Normal Dağılım , Z-Score Standardizasyon ve Normalizasyon

Gülcan Öğündür · Following

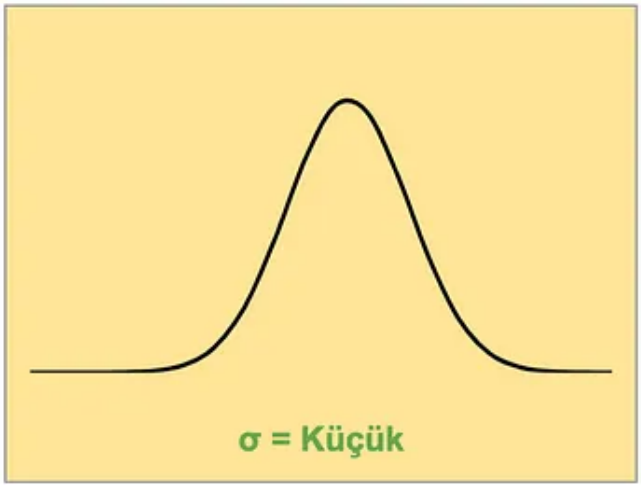
4 min read · Dec 25, 2019

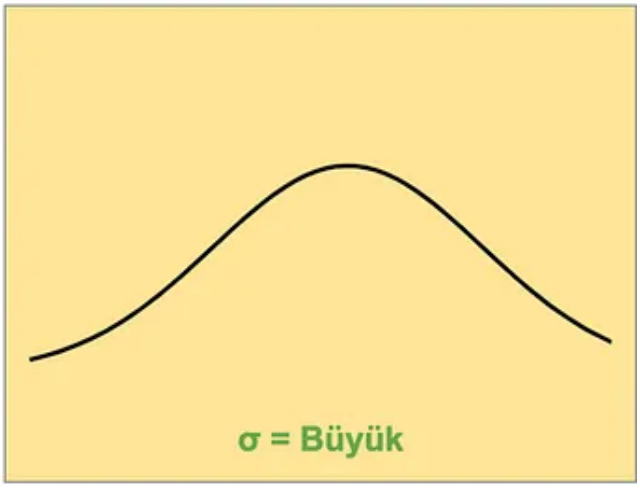
Normal Dağılım çan şeklinde özel bir yoğunluk eğrisidir. Bu sebeple çoğu zaman çan eğrisi olarak da isimlendirilmektedir. Normal dağılım, merkezi bir değer etrafında veri kümesi eğilimini tanımlar. Bu merkezi değer eğrinin her zaman orta noktasında bulunan popülasyon ortalama değeridir ***(μ)*** .

Normal dağılımda bazı veriler ortalamanın üzerinde , bazı veriler ortalamanın altında olabilir ancak verilerin büyük bir kısmının ortalama değere yakın alanlarda dağıldığı görülmektedir.

Popülasyonun standart sapması (σ) normal dağılımın yayılımını açıklar.

Standart sapma (σ) küçüldükçe eğri dikleşmeye, standart sapma büyüdükçe eğri düzleşmeye başlamaktadır. Bunun temel sebebi ise normal dağılım bir yoğunluk eğrisidir ve toplam alanı 1'e eşittir. Standart sapma arttıkça yükseklik azalmakta genişlik artmakta ancak toplam alan değişmemektedir.





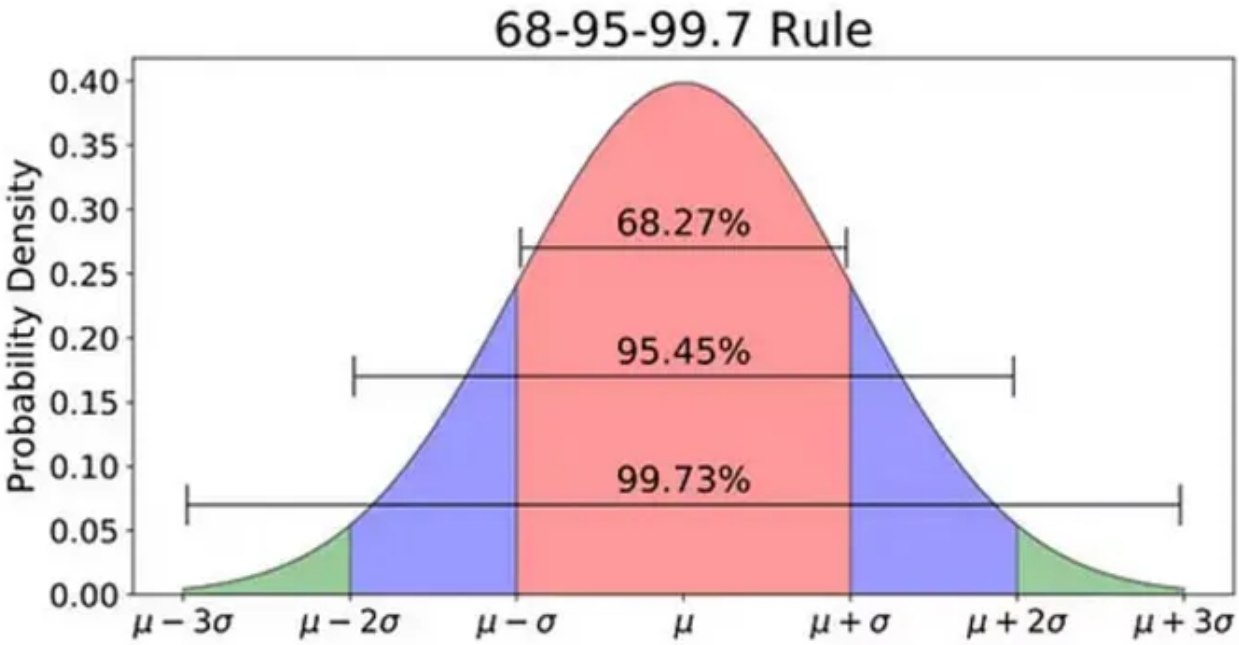
68–95- 99.7 Kuralı

Normal dağılımda en önemli veriler μ >ortalama ve σ>standart sapma verileridir.

Normal dağılımda ;

- +1(σ) ve -1(σ) arasındaki değerler arasında toplam popülasyonun %68,27'sinin olduğu
- +2(σ) ve -2(σ) arasındaki değerler arasında toplam popülasyonun %95.45'nin olduğu
- +3(σ) ve -3(σ) arasındaki değerler arasında toplam popülasyonun %99.7'nin olduğu görülmektedir.

Bu kural standart sapma değerinin küçük ya da büyük olmasından bağımsız her zaman böyle dağılmaktadır. 4(σ) ve -4(σ) sonrasında toplam popülasyonun çok az bir kısmı bulunduğu için daha sonrasındaki alanlar için bir alan hesaplaması yapılmamıştır.

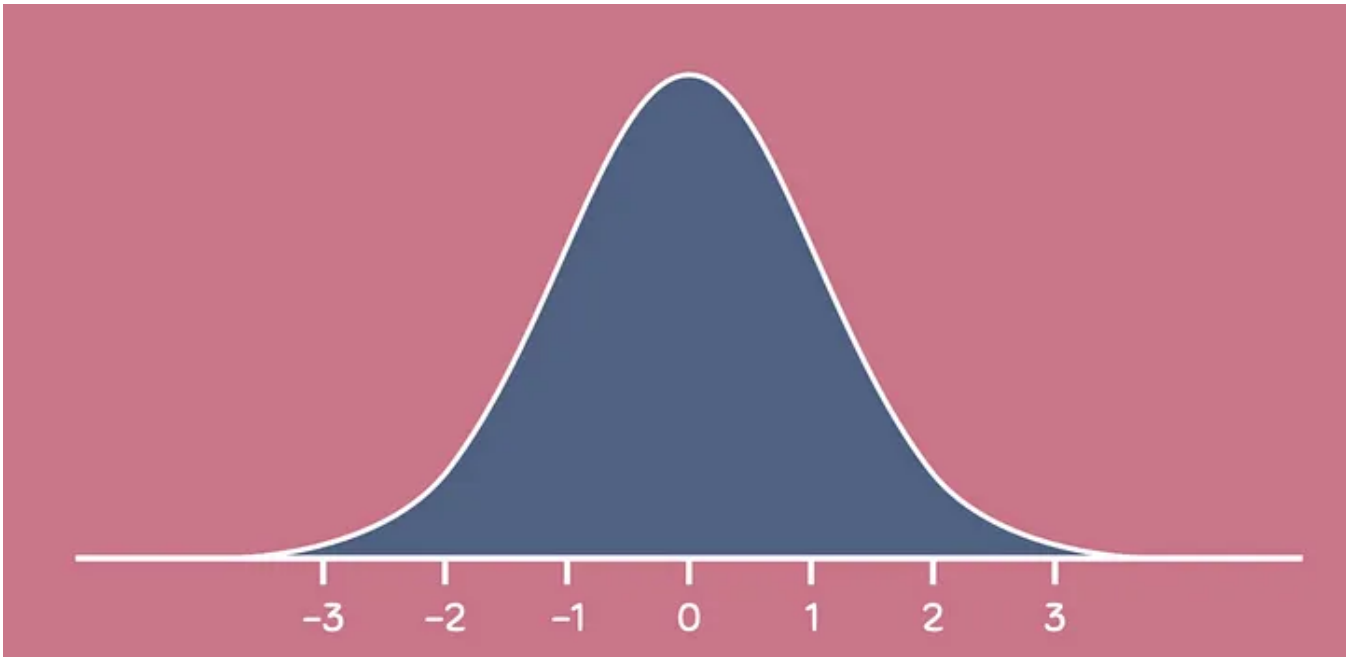


Standart Normal Dağılım

Standart Normal Dağılım μ=0 ve σ =1 olan özel bir normal dağılımdır.

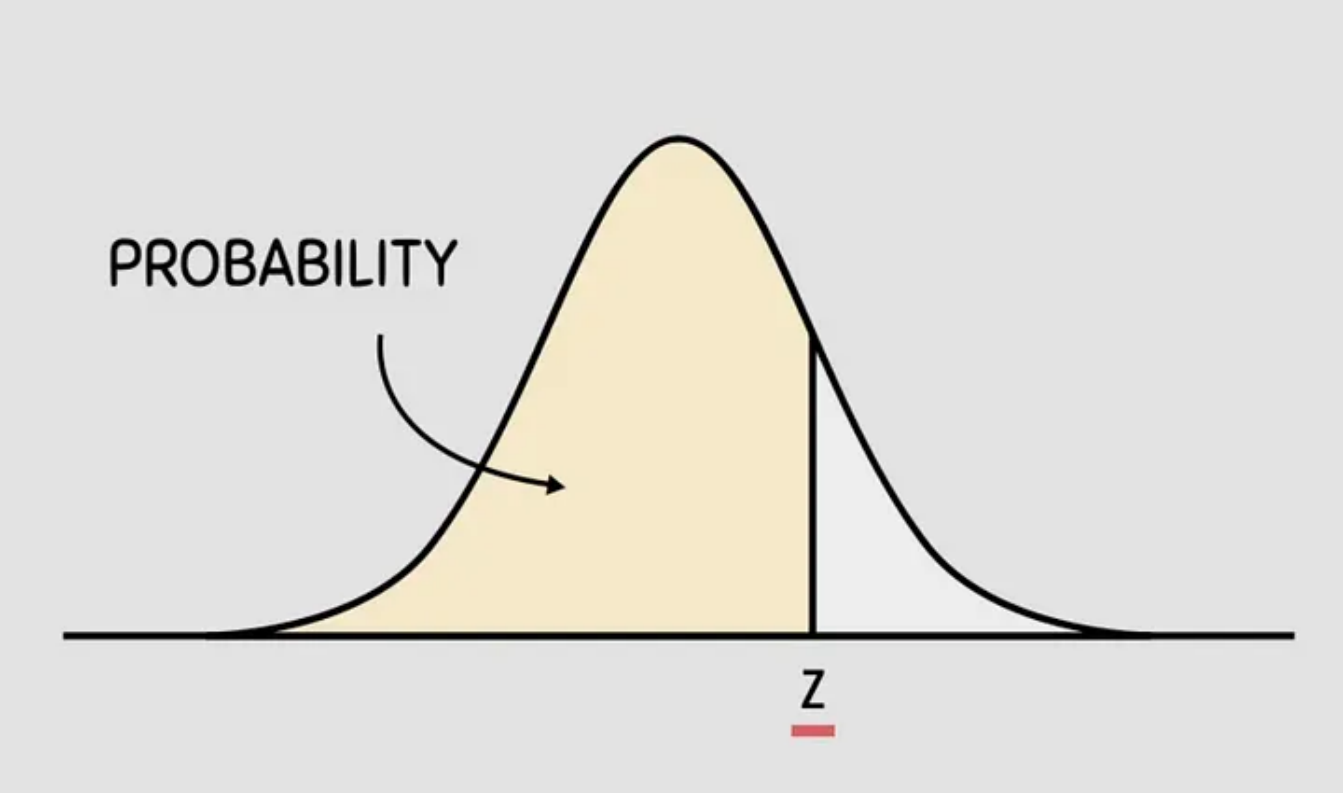
Standart Normal Dağılım'ın her zaman orta noktası (μ) 0'dır ve aralıklar birer birer artar.

Yatay eksendeki her sayı bir z-puanına karşılık gelir. z -puanı bize bir gözlemin ortalamadan(μ) kaç adet standart sapma uzak olduğunu göstermektedir.

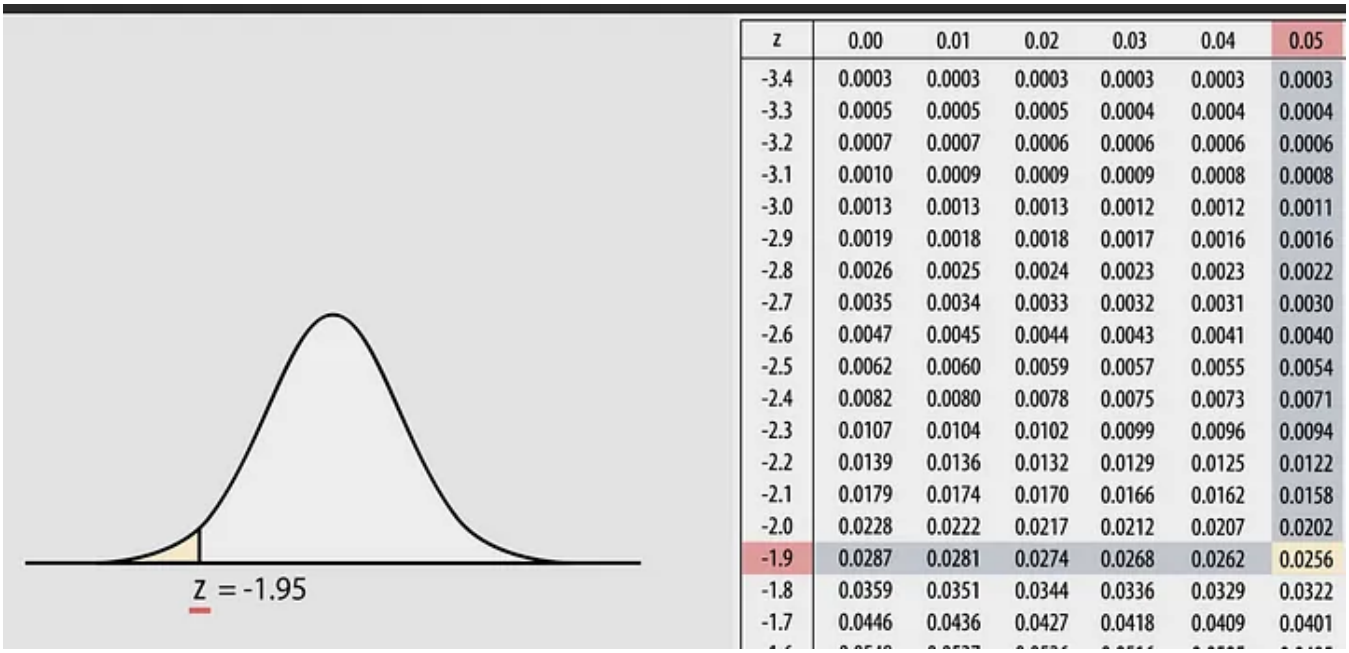


Standart Normal Dağılım

z-puan tablosu diğer adıyla standart dağılım tablosu sayesinde belirli bir z değerinin altında kalan alanı bulabiliriz.



Örneğin z puan değerinin -1.95 olduğu durumda z-puan tablosunda toplam popülasyonun %0.0256 oranla burada olduğu görülmektedir.



z	0.00	0.01	0.02	0.03	0.04	0.05
-3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
-3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004
-3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006
-3.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008
-3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011
-2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016
-2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022
-2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030
-2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040
-2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054
-2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071
-2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094
-2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122
-2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158
-2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202
-1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256
-1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322
-1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401

Burada unutulmaması gereken tüm normal dağılım tipleri μ ve σ'nin kaç olduğundan bağımsız olarak standart normal dağılıma dönüştürülebilmektedir. Bu dönüştürme işlemine **Standardizasyon** denilmektedir. Bu işlem bize z-puan

tablosunu kullanarak herhangi bir değer (gözlem) altındaki veya üstündeki toplam popülasyon oranını görmemizi sağlar.

$$z = \frac{x_i - \mu}{\sigma}$$

Standardizasyon Formülü

Formülde yer alan x(i) bizim gözlem değerimizdir.

Örneğin ; Bir matematik sınav sonuçlarının olduğu veri setimiz olduğunu düşünelim. Bu sınav sonucunda ortalamanın (μ) 60 olduğu ve standart sapmanın (σ) ise 10 olduğu tespit edilmiştir. Eğer 49 ve altında puan alan oranını bulmak istersek standardizasyon işlemi sonrası z-puan tablosunu kullanabiliriz.

$$Z = \frac{49 - 60}{10} = -1.1$$

z puanımız standardizasyon işlemi sonucunda -1.1 olarak bulundu. z -puan tablosuna bakıldığında toplam popülasyonun %13.57'sinin 49 ve daha altında puan aldığı tespit edilmiştir.

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
-1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
-1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
-1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
-1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379

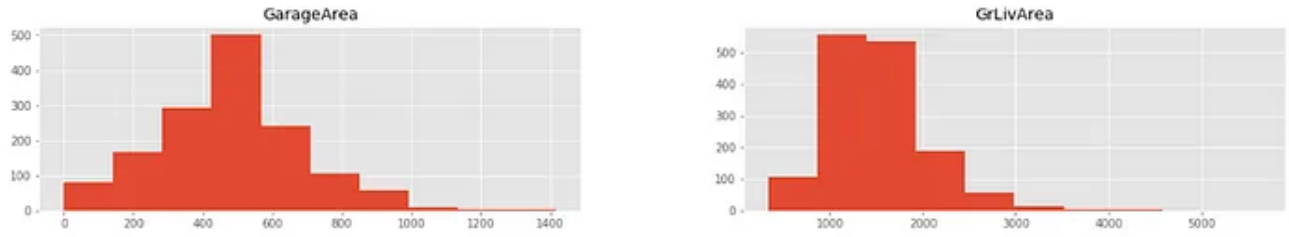
Standardizasyon işlemini Python'da StandartScaler fonksiyonu ile yapabiliriz. Makine öğrenmesi çalışmalarında en çok kullanılan houseprice veri setinde iki numeric değişkenin (GarageArea — GrLivArea)öncelikle dağılımına bakalım.

```
from sklearn import preprocessing
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv("/Users/gulcan.ogundur/Downloads/houseprice.csv")
df=df[['GarageArea','GrLivArea']]
df.head(5)
```

	GarageArea	GrLivArea
0	548	1710
1	460	1262
2	608	1786
3	642	1717
4	836	2198

```
continous_vars=df.describe().columns
print(continous_vars)
print("Continuous Variables Visualization","\n")
plt.style.use('ggplot')
df.hist(column=continous_vars,figsize=(20,3))
plt.show()
```

```
Index(['GarageArea', 'GrLivArea'], dtype='object')
Continuous Variables Visualization
```




```
df.describe()
```

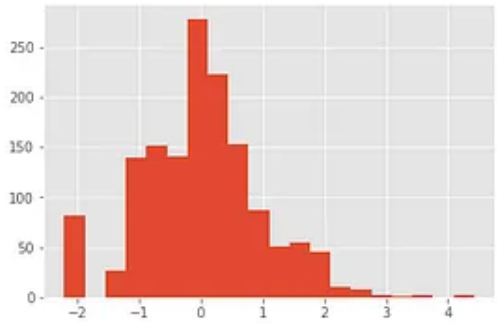
	GarageArea	GrLivArea
count	1460.0000	1460.0000
mean	472.9801	1515.4637
std	213.8048	525.4804
min	0.0000	334.0000
25%	334.5000	1129.5000
50%	480.0000	1464.0000
75%	576.0000	1776.7500
max	1418.0000	5642.0000

```
# Get column names first
names = df.columns
# Create the Scaler object
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
pd.set_option("display.precision", 4)
scaled_df = scaler.fit_transform(df)
scaled_df = pd.DataFrame(scaled_df, columns=names)
scaled_df.describe()
```

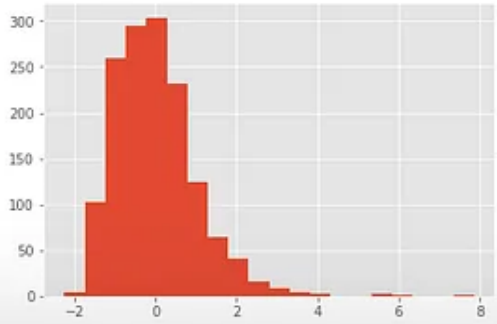
```
/anaconda3/lib/python3.7/site-packages/sklearn/preprocessing/data.py:625: DataConversion
Warning: Data with input dtype int64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/anaconda3/lib/python3.7/site-packages/sklearn/base.py:462: DataConversionWarning: Data
with input dtype int64 were all converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
```

	GarageArea	GrLivArea
count	1.4600e+03	1.4600e+03
mean	-2.0227e-17	-1.4463e-16
std	1.0003e+00	1.0003e+00
min	-2.2130e+00	-2.2491e+00
25%	-6.4792e-01	-7.3475e-01
50%	3.2844e-02	-9.7970e-02
75%	4.8201e-01	4.9740e-01
max	4.4215e+00	7.8556e+00

```
plt.style.use('ggplot')
plt.hist(scaled_df.GarageArea, bins=20)
plt.show()
```

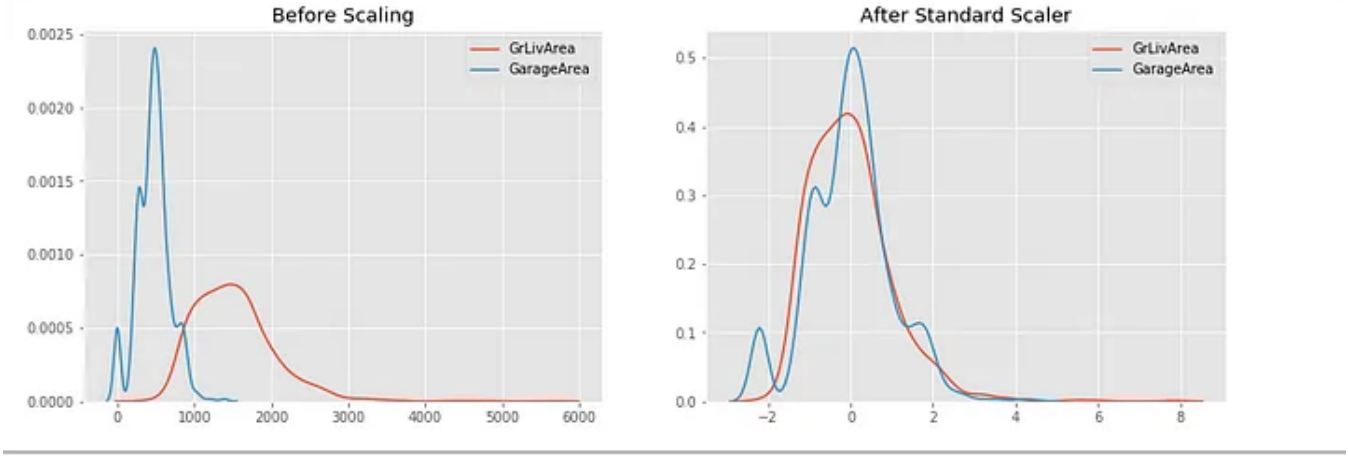



```
plt.style.use('ggplot')
plt.hist(scaled_df.GrLivArea, bins=20)
plt.show()
```



Standardizasyon işlemi sonrasında her iki verinin de normal dağılıma dönüştürüldüğü görülmektedir.

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(15, 5))
import seaborn as sns
ax1.set_title('Before Scaling')
sns.kdeplot(df['GrLivArea'], ax=ax1)
sns.kdeplot(df['GarageArea'], ax=ax1)
ax2.set_title('After Standard Scaler')
sns.kdeplot(scaled_df['GrLivArea'], ax=ax2)
sns.kdeplot(scaled_df['GarageArea'], ax=ax2)
plt.show()
```



Normalizasyon makine öğrenmesi projelerinde data hazırlama aşamasında yaygın olarak kullanılan bir işlemdir. Normalizasyonun amacı veri setindeki sayısal değişkenlerin değerlerini, değerler aralığındaki farklılıkları bozmadan ortak bir ölçeğe dönüştürmektir. Her veri setinin normalizasyon işlemine ihtiyacı yoktur. Sadece farklı aralıkta dağılan sayısal değişkenleri olduğunda gereklidir.

Örneğin, yaş (x1) ve gelir (x2) olmak üzere iki özellik içeren bir veri kümesini düşünün. Yaş 0–100 arasında değişirken, gelir 20,000 -500.000 arasında değişmektedir. Bu iki özellik çok farklı aralıklardadır. Örneğin, çok değişkenli doğrusal regresyon gibi bir analiz yaptığımızda, gelir değişkeni daha büyük değeri nedeniyle sonucu daha fazla etkileyecektir. Ancak bu durum gelir değişkenin daha önemli bir tahminleyici olduğu anlamına gelmez. Bunu önlemek amacıyla Normalizasyon işlemi yapmamız gerekmektedir.

Normalizasyon işlemi sonucunda tüm sayısal veriler 0 ile 1 arasında değer alacaktır. Eğer negatif veriler varsa -1 ile 1 arasında dağılacaktır.

$$z = \frac{x \ - \min(x)}{\max(x) \ - \min(x)}$$

Normalizasyon formülü

Normalizasyon için python'da sklearn kütüphanesinden MinMaxScaler yüklememiz gereklidir. data veri setindeki sayısal değerleri MinMaxScaler ile 0–1 aralığına dönüştürebiliriz.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)
```

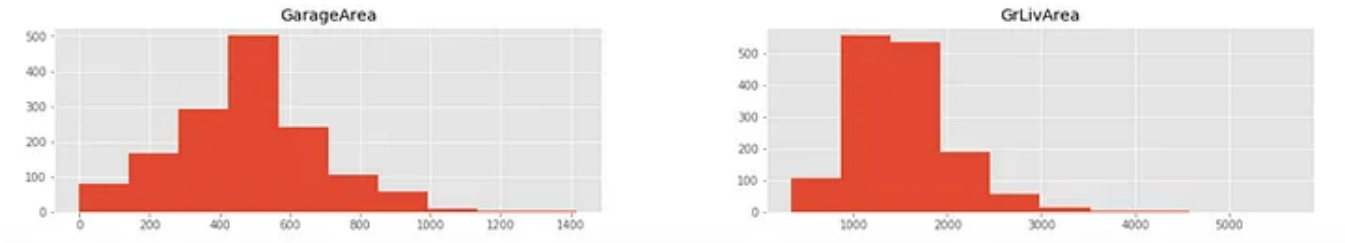
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
```

```
df=pd.read_csv("/Users/gulcan.ogundur/Downloads/houseprice.csv")
df=df[['GarageArea','GrLivArea']]
df.head(5)
```

	GarageArea	GrLivArea
0	548	1710
1	460	1262
2	608	1786
3	642	1717
4	836	2198

```
continous_vars=df.describe().columns
print(continous_vars)
print("Continuous Variables Visualization","\n")
plt.style.use('ggplot')
df.hist(column=continous_vars,figsize=(20,3))
plt.show()
```

```
Index(['GarageArea', 'GrLivArea'], dtype='object')
Continuous Variables Visualization
```



Normalizasyon işlemi sonucunda verilerin 0 -1 arasında dağıldığını görebiliriz.

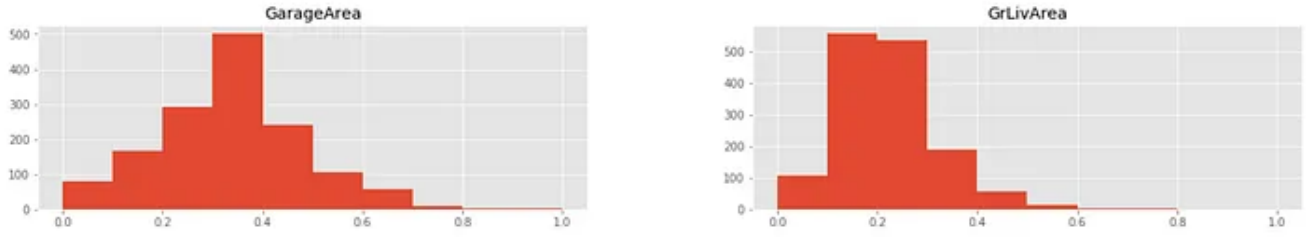
```
scaler=MinMaxScaler()
data_scaled=pd.DataFrame(scaler.fit_transform(df),columns=df.columns)
data_scaled.head(5)
```

```
/anaconda3/lib/python3.7/site-packages/sklearn/preprocessing/data.py:323: DataConversionWarning: Data with input dtyp
e int64 were all converted to float64 by MinMaxScaler.
  return self.partial_fit(X, y)
```

	GarageArea	GrLivArea
0	0.386460	0.259231
1	0.324401	0.174830
2	0.428773	0.273549
3	0.452750	0.260550
4	0.589563	0.351168

```
continous_vars=data_scaled.describe().columns
print(continous_vars)
print("Continuous Variables Visualization","\n")
plt.style.use('ggplot')
data_scaled.hist(column=continous_vars,figsize=(20,3))
plt.show()
```

```
Index(['GarageArea', 'GrLivArea'], dtype='object')
Continuous Variables Visualization
```



Standardization | Z Score | Normal Distribution | Normalization

Following

Written by Gülcan Öğündür

682 Followers

Business Intelligence Specialist at sahibinden.com in Istanbul. MS, Big Data and Business Analytics.
linkedin.com/in/gulcanogundur/

More from Gülcan Öğündür