

# cguvtxrdv

January 14, 2025

```
[5]: import numpy as np
import pandas as pd
import seaborn as sbn
```

```
[7]: banks = pd.read_excel("dataset.xlsx", header=1, index_col = False)
```

```
[9]: banks.head()
```

```
[9]:
```

	Unnamed: 0 (Özkaynak+Kar) / T.Aktifler \	
0	BANKS	RATIO1
1	Akbank	14.706635
2	Alternatif Bank A.Ş.	10.793131
3	Anadolubank A.Ş.	5.280553
4	Arap Türk Bankası A.Ş.	8.30348

	(Özkaynak+Kar) / (Mevduat+Mevduat Dışı Kaynaklar) \	
0		RATIO2
1		18.247825
2		13.446925
3		5.838857
4		9.335769

	Net Çalışma Sermayesi / T.Aktifler \	
0		RATIO3
1		12.201029
2		5.828228
3		2.712216
4		4.742442

	(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler) T.Krediler / T.Aktifler \		
0		RATIO4	RATIO5
1		8.063853	35.744739
2		2.500647	39.740475
3		1.737541	23.884281
4		6.422892	12.699995

	Takipteki Krediler / T.Krediler Donuk Aktifler / T.Aktifler \	
--	---	--

0	RATIO6	RATIO7
1	1.593786	7.882845
2	3.25354	4.964902
3	0.93584	10.907227
4	15.0021	3.77319

	YP Aktifler / YP Pasifler Likit Aktifler / T.Aktifler ... \	
0	RATIO8	RATIO9 ...
1	69.092799	50.230906 ...
2	31.049619	50.034446 ...
3	44.414074	61.265201 ...
4	89.391947	79.772911 ...

	YP Likit Aktifler / YP Pasifler Net Dönem Karı / Ortalama T.Aktifler \	
0	RATIO11	RATIO12
1	24.846358	5.867689
2	16.46403	2.015983
3	41.920168	0.917544
4	81.237353	2.375364

	Net Dönem Karı / Ortalama Özkaynaklar \	
0	RATIO13	
1	58.317577	
2	28.87684	
3	19.791565	
4	39.008862	

	Net Dönem Karı / Ortalama Ödenmiş Sermaye \	
0	RATIO14	
1	91.628267	
2	37.530201	
3	31.907878	
4	47.726316	

	Takipt.Alacak Sonrası Net Faiz Geliri / Ortalama T.Aktifler \	
0	RATIO15	
1	15.714988	
2	19.232776	
3	14.074734	
4	7.350322	

	Faiz Gelirleri / Faiz Giderleri Faiz Dışı Gelirler / Faiz Dışı Giderler \	
0	RATIO16	RATIO17
1	301.576162	-37.818069
2	283.928511	-123.559222
3	217.783512	-56.46911
4	207.173337	19.379085

	Toplam Gelirler / Toplam Giderler	Takip.Alacak.Karşılığı / T.Krediler \
0	RATIO18	RATIO19
1	170.411271	0.940234
2	117.279426	0.732466
3	109.566223	0.610884
4	131.46847	1.735113

	şüpheli alacak rasyosu
0	RATIO20
1	0.078191
2	0.087449
3	0.055839
4	0.203552

[5 rows x 21 columns]

Öncelikle Ratio4, Ratio5, Ratio6, Ratio8, Ratio11, Ratio12, Ratio13, Ratio14 sütunları dışındakileri silerek başlıyoruz. Çünkü bizim ilgilendiğimiz sütunlar bunlar.

```
[11]: print(banks.columns)
```

```
Index(['Unnamed: 0', '(Özkaynak+Kar) / T.Aktifler',
      '(Özkaynak+Kar) / (Mevduat+Mevduat Dışı Kaynaklar)',
      'Net Çalışma Sermayesi / T.Aktifler',
      '(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)',
      'T.Krediler / T.Aktifler', 'Takipteki Krediler / T.Krediler',
      'Donuk Aktifler / T.Aktifler', 'YP Aktifler / YP Pasifler',
      'Likit Aktifler / T.Aktifler',
      'Likit Aktifler / (Mevduat+Mevduat Dışı Kaynaklar)',
      'YP Likit Aktifler / YP Pasifler',
      'Net Dönem Karı / Ortalama T.Aktifler',
      'Net Dönem Karı / Ortalama Özkaynaklar',
      'Net Dönem Karı / Ortalama Ödenmiş Sermaye',
      'Takipt.Alacak Sonrası Net Faiz Geliri / Ortalama T.Aktifler',
      'Faiz Gelirleri / Faiz Giderleri',
      'Faiz Dışı Gelirler / Faiz Dışı Giderler',
      'Toplam Gelirler / Toplam Giderler',
      'Takip.Alacak.Karşılığı / T.Krediler', 'şüpheli alacak rasyosu'],
      dtype='object')
```

```
[13]: ###Almak istediğimiz sütunları seçiyoruz ve bankalar(string) değişkenlerini de_
      ↪ekliyoruz ve sonrasında analiz yapılmaya hazır olacak.
      selected_columns = [
          'Unnamed: 0',
          '(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)',
          'T.Krediler / T.Aktifler',
```

```

'Takipteki Krediler / T.Krediler',
'YP Aktifler / YP Pasifler',
'YP Likit Aktifler / YP Pasifler',
'Net Dönem Karı / Ortalama T.Aktifler',
'Net Dönem Karı / Ortalama Özkaynaklar',
'Net Dönem Karı / Ortalama Ödenmiş Sermaye'
]
clean_data = banks[selected_columns]
print(clean_data.head())

```

```

      Unnamed: 0 (Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler) \
0          BANKS          RATIO4
1          Akbank          8.063853
2  Alternatif Bank A.Ş.    2.500647
3    AnadoluBank A.Ş.      1.737541
4  Arap Türk Bankası A.Ş.   6.422892

```

```

T.Krediler / T.Aktifler Takipteki Krediler / T.Krediler \
0          RATIO5          RATIO6
1    35.744739          1.593786
2    39.740475          3.25354
3    23.884281          0.93584
4    12.699995          15.0021

```

```

YP Aktifler / YP Pasifler YP Likit Aktifler / YP Pasifler \
0          RATIO8          RATIO11
1    69.092799          24.846358
2    31.049619          16.46403
3    44.414074          41.920168
4    89.391947          81.237353

```

```

Net Dönem Karı / Ortalama T.Aktifler Net Dönem Karı / Ortalama Özkaynaklar \
0          RATIO12          RATIO13
1    5.867689          58.317577
2    2.015983          28.87684
3    0.917544          19.791565
4    2.375364          39.008862

```

```

Net Dönem Karı / Ortalama Ödenmiş Sermaye
0          RATIO14
1    91.628267
2    37.530201
3    31.907878
4    47.726316

```

[15]: *###Verilerimiz istediğimiz şekle geldikten sonra sadece Unnamed:0 indexini ↵  
↪Banks olarak değiştirmemiz gerekiyor*

```
banks_cleaned = clean_data.drop(index=0)
banks_cleaned
```

[15]: Unnamed: 0 \

1	Akbank
2	Alternatif Bank A.Ş.
3	Anadolubank A.Ş.
4	Arap Türk Bankası A.Ş.
5	Bank Ekspres A.Ş.
6	Bank Kapital Türk A.Ş.
7	Bayındırbank A.Ş.
8	Birleşik Türk Körfez Bankası A.Ş.
9	Citibank N.A.
10	Demirbank T.A.Ş.
11	Denizbank A.Ş.
12	Egebank A.Ş.
13	Eskişehir Bankası T.A.Ş.
14	Etibank A.Ş.
15	Finans Bank A.Ş.
16	İnterbank
17	İktisat Bankası T.A.Ş.
18	Kentbank A.Ş.
19	Koçbank A.Ş.
20	M.N.G. Bank A.Ş.
21	Milli Aydın Bankası T.A.Ş.
22	Osmanlı Bankası A.Ş.
23	Oyak Bank A.Ş.
24	Pamukbank T.A.Ş.
25	Sitebank A.Ş.
26	Sümerbank A.Ş.
27	Şekerbank T.A.Ş.
28	Tekstil Bankası A.Ş.
29	Toprakbank A.Ş.
30	Türk Dış Ticaret Bankası A.Ş.
31	Türk Ekonomi Bankası A.Ş.
32	Türk Eximbank
33	Türk Ticaret Bankası A.Ş.
34	Türkiye Garanti Bankası A.Ş.
35	Türkiye İmar Bankası T.A.Ş.
36	Türkiye İş Bankası A.Ş.
37	Türkiye Tütüncüler Bankası Yaşarbank A.Ş.
38	Türkiye Vakıflar Bankası T.A.O.
39	Ulusal Bank T.A.Ş.
40	Yapı ve Kredi Bankası A.Ş.
41	Yurt Ticaret ve Kredi Bankası A.Ş.

(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler) T.Krediler / T.Aktifler \

1	8.063853	35.744739
2	2.500647	39.740475
3	1.737541	23.884281
4	6.422892	12.699995
5	-25.086214	36.538072
6	-60.491801	9.867205
7	8.711221	20.678587
8	4.229683	20.325116
9	4.081897	20.637887
10	2.870137	49.225741
11	4.618989	33.99206
12	-26.099661	13.250004
13	-12.841719	14.91374
14	-42.763552	45.502131
15	2.899148	28.828147
16	-15.193375	6.299833
17	-32.707571	27.812725
18	3.519467	34.182315
19	1.279412	47.43986
20	5.397852	49.718117
21	4.808453	34.902066
22	3.299437	23.575748
23	20.617122	42.887314
24	9.560759	59.091285
25	1.785085	43.791263
26	-16.445236	3.581653
27	5.26575	34.822481
28	2.68874	27.883242
29	3.716428	26.250154
30	4.686868	24.229953
31	2.873826	22.560851
32	14.741803	86.711242
33	4.872459	15.559228
34	5.732308	38.001295
35	8.757769	37.935785
36	12.479918	37.417356
37	-11.802206	6.586629
38	4.792475	47.167677
39	-2.345342	3.099819
40	10.982783	39.21603
41	-45.391436	0.417507

Takipteki Krediler / T.Krediler YP Aktifler / YP Pasifler \

1	1.593786	69.092799
2	3.25354	31.049619
3	0.93584	44.414074
4	15.0021	89.391947

5	121.806065	59.445488
6	1115.069119	5.869196
7	6.141776	98.171635
8	0.243036	62.533701
9	1.37095	73.594305
10	5.455102	81.016144
11	2.397918	49.366842
12	205.263858	55.072241
13	118.765071	6.009752
14	14.270554	48.083889
15	1.884998	79.405914
16	635.94116	62.550701
17	247.045492	30.807839
18	2.62167	64.796334
19	1.310769	76.184934
20	2.851441	66.404911
21	66.035249	97.236641
22	2.470141	91.783172
23	9.902158	84.936022
24	0.81024	61.675445
25	15.308754	59.8123
26	911.936754	33.235903
27	11.07601	76.489811
28	1.082103	32.625576
29	15.790819	93.974618
30	5.889538	79.920568
31	0.862013	87.417207
32	0.38377	132.204412
33	80.956855	56.013174
34	2.655018	78.17227
35	1.151497	101.956616
36	5.631946	85.805949
37	87.238517	64.543799
38	5.874143	88.253955
39	0	29.433614
40	4.258398	83.496709
41	7554.584838	15.9223

YP Likit Aktifler / YP Pasifler Net Dönem Karı / Ortalama T.Aktifler \

1	24.846358	5.867689
2	16.46403	2.015983
3	41.920168	0.917544
4	81.237353	2.375364
5	2.724291	-31.038266
6	1.787967	-169.234991
7	80.50361	0.406922
8	39.474693	2.251823

9	68.125616	5.782273
10	22.880692	-5.632547
11	16.744227	0.952394
12	48.654352	-58.245168
13	2.188542	-57.96703
14	26.06938	-67.986968
15	40.204938	4.244646
16	9.03752	-71.983811
17	10.796045	-120.604068
18	42.179988	0.759748
19	37.595024	1.162783
20	32.72134	0.060399
21	45.800715	-23.124858
22	59.879094	1.190698
23	24.299542	-5.29762
24	20.00081	4.314358
25	32.815495	-13.340601
26	6.380278	-30.908953
27	54.302479	0.309327
28	23.372077	3.972918
29	36.011532	0.299045
30	67.540629	6.983347
31	52.191741	2.765991
32	0.79702	3.675192
33	51.78848	-8.66423
34	29.297457	3.684888
35	21.586158	0.163574
36	40.532161	4.069139
37	56.582586	-50.798801
38	37.67953	1.211496
39	27.300781	-20.972902
40	45.964323	4.130131
41	15.132954	-65.165792

Net Dönem Karı / Ortalama Özkaynaklar \

1	58.317577
2	28.87684
3	19.791565
4	39.008862
5	60.669572
6	223.111373
7	3.012231
8	25.828446
9	68.018069
10	-74.700962
11	8.589945
12	70.371115



13	107.935124
14	241.009731
15	61.901233
16	261.543806
17	222.473323
18	12.476781
19	15.886725
20	0.261961
21	-631.319507
22	16.922424
23	-19.880631
24	31.478631
25	-266.219639
26	81.75736
27	3.78634
28	44.761655
29	5.699442
30	72.565881
31	47.847597
32	23.914529
33	-157.096845
34	40.336625
35	1.95867
36	25.991568
37	92.340014
38	19.240521
39	-3994.78924
40	28.319729
41	83.748233

Net Dönem Karı / Ortalama Ödenmiş Sermaye

1	91.628267
2	37.530201
3	31.907878
4	47.726316
5	-263.13
6	-2424.905983
7	3.576
8	32.176471
9	88.774331
10	-64.393889
11	13.300464
12	-1441.724
13	-996.013333
14	-1705.18
15	77.426741
16	-782.951

17	-2913.603333
18	15.304372
19	26.022667
20	0.523636
21	-93.456336
22	22.216216
23	-20.017886
24	106.989375
25	-140.12549
26	-575.875
27	9.111111
28	59.736667
29	8.47278
30	94.285333
31	60.442139
32	27.631886
33	-48.887404
34	78.956538
35	2.247423
36	60.967147
37	-1093.107407
38	27.867222
39	-299.355556
40	69.084412
41	-2414.65

```
[17]: # Tüm sütunların sayısal olup olmadığını kontrol edin
print(banks_cleaned.dtypes)

# Veri kümesinde eksik değerler var mı?
print(banks_cleaned.isnull().sum())

# Veri kümenizin boyutlarını kontrol edin
print(banks_cleaned.shape)
```

Unnamed: 0	object
(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)	object
T.Krediler / T.Aktifler	object
Takipteki Krediler / T.Krediler	object
YP Aktifler / YP Pasifler	object
YP Likit Aktifler / YP Pasifler	object
Net Dönem Karı / Ortalama T.Aktifler	object
Net Dönem Karı / Ortalama Özkaynaklar	object
Net Dönem Karı / Ortalama Ödenmiş Sermaye	object
dtype: object	
Unnamed: 0	0

```

(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)    0
T.Krediler / T.Aktifler                               0
Takipteki Krediler / T.Krediler                       0
YP Aktifler / YP Pasifler                             0
YP Likit Aktifler / YP Pasifler                       0
Net Dönem Karı / Ortalama T.Aktifler                 0
Net Dönem Karı / Ortalama Özkaynaklar                0
Net Dönem Karı / Ortalama Ödenmiş Sermaye           0
dtype: int64
(41, 9)

```

```

[19]: # Sayısal hale getirilmesi gereken sütunları seçme
columns_to_convert = [
    "(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)",
    "T.Krediler / T.Aktifler",
    "Takipteki Krediler / T.Krediler",
    "YP Aktifler / YP Pasifler",
    "YP Likit Aktifler / YP Pasifler",
    "Net Dönem Karı / Ortalama T.Aktifler",
    "Net Dönem Karı / Ortalama Özkaynaklar",
    "Net Dönem Karı / Ortalama Ödenmiş Sermaye"
]

# Dönüştürme işlemi
for col in columns_to_convert:
    banks_cleaned[col] = pd.to_numeric(banks_cleaned[col], errors='coerce')

# Veri tiplerini kontrol etme
print(banks_cleaned.dtypes)

```

```

Unnamed: 0      object
(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)    float64
T.Krediler / T.Aktifler                               float64
Takipteki Krediler / T.Krediler                       float64
YP Aktifler / YP Pasifler                             float64
YP Likit Aktifler / YP Pasifler                       float64
Net Dönem Karı / Ortalama T.Aktifler                 float64
Net Dönem Karı / Ortalama Özkaynaklar                float64
Net Dönem Karı / Ortalama Ödenmiş Sermaye           float64
dtype: object

```

```

[21]: # Sadece object türünde olan sütunları seçiyoruz
object_columns = banks_cleaned.select_dtypes(include=['object']).columns

# Object türündeki sütunları veri setinden çıkartıyoruz
banks_cleaned_numeric = banks_cleaned.drop(columns=object_columns)

```

```
# Sonuç
print(banks_cleaned_numeric.head())
print(banks_cleaned_numeric.dtypes)
```

	(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)	T.Krediler / T.Aktifler \
1	8.063853	35.744739
2	2.500647	39.740475
3	1.737541	23.884281
4	6.422892	12.699995
5	-25.086214	36.538072

	Takipteki Krediler / T.Krediler	YP Aktifler / YP Pasifler \
1	1.593786	69.092799
2	3.253540	31.049619
3	0.935840	44.414074
4	15.002100	89.391947
5	121.806065	59.445488

	YP Likit Aktifler / YP Pasifler	Net Dönem Karı / Ortalama T.Aktifler \
1	24.846358	5.867689
2	16.464030	2.015983
3	41.920168	0.917544
4	81.237353	2.375364
5	2.724291	-31.038266

	Net Dönem Karı / Ortalama Özkaynaklar \
1	58.317577
2	28.876840
3	19.791565
4	39.008862
5	60.669572

	Net Dönem Karı / Ortalama Ödenmiş Sermaye
1	91.628267
2	37.530201
3	31.907878
4	47.726316
5	-263.130000

(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)	float64
T.Krediler / T.Aktifler	float64
Takipteki Krediler / T.Krediler	float64
YP Aktifler / YP Pasifler	float64
YP Likit Aktifler / YP Pasifler	float64
Net Dönem Karı / Ortalama T.Aktifler	float64
Net Dönem Karı / Ortalama Özkaynaklar	float64
Net Dönem Karı / Ortalama Ödenmiş Sermaye	float64

dtype: object

1) Çoklu bağlantılılık kontrolü:

Çoklu bağlantılılık (multicollinearity), bağımsız değişkenler arasında yüksek düzeyde ilişki olduğunda ortaya çıkar.

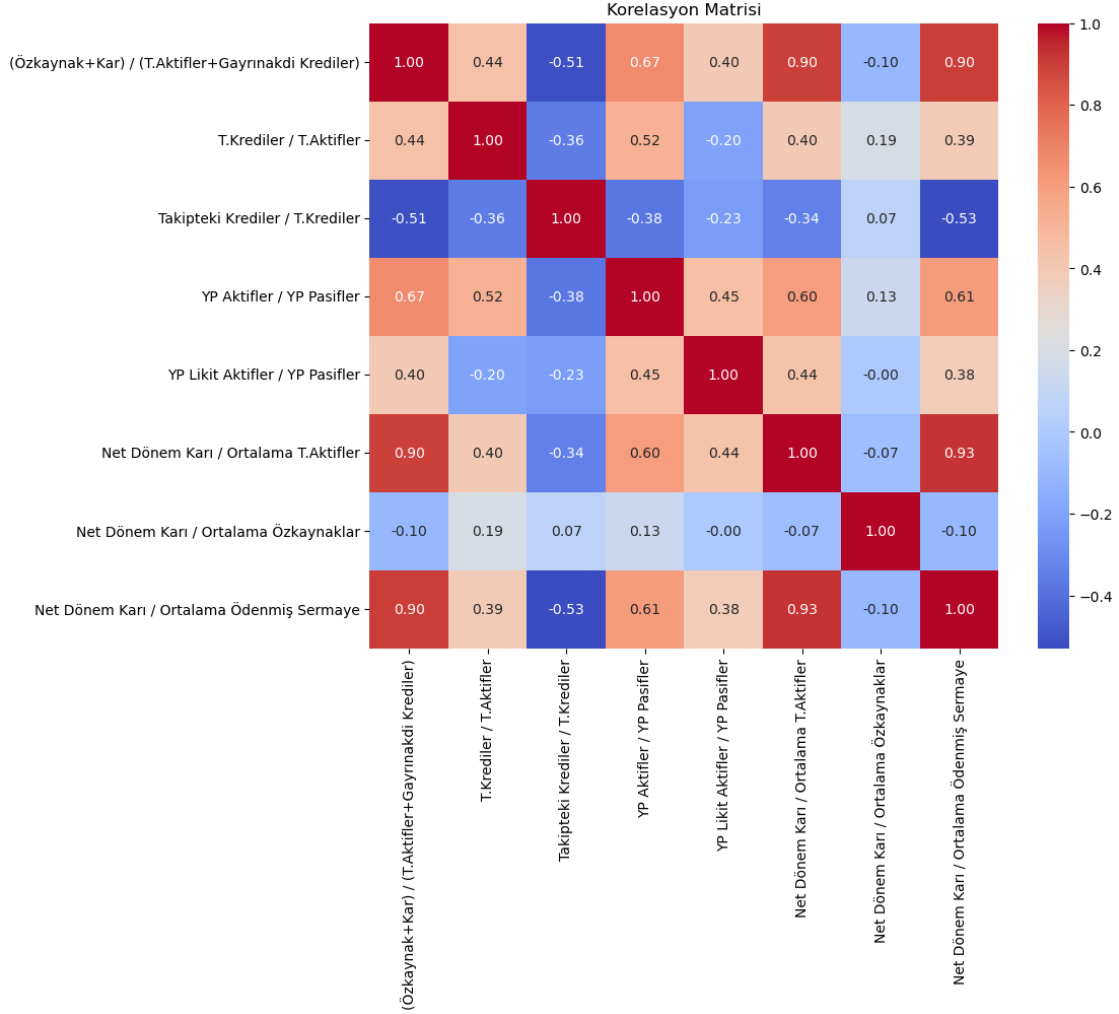
Hangi değişkenleri seçmeliyim?

Bağımsız değişkenlerin (predictors) ilişki durumunu incelemeliyiz. Genellikle bağımsız değişkenlerin birbirine çok yakın korelasyon göstermemesi gerekir. Örnek olarak: - (Özkaynak+Kar) / (T.Aktifler+Gayrimakdi Krediler) - T.Krediler / T.Aktifler - YP Aktifler / YP Pasifler - Net Dönem Karı / Ortalama T.Aktifler - Net Dönem Karı / Ortalama Özkaynaklar

```
[23]: import seaborn as sns
import matplotlib.pyplot as plt

# Korelasyon matrisi
correlation_matrix = banks_cleaned_numeric.corr()

# Korelasyon matrisi görselleştirme
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Korelasyon Matrisi")
plt.show()
```



```
[25]: from statsmodels.stats.outliers_influence import variance_inflation_factor

# VIF hesaplama
X = banks_cleaned_numeric.drop(columns=['Net Dönem Karı / Ortalama T.
↳Aktifler']) # Bağımlı değişkeni çıkar
vif_data = pd.DataFrame()
vif_data['Feature'] = X.columns
vif_data['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.
↳shape[1])]

print(vif_data)
```

	Feature	VIF
0	(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)	6.312724
1	T.Krediler / T.Aktifler	7.429679
2	Takipteki Krediler / T.Krediler	1.517477

3	YP Aktifler / YP Pasifler	17.044190
4	YP Likit Aktifler / YP Pasifler	5.991553
5	Net Dönem Karı / Ortalama Özkaynaklar	1.027053
6	Net Dönem Karı / Ortalama Ödenmiş Sermaye	6.729421

Takipteki Krediler / T.Krediler ve Net Dönem Karı / Ortalama Özkaynaklar sütunlarında çoklu doğrusal bağlantı sorunu yoktur çünkü  $-5 < VIF < +5$  arasındadır. YP Aktifler / YP Pasifler 17.044190 çok yüksek olmasının sebebi başka değişken ile çok yüksek korelasyon çıkmasındandır. Demek ki en çok ilişkideki sütunumuz YP Aktifler / YP Pasifler bağımsız değişkeniymiş. Genellikle bağımsız değişkenlerin birbirine çok yakın korelasyon göstermemesi gerekir. (Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler) ve Net Dönem Karı / Ortalama Ödenmiş Sermaye bunu biraz aştığından çoklu doğrusal bağlantı(multicollinearity) vardır.

[27]: #2)Bu parametre, verinin genel özelliklerine (özellikle verinin dağılımına ve ölçümlerine) göre, yüzün algılanan "duygusal" ya da "fiziksel" yapısını etkileyebilir. Eğer verilerde belirgin bir değişkenlik veya simetri varsa, bu görsel değişiklikler veriyi daha iyi anlamanızı sağlayabilir.

```
from ChernoffFace import chernoff_face
import matplotlib.pyplot as plt

# Chernoff yüzleri için görselleştirme
fig = chernoff_face(data=banks_cleaned_numeric.iloc[:10], # İlk 10 gözlemi
    ↳seçiyoruz
                    n_columns=5,
                    long_face=False,
                    figsize=(8, 8),
                    dpi=200)

plt.show()
```



3. Eksik verili bir versiyonunu oluşturuyoruz.

- Eksik verili bir veri setini ortalama ile doldurursak eğer bize sapmalı sonuç vermeyip analizimi etkilemez eğer farklı değerler atarsak küçük veri setleri için problem oluşabilir.
- En doğru yol ortalama değerlerini sütunların eksik veri yerlerine atamaktır.

4)Aşırı değer analizi yapacağız.

```
[29]: # Veriyi kopyalıyoruz
banks_with_missing = banks_cleaned_numeric.copy()

# Belirli bir yüzdede rastgele hücreleri NaN yapıyoruz (örneğin, %10)
missing_percentage = 0.1
missing_data = banks_with_missing.sample(frac=missing_percentage,
↪random_state=42)
```



```
# Eksik değerlerin oluşturulması
for col in banks_with_missing.columns:
    banks_with_missing.loc[missing_data.index, col] = np.nan

# Sonuç
print(banks_with_missing.isnull().sum())
```

```
(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)      4
T.Krediler / T.Aktifler                                  4
Takipteki Krediler / T.Krediler                         4
YP Aktifler / YP Pasifler                               4
YP Likit Aktifler / YP Pasifler                         4
Net Dönem Karı / Ortalama T.Aktifler                   4
Net Dönem Karı / Ortalama Özkaynaklar                  4
Net Dönem Karı / Ortalama Ödenmiş Sermaye              4
dtype: int64
```

```
[31]: banks_with_missing.apply(lambda x: x.fillna(x.mean())) # Varsayılan axis=0'dır.
      ↪ Bu sütun bazında demek.
```

```
[31]:      (Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler) \
1          8.063853
2          2.500647
3          1.737541
4          6.422892
5         -25.086214
6         -60.491801
7          8.711221
8          4.229683
9         -1.617070
10         2.870137
11         4.618989
12        -26.099661
13        -12.841719
14        -1.617070
15         2.899148
16        -15.193375
17        -32.707571
18         3.519467
19         1.279412
20         5.397852
21         4.808453
22         3.299437
23        20.617122
24         9.560759
25        -1.617070
```

26	-1.617070
27	5.265750
28	2.688740
29	3.716428
30	4.686868
31	2.873826
32	14.741803
33	4.872459
34	5.732308
35	8.757769
36	12.479918
37	-11.802206
38	4.792475
39	-2.345342
40	10.982783
41	-45.391436

	T.Krediler / T.Aktifler	Takipteki Krediler / T.Krediler \
1	35.744739	1.593786
2	39.740475	3.253540
3	23.884281	0.935840
4	12.699995	15.002100
5	36.538072	121.806065
6	9.867205	1115.069119
7	20.678587	6.141776
8	20.325116	0.243036
9	30.093424	279.573946
10	49.225741	5.455102
11	33.992060	2.397918
12	13.250004	205.263858
13	14.913740	118.765071
14	30.093424	279.573946
15	28.828147	1.884998
16	6.299833	635.941160
17	27.812725	247.045492
18	34.182315	2.621670
19	47.439860	1.310769
20	49.718117	2.851441
21	34.902066	66.035249
22	23.575748	2.470141
23	42.887314	9.902158
24	59.091285	0.810240
25	30.093424	279.573946
26	30.093424	279.573946
27	34.822481	11.076010
28	27.883242	1.082103
29	26.250154	15.790819

30	24.229953	5.889538
31	22.560851	0.862013
32	86.711242	0.383770
33	15.559228	80.956855
34	38.001295	2.655018
35	37.935785	1.151497
36	37.417356	5.631946
37	6.586629	87.238517
38	47.167677	5.874143
39	3.099819	0.000000
40	39.216030	4.258398
41	0.417507	7554.584838

	YP Aktifler / YP Pasifler	YP Likit Aktifler / YP Pasifler \
1	69.092799	24.846358
2	31.049619	16.464030
3	44.414074	41.920168
4	89.391947	81.237353
5	59.445488	2.724291
6	5.869196	1.787967
7	98.171635	80.503610
8	62.533701	39.474693
9	66.849890	34.108681
10	81.016144	22.880692
11	49.366842	16.744227
12	55.072241	48.654352
13	6.009752	2.188542
14	66.849890	34.108681
15	79.405914	40.204938
16	62.550701	9.037520
17	30.807839	10.796045
18	64.796334	42.179988
19	76.184934	37.595024
20	66.404911	32.721340
21	97.236641	45.800715
22	91.783172	59.879094
23	84.936022	24.299542
24	61.675445	20.000810
25	66.849890	34.108681
26	66.849890	34.108681
27	76.489811	54.302479
28	32.625576	23.372077
29	93.974618	36.011532
30	79.920568	67.540629
31	87.417207	52.191741
32	132.204412	0.797020
33	56.013174	51.788480

34	78.172270	29.297457
35	101.956616	21.586158
36	85.805949	40.532161
37	64.543799	56.582586
38	88.253955	37.679530
39	29.433614	27.300781
40	83.496709	45.964323
41	15.922300	15.132954

Net Dönem Karı / Ortalama T.Aktifler \

1	5.867689
2	2.015983
3	0.917544
4	2.375364
5	-31.038266
6	-169.234991
7	0.406922
8	2.251823
9	-17.052559
10	-5.632547
11	0.952394
12	-58.245168
13	-57.967030
14	-17.052559
15	4.244646
16	-71.983811
17	-120.604068
18	0.759748
19	1.162783
20	0.060399
21	-23.124858
22	1.190698
23	-5.297620
24	4.314358
25	-17.052559
26	-17.052559
27	0.309327
28	3.972918
29	0.299045
30	6.983347
31	2.765991
32	3.675192
33	-8.664230
34	3.684888
35	0.163574
36	4.069139
37	-50.798801

38	1.211496
39	-20.972902
40	4.130131
41	-65.165792

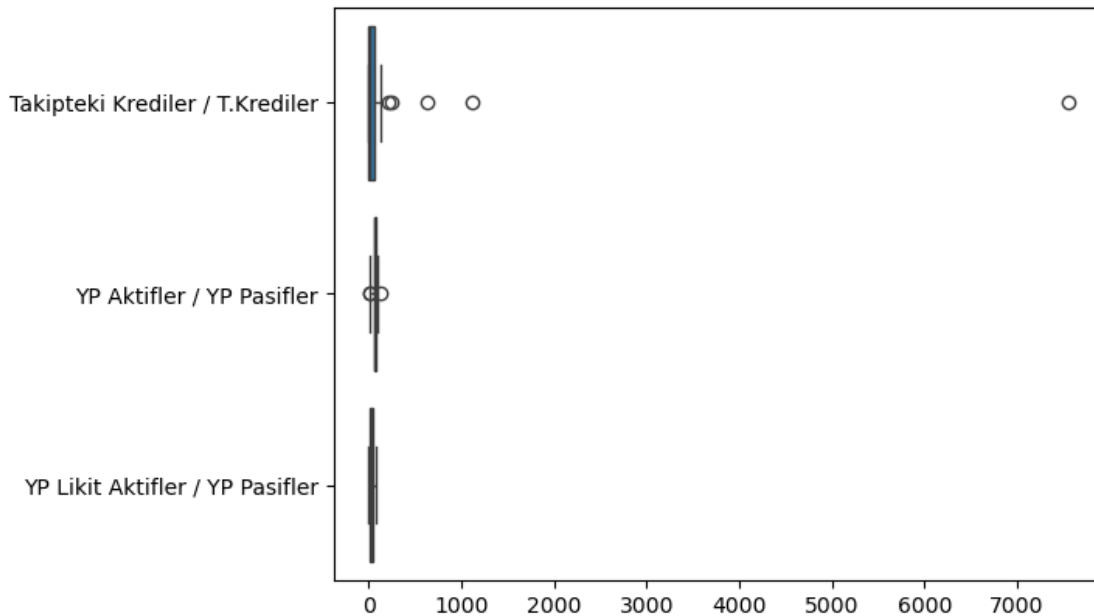
	Net Dönem Karı / Ortalama Özkaynaklar \
1	58.317577
2	28.876840
3	19.791565
4	39.008862
5	60.669572
6	223.111373
7	3.012231
8	25.828446
9	-84.292401
10	-74.700962
11	8.589945
12	70.371115
13	107.935124
14	-84.292401
15	61.901233
16	261.543806
17	222.473323
18	12.476781
19	15.886725
20	0.261961
21	-631.319507
22	16.922424
23	-19.880631
24	31.478631
25	-84.292401
26	-84.292401
27	3.786340
28	44.761655
29	5.699442
30	72.565881
31	47.847597
32	23.914529
33	-157.096845
34	40.336625
35	1.958670
36	25.991568
37	92.340014
38	19.240521
39	-3994.789240
40	28.319729
41	83.748233

	Net Dönem Karı / Ortalama Ödenmiş Sermaye
1	91.628267
2	37.530201
3	31.907878
4	47.726316
5	-263.130000
6	-2424.905983
7	3.576000
8	32.176471
9	-320.299050
10	-64.393889
11	13.300464
12	-1441.724000
13	-996.013333
14	-320.299050
15	77.426741
16	-782.951000
17	-2913.603333
18	15.304372
19	26.022667
20	0.523636
21	-93.456336
22	22.216216
23	-20.017886
24	106.989375
25	-320.299050
26	-320.299050
27	9.111111
28	59.736667
29	8.472780
30	94.285333
31	60.442139
32	27.631886
33	-48.887404
34	78.956538
35	2.247423
36	60.967147
37	-1093.107407
38	27.867222
39	-299.355556
40	69.084412
41	-2414.650000

```
[37]: sbn.boxplot(data = banks_with_missing[['Takipteki Krediler / T.Krediler', 'YP_
↳Aktifler / YP Pasifler', 'YP Likit Aktifler / YP Pasifler']], orient = 'h')
```

```
# Bu grafikten de anlayacağımız üzere Takipteki Krediler / T.Krediler de aykırı
↪ değerler mevcut. Sadece
# büyük değerlerde aykırı gözlemler mevcut.Burdan da aşırı derecede bazı
↪ bankalarda takipte kredi oranının etkileyen bazı kötü işlemler
# olduğunu gösterir
```

[37]: <Axes: >

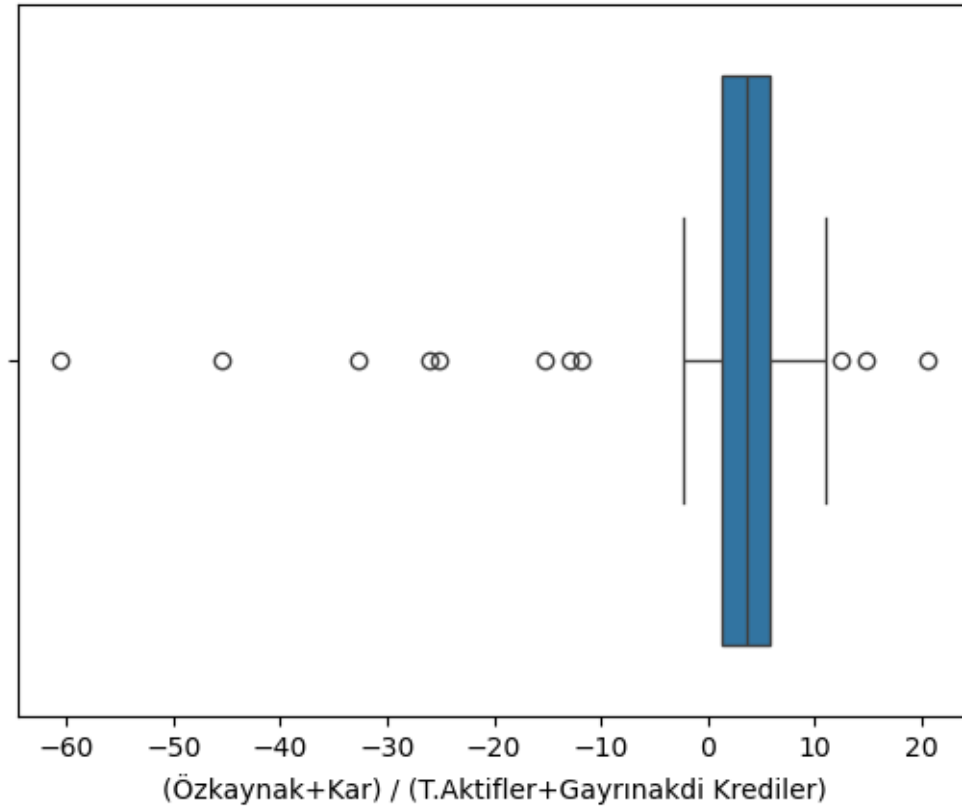


```
[48]: q1_x = banks_with_missing['Takipteki Krediler / T.Krediler'].quantile(0.25)
q3_x = banks_with_missing['Takipteki Krediler / T.Krediler'].quantile(0.75)
iqr_x = q3_x - q1_x
# x değişkeni için aykırı değerleri belirleyecek sınırları bulabiliriz.
lowerbound_x = q1_x - 1.5 * iqr_x # Alt sınır
upperbound_x = q3_x + 1.5 * iqr_x # Üst sınır
print(lowerbound_x)
print(upperbound_x)
```

```
-95.06840942457006
162.69744441157746
```

```
[39]: # Bağımlı değişkenimiz olan (Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)
↪ değişkeni için de boxplot çizdirelim:
sbn.boxplot(x = banks_with_missing['(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi
↪ Krediler)'])
```

[39]: <Axes: xlabel='(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)'>



[171]: *### Bağımlı değişkenimizde daha fazla uç değer olması ne kadar etkilendiğini  
→ bize göstermiş oluyor.*

[41]: `banks = banks_cleaned.drop(columns=["Unnamed: 0"]) # Sayısal olmayan sütunları  
→ listeleyerek çıkarabilirsiniz`  
banks

[41]:

	(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler) \
1	8.063853
2	2.500647
3	1.737541
4	6.422892
5	-25.086214
6	-60.491801
7	8.711221
8	4.229683
9	4.081897
10	2.870137
11	4.618989
12	-26.099661
13	-12.841719



14	-42.763552
15	2.899148
16	-15.193375
17	-32.707571
18	3.519467
19	1.279412
20	5.397852
21	4.808453
22	3.299437
23	20.617122
24	9.560759
25	1.785085
26	-16.445236
27	5.265750
28	2.688740
29	3.716428
30	4.686868
31	2.873826
32	14.741803
33	4.872459
34	5.732308
35	8.757769
36	12.479918
37	-11.802206
38	4.792475
39	-2.345342
40	10.982783
41	-45.391436

	T.Krediler / T.Aktifler	Takipteki Krediler / T.Krediler \
1	35.744739	1.593786
2	39.740475	3.253540
3	23.884281	0.935840
4	12.699995	15.002100
5	36.538072	121.806065
6	9.867205	1115.069119
7	20.678587	6.141776
8	20.325116	0.243036
9	20.637887	1.370950
10	49.225741	5.455102
11	33.992060	2.397918
12	13.250004	205.263858
13	14.913740	118.765071
14	45.502131	14.270554
15	28.828147	1.884998
16	6.299833	635.941160
17	27.812725	247.045492

18	34.182315	2.621670
19	47.439860	1.310769
20	49.718117	2.851441
21	34.902066	66.035249
22	23.575748	2.470141
23	42.887314	9.902158
24	59.091285	0.810240
25	43.791263	15.308754
26	3.581653	911.936754
27	34.822481	11.076010
28	27.883242	1.082103
29	26.250154	15.790819
30	24.229953	5.889538
31	22.560851	0.862013
32	86.711242	0.383770
33	15.559228	80.956855
34	38.001295	2.655018
35	37.935785	1.151497
36	37.417356	5.631946
37	6.586629	87.238517
38	47.167677	5.874143
39	3.099819	0.000000
40	39.216030	4.258398
41	0.417507	7554.584838

	YP Aktifler / YP Pasifler	YP Likit Aktifler / YP Pasifler \
1	69.092799	24.846358
2	31.049619	16.464030
3	44.414074	41.920168
4	89.391947	81.237353
5	59.445488	2.724291
6	5.869196	1.787967
7	98.171635	80.503610
8	62.533701	39.474693
9	73.594305	68.125616
10	81.016144	22.880692
11	49.366842	16.744227
12	55.072241	48.654352
13	6.009752	2.188542
14	48.083889	26.069380
15	79.405914	40.204938
16	62.550701	9.037520
17	30.807839	10.796045
18	64.796334	42.179988
19	76.184934	37.595024
20	66.404911	32.721340
21	97.236641	45.800715

22	91.783172	59.879094
23	84.936022	24.299542
24	61.675445	20.000810
25	59.812300	32.815495
26	33.235903	6.380278
27	76.489811	54.302479
28	32.625576	23.372077
29	93.974618	36.011532
30	79.920568	67.540629
31	87.417207	52.191741
32	132.204412	0.797020
33	56.013174	51.788480
34	78.172270	29.297457
35	101.956616	21.586158
36	85.805949	40.532161
37	64.543799	56.582586
38	88.253955	37.679530
39	29.433614	27.300781
40	83.496709	45.964323
41	15.922300	15.132954

Net Dönem Karı / Ortalama T.Aktifler \

1	5.867689
2	2.015983
3	0.917544
4	2.375364
5	-31.038266
6	-169.234991
7	0.406922
8	2.251823
9	5.782273
10	-5.632547
11	0.952394
12	-58.245168
13	-57.967030
14	-67.986968
15	4.244646
16	-71.983811
17	-120.604068
18	0.759748
19	1.162783
20	0.060399
21	-23.124858
22	1.190698
23	-5.297620
24	4.314358
25	-13.340601

26	-30.908953
27	0.309327
28	3.972918
29	0.299045
30	6.983347
31	2.765991
32	3.675192
33	-8.664230
34	3.684888
35	0.163574
36	4.069139
37	-50.798801
38	1.211496
39	-20.972902
40	4.130131
41	-65.165792

Net Dönem Karı / Ortalama Özkaynaklar \

1	58.317577
2	28.876840
3	19.791565
4	39.008862
5	60.669572
6	223.111373
7	3.012231
8	25.828446
9	68.018069
10	-74.700962
11	8.589945
12	70.371115
13	107.935124
14	241.009731
15	61.901233
16	261.543806
17	222.473323
18	12.476781
19	15.886725
20	0.261961
21	-631.319507
22	16.922424
23	-19.880631
24	31.478631
25	-266.219639
26	81.757360
27	3.786340
28	44.761655
29	5.699442

30	72.565881
31	47.847597
32	23.914529
33	-157.096845
34	40.336625
35	1.958670
36	25.991568
37	92.340014
38	19.240521
39	-3994.789240
40	28.319729
41	83.748233

Net Dönem Karı / Ortalama Ödenmiş Sermaye

1	91.628267
2	37.530201
3	31.907878
4	47.726316
5	-263.130000
6	-2424.905983
7	3.576000
8	32.176471
9	88.774331
10	-64.393889
11	13.300464
12	-1441.724000
13	-996.013333
14	-1705.180000
15	77.426741
16	-782.951000
17	-2913.603333
18	15.304372
19	26.022667
20	0.523636
21	-93.456336
22	22.216216
23	-20.017886
24	106.989375
25	-140.125490
26	-575.875000
27	9.111111
28	59.736667
29	8.472780
30	94.285333
31	60.442139
32	27.631886
33	-48.887404

```

34                                     78.956538
35                                     2.247423
36                                     60.967147
37                                    -1093.107407
38                                     27.867222
39                                    -299.355556
40                                     69.084412
41                                    -2414.650000

```

```

[55]: import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn import preprocessing
import xgboost
from xgboost import XGBRegressor
# Hedef ve özellik değişkenlerini ayırma
y = banks["Net Dönem Karı / Ortalama Özkaynaklar"]
x_ = banks.drop(["Takipteki Krediler / T.Krediler", "YP Aktifler / YP_
↳Pasifler", "YP Likit Aktifler / YP Pasifler", "Net Dönem Karı / Ortalama_
↳Ödenmiş Sermaye"], axis=1).astype("float64")
x = pd.concat([x_, banks[["Takipteki Krediler / T.Krediler", "YP Aktifler / YP_
↳Pasifler", "YP Likit Aktifler / YP Pasifler"]]], axis=1)
# Sütun bazlı silme işlemi yapması için axis = 1 yazıldı.

# Veri setini bölme
banks_x_train, banks_x_test, banks_y_train, banks_y_test = train_test_split(
    x, y, test_size=0.2, random_state=2017
)

```

```

[59]: # Create the model and fit it
xgb_model = XGBRegressor()
xgb_model.fit(banks_x_train, banks_y_train)

# Make predictions
xgb_y_pred = xgb_model.predict(banks_x_test)
xgb_y_pred

```

```

[59]: array([ 46.04296 ,  1.0472748, 16.60345 ,  1.0990962,  1.2809045,
        67.866325 , 60.41046 , -74.493965 , 24.798674 ], dtype=float32)

```

```

[ ]: ### Normalize edildiğinde, tahmin değerleri daha düzenli ve beklenebilir hale_
↳gelir. Bu, modelin daha dengeli çalıştığını gösterebilir.

```

```

[63]: #5)
import matplotlib.pyplot as plt

```

```

import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn import preprocessing

#Normalizasyon için MinMaxScaler() fonksiyonu kullanıp varsayımları inceleriz.

# min-max scaler ile değişkenlerdeki gözlemleri 0 ve 1 aralığında
↳ölçeklendiririz.
# min_max_scaler işlemi uygulayacak bir araç hazırlayalım ki her defasında bu
↳işlemi çağırmak kolay
# olsun:
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

# Farklı olduğunu düşündüğümüz değişkenlere yani sayısal değişkenlere bu scaler
↳işlemi yaptıralım:
banks.info()
numeric_vals = ["T.Krediler / T.Aktifler", "Takipteki Krediler / T.Krediler",
↳"Takipteki Krediler / T.Krediler", "YP Likit Aktifler / YP Pasifler",
↳"(Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)"]

# Ölçeklendirme yaparken fit etme işlemi train set üzerinden yapılır.
banks_x_train[numeric_vals] = scaler.fit_transform(banks_x_train[numeric_vals])
banks_x_test[numeric_vals] = scaler.fit_transform(banks_x_test[numeric_vals])

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41 entries, 1 to 41
Data columns (total 8 columns):
#   Column                                                                 Non-Null Count  Dtype
---  -
0   (Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)                  41 non-null    float64
1   T.Krediler / T.Aktifler                                             41 non-null    float64
2   Takipteki Krediler / T.Krediler                                    41 non-null    float64
3   YP Aktifler / YP Pasifler                                           41 non-null    float64
4   YP Likit Aktifler / YP Pasifler                                    41 non-null    float64
5   Net Dönem Karı / Ortalama T.Aktifler                              41 non-null    float64
6   Net Dönem Karı / Ortalama Özkaynaklar                             41 non-null    float64
7   Net Dönem Karı / Ortalama Ödenmiş Sermaye                         41 non-null    float64
dtypes: float64(8)
memory usage: 2.7 KB

```

```

[65]: #Min-Max dönüşümü
# Verinin en küçük değerleri 0 en büyük değerleri 1'e eşit yapan dönüşüm.
scaler00 = preprocessing.MinMaxScaler(feature_range=(0,1))

```

```
# Yukarıdaki kod satırı bir ölçeklendirici hazırlar. Bu ölçeklendiriciyi bir
    ↪ nesnede tutup o nesne
# üzerinden tanımladığımız bu ölçeklendirmeyi istediğimiz yerde daha rahat
    ↪ kullanabiliriz.

scaler00.fit_transform(banks)
# x_yeni = (x_gözlem - min(x)) / (max(x) - min(x)) şeklinde hesaplıyor.

# Değerleri 0 ve 1 arasındaki değerlerden oluşan bu yeni veriyi data frame
    ↪ yapısına dönüştürelim:
banksscaler00 = pd.DataFrame(scaler00.fit_transform(banks))
```

```
[71]: #RFE, özellik seçimi için kullanılan bir tekniktir. Özellik seçimi modelin
    ↪ performansını
# iyileştirmek, aşırı uyumu (overfitting) önlemek ve eğitim süresini kısaltmak
    ↪ amacıyla gereksiz
# veya anlamlı olmayan özellikleri (değişkenleri) veriden çıkarmak için yapılan
    ↪ bir işlemdir.

from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

# Bir lineer regresyon modeli oluşturalım:
lm_RFE = LinearRegression()
# Modeli x'lere karşılık y'lerle fit edelim:
lm_RFE.fit(banks_x_train, banks_y_train)

# Şimdi bu fit edilmiş model üzerinden RFE işlemi yapalım yani 7 bağımsız
    ↪ değişken seçelim:
rfe = RFE(lm_RFE, n_features_to_select=7)

rfe = rfe.fit(banks_x_train, banks_y_train)
list(zip(banks_x_train.columns, rfe.support_, rfe.ranking_)) # Hangi
    ↪ değişkenlerin modelde olduğuna dair bir liste verir.
```

```
[71]: [('Özkaynak+Kar) / (T.Aktifler+Gayrınakdi Krediler)', True, 1),
      ('T.Krediler / T.Aktifler', True, 1),
      ('Net Dönem Karı / Ortalama T.Aktifler', True, 1),
      ('Net Dönem Karı / Ortalama Özkaynaklar', True, 1),
      ('Takipteki Krediler / T.Krediler', True, 1),
      ('YP Aktifler / YP Pasifler', True, 1),
      ('YP Likit Aktifler / YP Pasifler', True, 1)]
```

```
[87]: # Toplamda 7 değişken var. 7 tanesinde 1 var. Çünkü tahminleri 7 değişken ile
    ↪ yaptı.
```



```
# Bu modelde işe yaramadığını düşündüğü modele dahil etmediği değişken yok  
↪hepsi true döndü.
```

```
# Tahminlerimizi bu 7 değişken ile elde edebiliriz.
```

```
y_pred_Train = rfe.predict(banks_x_train)  
y_pred_Train
```

```
y_pred_Test = rfe.predict(banks_x_test)  
y_pred_Test
```

```
[87]: array([ 47.84759715,   5.69944228,  16.92242448,   3.78633963,  
          1.95867026,  72.56588129,  61.90123349, -19.88063095,  
          31.47863121])
```

```
[107]: print("banks_y_train shape:", banks_y_train.shape)  
print("y_pred_Train shape:", y_pred_Train.shape)  
print("banks_y_test shape:", banks_y_test.shape)  
print("y_pred_Test shape:", y_pred_Test.shape)
```

```
banks_y_train shape: (32,)  
y_pred_Train shape: (32,)  
banks_y_test shape: (9,)  
y_pred_Test shape: (9,)
```

```
[127]: banks_y_train = banks_y_train.fillna(banks_y_train.mean())  
banks_y_test = banks_y_test.fillna(banks_y_test.mean())
```

```
[ ]: # RMSE ve MAPE hesaplama fonksiyonu  
def rmse(banks_y_train, y_pred_Train):  
    return np.sqrt(np.mean((banks_y_train - y_pred_Train) ** 2))  
  
def mape(y_true, y_pred_Train):  
    return np.mean(np.abs((banks_y_train - y_pred_Train) / banks_y_train)) * 100  
  
# Eğitim kümesi için RMSE ve MAPE  
rmse_train = rmse(banks_y_train, y_pred_Train)  
mape_train = mape(banks_y_train, y_pred_Train)  
  
# Test kümesi için RMSE ve MAPE  
rmse_test = rmse(banks_y_test, y_pred_Test)  
mape_test = mape(banks_y_test, y_pred_Test)
```