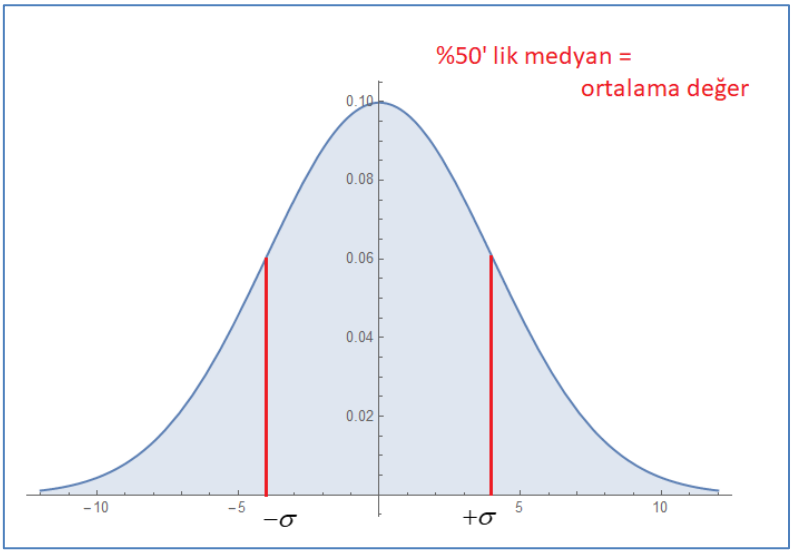
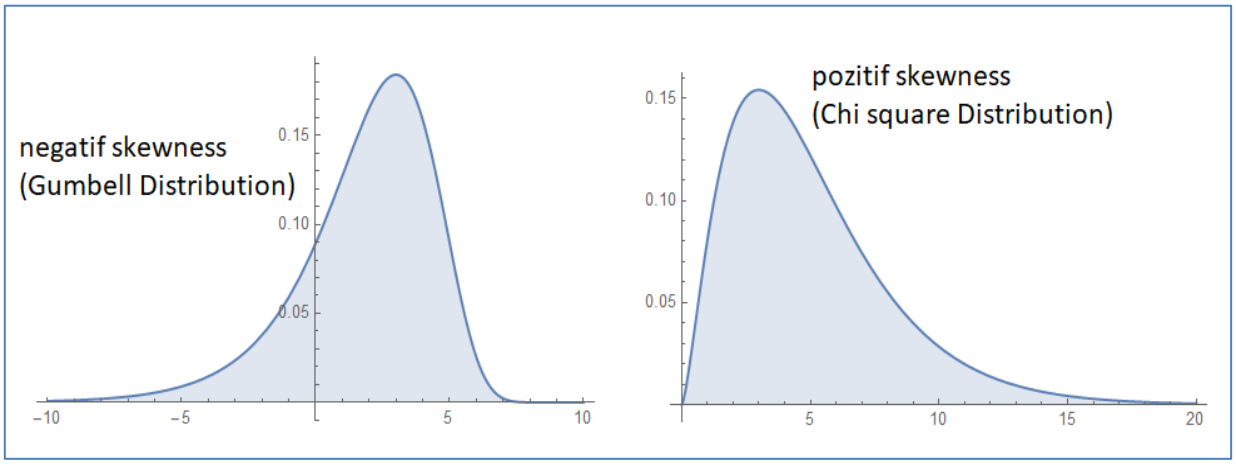
Skewness Durumlar *[İlgili jpynb dosyası bu nota ektir.]*

Doğadaki birçok olayın davranışı, Gauss dağılımı (Normal dağılıma) uygun davranır. Normal dağılım için %50' lik medyan değeri ile ortalama değer aynı değerdir.



Eğer veri setinin dağılımı, sağa doğru yığılmışsa bu durumda dağılımın kuyruk kısmı sola doğru uzamış demektir ve bu ***negatif çarpıklık (negative skewness)*** olarak adlandırılır. Tersi durumda Gauss dağılımı sola doğru yatık ise kuyruk kısmı sağa doğru uzamış demektir ve bu durum da ***pozitif çarpıklık (positive skewness)*** olarak adlandırılır.



Skewness, bir veri dağılımının simetrisini ölçen bir parametredir. Normal dağılıma sahip veriler için skewness yaklaşık olarak sıfır olmalıdır. Eğer veri dağılımı tek tepe noktalı ve sürekli ise, pozitif bir skewness değeri, dağılımın sağ kuyruğunda daha fazla ağırlık olduğunu gösterir.

Python'da skewness'i hesaplamak için `scipy.stats` kütüphanesindeki `skew` fonksiyonunu kullanabilir.

Skewness düzeltilmesi, veri analizi ve modelleme süreçlerinde önemlidir. İşte bazı yaygın skewness düzeltilme yöntemleri:

Box-Cox Dönüşümü: `Box-Cox` dönüşümü, veri setini normal dağılıma yaklaştırmak için kullanılır. Bu yöntem, pozitif ve negatif değerler içeren veri setleri için uygundur. Ancak, veri setinizde sıfır veya negatif değerler varsa `Box-Cox` dönüşümü uygun değildir.

Yeo-Johnson Dönüşümü: `Yeo-Johnson` dönüşümü, `Box-Cox` dönüşümünün genelleştirilmiş bir versiyonudur. Bu yöntem, pozitif, negatif ve sıfır değerler içeren veri setleri için uygundur. `Yeo-Johnson` dönüşümü, `scipy.stats` kütüphanesinde bulunan `PowerTransformer` sınıfı ile uygulanır.

Logaritmik Dönüşüm: Veri setiniz pozitif değerler içeriyorsa, logaritmik dönüşüm kullanabilirsiniz. Bu yöntem, veriyi daha simetrik hale getirir. Logaritmik düzeltme işleminde x-ekseninin menzili azaltılacağından çarpıklığı yani skewness durumları indirgemek için faydalı olabilir. Ancak veri setinde 0 (sıfır) olan değerler varsa bunlar nan (not a number – sayı olmayan) veri ile değiştirilmelidir. Aksi halde matematiksel olarak tanımsız durumlar oluşur. Farklı bir durum olarak nan olan veriler 1 (bir) ile değiştirilebilir. Bu durumda bunların logaritması 0 olacağından matematiksel olarak tanımsız durum oluşmaz.

Kare Kök Dönüşümü: Kare kök dönüşümü, pozitif değerler içeren veri setleri için kullanışlıdır. Veriyi daha normal dağılıma benzetmek için uygulanır. Değerlerin kara kökü alınarak verilerin daha küçük bir skalada değer alması sağlanabilir.

Rank Dönüşümü: Rank dönüşümü, veri setini sıralayarak orijinal değerlerin yerine sıralama değerlerini kullanır. Bu yöntem, veri setinin dağılımını düzeltebilir.

Örnek

Verilmiş data seti için Jupyter Notebook kullanarak skewness vurumları irdeleyiniz.

Çözüm

Detaylı çözüm Yapay zeka-4 ders notu Jupyter Notebook dosyasında yapılmıştır.

```
skewness_degerleri = df.skew()
skewness_degerleri
```

```
Energy    0.368923
target    0.784580
dtype: float64
```

Energy (Enerji) Kolonu:

Skewness değeri 0.368923 olarak hesaplanmış.

Bu değer, hafif bir sağa çekilme (sağa eğilim) olduğunu gösterir. Yani enerji değerleri sağ kuyrukta biraz daha yoğunlaşmış gibi görünüyor.

Enerji değerlerinin dağılımını daha simetrik hale getirmek için Yeo-Johnson dönüşümünü veya diğer yöntemleri deneyebilirsiniz.

Target (Hedef) sütunu:

Skewness değeri 0.784580 olarak hesaplanmış.

Bu değer de hafif bir sağa çekilme olduğunu gösterir. Hedef değerlerinin dağılımını düzeltmek için aynı yöntemleri kullanabilirsiniz.

Skewness değerleri, derin öğrenme modelleri üzerinde doğrudan bir etki yapmaz. Ancak, model performansını değerlendirmek için dikkate alınması gereken bir faktördür. Veri setinizin özelliklerine bağlı olarak en uygun dönüşüm yöntemini seçebilir.

Box-Cox Dönüşümü:

- Box-Cox dönüşümü, veri setini normal dağılıma yaklaştırmak için kullanılır.
- Bu dönüşüm, yalnızca pozitif değerler içeren veri setleri için uygundur.
- Lambda (λ) parametresi, dönüşümün doğrusallığını kontrol eder. Genellikle otomatik olarak tahmin edilir.
- Box-Cox dönüşümü, veri setinin simetrisini artırır ve regresyon modelleri gibi istatistiksel analizlerde kullanışlıdır.

$$T_{BC} = \begin{cases} \dfrac{y^{\lambda}-1}{\lambda}, & eğer \quad \lambda \neq 0 \\ \ln(y), & eğer \quad \lambda = 0 \end{cases}$$

Yeo-Johnson Dönüşümü:

- Yeo-Johnson dönüşümü, Box-Cox dönüşümünün genelleştirilmiş bir versiyonudur.
- Bu dönüşüm, pozitif, negatif ve sıfır değerler içeren veri setleri için uygundur.
- Yeo-Johnson, Box-Cox'tan farklı olarak pozitif ve negatif değerler için ayrı bir güç parametresi kullanır.
- Yeo-Johnson dönüşümü, veri setinin dağılımını daha simetrik hale getirir ve regresyon analizleri için tercih edilir.
- Bu dönüşümler, veri setinizin özelliklerine bağlı olarak seçilmelidir. İdeal olarak, hangi dönüşümün veri setiniz için daha iyi çalıştığını test edilmelidir.

$$T_{YL} = \begin{cases} \dfrac{(y+1)^{\lambda}-1}{\lambda}, & eğer \quad y \geq 0 \quad ve \quad \lambda \neq 0 \\ \ln(y+1), & eğer \quad y \geq 0 \quad ve \quad \lambda = 0 \\ -\dfrac{(-y+1)^{2-\lambda}-1}{2-\lambda}, & eğer \quad y < 0 \quad ve \quad \lambda \neq 2 \\ -\ln(-y+1), & eğer \quad y < 0 \quad ve \quad \lambda = 2 \end{cases}$$

Python da Yeo-Johnson dönüşümünü kullanmak amacıyla

```
from sklearn.preprocessing import PowerTransformer
```

kullanılır.

```
pt = PowerTransformer(method= “yeo-johnson”)
veri_duzeltilmis = pt.fit_transform(veri)
```

Makine öğrenmesi işlemi gerçekleştirildikten sonra elde edilen sonuçları orijinal veri setine dönüştürmek gerekir bu amaçla ters dönüşüm kullanılmalıdır:

```
veri_ters_duzeltilmis = pt.inverse_transform(veri_duzeltilmis)
```

ile sağlanır.

Makine öğrenmesi ve Derin öğrenme algoritmaları için veri ön işleme çok önemlidir. Bu veri ön işleme işleminden sonra yapay öğrenme için veri setinin 0-1 arasına dönüştürülmesi öğrenme oranını ciddi ve olumlu anlamda etkiler. Bu amaçla sklearn modülünden MinMax ölçeklendirme işlemi yapılır. Öğretme işleminden sonra üretilen modele ters dönüşüm uygulanarak gerçek veri düzeyine dönüştürülür. Bu durumda skewness düzeltmesi ile Min Max dönüşümünden hangisinin daha önce yapılması gerektiğini seçmek önemlidir. İ

1- Skewness Düzeltilmesi:

- Veri setinizde skewness varsa, bu durumu düzeltmek için bir dönüşüm yöntemi kullanmalısınız.
- Örneğin, Yeo-Johnson dönüşümünü uygulayabilirsiniz. Bu, veri setinin dağılımını daha simetrik hale getirecektir.
- Skewness düzeltilmiş veri setini kullanarak derin öğrenme modelinizi eğitebilirsiniz.

2- MinMax Ölçeklendirme:

- Veri setinizin özelliklerini aynı ölçekte olmasını sağlamak için MinMax ölçeklendirme kullanabilirsiniz.
- MinMaxScaler, her özelliği belirli bir aralığa (genellikle 0 ile 1 arasına) ölçekler.
- Bu, derin öğrenme modelleri için önemlidir, çünkü bazı algoritmalar özelliklerin ölçeğine duyarlıdır.

Bu nednele önce skewness düzeltilmesinin yapılması ve ardından da MinMax ölçeklendirmesinin yapılması mantıklı bir sıralama olacaktır.