



T.C.

**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
FEN FAKÜLTESİ
İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ BÖLÜMÜ**

2023-2024 AKADEMİK YILI BAHAR DÖNEMİ

**BİLGİSAYAR PROGRAMLAMA DERSİ
ÖDEV RAPORU**

**SPOTIFY VE YOUTUBE PLATFORMLARINDAKİ
TÜRKİYEDE EN ÇOK İZLENEN/DİNLENEN İLK 50 İÇERİĞİ
PYTHON DİLİNDE GÖRSELLEŞTİRİLMESİ VE ANALİZLERİ**

**Seyfettin Budak
BİLECİK 2024**

1 İçindekiler

1	<i>İçindekiler</i>	1
	<i>KONU</i>	2
	<i>ANALİZLER İÇİN GEREKLİ ADIMLAR VERİLERİN KULLANIMI</i>	2
	<i>YILLARA GÖRE SPOTIFY VERİ ANALİZLERİ</i>	4
	<i>YILLARA GÖRE YOUTUBE VERİ ANALİZLERİ</i>	11
	<i>KAYNAKÇA VE EKLER</i>	21

KONU

Spotify ve Youtube sosyal medya platformlarının Türkiye'de yıllara göre (2020-2021-2022-2023) en çok izlenen/dinlenen ilk 50 içeriği ile ilgili Python dilinde görselleştirme işlemi ve analizi.

Kaynak olarak veriler kaynakçada belirttiğim Javascript dili ile yazdığım kodlar yardımı ile Youtube API V3 ve Spotify WEB API kullanılarak elde edilmiştir.

Sonuçlar grafikler ile sunulmuş ve grafikler ile ilgili değerlendirmeler yapılarak tamamlanmıştır.

ANALİZLER İÇİN GEREKLİ ADIMLAR VERİLERİN KULLANIMI

Gerekli kütüphanelerin yüklenmesi için aşağıdaki kodları kullandım.

```
import numpy as np
import pandas as pd
!pip install isodate # isodate modülünü indirme
#(colabda dahili olmadığı için indiriyoruz)
import isodate
import matplotlib as mpl
import matplotlib.pyplot as plt
```

Sonrasında CSV dosyalarımın okunabilmesi için

```
data1 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/spotify2020.csv")
data2 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/spotify2021.csv")
data3 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/spotify2022.csv")
data4 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/spotify2023.csv")

data5 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/youtube2020.csv")
data6 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/youtube2021.csv")
data7 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/youtube2022.csv")
data8 =
pd.read_csv("https://raw.githubusercontent.com/bymayfe/YoutubeSpotifyStatsVirtualizat
ion/main/datasets/youtube2023.csv")
```

Grafikleri sunmadan önce verilerimin nasıl gözüktüğünü paylaşayım.

data7.head(14)

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	published	channelId	title	description	thumbnail	channelTitle	tags	categoryId	liveBroadcast	defaultLanguage	videoDuration	videoDimension	videoDefinition	videoCaption	videoLicense	videoProject	viewCount	likeCount	favouriteCount	commentCount
2	2022-11-2	UC4Zc-Jlni	ÖYmŞ Pit E Aquil	{'default': 'Pit Gamer'}	{'Pitbulldc			15	none	pt-BR	PT16S	2d	hd	TRUE	TRUE	rectangul	5.86E+08	11136818	0	25726
3	2022-06-0	UCOcBePi	ÖYŞ% #skincarero	{'default': 'Nursema				27	none	none	PT36S	2d	hd	FALSE	TRUE	rectangul	4.64E+08	7375881	0	10975
4	2022-07-0	UCH-ByEK	Watermelon ÖY% ic	{'default': 'Itsy bitsy	siblings			22	none	none	PT1M1S	2d	hd	FALSE	TRUE	rectangul	4.41E+08	7076181	0	
5	2022-07-1	UCSJ4gkVi	lofi hip hc ÖYZ%	{'default': 'Lofi Girl	{'lo-fi', 'lo			10	live	none	P0D	2d	sd	FALSE	TRUE	rectangul	3.95E+08	2725588	0	0
6	2022-08-1	UCSgWvjR	Katelyn Ohashi Flooi	{'default': 'International Gymna				17	none	none	PT14S	2d	hd	FALSE	FALSE	rectangul	3.63E+08	5634842	0	30606
7	2022-10-1	UCFvN0Lz	Cameram Cameram	{'default': 'Jumpers Junction				17	none	none	PT15S	2d	hd	FALSE	TRUE	rectangul	2.07E+08	7026387	0	83159
8	2022-05-2	UCR5wZc	Burak Buli Burak	{'default': 'netd mÄ%	{'netd MÄ			10	none	tr	PT2M20S	2d	hd	TRUE	TRUE	rectangul	1.89E+08	567945	0	17906
9	2022-06-3	UCR5wZc	KurtuluÄY KurtuluÄ	{'default': 'netd mÄ%	{'netd MÄ			10	none	tr	PT2M48S	2d	hd	TRUE	TRUE	rectangul	1.55E+08	352176	0	15409
10	2022-11-2	UC1Uuou	I saw an opportunity	{'default': 'Alex Murphy				24	none	none	PT24S	2d	hd	FALSE	TRUE	rectangul	1.55E+08	4669493	0	49506
11	2022-07-2	UCJWZnZi	Ara - Zeyn Harbiye	{'default': 'Zeynep Be	{'zeynep t			10	none	none	PT3M21S	2d	hd	TRUE	TRUE	rectangul	1.51E+08	535631	0	22582
12	2022-11-0	UCOcBePi	Landscape #drawing #	{'default': 'Nursema				27	none	none	PT58S	2d	hd	FALSE	TRUE	rectangul	1.4E+08		0	18671
13	2022-04-0	UCR5wZc	Semicenk Semicen	{'default': 'netd mÄ%	{'netd MÄ			10	none	tr	PT3M13S	2d	hd	TRUE	TRUE	rectangul	1.13E+08	519461	0	12482
14	2022-05-1	UCEicf154	Velet - ÄÄ 2022 -	{'default': 'VELET MU	{'velet', 'a			10	none	tr	PT2M4S	2d	hd	FALSE	TRUE	rectangul	92222547	472561	0	21956

Tablo 1 (youtube2022.csv) ayrıca tüm Youtube dataları buna benzemektedir.

data1.head(11)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	name	descriptio	owner	followers	total_trac	songName	songID	artistNam	artistGenr	songOrde	songAvail	songExpli	songPopu	songDuration
2	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	Bi Sonraki	5SYkeNpji	Murda, Ezhel		1 ['AR', 'AU']	FALSE		52	167350
3	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	Unutulaca	3aXTeIvni	Gazapizm		2 ['AR', 'AU']	FALSE		61	295637
4	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	UslanmÄ±	24BvISG2V	Zeynep BastÄ±k		3 []	FALSE		0	193082
5	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	Leila	1RqNAiL2	Reynmen		4 ['AR', 'AU']	FALSE		56	169714
6	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	AYA	4ywdchqX	Murda, Ezhel		5 ['AR', 'AU']	FALSE		52	196583
7	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	SaÄYÄ± Sc	5GBbUx3e	Gazapizm		6 ['AR', 'AU']	FALSE		55	264064
8	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	KazÄ±Ä±	5DJEXxOu	YÄ%zyÄ%zeyken Kor		7 ['AR', 'AU']	FALSE		62	208628
9	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	Nimet	3917xM4P	Didomido, Eglo G		8 []	FALSE		0	136730
10	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	Dance Mo	1rgnBhdG	Tones And I		9 ['AU', 'NZ']	FALSE		61	209754
11	YÄ±lÄ±n Ä	TÄ%rkiye	Cihan Ceri	1038	100	Kafemda	5emf5Wol	Äkling On Kala		10 []	FALSE		0	181370

Tablo 2 (spotify2020.csv) ayrıca tüm Spotify dataları buna benzemektedir.

```
statistics = data1['songPopularity'].describe().reset_index()
statistics.style.format({"number": "{:20,.0f}"}) .highlight_max(color='RED')
statistics = data2['songPopularity'].describe().reset_index()
statistics.style.format({"number": "{:20,.0f}"}) .highlight_max(color='BLUE')
```

	index	songPopularity
0	count	100.000000
1	mean	31.400000
2	std	29.923471
3	min	0.000000
4	25%	0.000000
5	50%	41.000000
6	75%	56.000000
7	max	89.000000

Tablo 3 (Kırmızı): 2020 yılına ait

	index	songPopularity
0	count	100.000000
1	mean	56.320000
2	std	15.462486
3	min	0.000000
4	25%	53.000000
5	50%	60.000000
6	75%	63.250000
7	max	84.000000

Tablo 4 (Mavi): 2021 yılına ait

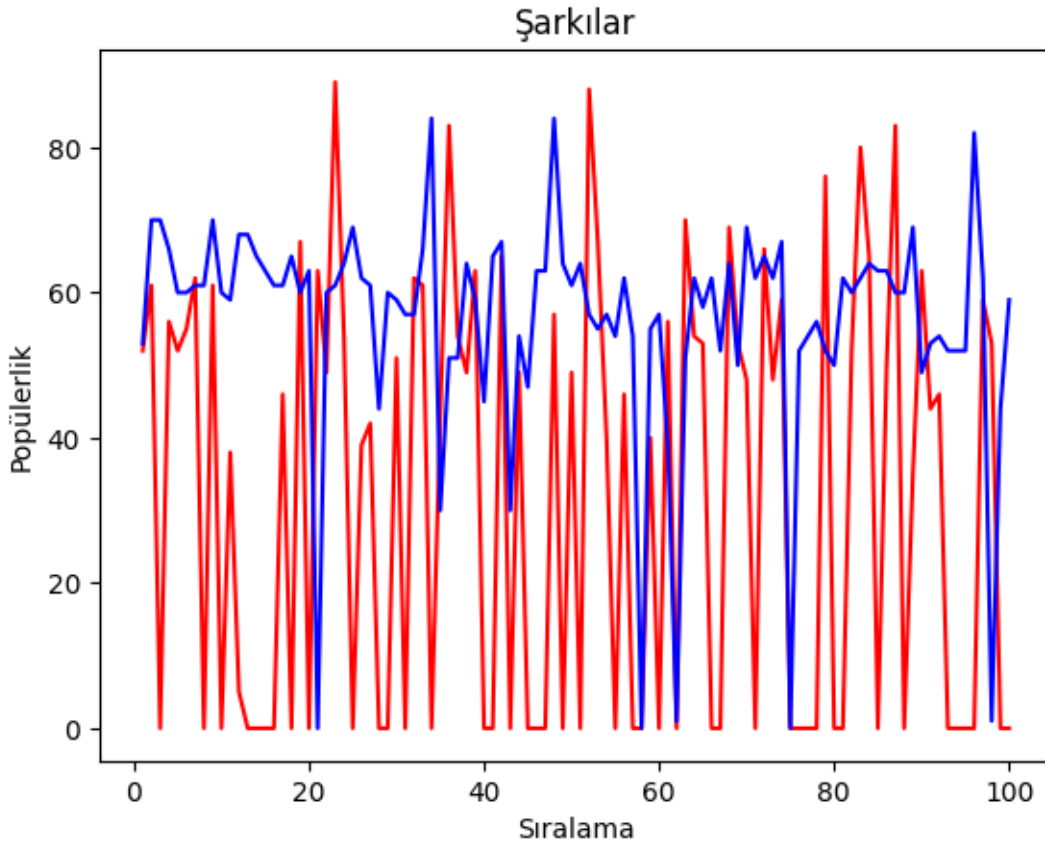
Ortalama, standart sapma, min/max değerleri ve çeyrekliklerini göstermektedir

YILLARA GÖRE SPOTIFY VERİ ANALİZLERİ

Bu değerleri bulduktan sonra, matplotlib ve pandas kütüphanelerini kullanarak aşağıda belirttiğim 20 grafiği oluşturdum ve altlarında açıklamasıyla beraber sizlere sundum.

Burada 2020 ve 2021 yılına ait Spotify şarkı liste sıralaması ve şarkının toplam popülarlığını matplotlib kütüphanesi kullanarak plot ile görselleştirdim.

```
plt.plot(data1['songOrder'], data1['songPopularity'] , label = '2020', color="red")
plt.plot(data2['songOrder'], data2['songPopularity'] , label = '2021' , color="blue")
plt.title('Şarkılar')
plt.xlabel('Sıralama')
plt.ylabel('Popülerlik')
plt.show()
```

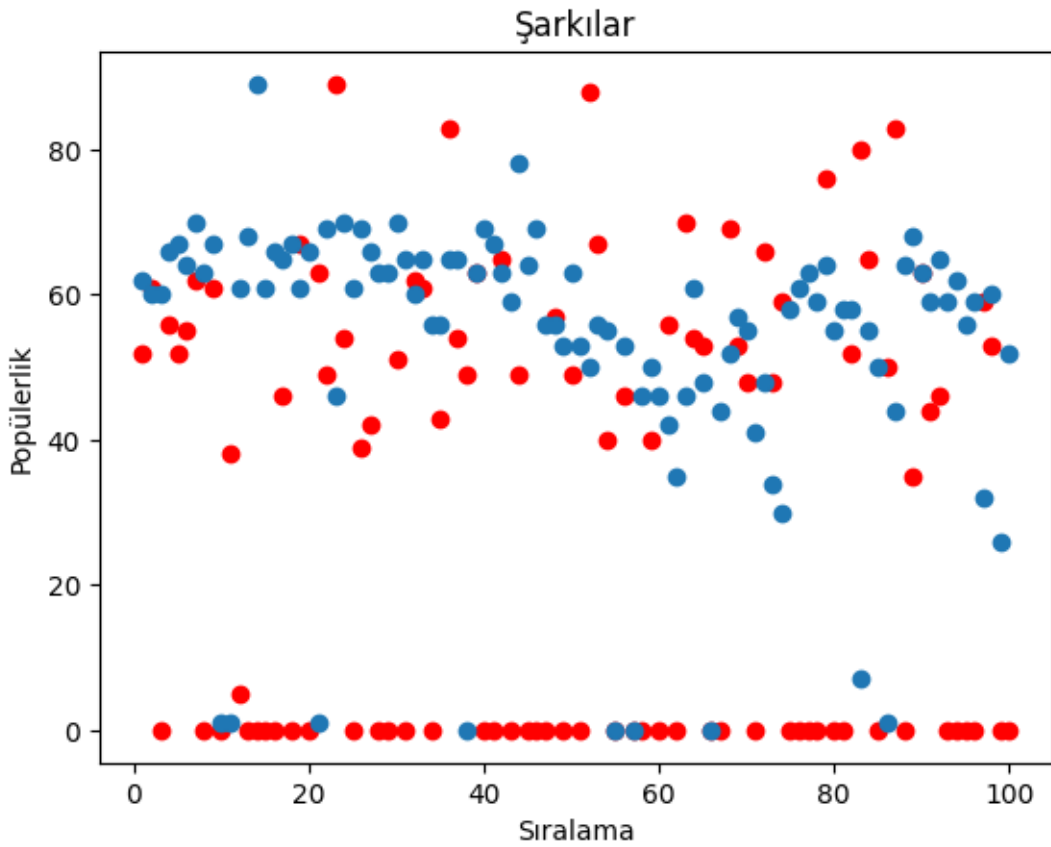


Tablo 5

Burada kırmızı olan 2020 mavi olan ise 2021 yılına ait veri olmak üzere aşağıdaki çıkarımları yapabiliriz. Spotify’da 2021 yılındaki en çok ilk 50 şarkı popülarlık konusunda 2020 yılındaki en çok dinlenen ilk 50 şarkıdan daha popülerdir.

Burada 2020 ve 2021 yılına ait Spotify şarkı liste sıralaması ve şarkının toplam popülerliğini matplotlib kütüphanesi kullanarak scatter ile görselleştirdim.

```
plt.scatter(data1['songOrder'], data1['songPopularity'], label = '2020', color="red")
plt.scatter(data3['songOrder'], data3['songPopularity'], label = '2022')
plt.title('Şarkılar')
plt.xlabel('Sıralama')
plt.ylabel('Popülerlik')
plt.show()
```

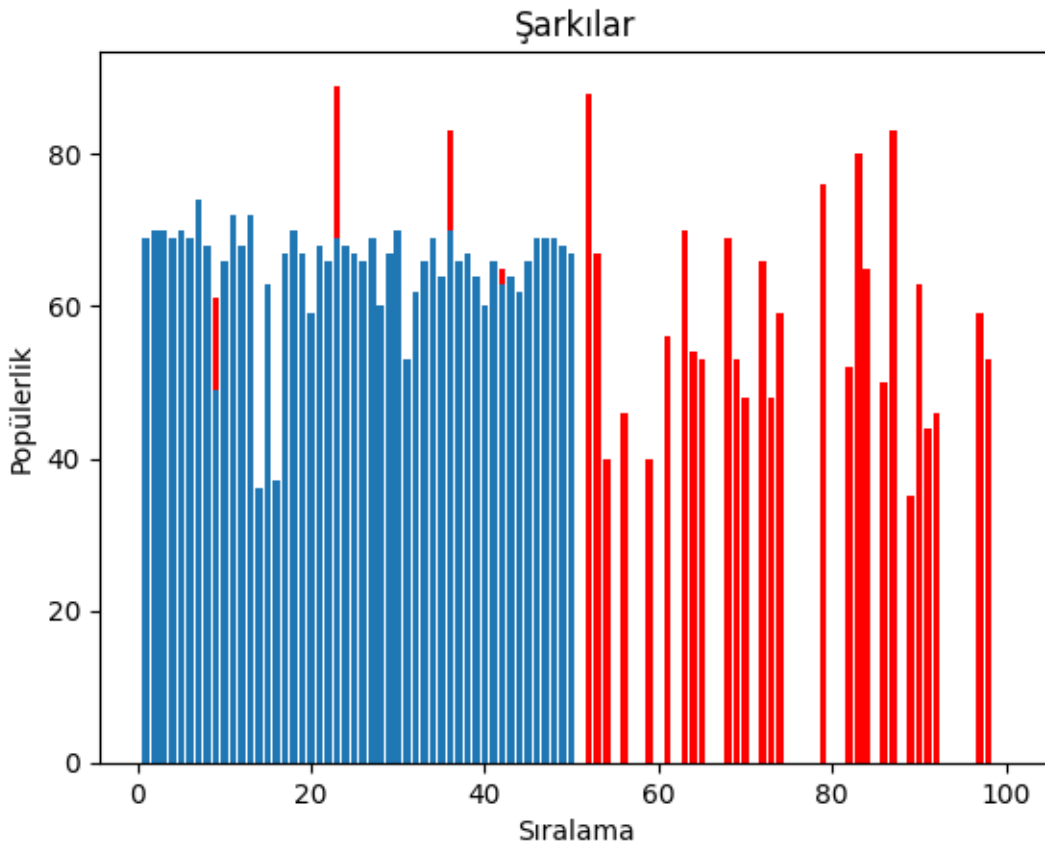


Tablo 6

Burada kırmızı olan 2020 mavi olan ise 2022 yılına ait veri olmak üzere aşağıdaki çıkarımları yapabiliriz. Spotify’da 2022 yılındaki en çok ilk 50 şarkı popülerlik konusunda 2020 yılındaki en çok dinlenen ilk 50 şarkıdan daha popülerdir.

Burada 2020 ve 2023 yılına ait Spotify şarkı liste sıralaması ve şarkının toplam popülerliğini matplotlib kütüphanesi kullanarak bar ile görselleştirdim.

```
plt.bar(data1['songOrder'], data1['songPopularity'], label = '2020', color="red")
plt.bar(data4['songOrder'], data4['songPopularity'], label = '2023')
plt.title('Şarkılar')
plt.xlabel('Sıralama')
plt.ylabel('Popülerlik')
plt.show()
```

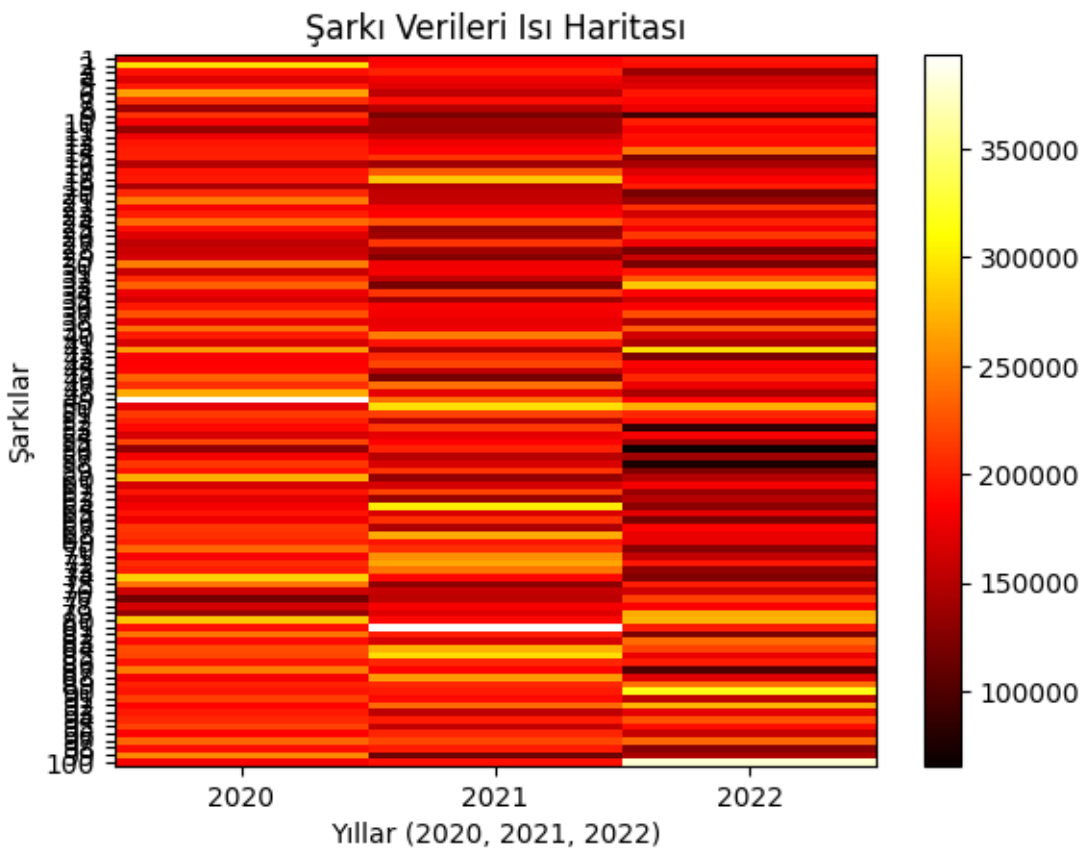


Tablo 7

Burada kırmızı olan 2020 mavi olan ise 2023 yılına ait veri olmak üzere aşağıdaki çıkarımları yapabiliriz. Spotify’da 2023 yılındaki en çok ilk 50 şarkı popülerlik konusunda 2020 yılındaki en çok dinlenen ilk 50 şarkıdan daha popülerdir.

Burada 2020, 2021 ve 2022 yılına ait Spotify şarkıların süreleri bakımından karşılaştırma ve çıkarım yapmak için matplotlib kütüphanesi kullanarak colorbar ile görselleştirdim.

```
dataset1 = [data1['songDuration'], data2['songDuration'], data3['songDuration']]
dataset1 = np.array(dataset1).T
plt.imshow(dataset1, aspect='auto', cmap='hot')
plt.colorbar() # Renk çubuğunu ekleme
plt.title('Şarkı Verileri Isı Haritası')
plt.xlabel('Yıllar (2020, 2021, 2022)')
plt.ylabel('Şarkılar')
plt.xticks(ticks=[0, 1, 2], labels=['2020', '2021', '2022'])
plt.yticks(ticks=range(len(data1['songDuration'])), labels=data1['songOrder'])
plt.show()
```

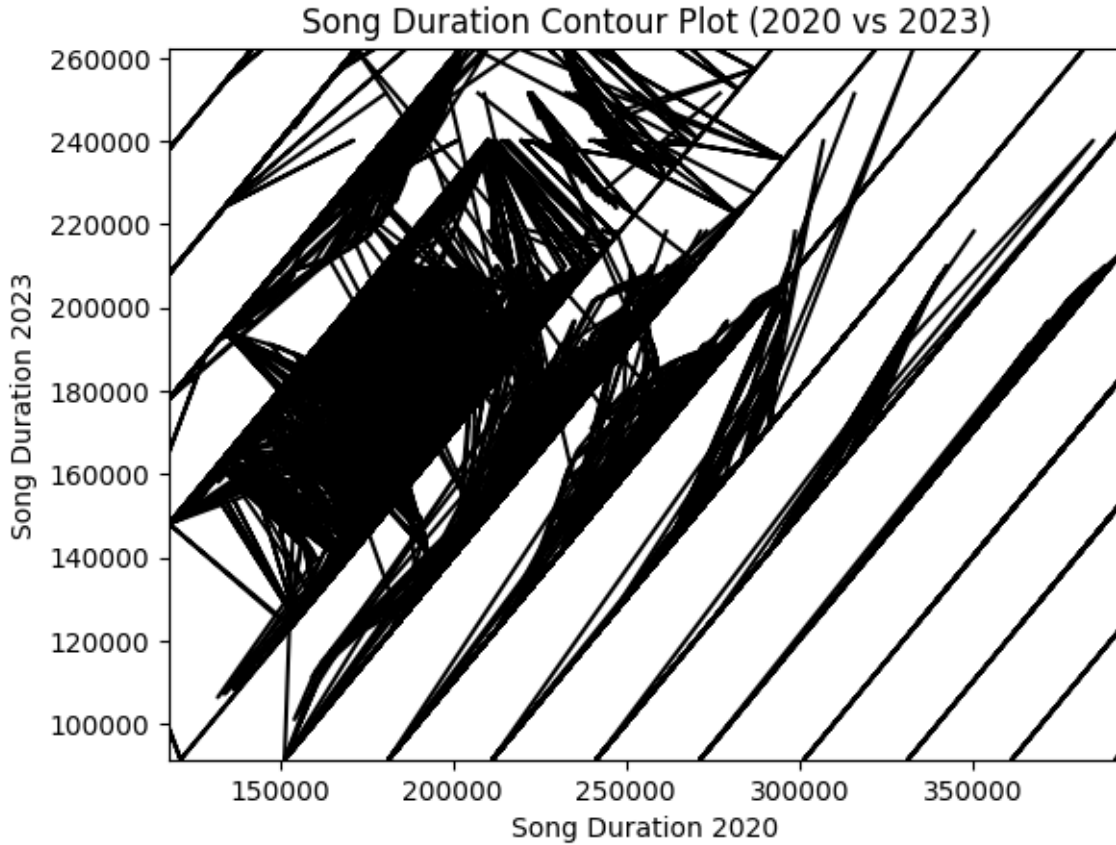


Tablo 8

Burada tüm yıllar için rengin hafif turuncuya gittiği ve kırmızı turuncu arasında olup 200000 milisaniye (3.33 dakika) ve 250000 milisaniye (4.16 dakika) olduğu çıkarımını ve 2020 de çıkan ve popüler olan şarkıların rengine bakarak şarkı uzunluklarının arttığı çıkarımına varıyoruz. Yıllar arttıkça şarkı süreleri kısalmaya başlamıştır.

Burada 2020 ve 2023 yılına ait Spotify şarkıların süreleri bakımından karşılaştırma ve çıkarım yapmak için matplotlib kütüphanesi kullanarak contour ile görselleştirdim.

```
duration_2020 = data1['songDuration']
duration_2023 = data4['songDuration']
# Ortak bir ölçek oluşturmak için verileri normalize edelim
popularity_2020 = (duration_2020 - np.min(duration_2020)) / (np.max(duration_2020) - np.min(duration_2020))
popularity_2021 = (duration_2023 - np.min(duration_2023)) / (np.max(duration_2023) - np.min(duration_2023))
X, Y = np.meshgrid(duration_2020, duration_2023)
Z = np.abs(X - Y)
plt.contour(X, Y, Z, levels=10, colors='black')
plt.title('Song Duration Contour Plot (2020 vs 2023)')
plt.xlabel('Song Duration 2020')
plt.ylabel('Song Duration 2023')
plt.show()
```

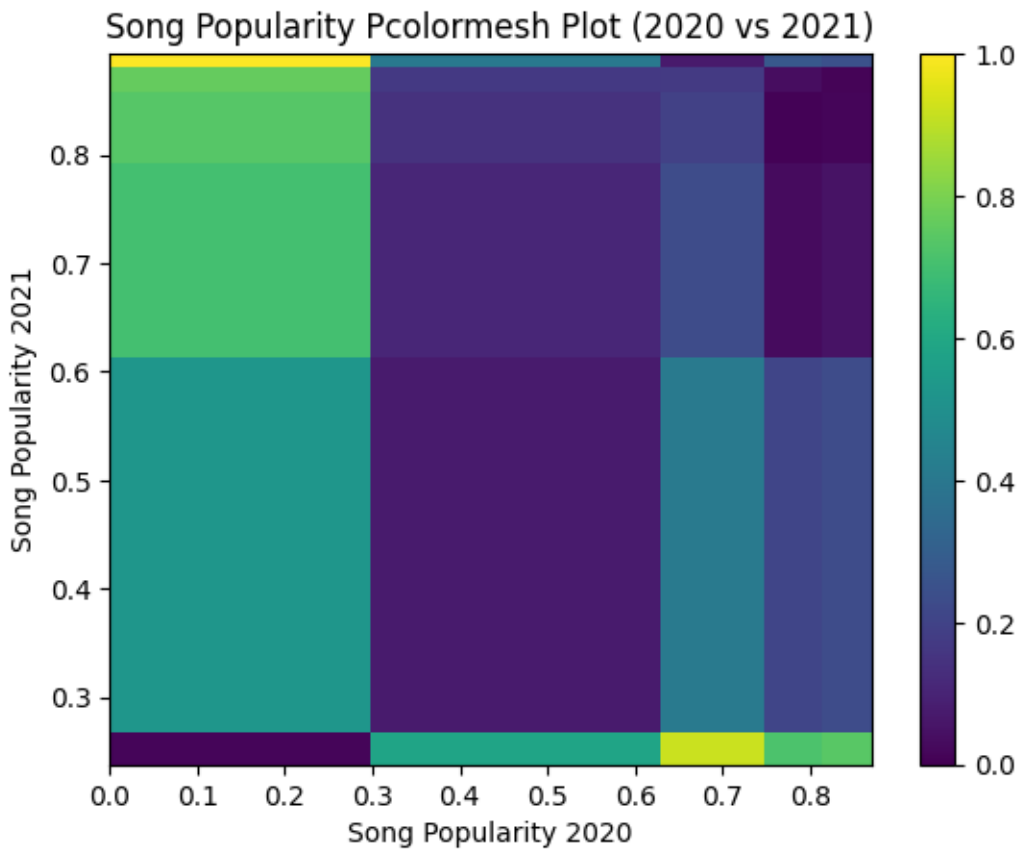


Tablo 9

Burada Spotify şarkılarının süreleri ile ilgili tekrar bir çıkarım yapmak gerekirse, 2020 ile 2023 yılındaki üstte yaptığım çıkarımı destekleyici bir grafik görüyoruz. Yılların şarkı süreleri ile ilgili kesiştiği yerler ortalamaya yakın ve 2023 yılında 260000 milisaniye max seviye ve oralarda biraz yoğunluk oluşumu var. 2020 yılına bakacak olursak max 350000 milisaniye olup yoğunluk oldukça az. Yani yıllar arttıkça popüler şarkıların şarkı süreleri azalmıştır.

Burada 2020 ve 2021 yılına ait Spotify şarkıların popülerlik indexleri bakımından karşılaştırma ve çıkarım yapmak için matplotlib kütüphanesi kullanarak pcolormesh ile görselleştirdim.

```
popularity_2020 = data1['songPopularity']
popularity_2021 = data2['songPopularity']
# Ortak bir ölçek oluşturmak için verileri normalize edelim
popularity_2020 = (popularity_2020 - np.min(popularity_2020)) /
(np.max(popularity_2020) - np.min(popularity_2020))
popularity_2021 = (popularity_2021 - np.min(popularity_2021)) /
(np.max(popularity_2021) - np.min(popularity_2021))
X, Y = np.meshgrid(popularity_2020, popularity_2021)
Z = np.abs(X - Y)
plt.pcolormesh(X, Y, Z, cmap='viridis', vmin=0, vmax=1)
plt.colorbar()
plt.title('Song Popularity Pcolormesh Plot (2020 vs 2021)')
plt.xlabel('Song Popularity 2020')
plt.ylabel('Song Popularity 2021')
plt.show()
```

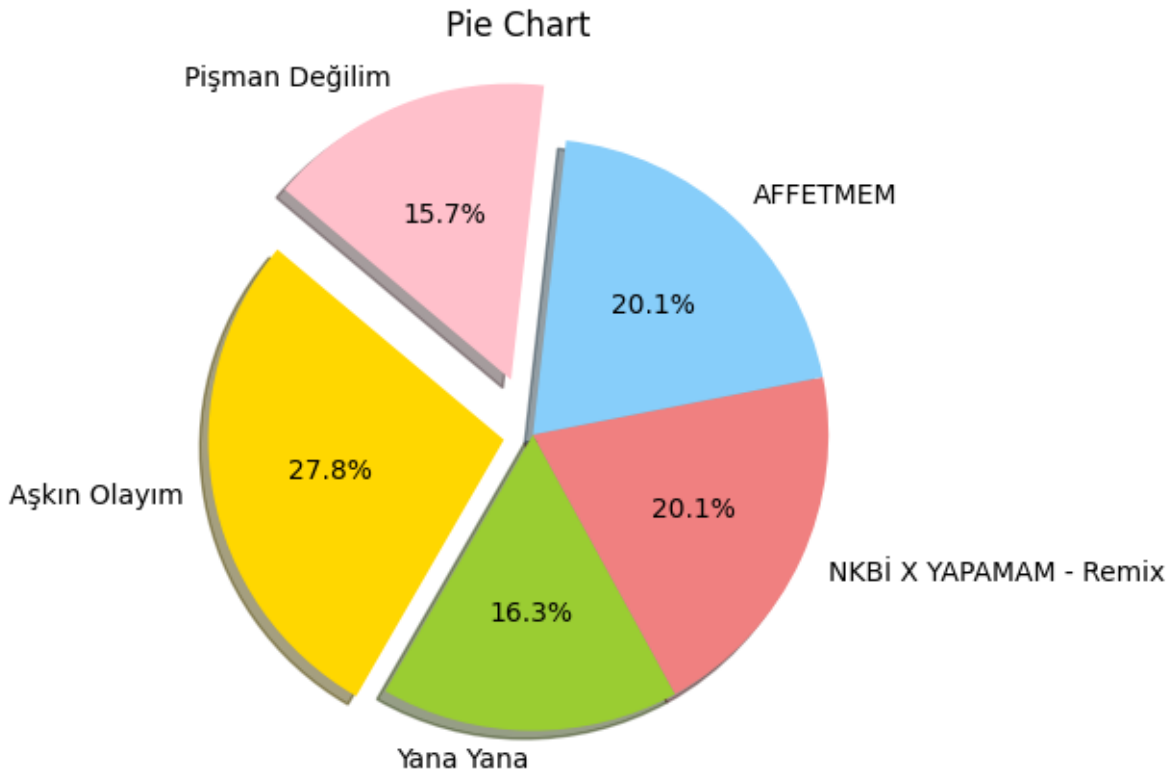


Tablo 10

Burada 2020 ve 2021 yılına ait verilerin popülerliğinin değer aralığını görüyoruz. Popülerlik indexi 0-100 arasında değer almaktadır. Popülerlik farklılıklarını listenin ilk sıralarında görüyoruz. Ve farklılıklar 0.6 (60) ile 1 (100) index değerleri arasında yer almakta. Listenin ilk sıralarının alan şarkılar yıllara göre popülerlik anlamında farklılık göstermiştir.

Burada 2023 yılına ait Spotify’da en çok dinlenen ilk 5 şarkının şarkı süreleri ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak pie ile görselleştirdim.

```
sizes = data4['songDuration'][:5] # İlk dört popülerite değeri
labels = data4['songName'][:5] # İlk dört şarkı ismi
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue', "pink"]
explode = (0.1, 0, 0, 0, 0.2) # İlk dilimi öne çıkar
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',
shadow=True, startangle=140)
plt.title('Pie Chart')
plt.axis('equal') # Daire şeklinde çizim
plt.show()
```



Tablo 11

Burada en çok dinlenen şarkının en çok şarkı süresine sahip olduğunu ve burada listede 5. Sırada bulunan şarkının ise en az şarkı süresine sahip olduğunu ilk 5 şarkı için sıralama azaldıkça şarkı süresinin azaldığını görebiliyoruz. Bu ilk 5 şarkı için geçerli olup listenin tamamına bakmamız gerektiği çıkarımını da göz ardı etmememiz gerekiyor.

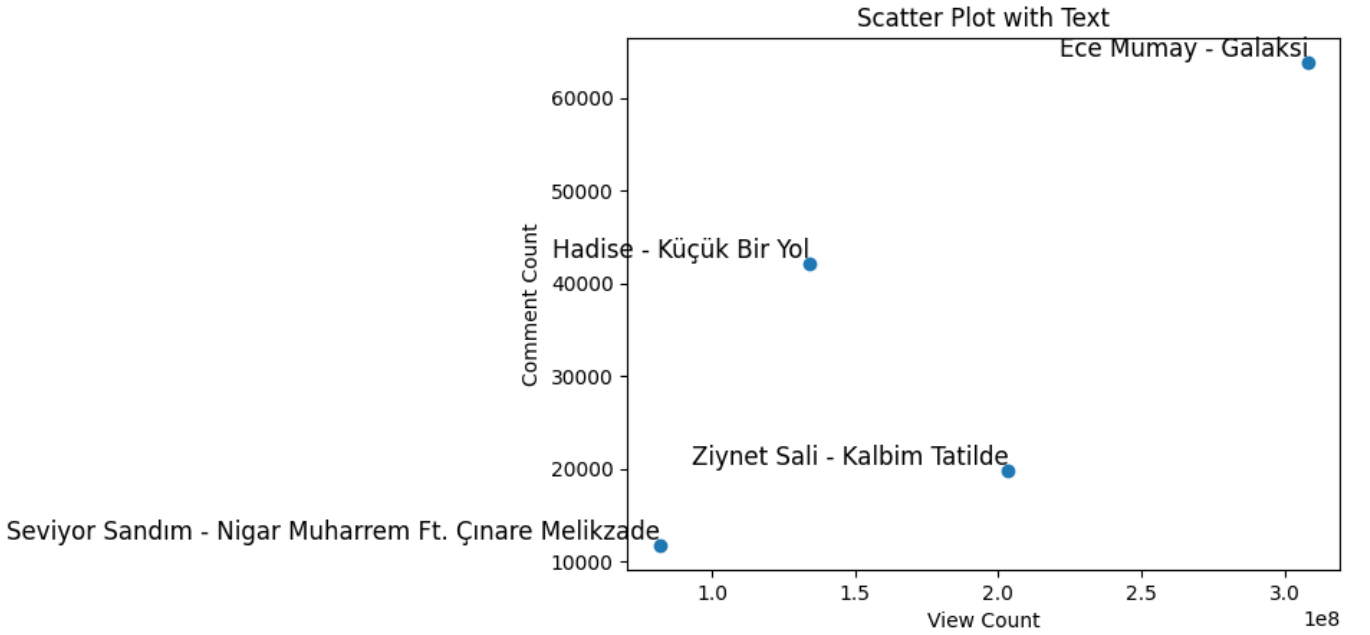
YILLARA GÖRE YOUTUBE VERİ ANALİZLERİ

Burada 2020 yılına ait Youtube’da en çok izlenen ilk 5 videonun yorum sayısı ve izlenme sayısı ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak scatter with text ile görselleştirdim.

```
# x = np.arange(len(data5['viewCount'][:5]))
x = data5['viewCount'][:5]
y = data5['commentCount'][:5]

# Scatter plot with text
plt.scatter(x, y)
for i in range(len(x)):
    plt.text(x[i], y[i], data5['title'][i], fontsize=12, ha='right', va='bottom')

plt.title('Scatter Plot with Text')
plt.xlabel('View Count')
plt.ylabel('Comment Count')
plt.show()
```

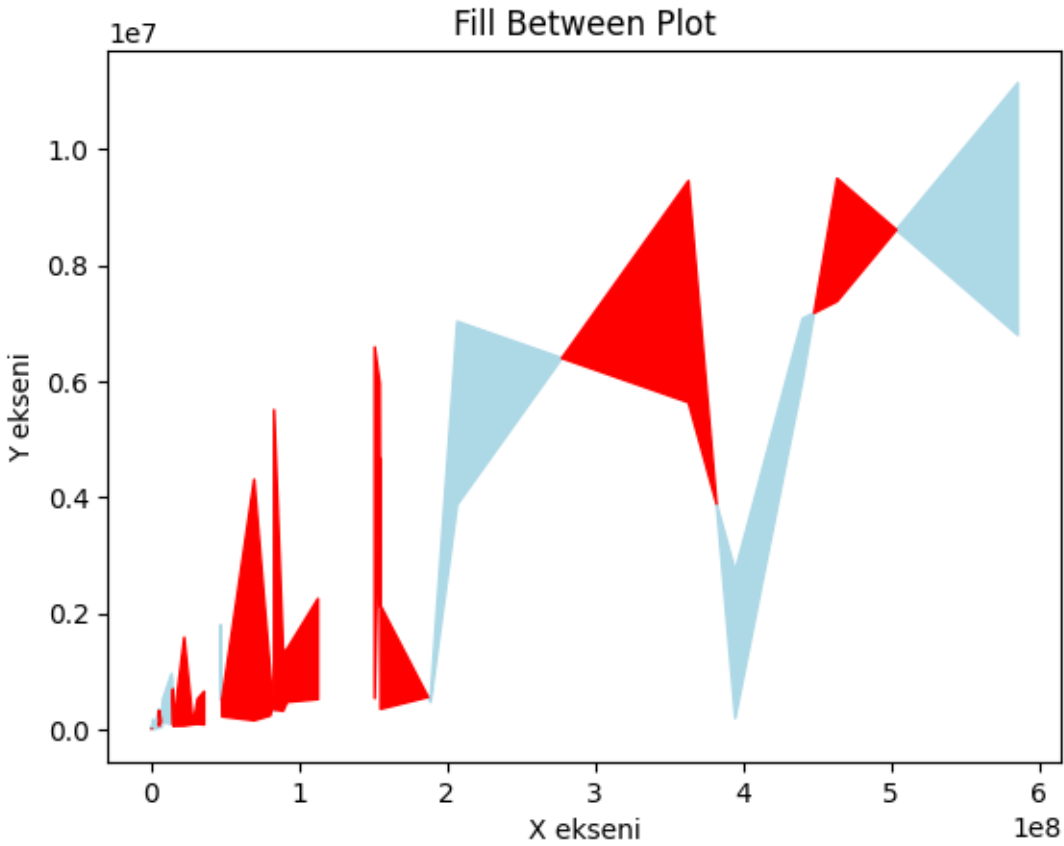


Tablo 12

Burada en çok izlenen ilk 5 videonun başlıklarıyla beraber izlenme sayılarının (x*100milyon) yorumlar üzerindeki etkisine bakacak olursak genellikle izlenme sayısı artınca yorum sayıları da artmış bulunmaktadır. Tümüne yorum yapmak için aldığımız değer sayısını arttırmak daha iyi olacaktır. Ben grafiği daha temiz sunmak amacı ile ilk 5 videoyu aldım. Siz kodda değişikliğe giderek sayıyı arttırıp tam sonuca ulaşabilirsiniz.

Burada 2022 ve 2023 yılına ait Youtube’da en çok izlenen ilk 50 videosu ile ilgili izlenme sayısının beğeni sayı üzerine etkisi ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak Fill between ile görselleştirdim.

```
x = data7['viewCount']
y1 = data7['likeCount']
y2 = data8['likeCount']
plt.fill_between(x, y1, y2, color='lightblue', where=(y1 > y2), interpolate=True)
plt.fill_between(x, y1, y2, color='red', where=(y1 <= y2), interpolate=True)
plt.title('Fill Between Plot')
plt.xlabel('X eksenı')
plt.ylabel('Y eksenı')
plt.show()
```

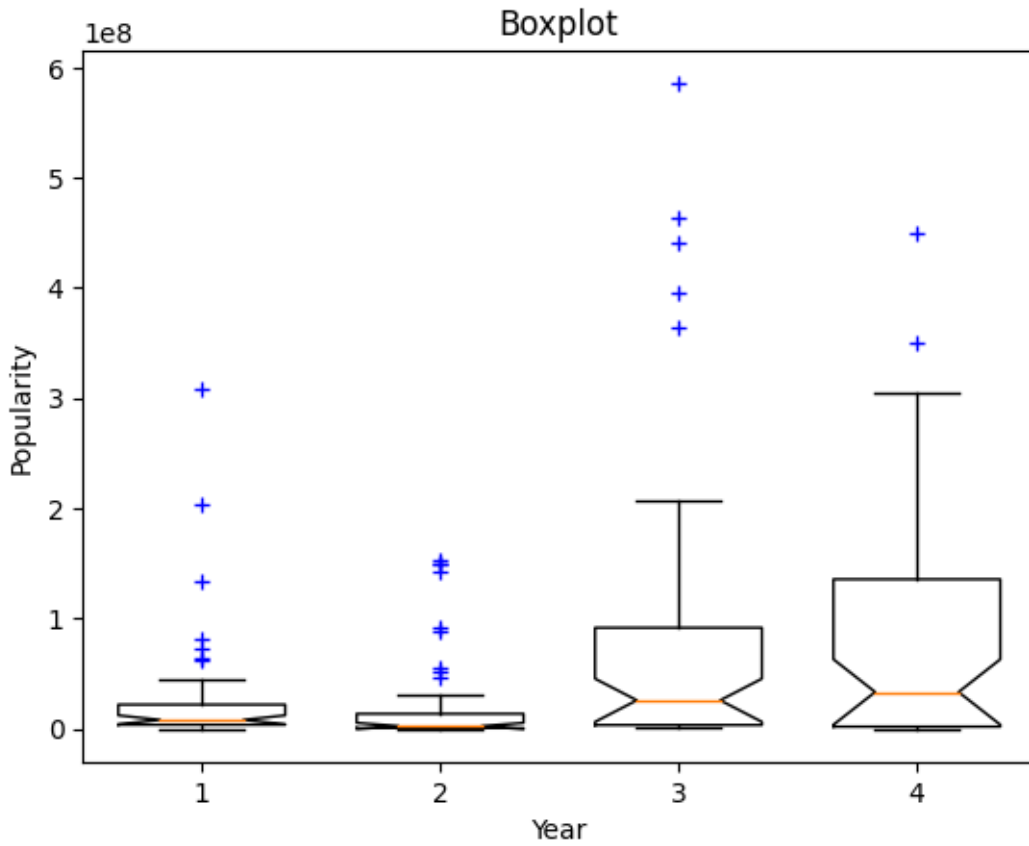


Tablo 13

Burada 2022 ve 2023 yılına ait Youtube’daki en çok izlenen ilk 50 videonun izlenme sayısı ile beğenme ve yorum sayısını karşılaştırmak için bir değer aralığı grafiği ile yorumluyoruz. Burada y1(2022 beğeni sayısı) y2(2023 beğeni sayısı) değerleri olmak üzere y1 ve y2 aralarındaki değerlerden büyük olanların kullanımıyla oluşturulmuştur. Yorum olarak izlenme sayısı arttıkça yorum sayısının da genel olarak arttığı söylenebilir.

Burada 2020, 2021, 2022 ve 2023 yılına ait Youtube’da en çok izlenen ilk 50 videosu ile ilgili izlenme sayısının popülerlik üzerine etkisi ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak Boxplot ile görselleştirdim.

```
data = [data5['viewCount'], data6['viewCount'], data7['viewCount'],  
data8['viewCount']]  
plt.boxplot(data, notch=True, sym='b+', widths=0.7)  
plt.title('Boxplot')  
plt.xlabel('Year')  
plt.ylabel('Popularity')  
plt.show()
```



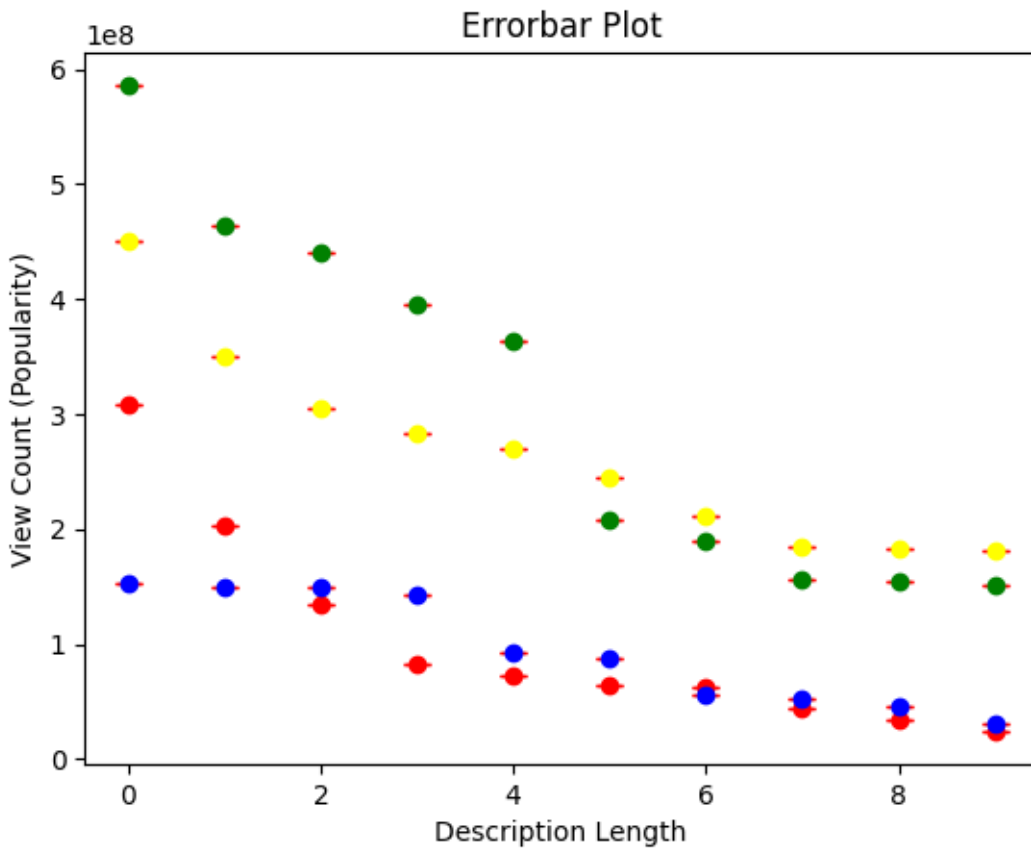
Tablo 14

Burada 2020, 2021, 2022 ve 2023 yıllarına ait Youtube’da en çok izlenen ilk 50 videodaki izlenmelerin popülerliği ve medyanları ile ilgili ve izlenmenin popülerlik üzerindeki etkisini bu grafik ile çıkarım yaparak şu yoruma varabiliriz. 2023 yılı diğer yıllardan daha fazla izleyici sayısına ulaşmış ve daha popüler olmuş aynı yorumu medyan (ortanca değer) için de yapabilirim. Ortanca değerleri de yıllar arttıkça artış gösteriyor. Ayrıca 2020 yılı izlenmeleri 2021 yılı izlenmelerinden daha fazla popülerlik sağlamış 2021 yılı harici diğer yıllar arttıkça izlenme sayılarına bağlı popülerlik artış gösteriyor.

Burada 2020, 2021, 2022 ve 2023 yılına ait Youtube’da en çok izlenen ilk 10 videosunun açıklama kısmındaki karakter sayısının videolardaki izlenme sayısına etkisi ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak errorbar ile görselleştirdim.

```
datas= [data5, data6, data7, data8]
colors = ['red', 'blue', "green", "yellow"]
count= 0
for data in datas:
    x=np.arange(len(data['description'][:10]))
    y = data['viewCount'][:10]
    yerr = np.random.rand(len(data['title'][:10])) * 10
    plt.errorbar(x,y, yerr=yerr, fmt='o', ecolor='r', capsize=5, color=colors[count])
    count+=1

plt.title('Errorbar Plot')
plt.xlabel('Description Length')
plt.ylabel('View Count (Popularity)')
plt.show()
```

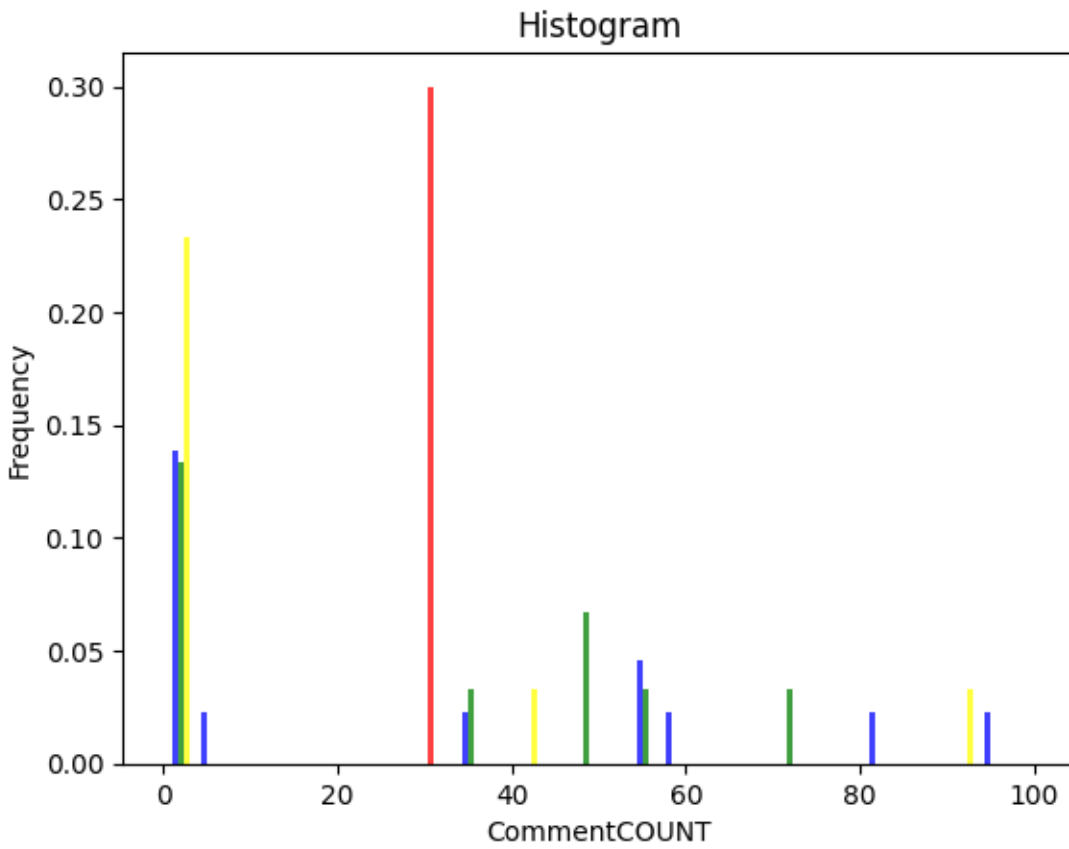


Tablo 15

Burada izlenme sayısı ile Açıklama kısmındaki yazı uzunluğu arasında ters orantı var. Çok izlenen videoların açıklama kısmındaki karakter sayısı az olup izlenme sayısı arttıkça açıklama kısmındaki karakter sayısı da azalıyor. Youtube’da çok izlenmek için açıklama kısmını kısa tutmalıyız.

Burada 2020, 2021, 2022 ve 2023 yılına ait Youtube’da en çok izlenen ilk 50 videonun yorum sayısı ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak hist ile görselleştirdim.

```
data = [data5['commentCount'], data6["commentCount"], data7['commentCount'],  
data8['commentCount']]  
colors = ['red', 'blue', "green", "yellow"]  
# Histogram plot  
plt.hist(data, bins=30, range=(0, 100), density=True, color=colors, alpha=0.75)  
plt.title('Histogram')  
plt.xlabel('CommentCOUNT')  
plt.ylabel('Frequency')  
plt.show()
```

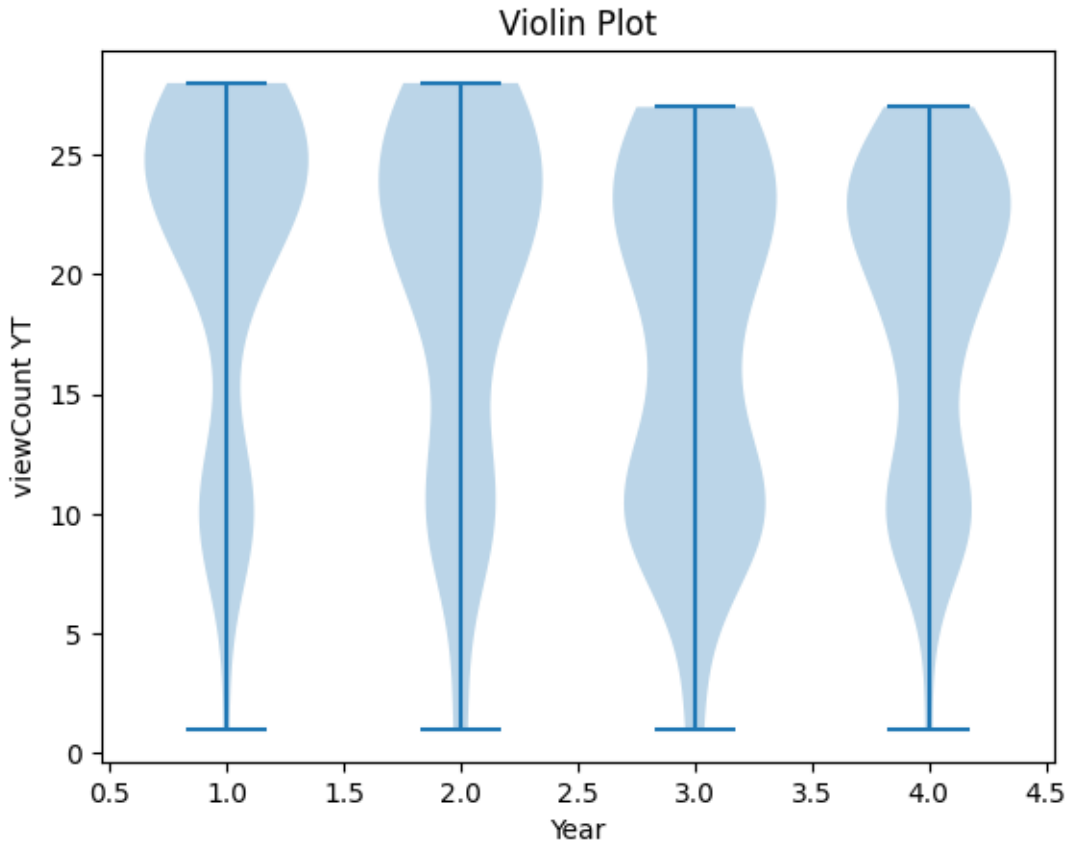


Tablo 16

Burada yıllara göre yorum sayılarının hangi frekanslarda olduğunu görüyoruz. Burada 2019 yılından sonra yorum kapatma özelliği geldiği için anlamlı çok yorum yapamıyoruz. En çok izlenen videolarda yıllar arttıkça yorum sayısı 0 üzerinde yoğunlaşmış yani günümüze geldiğimizde çok izlenen videolarda yorumlar devredışı bırakılıyor. Ayrıca 2020 yılında en çok izlenen videolarda neredeyse hiç kapatılmadığı çıkarımına ulaşıyoruz.

Burada 2020, 2021, 2022 ve 2023 yılına ait Youtube’da en çok izlenen ilk 50 videonun categoryId’leri ile ilgili çıkarım yapmak için matplotlib kütüphanesi kullanarak violinplot ile görselleştirdim.

```
data = [data5['categoryId'], data6["categoryId"], data7['categoryId'],  
data8['categoryId']]  
# Violin plot  
plt.violinplot(data, positions=[1, 2, 3, 4], widths=0.7, vert=True)  
plt.title('Violin Plot')  
plt.xlabel('Year')  
plt.ylabel('viewCount YT')  
plt.show()
```



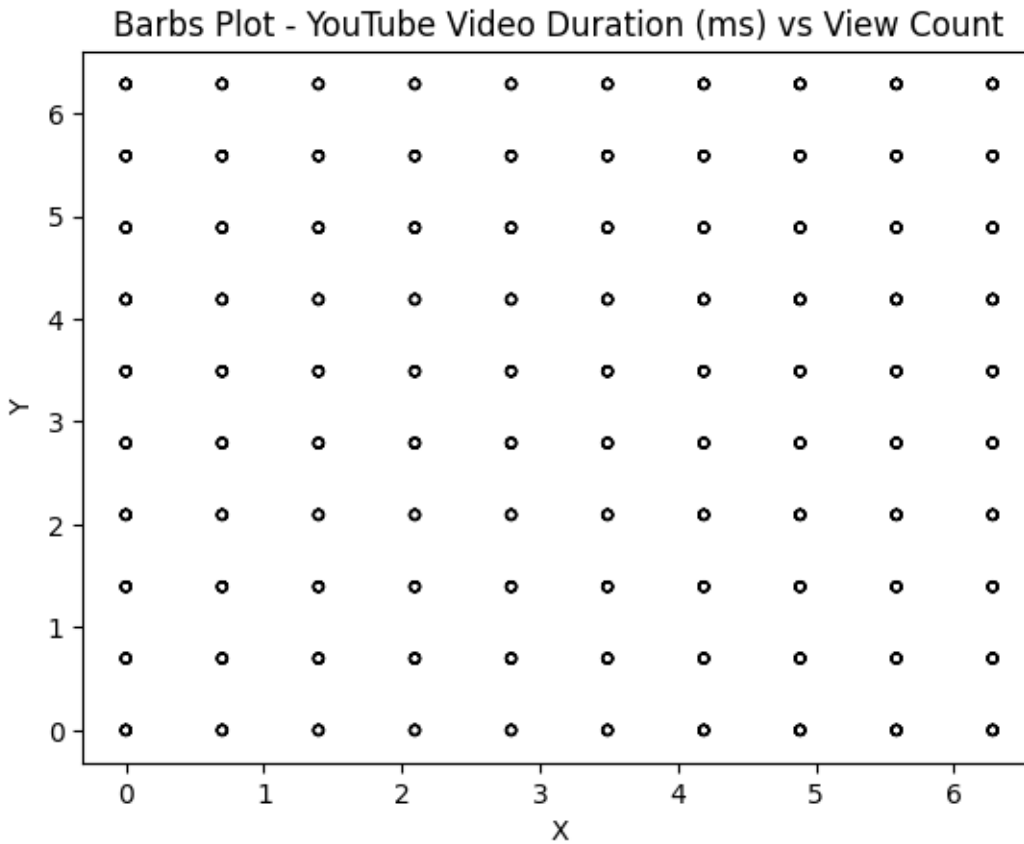
Tablo 17

Burada en çok izlenen videolarla ilgili hangi kategori üzerinde yığılım yaptığı çıkarımına varabiliriz. 2020 ve 2021 yılında 25 - News & Politics kategorisinde yığılım olduğunu görebiliyoruz. Diğer yıllarda ise 25 - News & Politics kategorisinin yanında 20 – Gaming ve 10 – Music kategorilerinde yığılmalar artmış ve o yönlere kaymış durumda. Tüm kategorileri ve IDleri görebilmek için github sayfamda hangi id hangi kategoriye ait paylaştım. Detaylıca inceleyebilirsiniz.

[YoutubeSpotifyStatsVirtualization/datasets/youtubeCategoriesId.txt at main · bymayfe/YoutubeSpotifyStatsVirtualization \(github.com\)](https://github.com/bymayfe/YoutubeSpotifyStatsVirtualization/datasets/youtubeCategoriesId.txt)

Burada tüm Youtube yıl verilerini birleştirdim iso formatındaki videoDuration verimi isodate modülü ile milisaniyeye dönüştürdüm. Anlamlı bir grafik çıkarmak için normalize edip izlenme sayısı verisi ile analiz yapabilmek için matplotlib modülünün barbs grafiği ile görselleştirdim. Fakat yorum yapamadım.

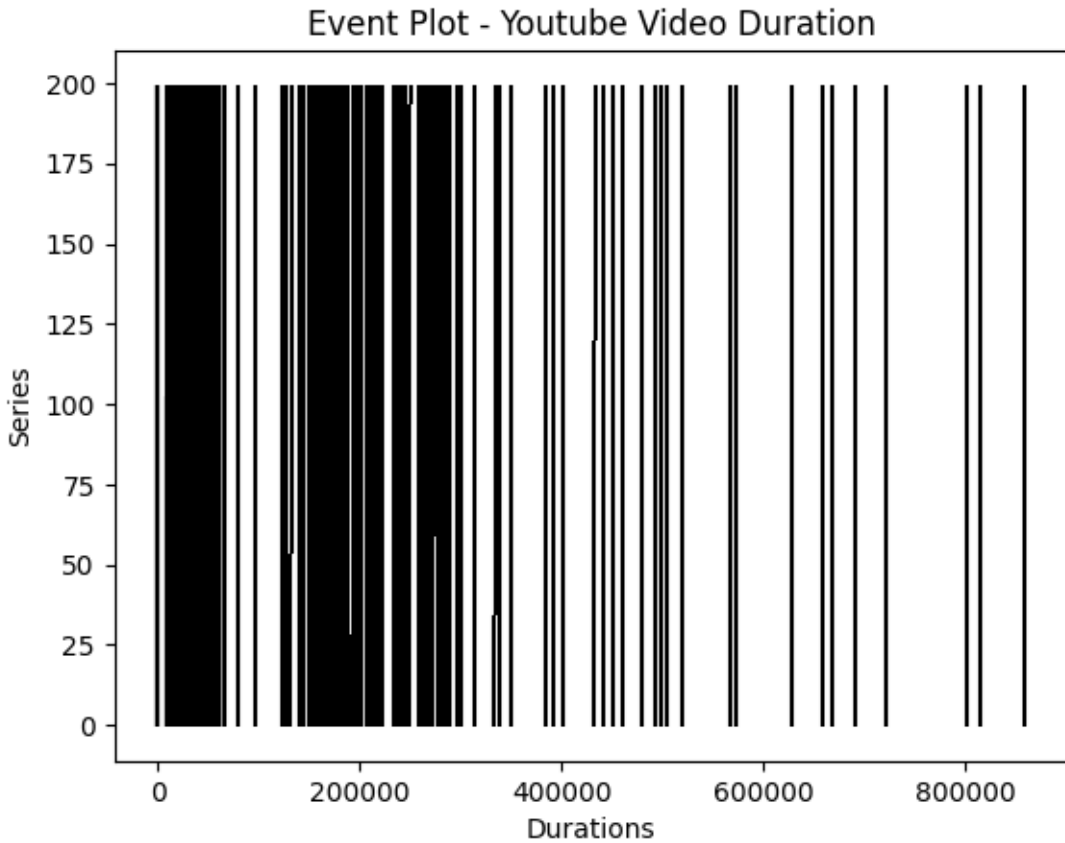
```
data = pd.concat([data5, data6, data7, data8]) # Verileri birleştirme
def iso_to_ms(duration): # ISO 8601 formatından milisaniyelere dönüştürme
    duration_obj = isodate.parse_duration(duration)
    return duration_obj.total_seconds() * 1000
data['videoDurationMS'] = data['videoDuration'].apply(iso_to_ms)
num_points = 10 # Daha küçük bir numara kullanarak meshgrid boyutunu belirleme
x = np.linspace(0, 2 * np.pi, num_points)
y = np.linspace(0, 2 * np.pi, num_points)
X, Y = np.meshgrid(x, y)
# Özellikleri normalize etme
U = data['videoDurationMS'].values[:num_points] / max(data['videoDurationMS'].values)
V = data['viewCount'].values[:num_points] / max(data['viewCount'].values)
U = np.resize(U, (num_points, num_points))
V = np.resize(V, (num_points, num_points))
plt.barbs(X, Y, U, V, length=5, pivot='middle')
plt.title('Barbs Plot - YouTube Video Duration (ms) vs View Count')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



Tablo 18

Burada tüm youtube yıl verilerini birleştirdim iso formatındaki videoDuration verimi isodate modülü ile milisaniyeye dönüştürdüm. Video süresi verisi ile analiz yapabilmek için matplotlib modülünün evenplot ile görselleştirdim.

```
data = pd.concat([data5, data6, data7, data8])
def iso_to_ms(duration):
    duration_obj = isodate.parse_duration(duration)
    return duration_obj.total_seconds() * 1000
data['videoDurationMS'] = data['videoDuration'].apply(iso_to_ms)
sinif = data["videoDurationMS"]
datalar = [sinif + i for i in range(len(sinif))] # Farklı seriler oluşturma
plt.eventplot(datalar, orientation='horizontal', lineoffsets=1, colors='black')
plt.title('Event Plot - Youtube Video Duration')
plt.xlabel('Durations')
plt.ylabel('Series')
plt.show()
```

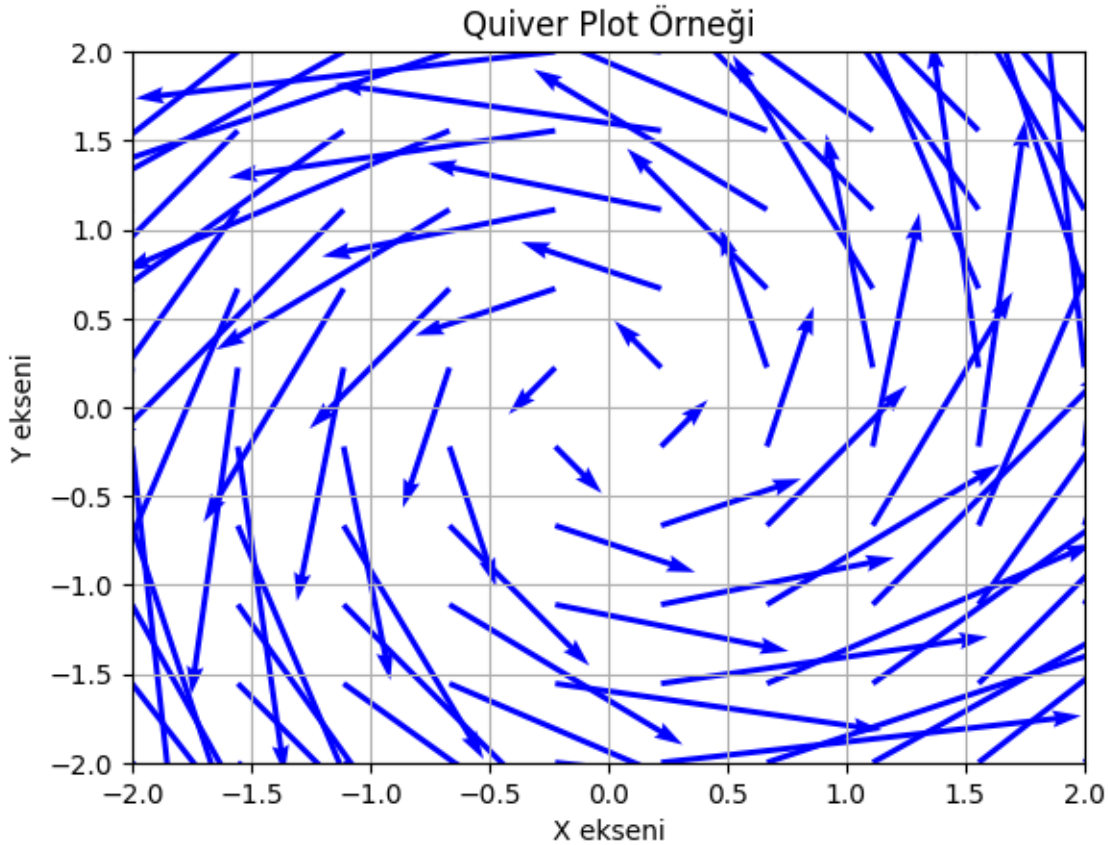


Tablo 19

Burada tüm yıl değerlerine ait verilerin video süreleri ile ilgili yorum yapacak olursak. Video süreleri 160000 milisaniye (2.66 dakika) ile 300000 milisaniye (5 dakika) arasında yoğunluk göstermektedir. Genel olarak video süreleri bu milisaniye aralığında yer almaktadır. Ayrıca kısa uzunluktaki videolar da en çok izlenen videolar listesine girmeyi başarmıştır.

Kullandığım veri Quiver Plot için uygun olmadığından kendim yapay bir veri seti oluşturarak o tablo kullanımını göstermek istedim.

```
# Yapay veri oluştur
x = np.linspace(-2, 2, 10)
y = np.linspace(-2, 2, 10)
X, Y = np.meshgrid(x, y)
U = -Y
V = X
plt.figure()
plt.quiver(X, Y, U, V, units='xy', scale=1, color='blue')
plt.title('Quiver Plot Örneği')
plt.xlabel('X eksenini')
plt.ylabel('Y eksenini')
plt.xlim([-2, 2])
plt.ylim([-2, 2])
plt.grid()
plt.show()
```



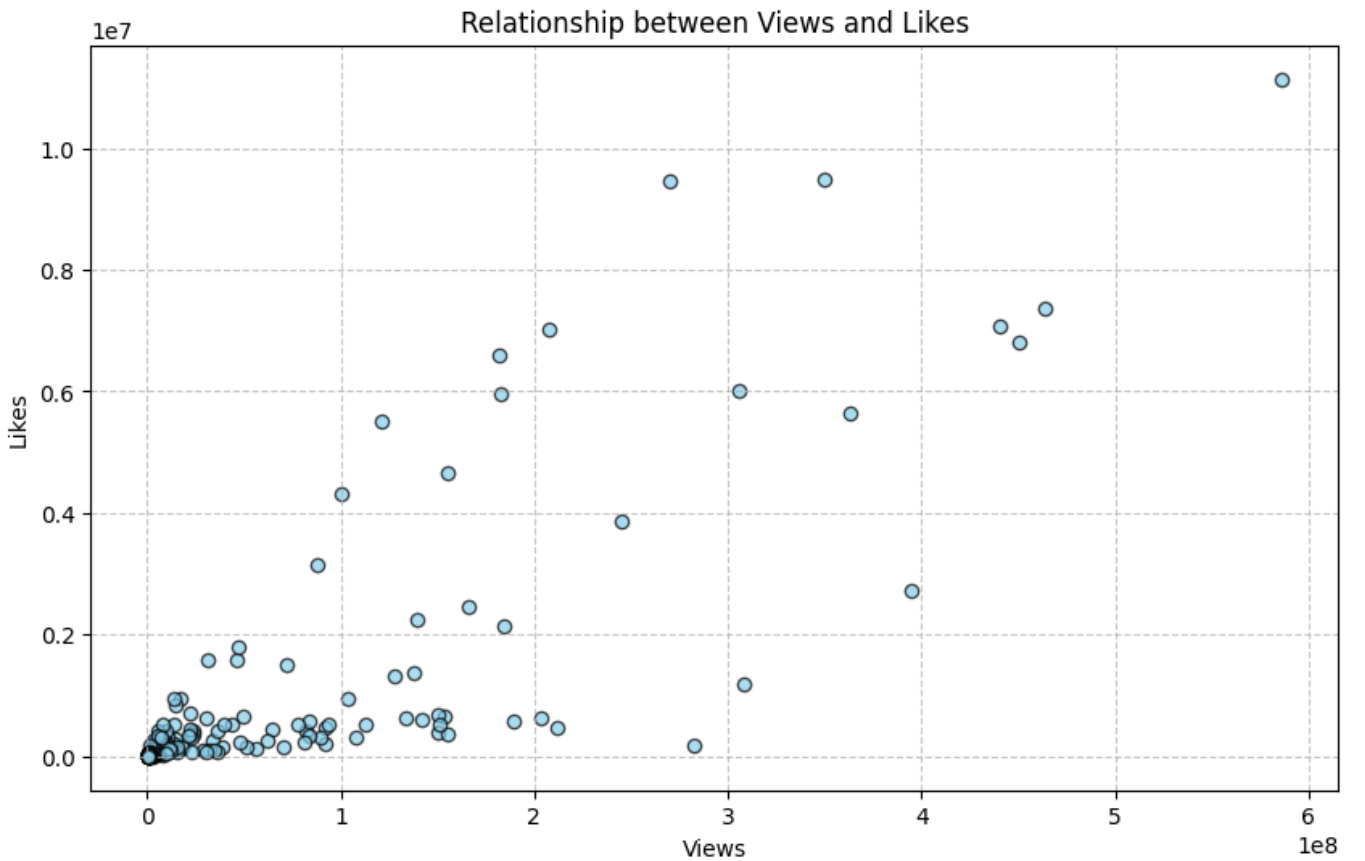
Tablo 20

Burada tüm youtube yıl verilerini birleştirdim. İzlenme sayısı ve beğeni sayısı ile ilgili analiz yapabilmek için matplotlib modülünün figure ve numpy modülünün polyfit fonksiyonu ile görselleştirdim.

```
data = pd.concat([data5, data6, data7, data8])
# Scatter plot of Views vs. Likes
plt.figure(figsize=(10, 6))
plt.scatter(data['viewCount'], data['likeCount'], color='skyblue', alpha=0.7,
            edgecolors='black')
plt.title('Relationship between Views and Likes')
plt.xlabel('Views')
plt.ylabel('Likes')
plt.grid(True, linestyle='--', alpha=0.7)

# Adding a trendline
z = np.polyfit(data['viewCount'], data['likeCount'], 1)
p = np.poly1d(z)
plt.plot(data['viewCount'], p(data['likeCount']), color="red")

plt.show()
```



Tablo 21

Burada Youtubedaki videoların izlenme ve beğeni sayısı ile ilgili bağıntıyı yorumlamak için bir grafik oluşturdum. İzlenme sayısı azaldıkça beğeni sayısı azalmış ve yoğunluk az kesimde. İzlenme arttıkça beğeni sayısı da artmış durumda. Ama yüksek kesimde yoğunluk az.

KAYNAKÇA VE EKLER

[YouTube Data API | Google for Developers](#)

[Web API | Spotify for Developers](#)

[Colab | Google Notebook Service](#)

VERİYİ ELDE ETTİĞİM KODLAR

Veriyi elde ettiğim kodlar javascript ve python dilinde olup çok fazla kod olduğu için bir github sayfası oluşturdum. Github sayfasında javascript klasörünün içinde veriyi elde edip düzenlediğim kodlar bulunmakta. (Javascript/main.js, Javascript/youtube.js, Javascript/spotify.js)

Ayrıca veriyi json halinde kaydedip sonrasında py koduyla csv dosyasına çevirdim.

(Javascript/main.js, Javascript/savejson.js) (Python/JSONtoCSV/main.py)

Buradaki kodlarla (Javascript/JSON) klasörünün içindeki json dosyalarını csv dosyasına dönüştürdüm.

Sonrasına hem üstte sunumda belirttiğim kodlarla üstteki tabloları hazırlayıp yorumladım hem de aşağıdaki github sayfasına colabdan hazırladığım kodları paylaştım. (Python/Colab/IST204Odev.ipynb)

[bymayfe/YoutubeSpotifyStatsVirtualization \(github.com\)](#)