

Final Projesi İlerleme Raporu

EYON407 - Veri Madenciliği

Dersin Öğretim Üyesi: Dr. Nesibe Manav Mutlu

Proje Üyesi: Yasin Top

Proje Konu Başlığı: Sürdürülebilirlik Amaçlarında Nükleer Enerjinin Konumu

Veri Kaynağı ve Adı: Kaggle- World Energy Consumption: Data on Energy by Our World in Dat

(<https://www.kaggle.com/datasets/pralabhpoudel/world-energy-consumption?resource=download>)

Gönderme Tarihi: 18.11.2022

Özet:

Zaman çizelgesi ve kısaca yapılacak görevler

Hafta 1-2 - (20 - 26 Ekim, 2022)

- Grup üyelerinin belirlenmesi.
- Sürdürülebilirliğin tanımı, kapsamı, alt başlıkları, etki ettiği alanlar.

Hafta 3-4 - (27 Ekim - 2 Kasım, 2022)

- Sürdürülebilir kalkınma amaçlarına giriş, kısaca amaçlarından bahsetme.
- Sürdürülebilir kalkınma amaçlarından madde seçimi(Yenilenebilir Enerji)
- Seçilen maddenin içerdiği kısımlardan bahsetme(Rüzgar, Güneş, Nükleer Enerji)
- Bu kısımlardan yenilenebilir enerjide nükleer enerji grubun konusu.
- Konunun anlaşılması için kapsamların belirlenmeye çalışılması, dahil olabilecek konuların hızlı bir taramasının yapılması
 - Yenilenebilir Enerji ve Sürdürülebilir Kalkınma İlişkisi
 - Yenilenebilir Enerjinin Sürdürülebilirlik Üzerindeki Rolü
 - Sürdürülebilir Gelişme Gündeminde Nükleer Enerjinin Sorunları
 - Nükleer Enerjinin Konumu

- İçeriğe giriş

Hafta 5 - (2 - 8 Kasım, 2022)

- Hafta 4 e ek olarak aşağıdaki konulardan devam edilmiştir.
 - Dünya enerji üretimi ve tüketimi

Türkiye'nin Enerji Bağımlılığı

Nükleer Enerji Gelecekteki Enerji İhtiyacına Çözüm Olabilir Mi?

- Veri setlerinin belirlenmesi ve indirilmesi:

Wind, Solar and Nuclear Energy Production

(<https://www.kaggle.com/code/vtorreslopez/wind-solar-and-nuclear-energy-production/notebook>)

World Energy Consumption: Data on Energy by Our World in Data

(<https://www.kaggle.com/datasets/pralabhpoudel/world-energy-consumption?resource=download>)

Hafta 6 - (9 - 15 Kasım, 2022)

- Python Pandas kütüphaneleri ile veri setleri üzerinde denemelere başlandı.
- Kütüphaneler kuruldu.
- Değişkenler belirlenmeye başlandı.
- Data setleri veri görselleştirme için hazırlanmaya başlandı.

```
G8 ülkeleri ve Türkiye Enerji Tüketimleri Enerji Kullanılan Maddelerin Payları İncelenmiştir Grup Üyeleri: Yasin Top - 20184029009  
Esmenur Akyel - 20194029011
```

```
[1] import pandas as pd  
import seaborn as sns  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn import tree  
from sklearn import metrics  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import confusion_matrix  
from sklearn import utils  
from sklearn.linear_model import LinearRegression  
from sklearn.linear_model import LogisticRegression  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis  
from sklearn.naive_bayes import GaussianNB  
from sklearn.svm import SVC  
from sklearn.model_selection import cross_val_score  
from sklearn.cluster import KMeans  
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

Python

```
[2] dataset=pd.read_excel("G8_and_Turkey_Dataset1234.xlsx")
```

```
[3] dataset5=dataset[["population","coal_share_energy","coal_consumption","renewables_share_energy","renewables_consumption"]]
```

Python

Veri seti tanımlandı.

```
dataf=pd.DataFrame(dataset)

print(dataf)
```

[4] Python

... Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

	iso_code	country	year	population	gdp \
0	TUR	Turkey	2009	71321000	1130000000000
1	TUR	Turkey	2010	72327000	1250000000000
2	TUR	Turkey	2011	73443000	1400000000000
3	TUR	Turkey	2012	74651000	1490000000000
4	TUR	Turkey	2013	75925000	1560000000000
..
67	GBR	United Kingdom	2012	64525000	2290000000000
68	GBR	United Kingdom	2013	64984000	2430000000000
69	GBR	United Kingdom	2014	65423000	2480000000000
70	GBR	United Kingdom	2015	65860000	2530000000000
71	GBR	United Kingdom	2016	66298000	2580000000000

	coal_share_energy	coal_consumption	hydro_electricity \
0	30.254	359.344	35.598
1	29.241	365.624	51.423
2	29.488	394.011	51.155
3	29.924	424.437	56.669
4	26.082	367.092	58.225
..
67	19.069	453.082	5.310
68	18.186	429.998	4.701
69	15.532	346.043	5.888
70	11.911	268.353	6.297
71	5.790	128.894	5.394
...
70	19.056	12.010	4.522
71	26.174	11.097	4.197

[72 rows x 24 columns]

Veri seti gösterildi.

```
dataset.tail(10)
```

[6] Python

...

Kaç adet satır kaç adet sütun olduğu hesaplandı.

```
dataset.dtypes

[8] Python

... iso_code          object
     country          object
     year             int64
     population        int64
     gdp               int64
     coal_share_energy float64
     coal_consumption  float64
     hydro_electricity float64
     nuclear_electricity float64
     renewables_electricity float64
     solar_electricity float64
     wind_electricity  float64
     hydro_share_elec  float64
     hydro_share_energy float64
     nuclear_share_energy float64
     oil_share_energy  float64
     renewables_share_elec float64
     renewables_share_energy float64
     renewables_consumption float64
     solar_share_elec  float64
     solar_share_energy float64
     solar_consumption float64
     wind_share_elec   float64
     wind_share_energy float64
     dtype: object
```

Verilerin türleri bulundu

```
dataset.isnull().sum

[11] Python

... Output exceeds the size limit. Open the full output data in a text editor
<bound method NDFrame._add_numeric_operations.<locals>.sum of iso_code country year population gdp
coal_share_energy \
0 False False False False False False
1 False False False False False False
2 False False False False False False
3 False False False False False False
4 False False False False False False
.. ... ..
67 False False False False False False
68 False False False False False False
69 False False False False False False
70 False False False False False False
71 False False False False False False

coal_consumption hydro_electricity nuclear_electricity \
0 False False False
1 False False False
2 False False False
3 False False False
4 False False False
.. ... ..
67 False False False
68 False False False
69 False False False
70 False False False
71 False False False

...
69 False
70 False
71 False

[72 rows x 24 columns]>
```

Boş değerler gösterildi True= boş değer, False = boş değer değil.

[17432 rows x 122 columns]>

In [9]: vs.dropna(axis="columns",how="all")

Out[9]:

	iso_code	country	year	coal_prod_change_pct	coal_prod_change_twh	gas_prod_change_pct	gas_prod_change_twh	oil_prod_change_pct	oil_prod_c
0	AFG	Afghanistan	1900	NaN	NaN	NaN	NaN	NaN	
1	AFG	Afghanistan	1901	NaN	0.000	NaN	NaN	NaN	
2	AFG	Afghanistan	1902	NaN	0.000	NaN	NaN	NaN	
3	AFG	Afghanistan	1903	NaN	0.000	NaN	NaN	NaN	
4	AFG	Afghanistan	1904	NaN	0.000	NaN	NaN	NaN	
...
17427	ZWE	Zimbabwe	2015	-25.013	-10.847	NaN	NaN	NaN	
17428	ZWE	Zimbabwe	2016	-37.694	-12.257	NaN	NaN	NaN	
17429	ZWE	Zimbabwe	2017	8.375	1.697	NaN	NaN	NaN	
17430	ZWE	Zimbabwe	2018	22.555	4.952	NaN	NaN	NaN	
17431	ZWE	Zimbabwe	2019	-35.015	-9.422	NaN	NaN	NaN	

17432 rows x 122 columns

In []:

Sütünlardan tüm indisleri boş olan veriler silinmeye çalışıldı. Tüm verileri boş olan sütunumuz olmadığı için herhangi bir sütunumuz silinmedi.

In [12]: vs.dropna(axis="columns",how="any")

Out[12]:

	country	year
0	Afghanistan	1900
1	Afghanistan	1901
2	Afghanistan	1902
3	Afghanistan	1903
4	Afghanistan	1904
...
17427	Zimbabwe	2015
17428	Zimbabwe	2016
17429	Zimbabwe	2017
17430	Zimbabwe	2018
17431	Zimbabwe	2019

17432 rows x 2 columns

In []:

Herhangi bir sütunu boş olan her bir sütunu sildik.
Herhangi boş veri barındırmayan sadece iki sütunumuzun olduğunu gördük.

```
In [8]: vs.dropna(axis="index",how="any")
```

Out[8]:

	iso_code	country	year	coal_prod_change_pct	coal_prod_change_twh	gas_prod_change_pct	gas_prod_change_twh	oil_prod_change_pct	oil_prod_cha
16231	GBR	United Kingdom	2003	-6.230	-13.628	-0.627	-6.803	-8.514	
16232	GBR	United Kingdom	2004	-11.579	-23.750	-6.394	-68.930	-10.086	
16233	GBR	United Kingdom	2005	-18.468	-33.493	-8.497	-85.741	-11.170	
16234	GBR	United Kingdom	2006	-10.193	-15.071	-9.303	-85.906	-9.612	
16235	GBR	United Kingdom	2007	-6.318	-8.389	-9.858	-82.556	-0.004	
16236	GBR	United Kingdom	2008	5.690	7.078	-3.606	-27.223	-6.250	
16237	GBR	United Kingdom	2009	-2.358	-3.100	-15.892	-115.648	-5.001	
16238	GBR	United Kingdom	2010	3.501	4.495	-5.400	-33.048	-7.679	
16239	GBR	United Kingdom	2011	0.935	1.242	-20.411	-118.184	-17.454	
16240	GBR	United Kingdom	2012	-8.231	-11.039	-14.952	-68.901	-14.261	
16241	GBR	United Kingdom	2013	-24.658	-30.349	-5.644	-22.120	-8.784	
16242	GBR	United Kingdom	2014	-8.581	-7.957	1.220	4.511	-1.768	
16243	GBR	United Kingdom	2015	-26.135	-22.155	8.628	32.297	13.424	
16244	GBR	United Kingdom	2016	-51.098	-31.996	2.649	10.770	4.763	

14 rows x 122 columns

Index'inde boş eleman olanları çıkarınca geriye 14 sütun kaldığını görüyoruz.

Hafta 7 - (16 - 22 Kasım, 2022)

```
In [9]: vs.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 17432 entries, 0 to 17431  
Columns: 122 entries, iso_code to wind_energy_per_capita  
dtypes: float64(119), int64(1), object(2)  
memory usage: 16.2+ MB
```

In []:

Dataframe hakkında bilgi aldık

Bilgiler sütun sayılarını, sütun etiketlerini, sütun veri türlerini, bellek kullanımını, aralık dizinini ve her sütundaki boş olmayan hücre sayılarını içerir

```
In [12]: vs.describe().T
```

```
Out[12]:
```

	count	mean	std	min	25%	50%	75%	max
year	17432.0	1.973094e+03	3.433400e+01	1900.000	1946.000	1983.0000	2002.00000	2.020000e+03
coal_prod_change_pct	7445.0	2.083077e+01	6.971787e+02	-100.000	-1.532	0.0000	7.69000	4.496575e+04
coal_prod_change_twh	10394.0	8.798102e+00	1.355037e+02	-2326.870	0.000	0.0000	0.33400	3.060593e+03
gas_prod_change_pct	4862.0	1.921623e+14	1.339910e+16	-100.000	0.000	2.5835	9.70350	9.342930e+17
gas_prod_change_twh	7893.0	1.436902e+01	8.541565e+01	-1054.320	0.000	0.0000	2.55900	2.112975e+03
...
wind_share_energy	4284.0	3.454057e-01	1.337962e+00	0.000	0.000	0.0000	0.02000	2.066000e+01
wind_cons_change_twh	4207.0	2.163830e+00	1.701315e+01	-10.409	0.000	0.0000	0.01000	4.287360e+02
wind_consumption	4290.0	1.508094e+01	1.266741e+02	0.000	0.000	0.0000	0.14100	3.540051e+03
wind_elec_per_capita	5499.0	5.362578e+01	2.095435e+02	0.000	0.000	0.0000	3.04650	2.825425e+03
wind_energy_per_capita	4290.0	1.340031e+02	5.133680e+02	0.000	0.000	0.0000	4.74475	6.928363e+03

120 rows x 8 columns

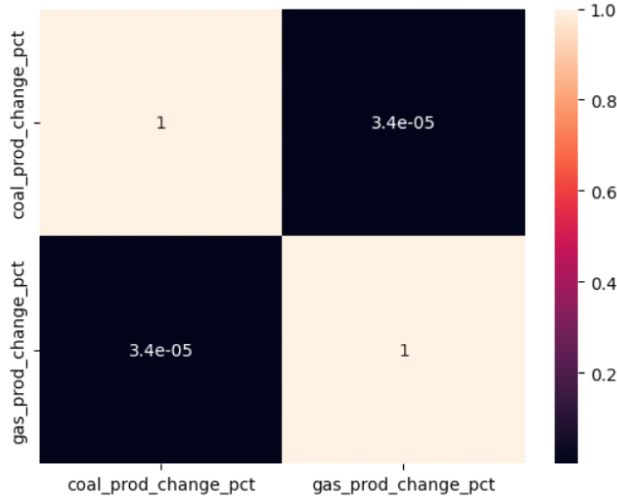
```
In [ ]:
```

DataFrame'in sayısal değerlerinin yüzdelik, ortalama ve std (kartilleri) gibi bazı istatistiksel verileri hesaplamak için kullanılır. Ülke gibi kategorik değerleri işleme almaz.

¶

```
In [18]: import seaborn as sns
dataaboutheat=vs[["coal_prod_change_pct","gas_prod_change_pct"]]
sns.heatmap((dataaboutheat.corr()),annot=True)
```

```
Out[18]: <AxesSubplot:>
```



```
In [ ]:
```

Daha fazla görselleştirme için Seaborn kütüphanesini çalışmaya aktardık.

İki değişkenin aralarındaki ısı haritasını çıkardık. Korelasyon değeri olarak alışmadık bir değer çıktı. Sebebi araştırıyoruz.

Verilerimizi Kırılgan Beşli birliğindeki ülkelere göre filitrelendi. (Brezilya, Hindistan, Endonezya, Türkiye, Güney Afrika)

Hafta 8-9 (23 Kasım- 7 Aralık, 2022 -Ara Sınavlar)

Hafta 10 - (8 - 14 Aralık, 2022)

Hafta 11 - (15 - 21 Aralık, 2022)

Verilerin fazlalığından dolayı hem karışık hem de boş tablolarla karşılaşmaya başladık bu sebepten veri setimizi kısıtladık. **Tablo baştan değişmiştir. Ülke filtremiz düzeltildi. G8 ülkeleri ve Türkiye verileridir.,**

```
In [2]: dataset=pd.read_excel("C:\G8_and_Turkey_Dataset.xlsx")

In [4]: pd.DataFrame(dataset)

Out[4]:
```

	iso_code	country	year	population	gdp	coal_share_energy	coal_consumption	hydro_electricity	nuclear_electricity	renewables_electricity	...
0	TUR	Turkey	2009	71321000	1130000000000	30.254	359.344	35.598	0.000	37.944	...
1	TUR	Turkey	2010	72327000	1250000000000	29.241	365.624	51.423	0.000	55.353	...
2	TUR	Turkey	2011	73443000	1400000000000	29.488	394.011	51.155	0.000	56.927	...
3	TUR	Turkey	2012	74651000	1490000000000	29.924	424.437	56.669	0.000	63.993	...
4	TUR	Turkey	2013	75925000	1560000000000	26.082	367.092	58.225	0.000	68.087	...
...
67	GBR	United Kingdom	2012	64525000	2290000000000	19.069	453.082	5.310	70.405	41.249	...
68	GBR	United Kingdom	2013	64984000	2430000000000	18.186	429.998	4.701	70.607	53.214	...
69	GBR	United Kingdom	2014	65423000	2480000000000	15.532	346.043	5.888	63.748	64.523	...
70	GBR	United Kingdom	2015	65860000	2530000000000	11.911	268.353	6.297	70.345	82.576	...
71	GBR	United Kingdom	2016	66298000	2580000000000	5.790	128.894	5.394	71.726	81.608	...

72 rows x 24 columns

**2019 yılından sonraki dönemde pandemi etkisini gösterdiği için, ayrıca sütunlarda 2017 yılından sonraki GDP verileri boş olduğundan dolayı 2017 yılından sonraki veriler tablodan çıkarılmıştır. Lejant oluşturulmuştur.*



Lejant Açıklamaları.txt

Metin.1

Metin 1 de tablodaki sütunların açıklamaları yer alıyor.

Hafta 12 - (22 - 28 Aralık, 2022)

```
In [7]: dataset.dtypes
```

```
Out[7]: iso_code      object
country    object
year       int64
population int64
gdp        int64
coal_share_energy float64
coal_consumption float64
hydro_electricity float64
nuclear_electricity float64
renewables_electricity float64
solar_electricity float64
wind_electricity float64
hydro_share_elec float64
hydro_share_energy float64
nuclear_share_energy float64
oil_share_energy float64
renewables_share_elec float64
renewables_share_energy float64
renewables_consumption float64
solar_share_elec float64
solar_share_energy float64
solar_consumption float64
wind_share_elec float64
wind_share_energy float64
dtype: object
```

Tablo kategorik (nominal) ve numerik (tam sayı ve ondalıklı sayı) türlerinden oluşuyor.

```
In [12]: dataset.dropna(axis="index", how="any")
```

```
Out[12]:
```

	iso_code	country	year	population	gdp	coal_share_energy	coal_consumption	hydro_electricity	nuclear_electricity	renewables_electricity	...
0	TUR	Turkey	2009	71321000	1130000000000	30.254	359.344	35.598	0.000	37.944	...
1	TUR	Turkey	2010	72327000	1250000000000	29.241	365.624	51.423	0.000	55.353	...
2	TUR	Turkey	2011	73443000	1400000000000	29.488	394.011	51.155	0.000	56.927	...
3	TUR	Turkey	2012	74651000	1490000000000	29.924	424.437	56.669	0.000	63.993	...
4	TUR	Turkey	2013	75925000	1560000000000	26.082	367.092	58.225	0.000	68.087	...
...
67	GBR	United Kingdom	2012	64525000	2290000000000	19.069	453.082	5.310	70.405	41.249	...
68	GBR	United Kingdom	2013	64984000	2430000000000	18.186	429.998	4.701	70.607	53.214	...
69	GBR	United Kingdom	2014	65423000	2480000000000	15.532	346.043	5.888	63.748	64.523	...
70	GBR	United Kingdom	2015	65860000	2530000000000	11.911	268.353	6.297	70.345	82.576	...
71	GBR	United Kingdom	2016	66298000	2580000000000	5.790	128.894	5.394	71.726	81.608	...

72 rows × 24 columns

Tablodaki az sayıda null değerindeki elemanları sıfır (0)ile doldurduk.

Boş hücremiz yoktur.

```
In [35]: dataset.groupby("year")["coal_share_energy"].mean()
```

```
Out[35]: year
2009    16.068333
2010    16.297222
2011    16.420000
2012    16.989778
2013    16.402000
2014    16.246444
2015    15.182667
2016    14.197889
Name: coal_share_energy, dtype: float64
```

Ülkeleri tek olarak alıp daha ayrıntılı şekilde ilişkilerine bakalım:

Türkiye için yıllara göre kömür tüketimine (coal_consumption) ve kömürün paylarına (coal_share_energy) bakalım :

(coal_share_energy - Kömürün payı : Enerji üretilirken kullanılan kömürün enerji üretiminde kullanılan diğer tüm ham maddelere oranını ifade eder.)

```
In [48]: turkiye = dataset[dataset["country"] == "Turkey"]
```

```
In [54]: print(turkiye)|
```

	iso_code	country	year	population	gdp	coal_share_energy	\
0	TUR	Turkey	2009	71321000	1130000000000	30.254	
1	TUR	Turkey	2010	72327000	1250000000000	29.241	
2	TUR	Turkey	2011	73443000	1400000000000	29.488	
3	TUR	Turkey	2012	74651000	1490000000000	29.924	
4	TUR	Turkey	2013	75925000	1560000000000	26.082	
5	TUR	Turkey	2014	77229000	1650000000000	28.910	
6	TUR	Turkey	2015	78529000	1720000000000	25.414	
7	TUR	Turkey	2016	79828000	1570000000000	26.788	

	coal_consumption	hydro_electricity	nuclear_electricity	\
0	359.344	35.598	0.0	
1	365.624	51.423	0.0	
2	394.011	51.155	0.0	
3	424.437	56.669	0.0	
4	367.092	58.225	0.0	
5	420.146	39.750	0.0	
-	

```
In [58]: turkiye.groupby("year")["coal_consumption"].mean()
```

```
Out[58]: year
2009    359.344
2010    365.624
2011    394.011
2012    424.437
2013    367.092
2014    420.146
2015    404.026
2016    447.261
Name: coal_consumption, dtype: float64
```

```
In [59]: turkiye.groupby("year")["coal_share_energy"].mean()
```

```
Out[59]: year
2009    30.254
2010    29.241
2011    29.488
2012    29.924
2013    26.082
2014    28.910
2015    25.414
2016    26.788
Name: coal_share_energy, dtype: float64
```

```
In [ ]:
```

Türkiye’de yıllara göre kömür tüketim filtresiyle değişken oluşturalım .
Türkiye’de yıllara göre üretilen enerjide kömürün payı filtresiyle değişken oluşturalım .

```

In [ ]:

In [68]: yillaragoretuketim=turkiye.groupby("year")["coal_consumption"].mean()

In [69]: print(yillaragoretuketim)

year
2009    359.344
2010    365.624
2011    394.011
2012    424.437
2013    367.092
2014    420.146
2015    404.026
2016    447.261
Name: coal_consumption, dtype: float64

In [71]: yillaragorekomurunpayi=turkiye.groupby("year")["coal_share_energy"].mean()

In [72]: print(yillaragorekomurunpayi)

year
2009     30.254
2010     29.241
2011     29.488
2012     29.924
2013     26.082
2014     28.910
2015     25.414
2016     26.788
Name: coal_share_energy, dtype: float64

```

Türkiye’de kömür tüketimi düzenli olarak artmış, ama enerji üretiminde kömürün payı azalmıştır.
Sebebi;

```

In [77]: turkiye.groupby("year")["wind_share_elec", "hydro_share_elec", "solar_share_elec"].mean()
Out[77]:

```

year	wind_share_elec	hydro_share_elec	solar_share_elec
2009	0.808	19.228	0.000
2010	1.444	25.533	0.000
2011	2.165	23.484	0.000
2012	2.564	24.888	0.000
2013	3.298	25.483	0.000
2014	3.554	16.649	0.007
2015	4.650	26.422	0.077
2016	5.874	25.467	0.387

Tüm Dünya’da olduğu gibi Türkiye’de de *yenilenebilir enerjinin payının artmasıdır.*

Diğer ülkelere de bakalım:

```

In [79]: dataset.groupby("year")["wind_share_elec", "hydro_share_elec", "solar_share_elec"].mean()
Out[79]:

```

year	wind_share_elec	hydro_share_elec	solar_share_elec
2009	1.899556	16.174556	0.189444
2010	2.141222	16.439556	0.335444
2011	2.801444	15.944000	0.875444
2012	3.256333	16.214778	1.404667
2013	3.891556	17.089667	1.730000
2014	4.386667	16.112222	2.166111
2015	5.292667	16.383667	2.532111
2016	5.586889	16.337222	2.805778

Beklenildiği üzere düzenli artış.

```
In [80]: almanya = dataset[dataset["country"] == "Germany"]
```

```
In [82]: almanya.groupby("year")[["population", "wind_share_elec", "hydro_share_elec", "solar_share_elec"]].mean()
```

```
Out[82]:
```

	population	wind_share_elec	hydro_share_elec	solar_share_elec
year				
2009	80900000.0	6.691	3.215	1.117
2010	80827000.0	6.158	3.350	1.867
2011	80856000.0	8.214	2.914	3.226
2012	80973000.0	8.281	3.499	4.237
2013	81174000.0	8.327	3.634	4.898
2014	81450000.0	9.405	3.151	5.804
2015	81787000.0	12.547	2.958	6.024
2016	82194000.0	12.407	3.179	5.909

```
In [85]: almanya.groupby("year")[["population", "renewables_share_energy"]].mean()
```

```
Out[85]:
```

	population	renewables_share_energy
year		
2009	80900000.0	7.705
2010	80827000.0	8.044
2011	80856000.0	9.675
2012	80973000.0	10.901
2013	81174000.0	11.048
2014	81450000.0	12.221
2015	81787000.0	13.671
2016	82194000.0	13.436

```
In [ ]:
```

82.Satırda yine yıllara göre üç çeşit yenilenebilir enerjinin toplam enerjideki paylarına ve ek olarak nüfusa bakıyoruz. Bu kez Almanya için:

Her yenilenebilir enerji için ayrı ayrı sentez yapmak yerine:

veri setimizdeki "renewables_share_energy" sütununu kullanabiliriz. Bu sütun yenilenebilir enerjilerin ortalamasıdır.

Genel itibariyle şu ana kadar :

Nüfus arttıkça (yıl ilerledikçe) ülkelerin sonlu hacimlere sahip kirli enerji kaynaklarını azaltmaya; temiz enerji kaynaklarını daha fazla etkin kullanmaya ve paylarını arttırmaya çalıştığını görebiliyoruz.

Hafta 13 - (29 Aralık - 4 Ocak, 2023)

```
In [27]: dataset.groupby("year")["coal_consumption"].mean()
```

```
Out[27]: year
2009    1091.519333
2010    1156.561889
2011    1118.893333
2012    1078.663444
2013    1086.433000
2014    1065.400667
2015     984.178222
2016     922.175222
Name: coal_consumption, dtype: float64
```

```
In [28]: dataset2=dataset.groupby("country")[["gdp","coal_share_energy","coal_consumption","renewables_share_energy","renewables_consumption"]
print(dataset2)
```

country	gdp	coal_share_energy	coal_consumption
Germany	3.606250e+12	24.383375	908.952625
Italy	2.087500e+12	8.302875	155.729250
Japan	4.491250e+12	24.645875	1345.628375
Russia	3.213750e+12	13.552875	1067.894375
Turkey	1.471250e+12	28.262625	397.742625
United Kingdom	2.371250e+12	14.357125	337.287625
United States	1.603750e+13	19.581375	4989.671500

country	renewables_share_energy	renewables_consumption
Canada	27.930125	1053.679875
France	8.705625	245.223750
Germany	10.837625	404.173375
Italy	14.067875	261.004375
Japan	5.830250	317.595500
Russia	5.622375	442.865125
Turkey	11.483500	164.206375
United Kingdom	6.233750	143.639500
United States	6.581125	1677.809000

```
In [ ]:
```

Dataset2 değişkeninde ülkelerin istenilen özelliklerinin ortalamaları alındı. Sonrasında da sütunlar Türkçeleştirildi.

```
7- 2021:
```

Dataset2 değişkeninde ülkelerin istenilen özelliklerinin ortalamaları alındı. Sonrasında da sütunlar Türkçeleştirildi.

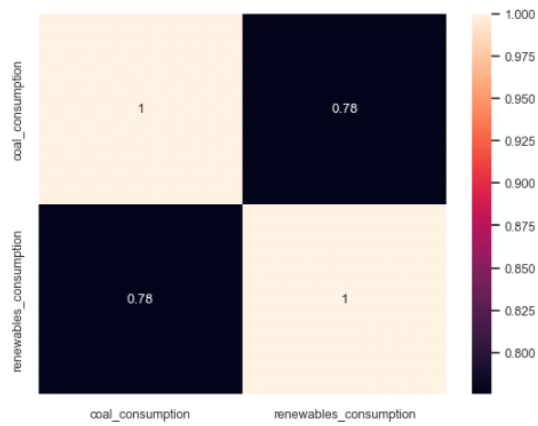
```
In [29]: dataset2.rename(columns ={'gdp':'yurtici_milli_hasıla','coal_share_energy':'kömürün_enerji_payı','coal_consumption':'kömür_tüketimi'})
```

```
Out[29]:
```

country	yurtici_milli_hasıla	kömürün_enerji_payı	kömür_tüketimi	yenilenebilir_enerjideki_oranı	yenilenebilir_enerji_tüketimi
Canada	1.467500e+12	6.594375	247.830750	27.930125	1053.679875
France	2.450000e+12	4.099375	116.066125	8.705625	245.223750
Germany	3.606250e+12	24.383375	908.952625	10.837625	404.173375
Italy	2.087500e+12	8.302875	155.729250	14.067875	261.004375
Japan	4.491250e+12	24.645875	1345.628375	5.830250	317.595500
Russia	3.213750e+12	13.552875	1067.894375	5.622375	442.865125
Turkey	1.471250e+12	28.262625	397.742625	11.483500	164.206375
United Kingdom	2.371250e+12	14.357125	337.287625	6.233750	143.639500
United States	1.603750e+13	19.581375	4989.671500	6.581125	1677.809000

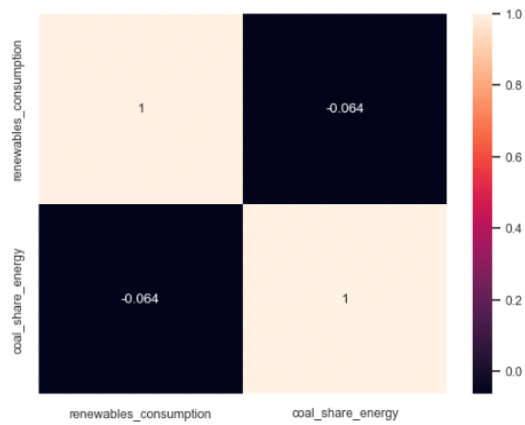
```
In [30]: korelasyon=dataset[["coal_consumption","renewables_consumption"]]  
sns.set(font_scale=0.8)  
sns.heatmap(korelasyon.corr(),annot=True)
```

Out[30]: <AxesSubplot:>



```
In [31]: korelasyon=dataset[["renewables_consumption","coal_share_energy"]]  
sns.set(font_scale=0.8)  
sns.heatmap(korelasyon.corr(),annot=True)
```

Out[31]: <AxesSubplot:>

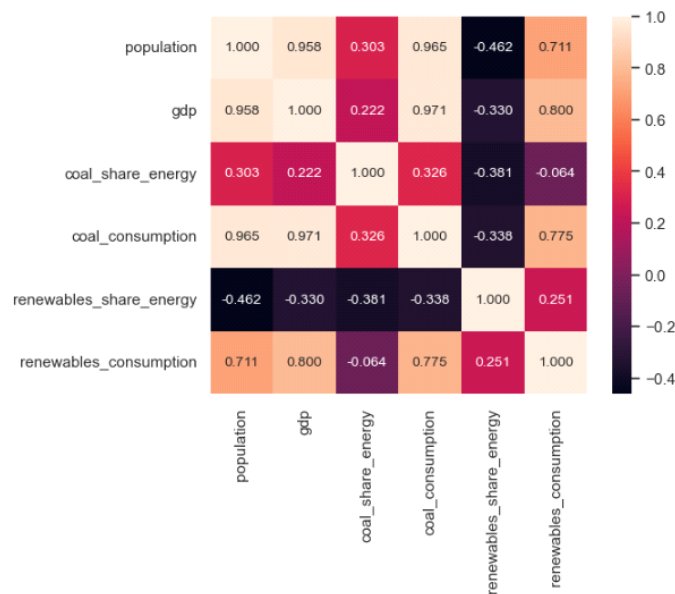


Daha fazla ilişkiye sahip bir ısı haritası oluşturalım, tablomuzu tekrar getirip karşılaştıralım.

```
In [32]: dataset
```

```
In [33]: cols=['population', 'gdp', 'coal_share_energy', 'coal_consumption', 'renewables_share_energy', 'renewables_consumption']
cm= np.corrcoef(dataset[cols].values.T)
sns.set(font_scale=1.0)
hm=sns.heatmap(cm,
               cbar=True,
               annot=True,
               square=True,
               fmt='.3f',
               annot_kws={'size':10},
               yticklabels=cols,
               xticklabels=cols)

plt.show()
```



```
plt.figure(figsize=(10, 5))
sns.heatmap(dataset5.corr().abs(), annot=True, annot_kws={'size':10})
sns.pairplot(dataset5, kind="reg")

sns.jointplot(x="coal_consumption",y="coal_share_energy",data=dataset5, kind="reg", color="purple")

sns.jointplot(x="renewables_consumption",y="renewables_share_energy",data=dataset5, kind="reg", color="green")

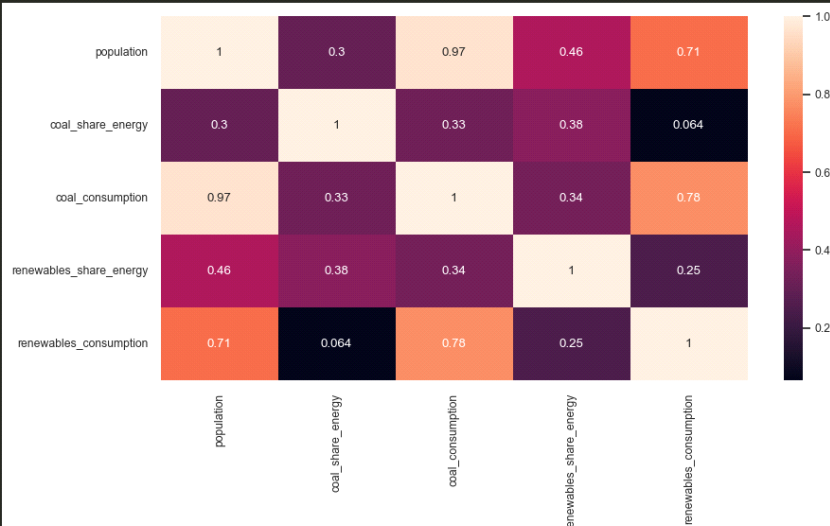
sns.jointplot(x="coal_consumption",y="renewables_share_energy",data=dataset5, kind="reg", color="red")
```

[30]

Python

... <seaborn.axisgrid.JointGrid at 0x1c3feae8970>

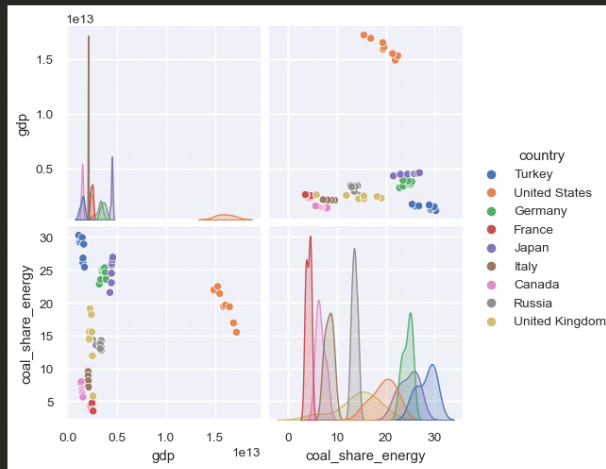
</>



```
datasetpair2=dataset[["country","gdp","coal_share_energy"]]  
sns.set(font_scale=0.9)  
sns.pairplot(datasetpair2,hue="country",diag_kind="kde")  
plt.show()
```

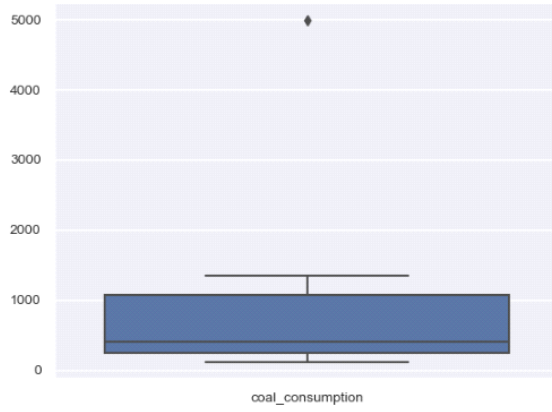
[31]

Python



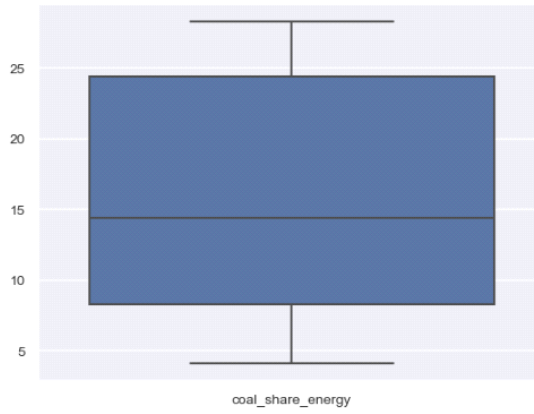

```
In [35]: sns.boxplot(data=dataset2[["coal_consumption"]])
```

```
Out[35]: <AxesSubplot:>
```



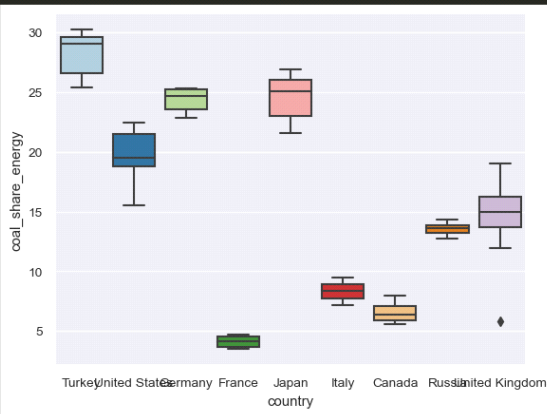
```
In [36]: sns.boxplot(data=dataset2[["coal_share_energy"]])
```

```
Out[36]: <AxesSubplot:>
```



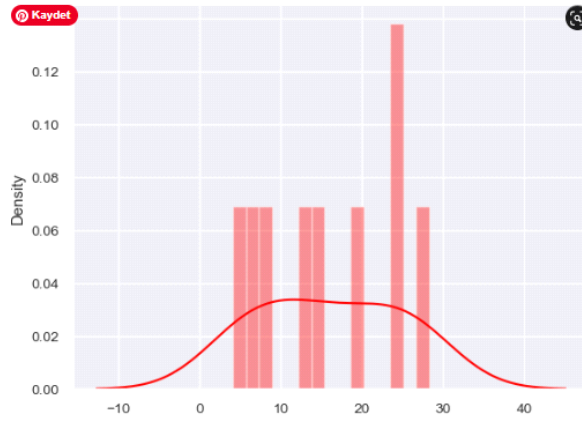
distplot ile görselleştirilim

```
sns.boxplot(x=dataf["country"], y=dataf["coal_share_energy"], palette="Paired")  
plt.show()
```



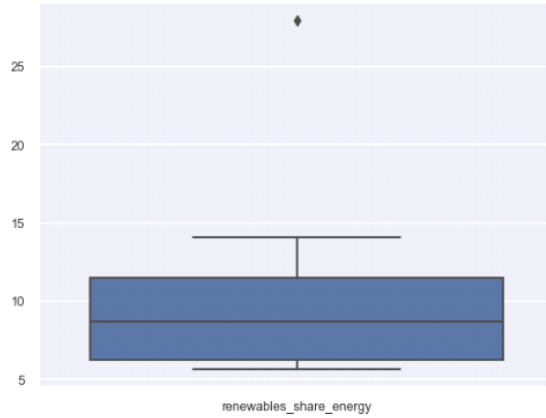
```
In [37]: sns.distplot(dataset2[["coal_share_energy"]],bins=15,color="red")
sns.set(font_scale=0.8)
```

C:\Users\steam\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)



```
In [38]: sns.boxplot(data=dataset2[["renewables_share_energy"]])
```

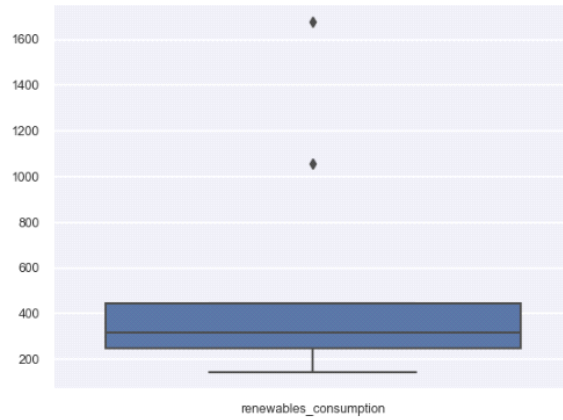
Out[38]: <AxesSubplot:>



Bir ülkenin enerji üretiminde yenilenebilir enerjiyi diğer ülkelerden açık ara daha fazla kullandığını görüyoruz.

Bir ülkenin enerji üretiminde yenilenebilir enerjiyi diğer ülkelerden açık ara daha fazla kullandığını görüyoruz.

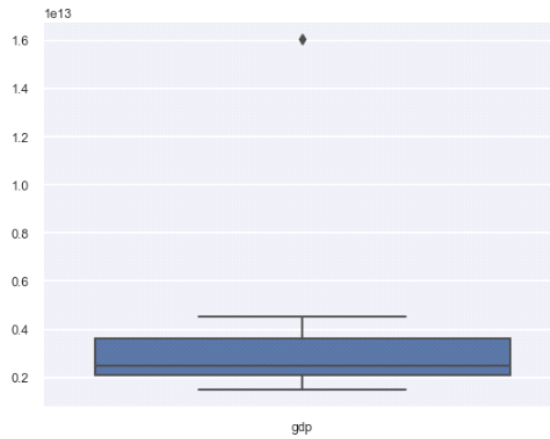
```
In [39]: sns.boxplot(data=dataset2[["renewables_consumption"]])
sns.set(font_scale=0.8)
```



İki ülke yenilenebilir enerjiyi daha fazla kullanmıştır.

```
In [40]: sns.boxplot(data=dataset2[["gdp"]])
```

Out[40]: <AxesSubplot:>

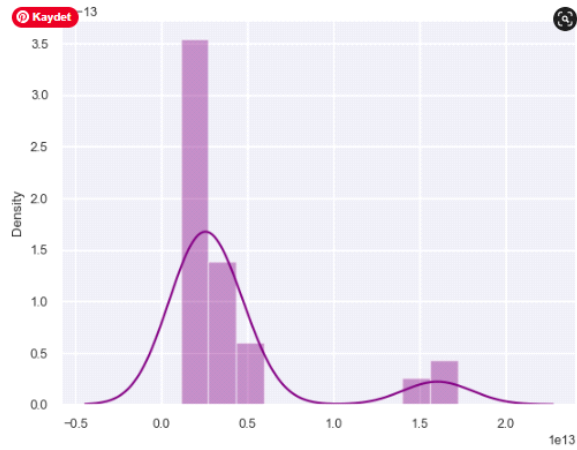


gdp

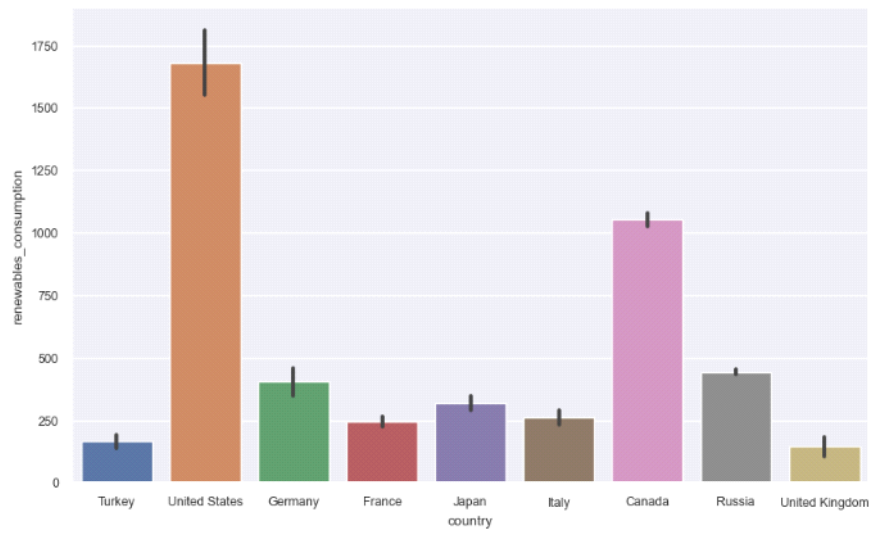
```
In [41]: sns.distplot(dataset[["gdp"]],bins=10,color="purple")
```

C:\Users\steam\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

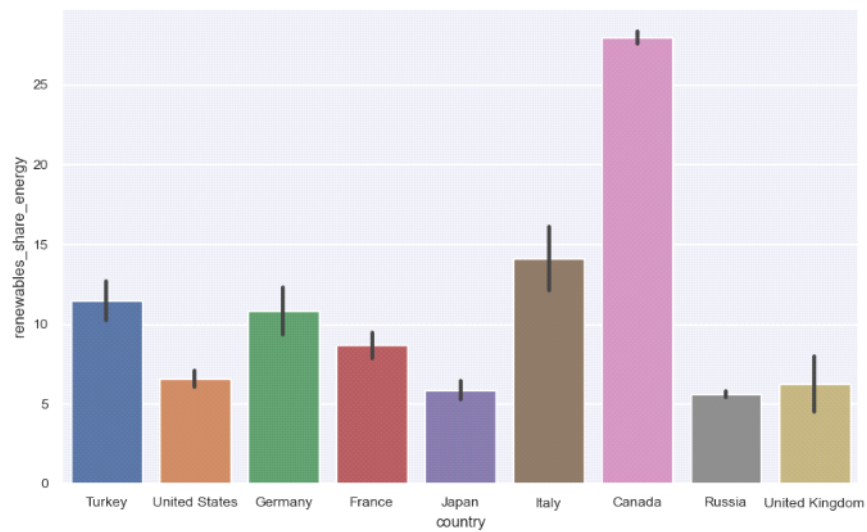
```
Out[41]: <AxesSubplot:ylabel='Density'>
```



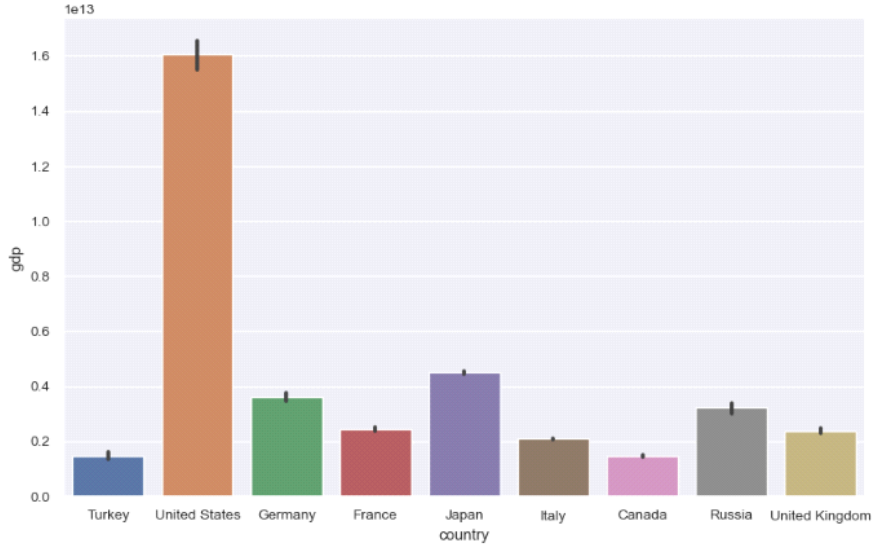
```
In [42]: plt.figure(figsize=(10,6))
sns.barplot(x="country",y="renewables_consumption",data=dataset)
sns.set(font_scale=0.9)
plt.show()
```



```
In [43]: plt.figure(figsize=(10,6))
sns.barplot(x="country",y="renewables_share_energy",data=dataset)
sns.set(font_scale=0.9)
plt.show()
```



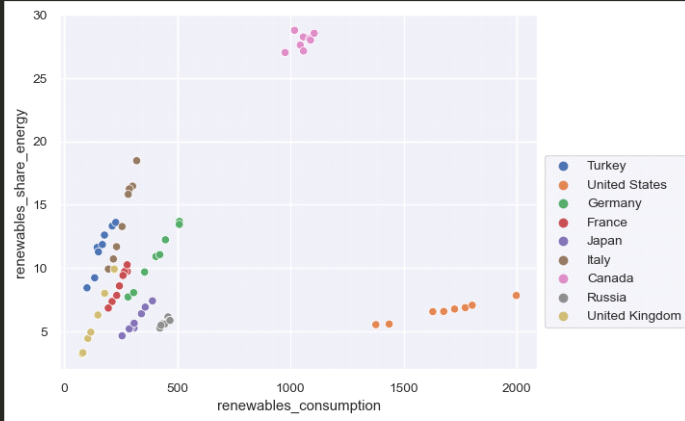
```
In [44]: plt.figure(figsize=(10,6))
sns.barplot(x="country",y="gdp",data=dataset)
sns.set(font_scale=0.9)
plt.show()
```



Miktar olarak yenilenebilir enerjiyi en fazla kullanan ülke Amerika, ama enerji üretiminde yenilenebilir enerjinin payı en yüksek ülke Kanada. Bu karşılaştırmayı bir de saçılım grafiği ile gözlemleyelim:

Vilları geçtikçe kömür üretimi de yenilenebilir enerjiler de artmış ama kömürün payı azalmıştır çünkü ; enerji üretimi payında yenilenebilir enerji her sene daha fazla yer edinmiştir.

```
sns.scatterplot(x="renewables_consumption", y="renewables_share_energy", data=dataset, hue=dataset.country)
plt.legend(loc='lower left', bbox_to_anchor=(1,0.1))
sns.set(font_scale=1.3)
plt.show()
```



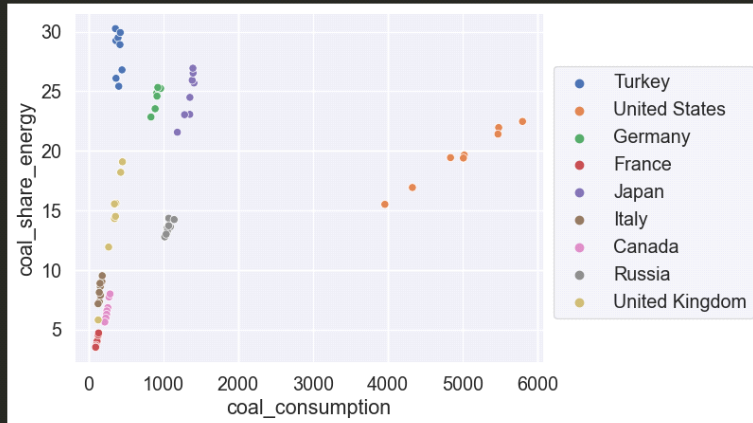
barplot grafiği ile paralel bir görüntü var. Yani yenilenebilir enerji tüketimi en fazla olan ülke Amerika ama yenilenebilir enerjiyi diğer enerjilerden daha üstün bir payda tutmamış. Hata diğer ülkelere göre bu enerjinin payları diğer ülkelerin yenilenebilir enerji payından daha az olmuş. Kanada ise yenilenebilir enerjinin diğer enerjilere oranı özelliğini (diğer alternatif enerji kaynaklarına oranla bu enerji türünün payını) tüm ülkelerden daha fazla paya yerleştirmiştir, yani oldukça ön planda tutmuştur, ve tüketim olarak da neredeyse en çok yenilenebilir enerji tüketen ülke konumunda olmuştur.

```
sns.scatterplot(x="coal_consumption", y="coal_share_energy", data= dataset, hue=dataset.country)
plt.legend(loc='lower left', bbox_to_anchor=(1,0.1))
sns.set(font_scale=1.3)
plt.show()
```

[42]

Python

...



K-Means Modellemesi

Kümeleme modelimizi oluşturuyoruz. Öncelikle sadece verilerimizin bir kopyasını oluşturuyoruz ve onu x değişkeninde saklıyoruz. Şimdi değişken k-means yaratacağız ve K-Means argümanı 2 olsun dedik.Yani 2 küme oluşturmak istediğimizi söyledik. Bu kod satırını k-means algoritmasını uygulayacak ve verilerimizi iki kümeye ayıracaktır. Kümeleme argümanları : 'coal_share_energy' ve 'renewables_share_energy'

```
# dataset5 birkaç satır sonrasında tanımlanmıştır ve sütunları : population,coal_share_energy, coal_consumption, renewables_share_energy, renewables
#Hızlıca ilişkileri tekrar göstermek istiyoruz.
```

[43]

Python

```
x=dataset5.copy()
kmeans=KMeans(5)# 5 küme
kmeans.fit(x) #model
```

[44]

Python

... KMeans(n_clusters=5)

```
clusters=x.copy()
clusters['cluster_pred']=kmeans .fit_predict(x) # Bir önceki satırda modelimizi oluşturmştuk, bu satırda ise x datasına göre
#modelimizi tahminliyoruz.
```

[45]

Python

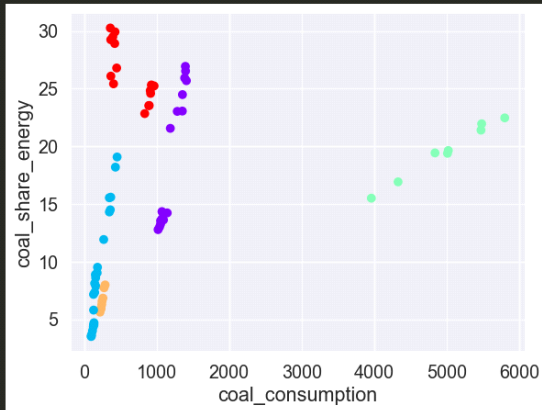
```
plt.scatter(clusters['coal_consumption'],clusters['coal_share_energy'],c=clusters['cluster_pred'],cmap='rainbow')
plt.xlabel('coal_consumption')
plt.ylabel('coal_share_energy')
plt.show
```

[46]

Python

```
... <function matplotlib.pyplot.show(close=None, block=None)>
```

</>



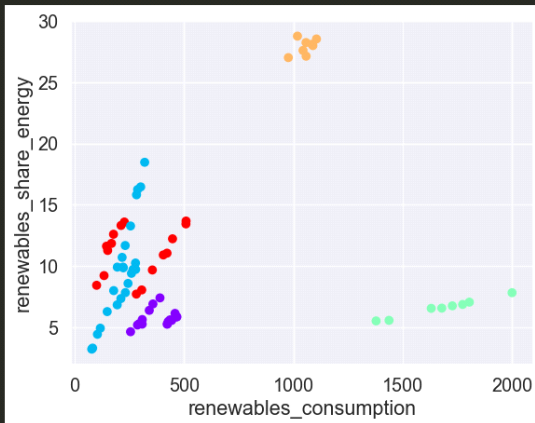
```
plt.scatter(clusters['renewables_consumption'],clusters['renewables_share_energy'],c=clusters['cluster_pred'],cmap='rainbow')
plt.xlabel('renewables_consumption')
plt.ylabel('renewables_share_energy')
plt.show
```

[47]

Python

```
... <function matplotlib.pyplot.show(close=None, block=None)>
```

</>



```
# K-Means kümeleme modelimiz için bizim kullanacağımız argümanlar : 'coal_share_energy' ve 'renewables_share_energy'
#
# Ve bu iki argümanı barındıracak olan veri setimizin ismini de 'datasetknn' yapıyoruz.

datasetk=dataset5[["coal_share_energy","renewables_share_energy"]]
```

[48]

Python


```
[49] #kmean kümeler arası uzaklığın yüksek, aynı kümedeki elemanlar arası yakın olması Python
```

Şimdi standartlaştırma yapmamız gerekiyor. Standartlaştırma, bizim daha sağlıklı ve doğru işlemler yapmamızı sağlar. Datamızda gördüğünüz üzere değerler birbirinden oldukça farklıdır. Bu bizim algoritmamızın bazı alt değerlerini görmezden gelmesine sebep olabilir, bu yüzden değerleri birbirine yakın seviyesine indirmemiz gerekiyor.

```
[50] # İlişkiyi incelemek için artık sadece bahsettiğimiz iki argümanı kullanarak tekrar model yapacağız. İki küme olacak  
# K-Means kümeleme modelimiz için bizim kullanacağımız argümanlar : 'coal_share_energy' ve 'renewables_share_energy'  
  
# Aynı işlemi tekrar yapıyoruz:  
Python
```

```
[51] # 2,3,4 civarında küme olabilir Python
```

```
[52] x=datasetk.copy()  
kmeans=KMeans(2) # 2 küme seçelim  
kmeans.fit(x) #model Python
```

```
... KMeans(n_clusters=2)
```

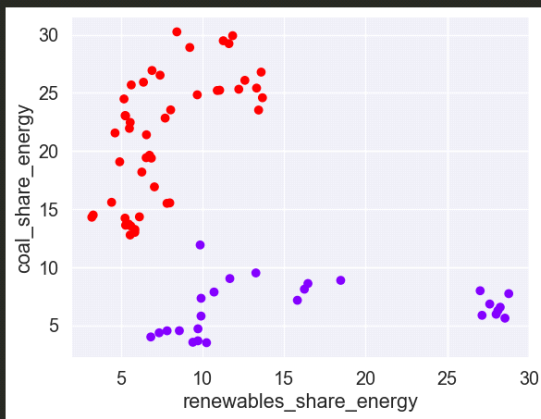
```
[53] # İki küme kullanmamızın sebebi analizlerin zorlaşmasıdır ; hem veri karmaşasını engellemek hem de  
# sadece gerekli argümanları karşılaştırmak için küme sayısını minimum tuttuk. Python
```

```
[54] clusters=x.copy()  
clusters['cluster_pred']=kmeans .fit_predict(x) # Bir önceki satırda modelimizi oluşturmştuk, bu satırda ise x datasına göre  
#modelimizi tahminliyoruz. Python
```

```
[55] plt.scatter(clusters['renewables_share_energy'],clusters['coal_share_energy'],c=clusters['cluster_pred'],cmap='rainbow')  
plt.xlabel('renewables_share_energy')  
plt.ylabel('coal_share_energy')  
plt.show  
Python
```

```
... <function matplotlib.pyplot.show(close=None, block=None)>
```

```
</>
```



Veriler sıfır ortalama civarında ölçeklenir. Şimdi her iki verinin de eşit şekilde ölçeklendiğini ve şimdi ikisinin de özellik olarak seçilme şansının eşit olacağını görebiliyoruz.

Dirsek Yöntemi

Markdown

```
from sklearn import preprocessing
x_scaled=preprocessing.scale(x)
x_scaled
```

[56]

Python

... Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
array([[ 1.69964768, -0.34736288],
       [ 1.57986441,  0.11758297],
       [ 1.60846626,  0.06677956],
       [ 1.66036586,  0.15048146],
       [ 1.20303033,  0.25951228],
       [ 1.53966356, -0.23207261],
       [ 1.12351442,  0.36417604],
       [ 1.2870696 ,  0.40493523],
       [ 0.71093634, -0.7708217 ],
       [ 0.77188267, -0.76397998],
       [ 0.64522858, -0.61884817],
       [ 0.41013288, -0.6214684 ],
       [ 0.43524943, -0.59162685],
       [ 0.40596663, -0.57517761],
       [ 0.11147206, -0.54781073],
       [-0.05708262, -0.43528626],
       [ 0.81616399, -0.45202664],
       [ 0.90020326, -0.40267892],
       [ 1.05471172, -0.16525666],
       [ 1.09887401,  0.01321034],
       [ 1.10173087,  0.03460892],
       [ 1.11184891,  0.20536079],
       [ 1.0247147 ,  0.41643514],
       [ 0.89841772,  0.38222654],
       [-1.38373659, -0.50515915],
       ...
       [ 0.36823228, -0.85758054],
       [ 0.26312367, -0.66018962],
       [-0.05279734, -0.41053962],
       [-0.48382596, -0.13934546],
       [-1.21244408, -0.13148476]])
```

```
wcss=[] # liste oluşturduk.
# WCSS (Within Cluster Sum of Square) her bir noktanın küme merkezine olan uzaklığının karesinin toplamını alınarak hesaplanır.
```

[57]

Python

```
#for i in range(1,72): #72 adet örneklememiz olduğu için 1'den 72'ye kadar gitmesini sağlıyoruz.
#kmeans=KMeans(1)
#kmeans.fit(x_scaled) #standortlaştırdığımız datamıza göre fit ettir, yani modelimizi oluşturun.
#wcss.append(kmeans.inertia_) #inertia ; Her bir k değeri için wcss değerini (her bir noktanın küme merkezine uzaklığı) bul.
#wcss
```

[58]

Python

```
wcss = []
kume_sayisi_listesi = range(1, 72)
for i in kume_sayisi_listesi :
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(x_scaled)
    wcss.append(kmeans.inertia_)
```

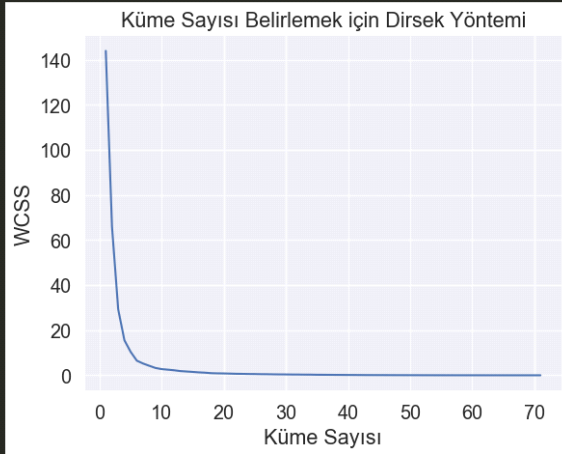
[59]

Python

... C:\Users\steam\anaconda3\lib\site-packages\sklearn\cluster_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
warnings.warn(

```
plt.plot(kume_sayisi_listesi, wcss)
plt.title('Küme Sayısı Belirlemek için Dirsek Yöntemi')
plt.xlabel('Küme Sayısı')
plt.ylabel('WCSS')
plt.show()
```

Python

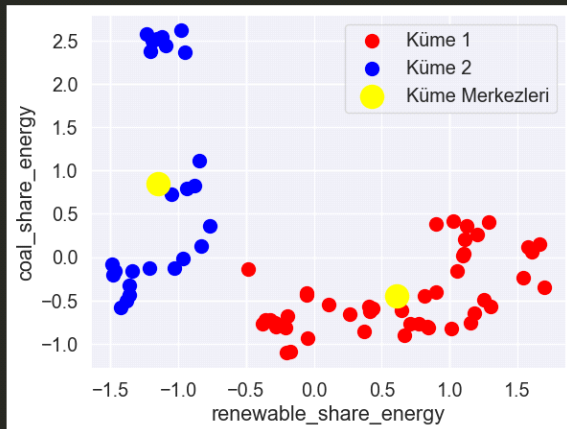


```
kmeans = KMeans(n_clusters = 2, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(x_scaled)
```

Python

```
plt.scatter(x_scaled[y_kmeans == 0], x_scaled[y_kmeans == 1], s = 100, c = 'red', label = 'Küme 1')
plt.scatter(x_scaled[y_kmeans == 1, 0], x_scaled[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Küme 2')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Küme Merkezleri')
plt.title('')
plt.xlabel('renewable_share_energy')
plt.ylabel('coal_share_energy')
plt.legend()
plt.show()
```

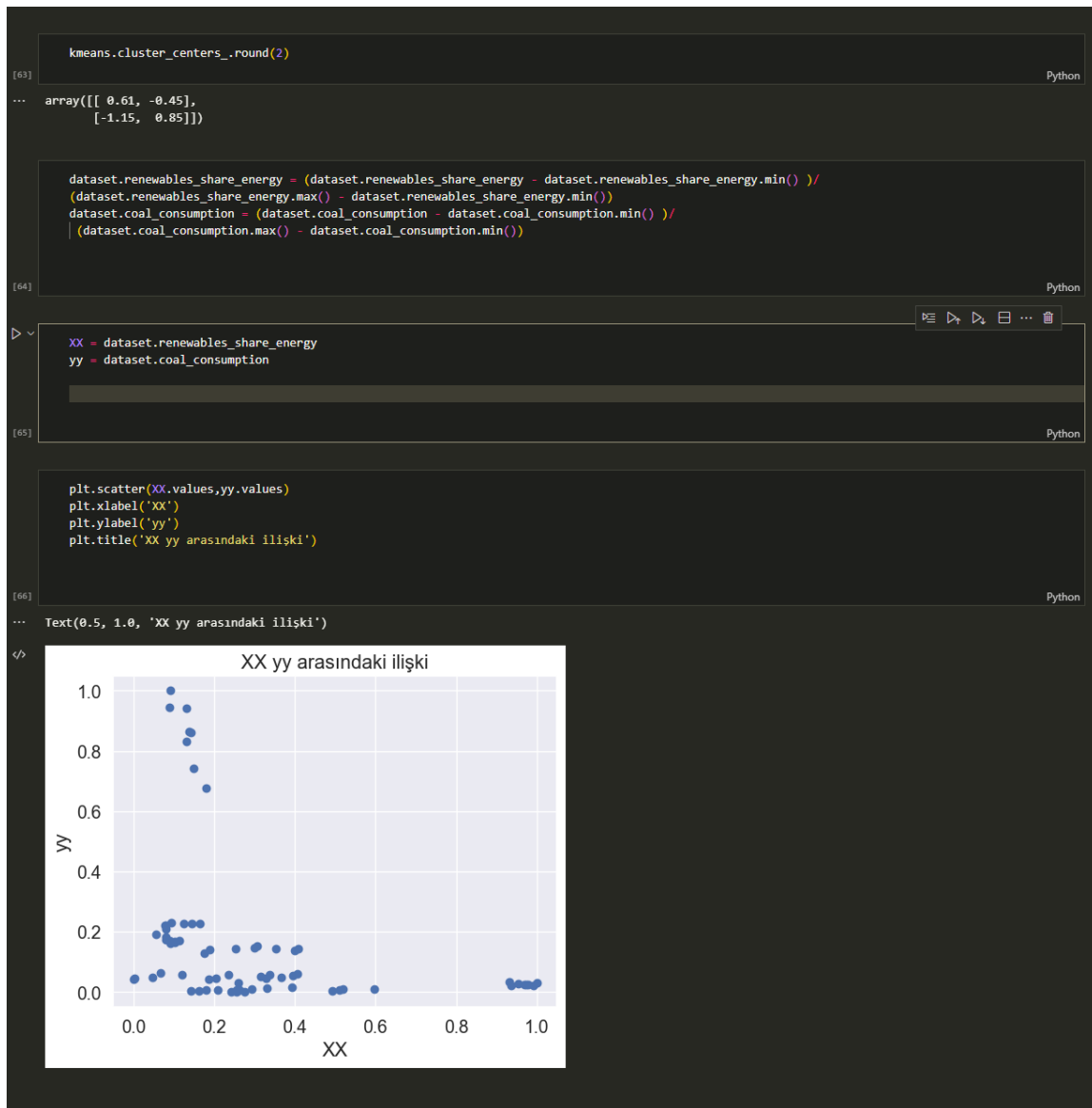
Python



```
kmeans.cluster_centers_.round(2)
```

Python

```
array([[ 0.61, -0.45],
       [-1.15,  0.85]])
```



Lineer Regresyon

```
[67] # Grafikte x ile y arasında doğrusal olmayan bir ilişki olduğu görülüyor. Şimdi en uygun çizgiyi çizdirmeye çalışalım. Python

linear_regresyon = LinearRegression()
linear_regresyon.fit(XX.values.reshape(-1,1),yy.values.reshape(-1,1))

[68] Python
... LinearRegression()

[69] # Modelimiz kuruldu ve verilerimize uyduruldu. Şimdi lineer modelimizi görüntüleyelim. Python

print(lineer_regresyon.intercept_)
print(lineer_regresyon.coef_)

[70] Python
... [0.26548598]
    [-0.32227264]

[71] # Şimdi regresyonun performansına bakalım. Python

yy_predicted = linear_regresyon.predict(XX.values.reshape(-1,1))
r2_score(yy,yy_predicted)

[72] Python
... 0.11429313072274194

+ Kod + Markdown
```

Kullanılan diğer model Lineer Regresyon oldu.

Modeli fit ettik.

Sıradığımız zaman model başarısını 0.11 olarak bulduk.

Sonuç olarak Lineer Regresyon'un bu veri seti için uygun bir model olmadığını kanıtladık.

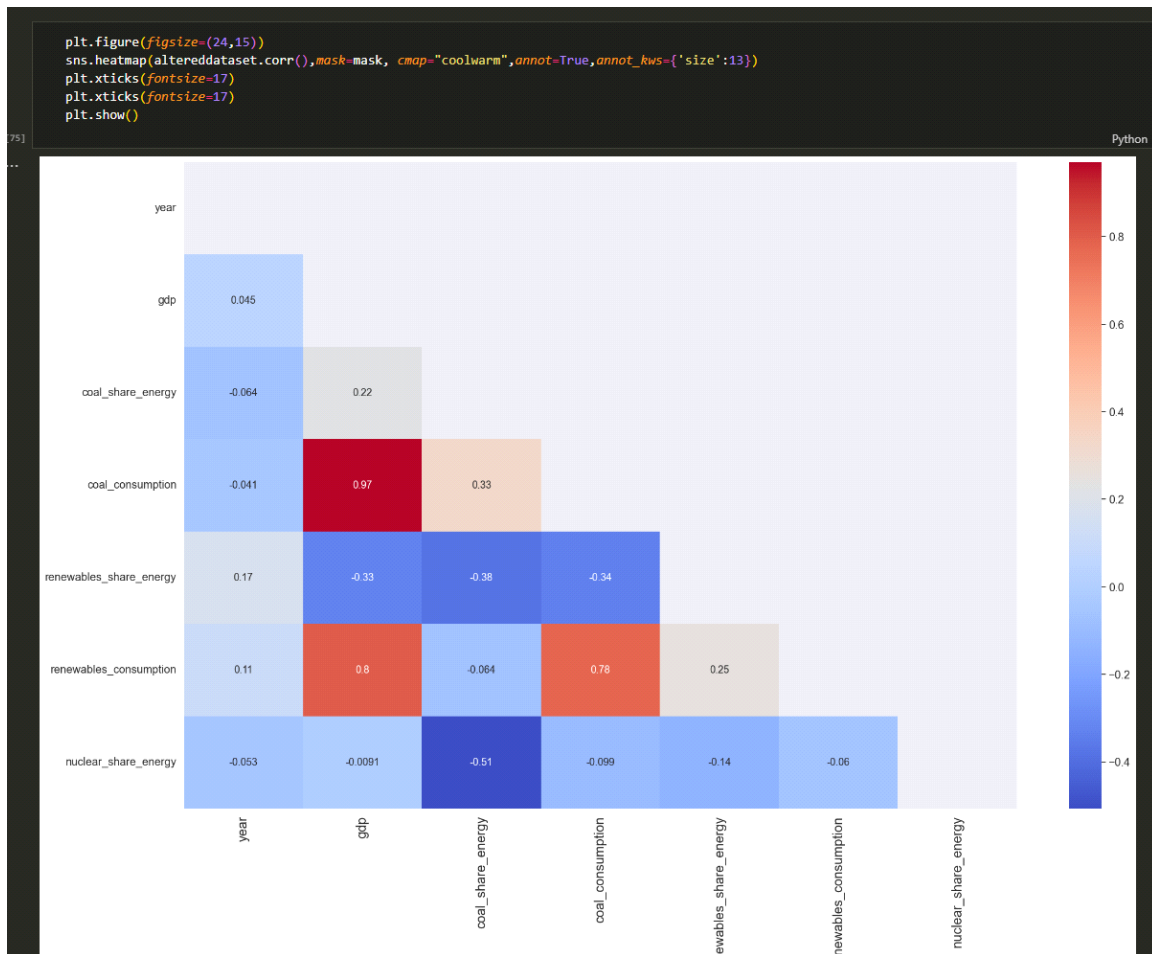
```
[73] dataset[["country","year","gdp","coal_share_energy","coal_consumption","renewables_share_energy","renewables_consumption","nuclear_share_energy"]] Python

mask=np.zeros_like(altereddataset.corr())
triangle_indices=np.triu_indices_from(mask)
mask[triangle_indices]=True
mask

[74] Python
... array([[1., 1., 1., 1., 1., 1., 1.],
    [0., 1., 1., 1., 1., 1., 1.],
    [0., 0., 1., 1., 1., 1., 1.],
    [0., 0., 0., 1., 1., 1., 1.],
    [0., 0., 0., 0., 1., 1., 1.],
    [0., 0., 0., 0., 0., 1., 1.],
    [0., 0., 0., 0., 0., 0., 1.]])

plt.figure(figsize=(24,15))
sns.heatmap(altereddataset.corr(),mask=mask, cmap="coolwarm",annot=True,annot_kws={'size':13})
plt.xticks(fontsize=17)
plt.yticks(fontsize=17)
plt.show()

[75] Python
```



```
dataset_pred=dataset[["population","coal_share_energy","coal_consumption","renewables_share_energy","renewables_consumption"]]
print(dataset_pred)
```

76] Python

```
...   population  coal_share_energy  coal_consumption  renewables_share_energy  \
0      71321000      30.254      0.046351      0.203959
1      72327000      29.241      0.047452      0.328900
2      73443000      29.488      0.052427      0.315248
3      74651000      29.924      0.057760      0.337741
4      75925000      26.082      0.047709      0.367040
...   ...
67    64525000      19.069      0.062780      0.066852
68    64984000      18.186      0.058735      0.119895
69    65423000      15.532      0.044020      0.186982
70    65860000      11.911      0.030404      0.259858
71    66298000       5.790      0.005962      0.261970

renewables_consumption
0      100.062
1      145.270
2      150.571
3      167.994
4      177.238
...   ...
67     116.866
68     148.359
69     178.008
70     221.972
71     220.545

[72 rows x 5 columns]
```

```
X = dataset_pred.iloc[:,[1]]
y = dataset_pred.iloc[:,[3]]

print(X)
print(y)
```

[77]

Python

```
...      coal_share_energy
0         30.254
1         29.241
2         29.488
3         29.924
4         26.082
..      ...
67        19.069
68        18.186
69        15.532
70        11.911
71         5.790
```

```
[72 rows x 1 columns]
      renewables_share_energy
0         0.203959
1         0.328900
2         0.315248
3         0.337741
4         0.367040
..      ...
67        0.066852
68        0.119895
69        0.186982
70        0.259858
71        0.261970
```

```
[72 rows x 1 columns]
```

Her enerji türünün tüketimi düzenli şekilde artmıştır. Nüfus arttıkça birim cinsinden tüketim de artmıştır. Bazı ülkeler yenilenebilir enerjinin önemini daha erken anlamış, bazıları da enerji paylarında yenilenebilir enerjilere daha fazla yer vermiştir.

Karar ağaçları

Type Markdown and LaTeX: α^2

```
In [49]: dataset_pred=dataset[["population","coal_share_energy","coal_consumption","renewables_share_energy","renewables_consumption"]]

X = dataset_pred.iloc[:,[0,-5]].values
y = dataset_pred.iloc[:,4].values

In [50]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

In [51]: from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

In [52]: from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state=0)
classifier.fit(X_train, y_train)

-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8848\3726176928.py in <module>
      1 from sklearn.tree import DecisionTreeClassifier
      2 classifier = DecisionTreeClassifier(criterion = 'entropy', random_state=0)
----> 3 classifier.fit(X_train, y_train)

~\anaconda3\lib\site-packages\sklearn\tree\_classes.py in fit(self, X, y, sample_weight, check_input, X_idx_sorted)
    935     """
    936
--> 937     super().fit(
    938         X,
    939         y,

~\anaconda3\lib\site-packages\sklearn\tree\_classes.py in fit(self, X, y, sample_weight, check_input, X_idx_sorted)
    201
    202     if is_classification:
--> 203         check_classification_targets(y)
    204         y = np.copy(y)
    205

~\anaconda3\lib\site-packages\sklearn\utils\multiclass.py in check_classification_targets(y)
    195     "multilabel-sequences",
    196     ]:
--> 197         raise ValueError("Unknown label type: %r" % y_type)
    198
    199

ValueError: Unknown label type: 'continuous'

In [ ]: y_pred = classifier.predict(X_test)

In [ ]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

In [ ]: print(dataf)

In [ ]: print(dataf.dtypes)

In [ ]: dataf['country'] = dataf['country'].apply(lambda x: float(x.split()[0].replace(',', '')))
```

Bir başka modelleme denememiz Karar Ağaçları oldu. Veri setinin modele uygun olmamasından dolayı çalışma hatalarıyla karşılaştık ve Karar Ağaçları modelinin başarı sonucuna ulaşamadık.

Her enerji türünün tüketimi düzenli şekilde artmıştır. Nüfus arttıkça birim cinsinden tüketim de artmıştır. Bazı ülkeler yenilenebilir enerjinin önemini daha erken anlamış, bazıları da enerji paylarında yenilenebilir enerjilere daha fazla yer vermiştir.

LEGEND / LEJANT

country year-Geographic location -Coğrafi Lokasyon

population- Nüfus

gdp- Gayrisafi Yurtiçi Hasıla (Trillion USD)

coal_share_energy : Share of electricity consumption that comes from coal. Kömürden gelen elektrik tüketiminin payı.

coal_consumption : Primary energy consumption from coal, measured in terawatt-hours. Kömürden birincil enerji tüketimi, terawatt-saat cinsinden ölçülür.

hydro_electricity : Electricity generation from hydropower, measured in terawatt-hours. Hidroelektrik enerjiden terawatt-saat cinsinden ölçülen elektrik üretimi.

nuclear_electricity : Electricity generation from nuclear power, measured in terawatt-hours. Nükleer enerjiden terawatt-saat cinsinden ölçülen elektrik üretimi.

renewables_electricity : Electricity generation from renewables, measured in terawatt-hours. Yenilenebilir enerjiden terawatt-saat cinsinden ölçülen elektrik üretimi.

solar_electricity : Electricity generation from solar, measured in terawatt-hours. Güneş enerjisinden terawatt-saat cinsinden ölçülen elektrik üretimi.

wind_electricity : Electricity generation from wind, measured in terawatt-hours. Terawatt-saat cinsinden ölçülen rüzgardan elektrik üretimi.

hydro_share_elec : Share of electricity consumption that comes from hydropower. Hidroelektrik enerjiden gelen elektrik tüketiminin payı.

hydro_share_energy : Share of primary energy consumption that comes from hydropower. Hidroelektrik enerjiden gelen birincil enerji tüketiminin payı.

nuclear_share_energy : Share of electricity consumption that comes from nuclear power. Nükleer enerjiden gelen elektrik tüketiminin payı.

oil_share_energy : Share of electricity consumption that comes from oil. Petrolen gelen elektrik tüketiminin payı.

renewables_share_elec : Share of electricity consumption that comes from renewables. Yenilenebilir kaynaklardan gelen elektrik tüketiminin payı.

renewables_share_energy : Share of primary energy consumption that comes from renewables. Yenilenebilir kaynaklardan gelen birincil enerji tüketiminin payı.

renewables_consumption : Primary energy consumption from renewables, measured in terawatt-hours. Yenilenebilir kaynaklardan elde edilen birincil enerji tüketimi, terawatt-saat cinsinden ölçülür.

solar_share_elec : Share of electricity consumption that comes from solar. Güneşten gelen elektrik tüketiminin payı.

solar_share_energy : Share of primary energy consumption that comes from solar. Güneşten gelen birincil enerji tüketiminin payı.

solar_consumption : Primary energy consumption from solar, measured in terawatt-hours. Terawatt-saat cinsinden ölçülen güneşten gelen birincil enerji tüketimi.

wind_share_elec : Share of electricity consumption that comes from wind. Rüzgardan gelen elektrik tüketiminin payı.

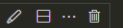
wind_share_energy : Share of primary energy consumption that comes from wind. Rüzgardan gelen birincil enerji tüketiminin payı.

This data has been collected, aggregated, and documented by Hannah Ritchie, Max Roser and Edouard Mathieu.(Hannah Ritchie and Max Roser (2020) - "Energy". Published online at OurWorldInData.org. Retrieved from: '<https://ourworldindata.org/energy>' [Online Resource])

The mission of Our World in Data is to make data and research on the world's largest problems understandable and accessible.

standardize names of countries and regions. Since the names of countries and regions are different in different data sources, standardize all names to the Our World in Data standard entity names. recalculate primary energy in terawatt-hours. The primary data sources on energy—the BP Statistical Review of World Energy, for example—typically report consumption in terms of exajoules. recalculated these figures as terawatt-hours using a conversion factor of 277.8. calculate per capita figures. All of our per capita figures are calculated from our metric Population, which is included in the complete dataset. These population figures are sourced from Gapminder and the UN World Population Prospects (UNWPP).

Data sources Energy consumption (primary energy, energy mix and energy intensity): this data is sourced from a combination of two sources—the BP Statistical Review of World Energy and SHIFT Data Portal. Electricity consumption (electricity consumption, and electricity mix): this data is sourced from a combination of two sources—the BP Statistical Review of World Energy and EMBER – Global Electricity Dashboard. Other variables: this data is collected from a variety of sources (United Nations, World Bank, Gapminder, Maddison Project Database, etc.).



Yasin Top 20184029009

Esmanur Akyel 20194029011

Hafta 14 - (5 - 12 Ocak, 2023)

Hafta 15 - 16 (16 - 27 Ocak, 2023)

