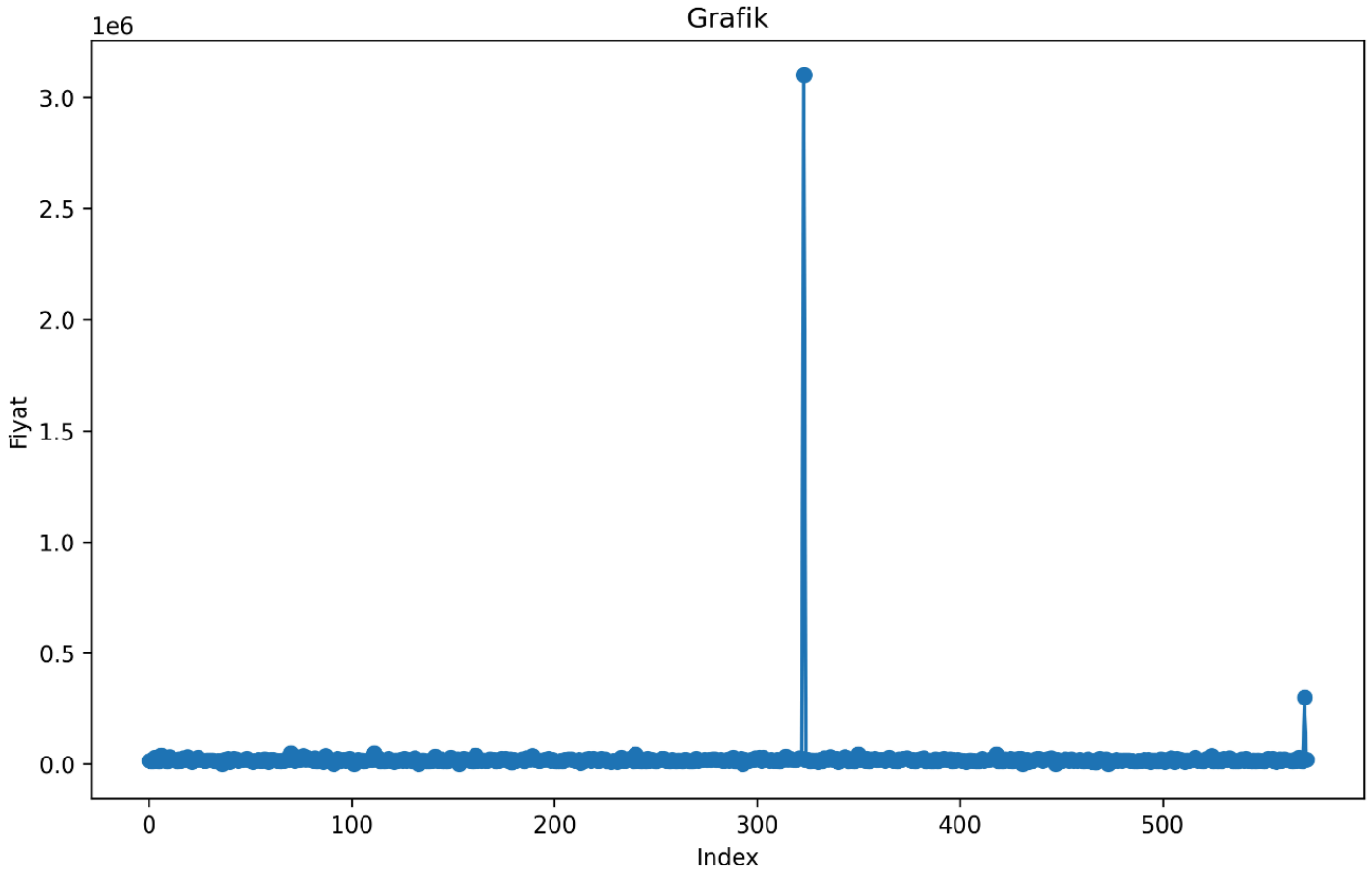




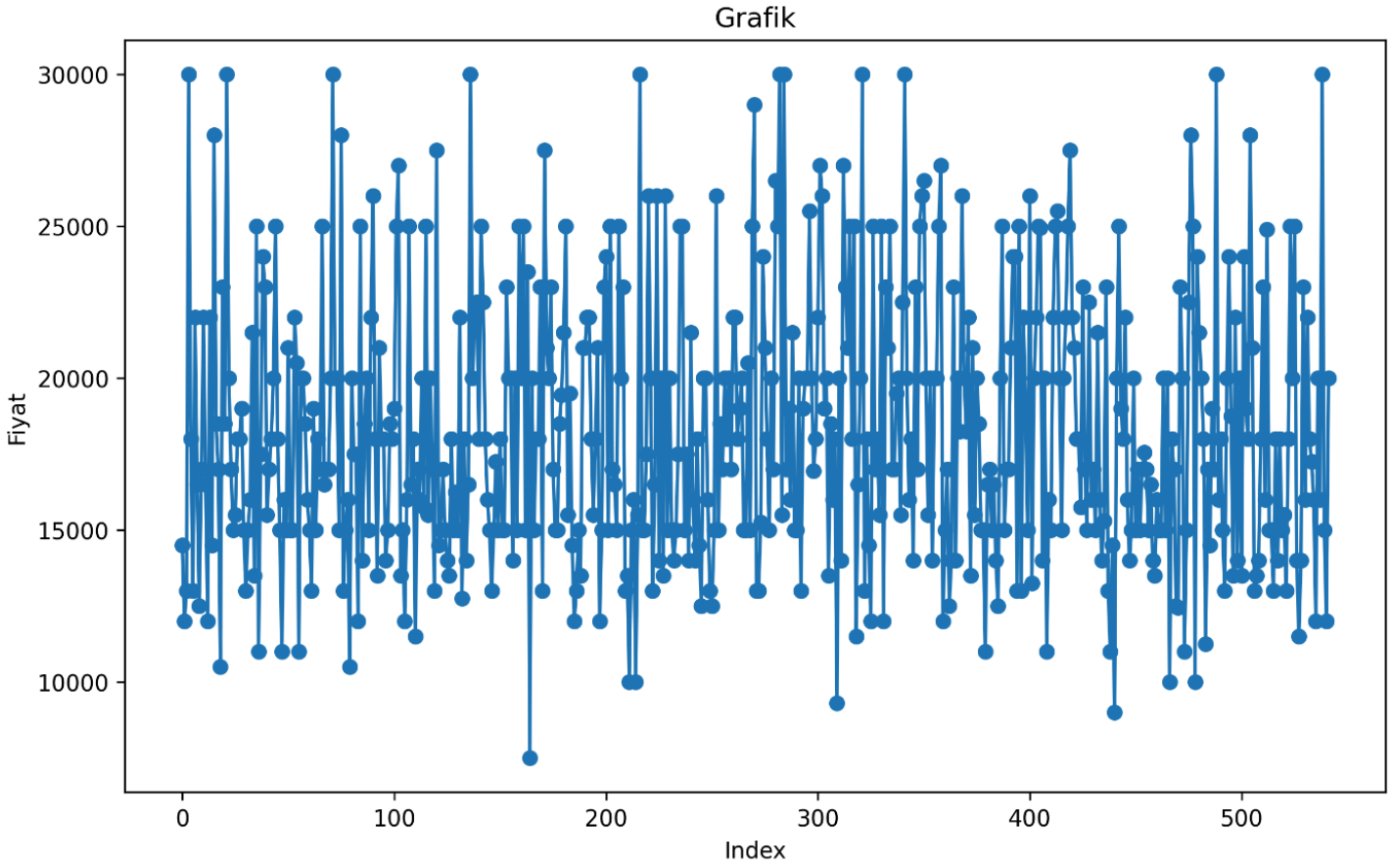
Daha sonra fiyat tablosuna biraz daha detaylı baktığımda kiralık daire ilanlarında günlük kiralık dairelerin de bulunduğunu ve bu ilanlarda fiyat olarak da günlük fiyat yazdığını fark ettim. Yani kira bedeli 200 TL, 300 TL olan pek çok ilan vardı. Bu gürültülerden kurtulmak gerekiyordu.

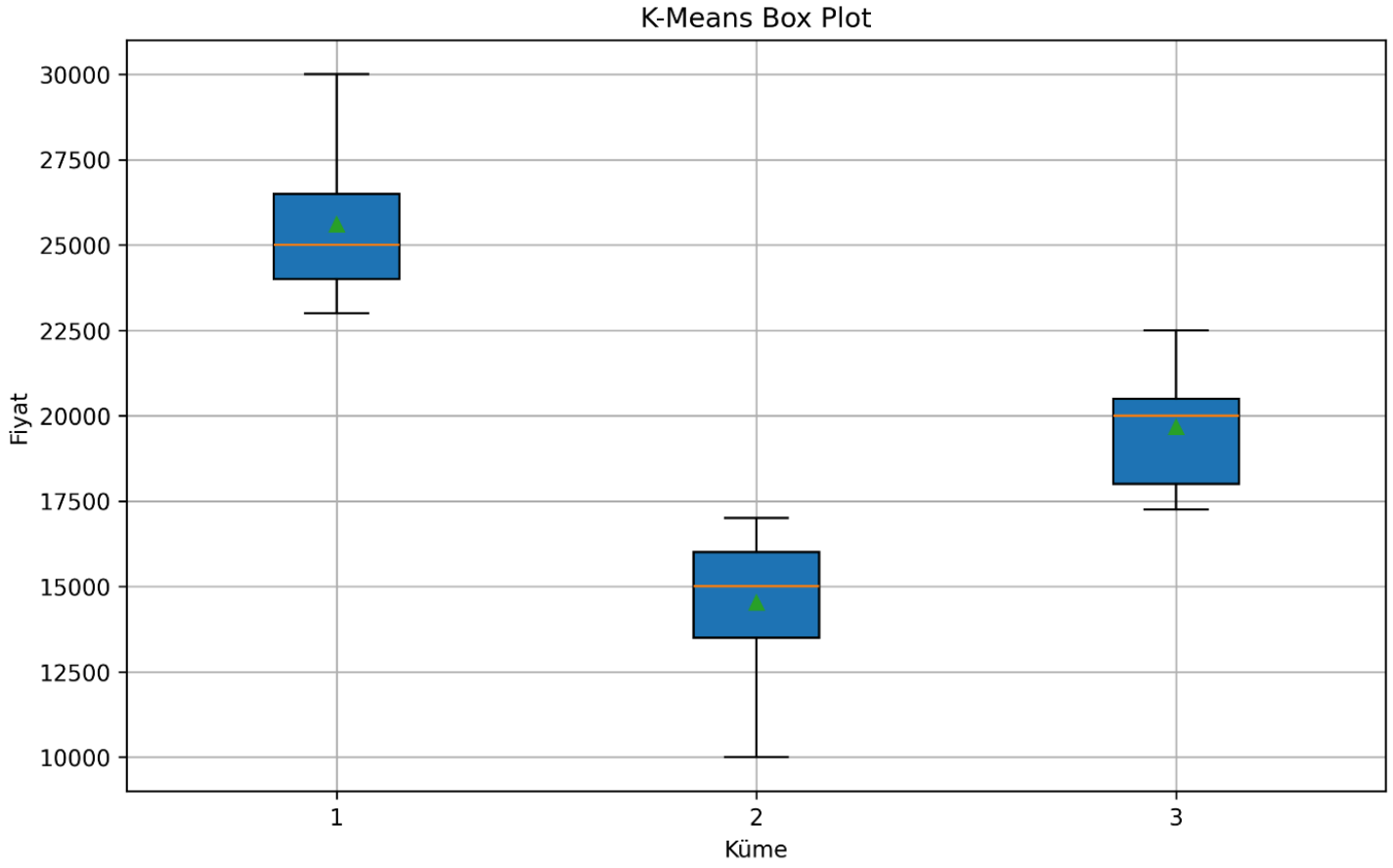
C1	
Fiyat	
200	
200	
200	
300	
300	
350	
400	
400	
400	
5000	
7500	
9000	
9300	
10000	
10000	
10000	
10000	
10500	

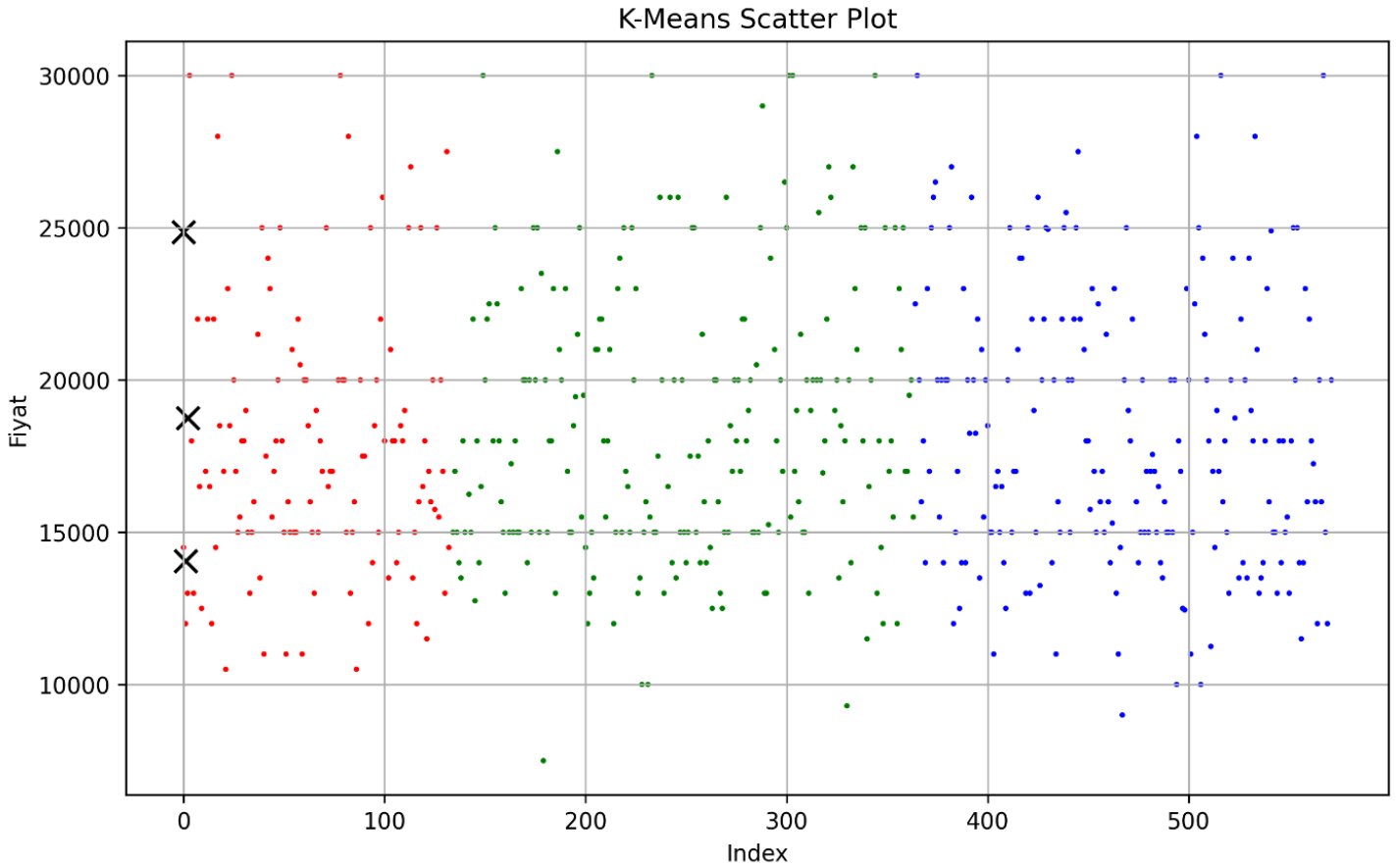
Gürültülerden kurtulmak için 2 yol kullanacaktım. Bunlardan birisi IQR yöntemi diğeri ise Z-Score yöntemi. IQR yöntemi eğilimli dağılım gösteren veriler, Z-Score ise normale yakın dağılım gösteren veriler için daha uygun. Verilerimin dağılımını gözlemlemek için grafik çizdim.



Bu grafik bana hiçbir bilgi vermedi. O yüzden iki yöntemi de denemeye karar verdim. Önce IQR ile başladım ve çarpanı genel olarak kullanılan 1.5 olarak belirledim. Verimi temizledim ve ardından bu grafikleri oluşturdum.

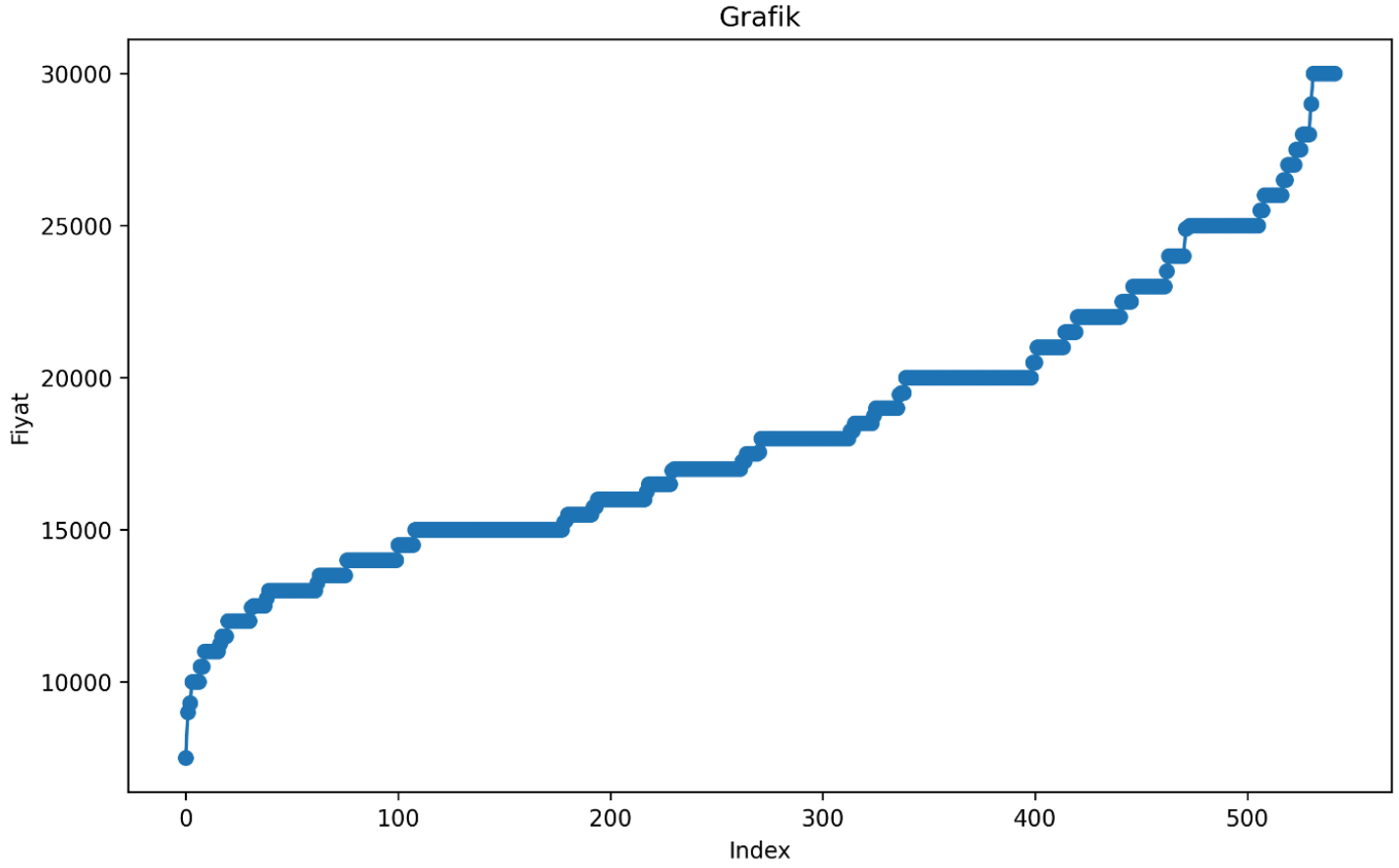

Grafik
Fiyat
Index
30000
25000
20000
15000
10000
0
100
200
300
400
500

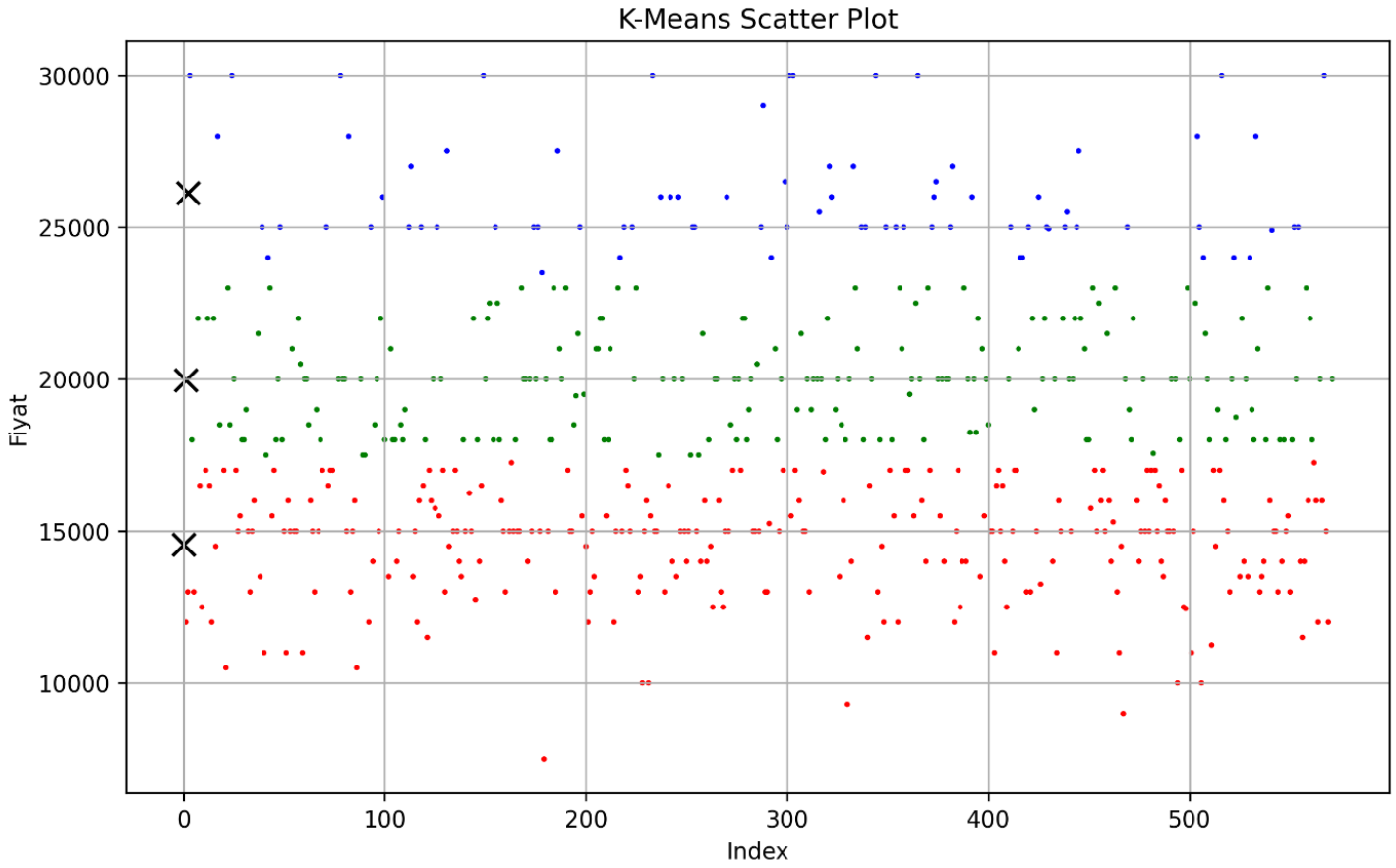

K-Means Box Plot
Fiyat
Küme
30000
27500
25000
22500
20000
17500
15000
12500
10000
1
2
3

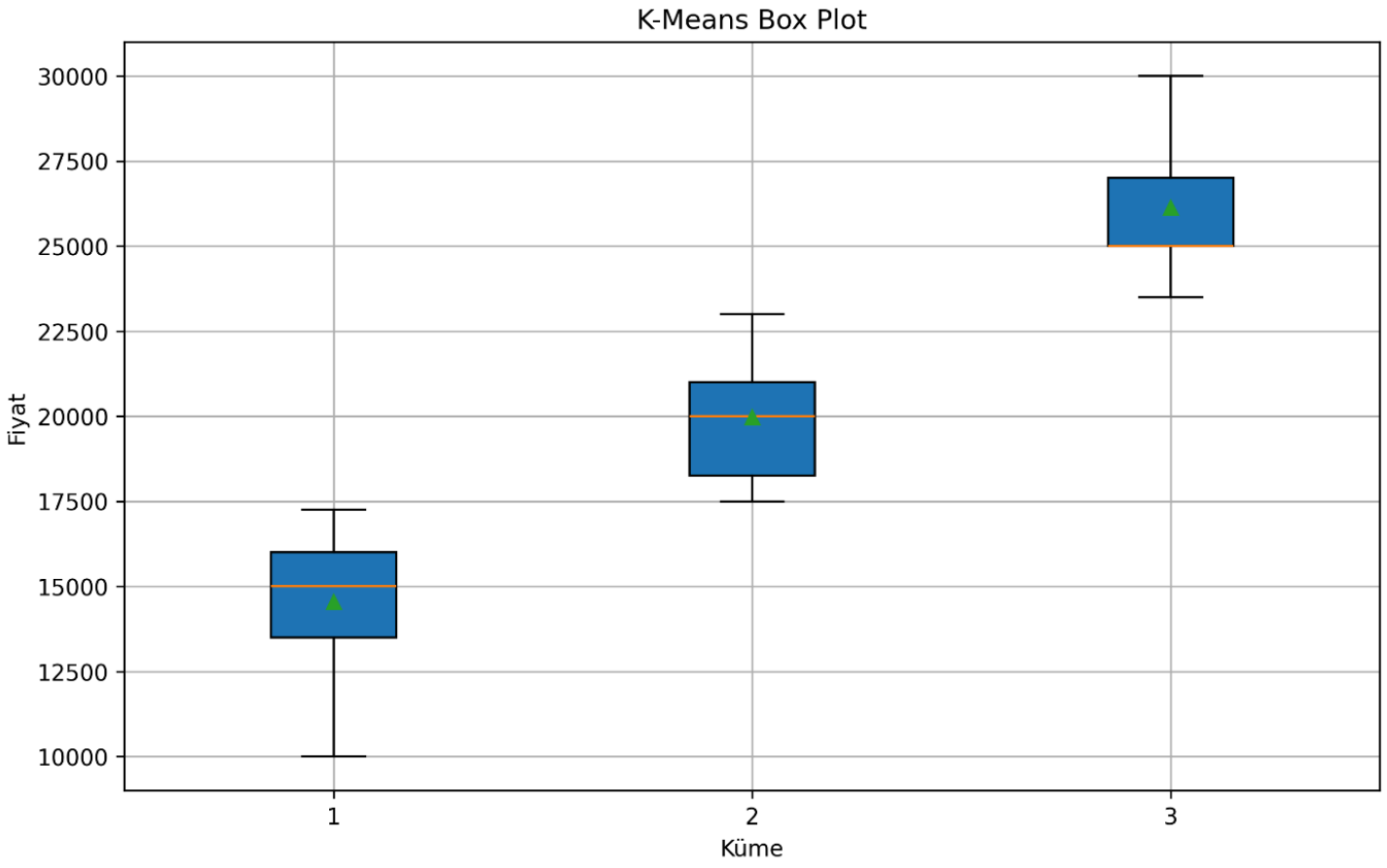


İlk grafiğin inişleri ve çıkışlarının x ekseninin zaman belirtmediği için anlamsız olduğunu fark ettim. Veriler sıralı olsaydı grafiğimiz çok daha düzgün görünürdü. Ayrıca scatter grafiğine bakınca fark ettim ki kümeleme algoritması indexlerin sıralı olmamasından ötürü kötü etkilenmiş.

Verileri sıraladım ve tekrar grafik çıktıları aldım.





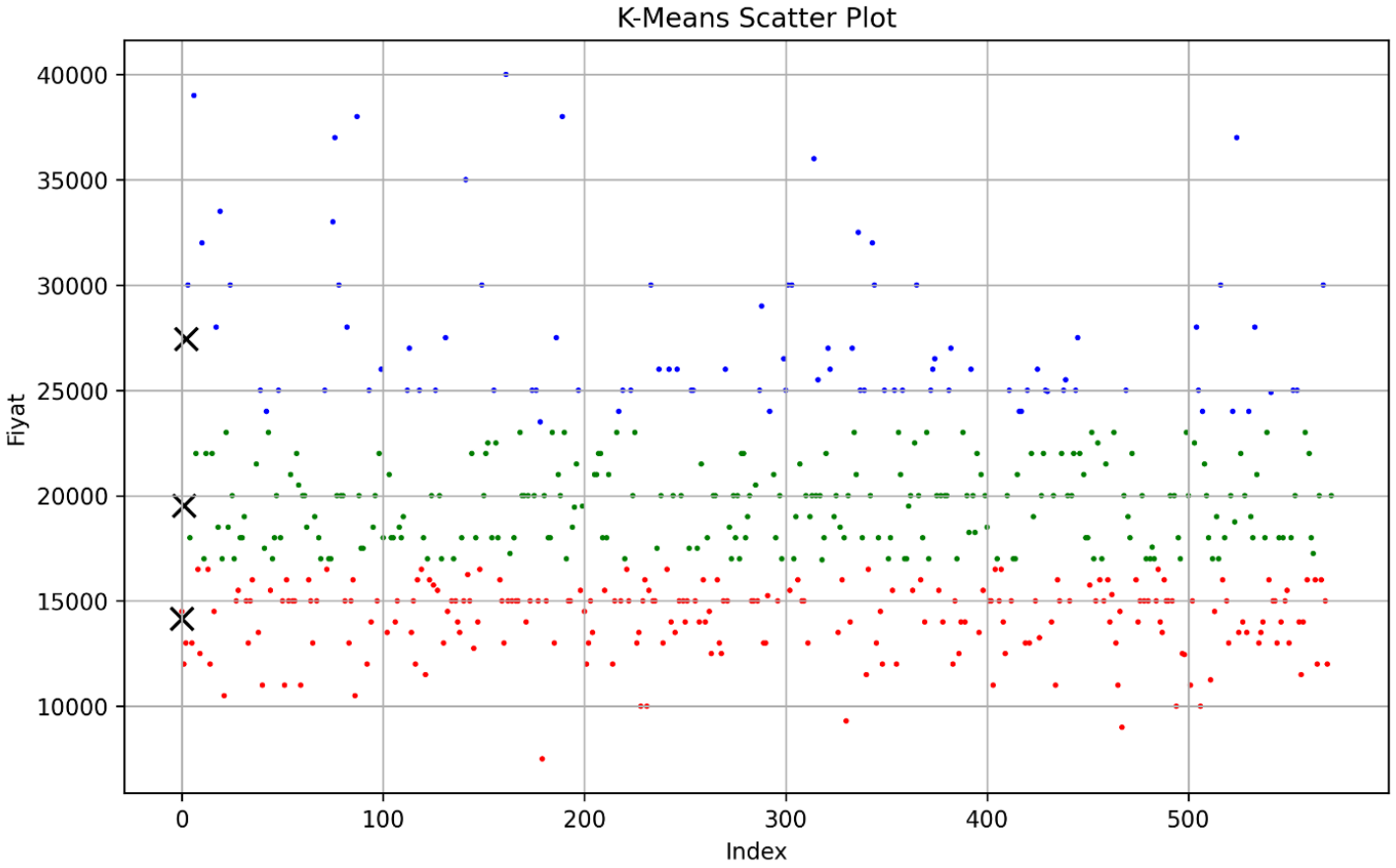


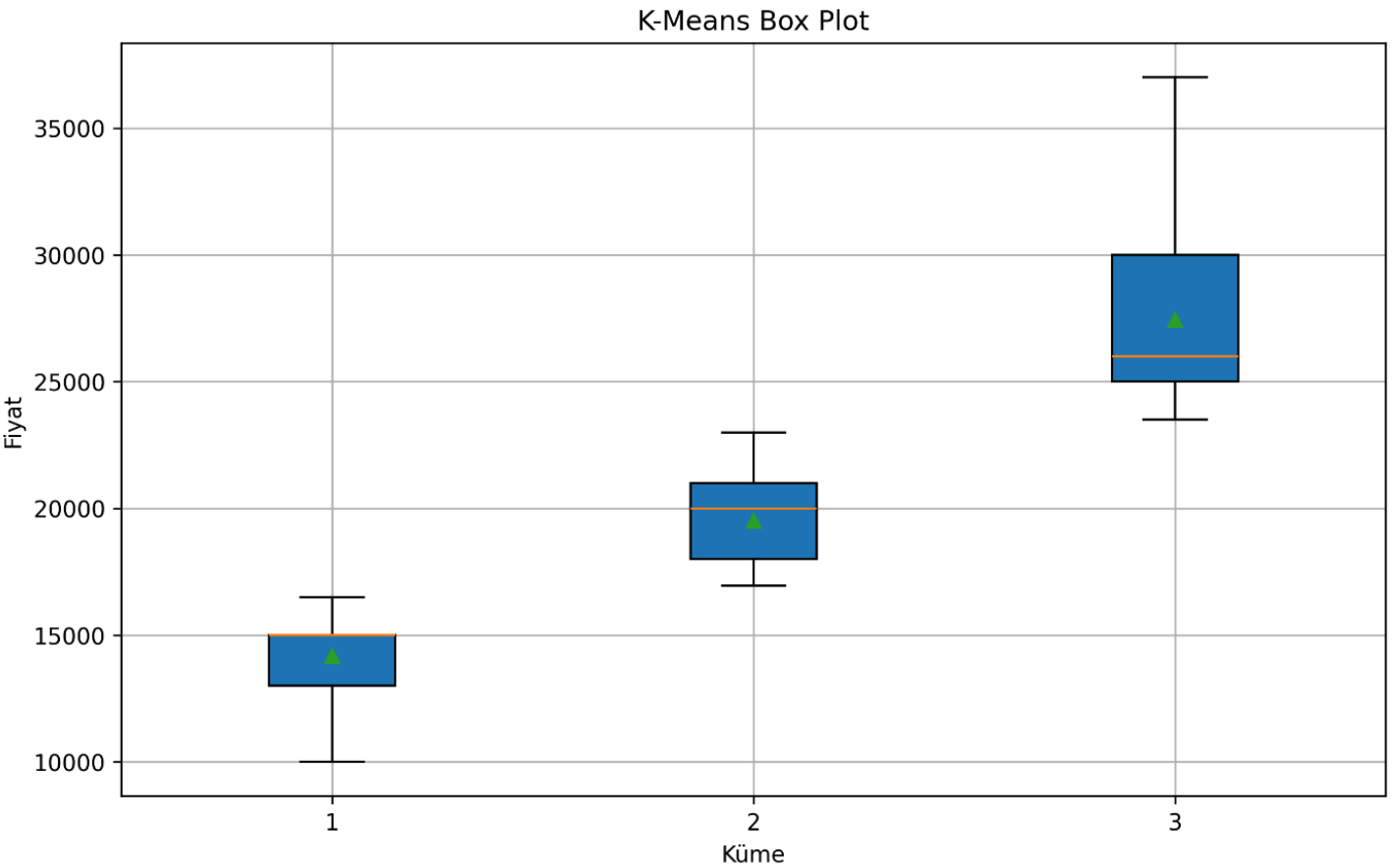
Outlierları grafiklerde belirtmem ekstrem değerlerden ötürü her şeyi inanılmaz küçültüyordu ve grafikleri bozuyordu. Bu yüzden outlierları grafikte göstermek yerine yazdırmaya karar verdim:

```
Outlierlar (IQR yöntemi):
       Fiyat
36       200
133      200
153      200
431      300
293      300
101      350
447      400
473      400
91       400
213     5000
343    32000
314    36000
76     37000
524    37000
87     38000
189    38000
6      39000
161    40000
418    45000
240    45000
350    45000
70     50000
111    50000
570   300000
323  3100000
```

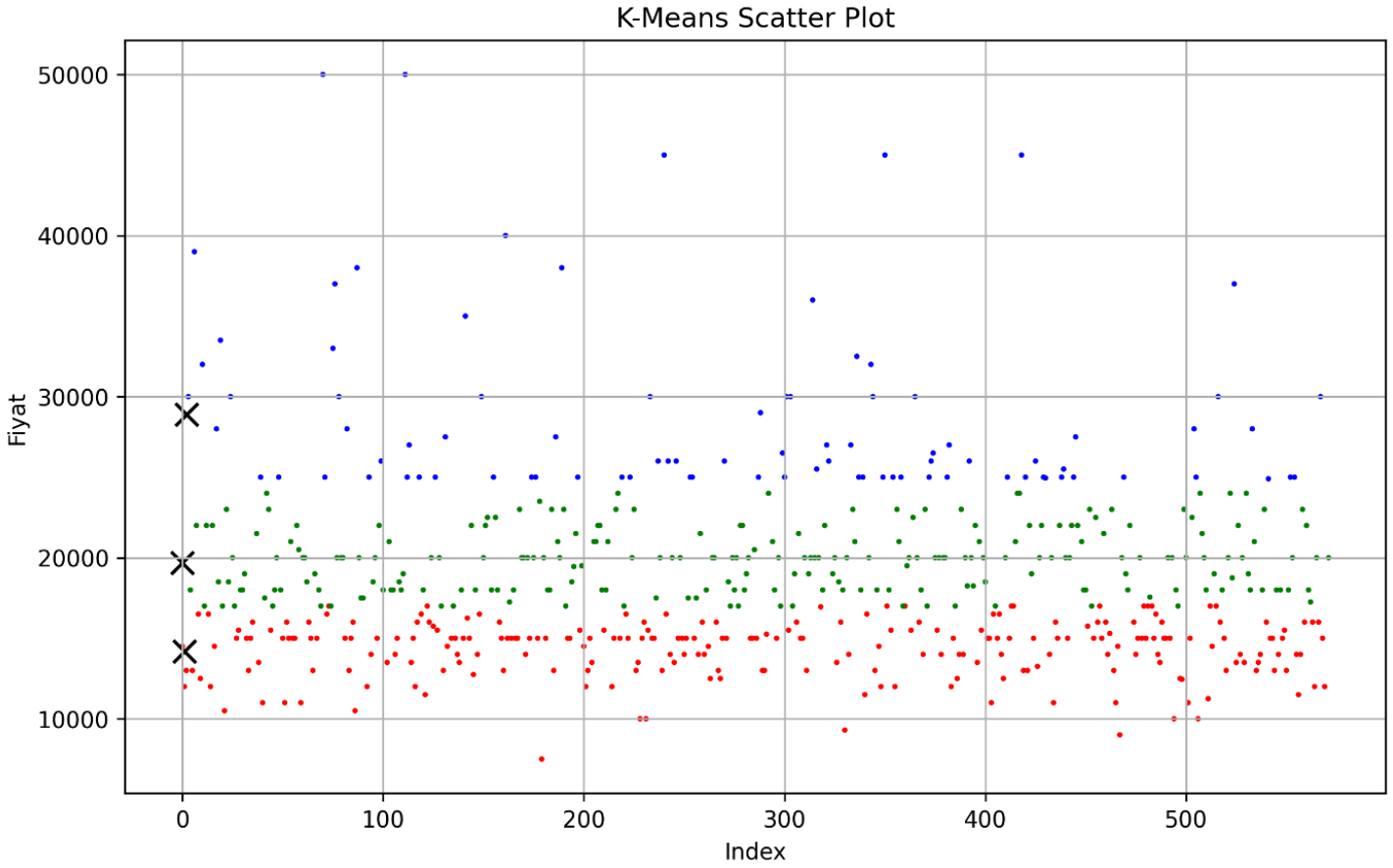
Soldaki sayılar indexleri sağdakiler ise fiyatı temsil ediyor. Buraya hepsini sığdıramadım ancak genel olarak şunu fark ettim: IQR yöntemi düşük değerleri iyi filtrelese de yüksek değerleri olması gerektiğinden daha fazla filtreliyordu. Günlük kiralık daireler ayrışmıştı ancak yüksek kiralı normal ilanlar da onlarla birlikte gitmişti. 32 bin ve 50 bin arasındaki değerler mantıklı değerler. Bu yüzden üst tabanın katsayısını 1.5'dan 3'e çıkardım. Outlierlar aşağıdaki gibi oldu ve yeni grafikleri oluşturdum.

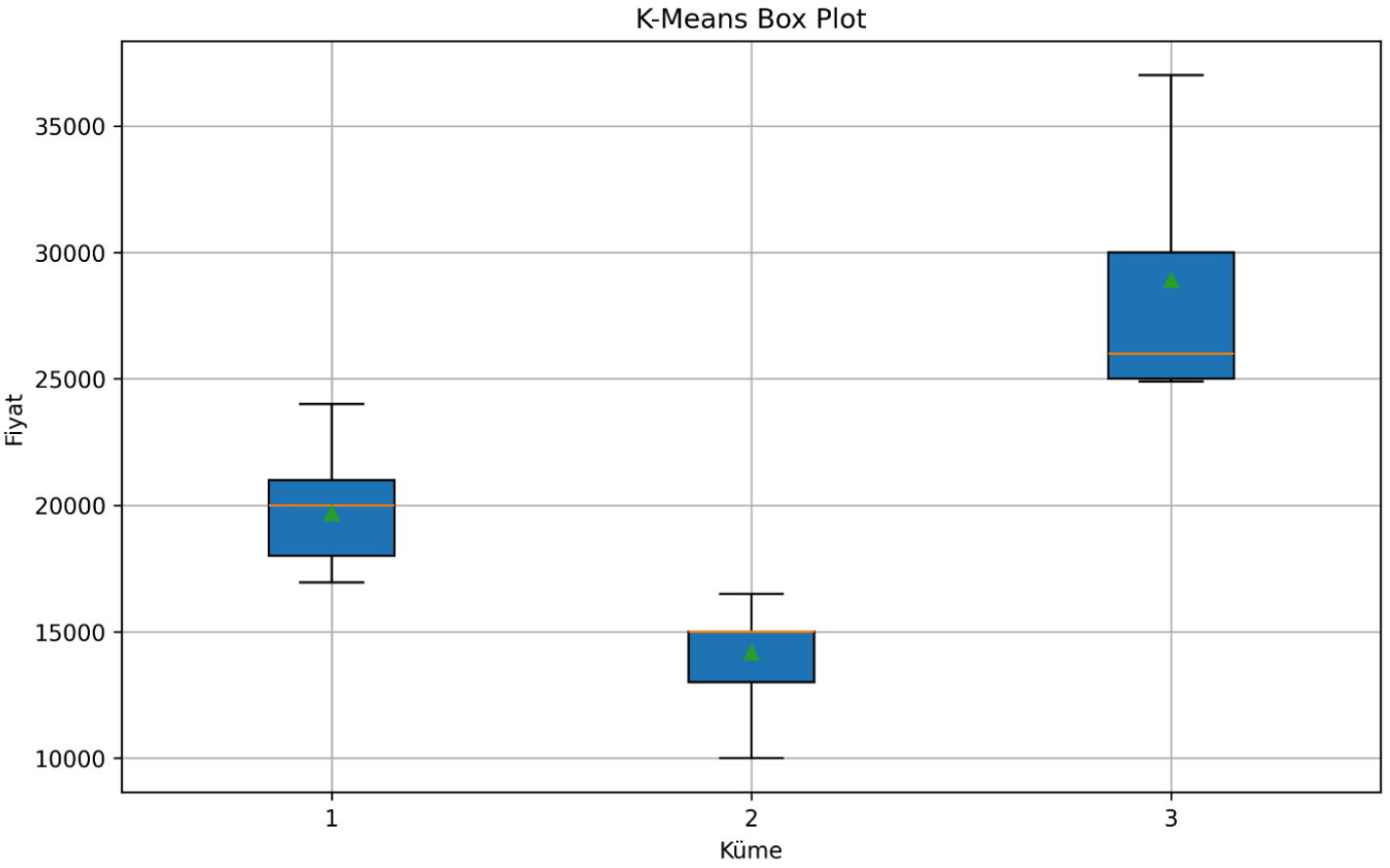
```
       Fiyat
36       200
133      200
153      200
431      300
293      300
101      350
447      400
473      400
91       400
213     5000
418    45000
240    45000
350    45000
70     50000
111    50000
570   300000
323  3100000
```





Outlier olarak seçilen 45 ve 50 bin liralık ilanları da veri setime katmak istedim ve bu yüzden üst tabanın katsayısını 6 yaparak bir daha grafikler oluşturdum.



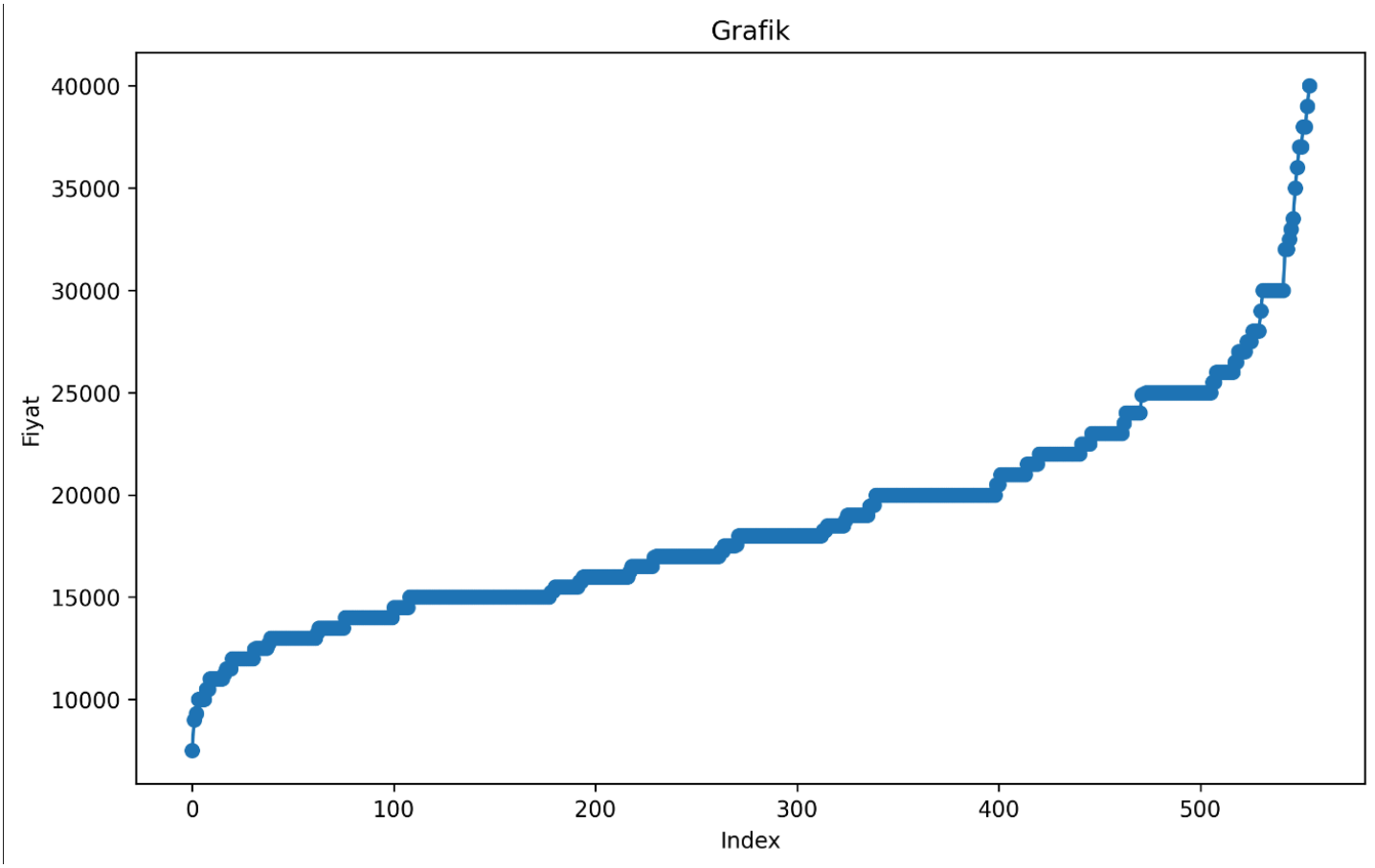


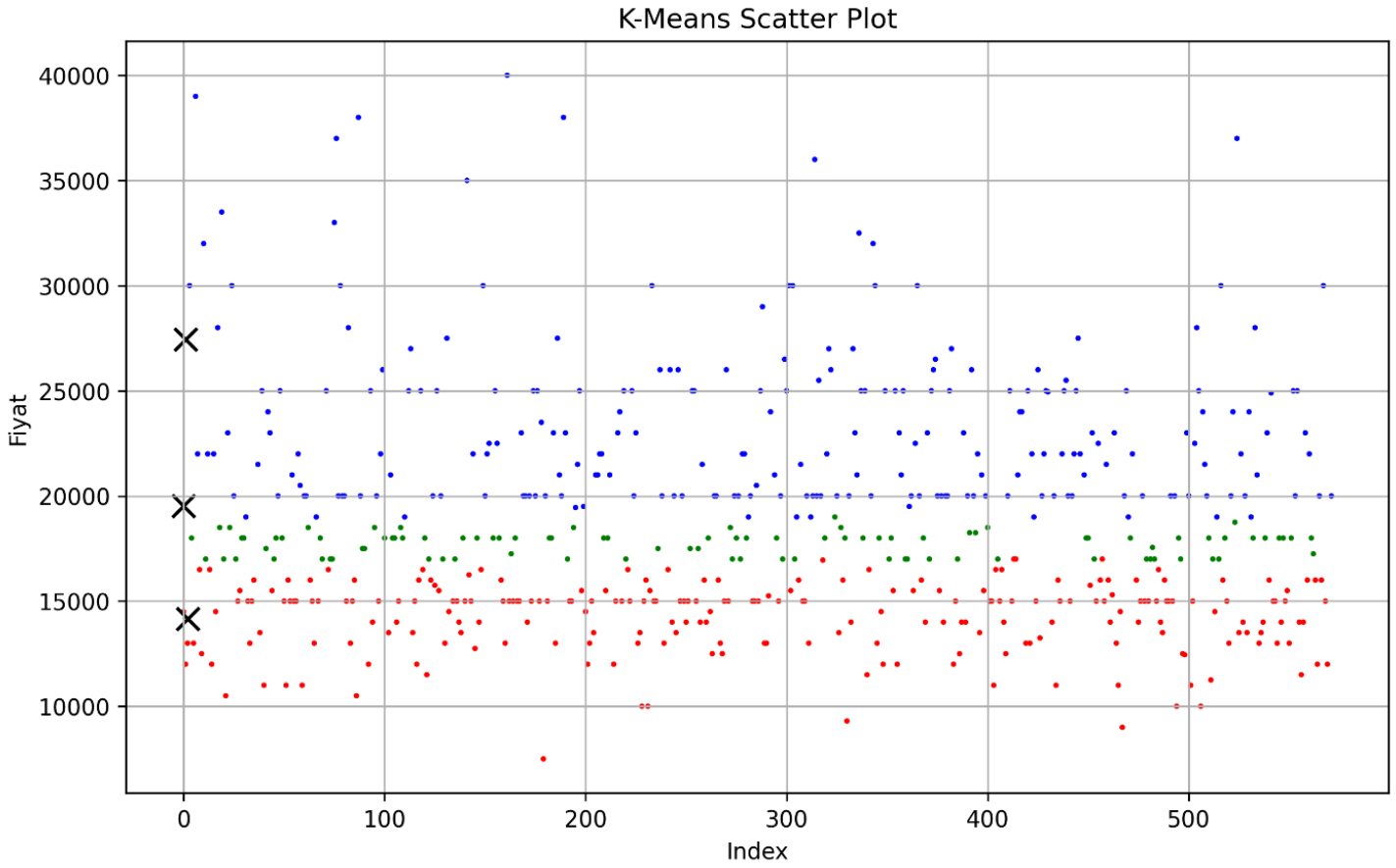
Genel dağılımı biraz bozdukları için günün sonunda üst değer katsayısını 3 yapmaya karar verdim.

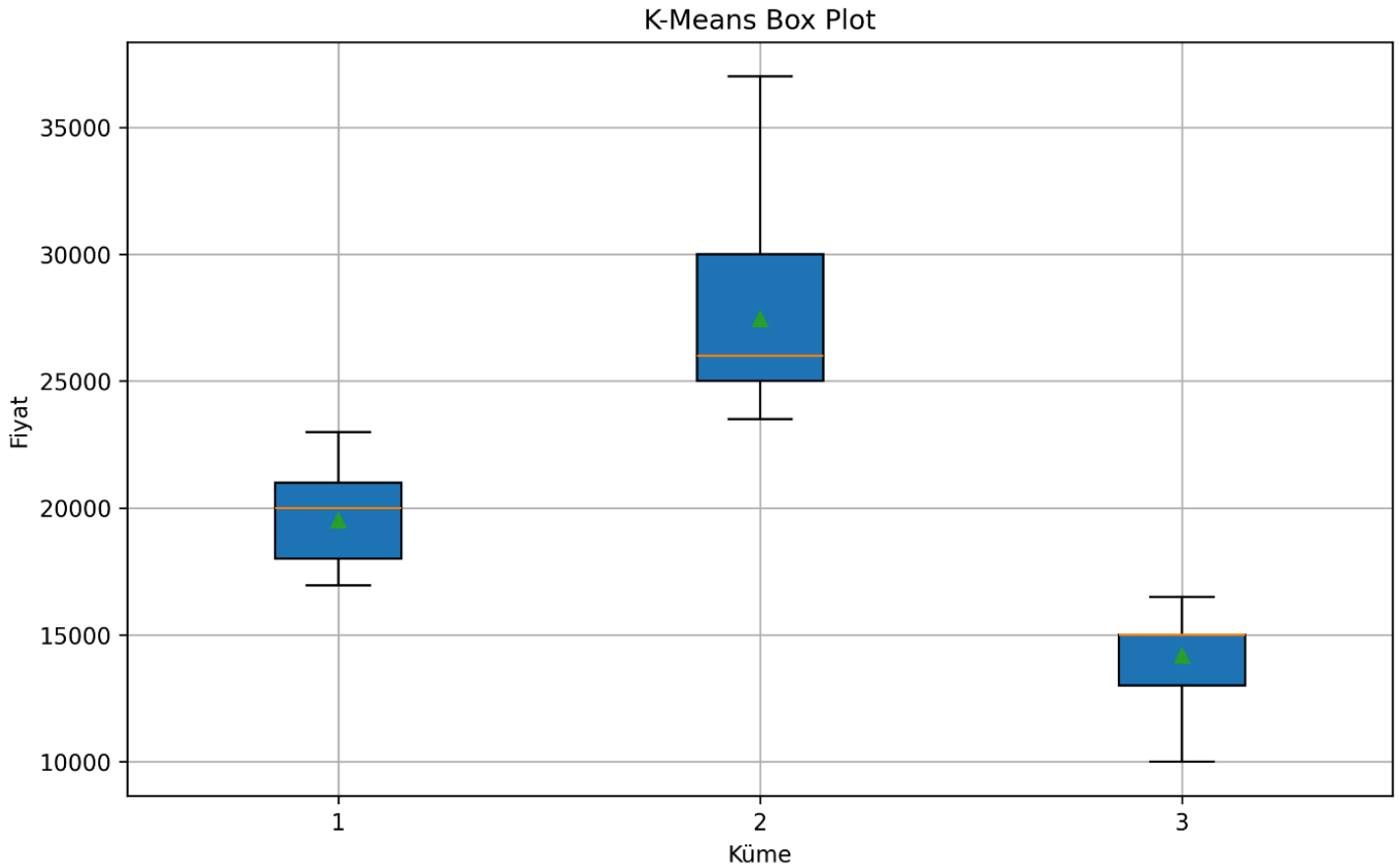
Sıralı grafikte verimin dağılımının normal olmadığını fark ettiğim için uygulamama z-score eklemedim ancak yine de deneyip görmek istedim o yüzden buraya ekliyorum:

```
# Fiyat sütununu z-score'a çevir
df["Z-Score"] = (df["Fiyat"] - df["Fiyat"].mean()) / df["Fiyat"].std()

z_score_modifier = 0.15

# Z-Score değeri z_score_modifier'dan büyük veya eksi z_score_modifier'dan
küçük olan satırları seç
outliers = df[(df["Z-Score"] > z_score_modifier) | (df["Z-Score"] <
-z_score_modifier)]
print("\nOutlierlar (Z-Score yöntemi):")
print(outliers[["Fiyat"]])

# Outlierları kaldır
df = df[(df["Z-Score"] <= z_score_modifier) & (df["Z-Score"] >=
-z_score_modifier)]
```

z_score_modifier 0.15 olduğunda alt değer katsayısı 1.5 ve üst değer katsayısı 3 olan IQR yöntemi ile hemen hemen aynı değerler outlier olarak tespit edildi ve benzer grafikler elde ettim.







Farklı yol da aynı durumu ortaya çıkarınca uygun analizi yaptığımı düşünüp daha fazla deneme yapmayı bıraktım. Elbette random başlangıçtan ötürü her denememde kümeler biraz değişiyordu, bunu da birkaç deneme yapıp daha dengeli dağılım gözlemlediğim halinde durarak hallettim.

Benim için öğretici ve uğraşması keyifli bir ödevdi. İlk başta ham verileri elde etmek için modemimle defalarca uğraşmak biraz sinir bozucuydu ancak farklı yolları deneyip farklı grafikler yazdırmak, bunları değerlendirmek, parametrelerle uğraşıp yeni grafikler elde etmek güzeldi. Ayrıca k-means algoritmamı kendim yazdığım için de sevinçliyim. Kütüphaneler karmaşık syntax ve inanılmaz fazla sayıdaki parametreleri ile gözümü çok korkutmuştu.

5- Kaynaklar

Chat GPT
Ders Slaytları
https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html
https://www.w3schools.com/python/python_ml_k-means.asp
https://matplotlib.org/stable/index.html#matplotlib-release-documentation
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html
https://www.w3schools.com/python/matplotlib_scatter.asp
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html
https://www.geeksforgeeks.org/box-plot-in-python-using-matplotlib/
https://regexr.com/
https://medium.com/analytics-vidhya/removing-outliers-understanding-how-and-what-behind-the-magic-18a78ab480ff
https://medium.com/@datasciencejourney100_83560/z-score-to-identify-and-remove-outliers-c17382a4a739