

# Dataset Normalizer Programı

## Normalizasyon nedir ?

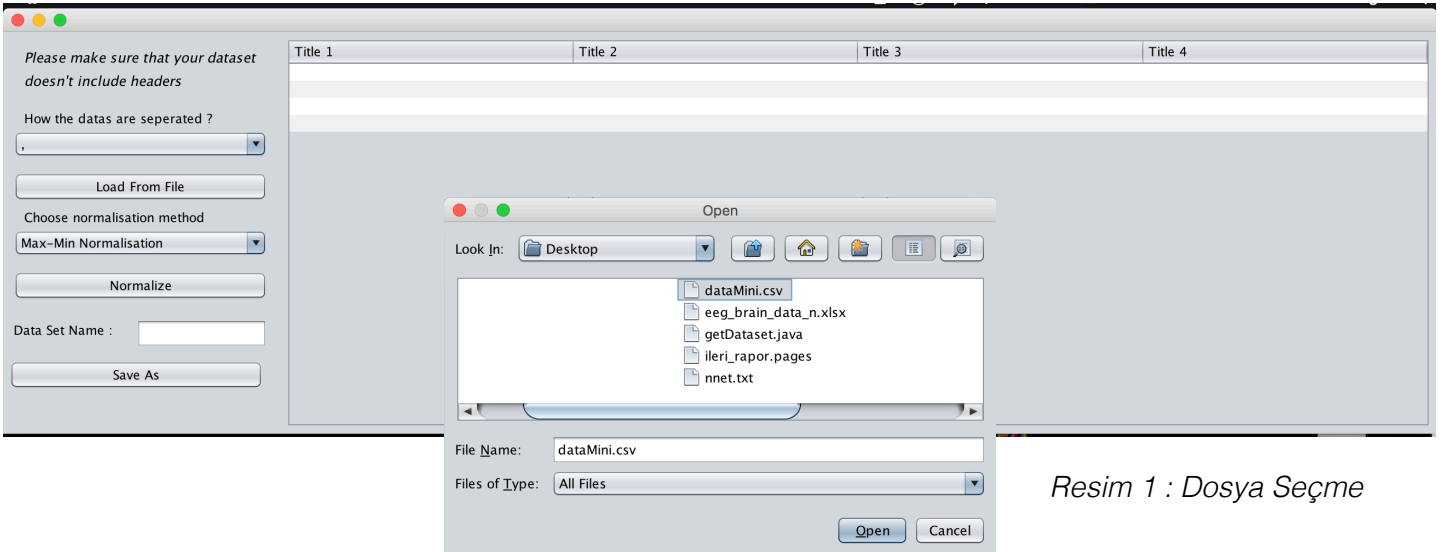
İstatistikte ve istatistik uygulamalarında, normalleştirme bir dizi anlam ifade edebilir. En basit hallerde, derecelendirmelerin normalleştirilmesi, farklı ölçeklerde ölçülen değerlerin, genellikle ortalamadan önce, kavramsal olarak ortak bir ölçekte ayarlanması anlamına gelir. Daha karmaşık durumlarda, normalizasyon, düzeltilmiş değerlerin tüm olasılık dağılımlarını hizaya getirmek amacıyla daha sofistike düzenlemelere işaret edebilir.

## Dataset (Veri Seti) nedir ?

İstatistikte, veri setleri genellikle istatistiksel bir popülasyondan örnek alınarak elde edilen gerçek gözlemlerden gelen verileri içeren setlerdir, ve her satır bu popülasyonun bir elemanı üzerindeki gözlemlere karşılık gelir. Veri setleri, belirli yazılım türlerini test etmek amacıyla algoritmalar ile üretilebilir.

## Program

Program, kullanıcının seçtiği dataset'in içindeki verileri, seçtiği normalizasyon metodu ile normalize etmesini sağlıyor. Kullanıcı, datalarının önceki ve sonraki hallerini tablo'da görebiliyor.



Resim 1 : Dosya Seçme

Please make sure that your dataset doesn't include headers														
How the datas are seperated ?														
,														
Load From File														
Choose normalisation method														
Max-Min Normalisation														
Normalize														
Data Set Name :														
Save As														
Kolon 0	Kolon 1	Kolon 2	Kolon 3	Kolon 4	Kolon 5	Kolon 6	Kolon 7	Kolon 8	Kolon 9	Kolon 10	Kolon 11	Kolon 12	Kolon 13	Kolon 14
0.0	0.0	56.0	43.0	278.0	301963.0	90612.0	33735.0	23991.0	27946.0	45097.0	33228.0	8293.0	0.0	0.0
0.0	0.0	40.0	35.0	-50.0	73787.0	28083.0	1439.0	2240.0	2746.0	3687.0	5293.0	2740.0	0.0	0.0
0.0	0.0	47.0	48.0	101.0	758353.0	383745.0	201999.0	62107.0	36293.0	130536.0	57243.0	25354.0	0.0	0.0
0.0	0.0	47.0	57.0	-5.0	201224...	129350.0	61236.0	17084.0	11488.0	62462.0	49960.0	33932.0	0.0	0.0
0.0	0.0	44.0	53.0	-8.0	100514...	354328.0	37102.0	88881.0	45307.0	99603.0	44790.0	29749.0	0.0	0.0
0.0	0.0	44.0	66.0	73.0	178644...	176766.0	59352.0	26157.0	15054.0	33669.0	33782.0	31750.0	0.0	0.0
0.0	0.0	43.0	69.0	130.0	635191.0	122446.0	90107.0	65072.0	36230.0	53019.0	62938.0	59307.0	0.0	0.0
0.0	0.0	40.0	61.0	-2.0	161098.0	12119.0	1963.0	809.0	1277.0	3186.0	3266.0	2518.0	0.0	0.0
0.0	0.0	43.0	69.0	17.0	492796.0	120998.0	63697.0	68242.0	10769.0	88403.0	73756.0	22676.0	0.0	0.0
0.0	0.0	47.0	69.0	-59.0	82048.0	116131.0	47317.0	26197.0	41642.0	28866.0	32551.0	41810.0	0.0	0.0
0.0	0.0	48.0	38.0	-14.0	757165.0	186196.0	3242.0	3841.0	18854.0	43021.0	46799.0	11928.0	0.0	0.0
0.0	0.0	44.0	48.0	72.0	667513.0	141854.0	75050.0	16234.0	45926.0	34496.0	74875.0	31839.0	0.0	0.0
0.0	0.0	47.0	34.0	121.0	165360.0	42119.0	3158.0	6256.0	7270.0	19462.0	10984.0	8148.0	0.0	0.0
0.0	0.0	48.0	30.0	-52.0	737665.0	84275.0	2235.0	38748.0	21705.0	28343.0	80927.0	24735.0	0.0	0.0
0.0	0.0	47.0	48.0	69.0	577024.0	179555.0	25937.0	21604.0	43790.0	44432.0	69203.0	25586.0	0.0	0.0
0.0	0.0	57.0	43.0	44.0	234964.0	80944.0	7991.0	32568.0	11828.0	63682.0	41013.0	18147.0	0.0	0.0
0.0	0.0	57.0	40.0	144.0	671467.0	133227.0	7142.0	14300.0	23373.0	65591.0	47860.0	16501.0	0.0	0.0
0.0	0.0	70.0	40.0	25.0	33290.0	14729.0	10980.0	10407.0	14316.0	37112.0	60378.0	36306.0	0.0	0.0
0.0	0.0	74.0	44.0	-138.0	102541...	268122.0	67285.0	36996.0	10083.0	80826.0	90350.0	26345.0	0.0	0.0
0.0	0.0	67.0	53.0	25.0	756442.0	97387.0	59785.0	32241.0	16122.0	53157.0	44602.0	14331.0	0.0	0.0
0.0	0.0	70.0	66.0	124.0	87109.0	7200.0	4987.0	2817.0	2130.0	11413.0	9422.0	4922.0	0.0	0.0

Resim 2 : Ham datayı görüntüleme

Please make sure that your dataset doesn't include headers														
How the datas are seperated ?														
,														
Load From File														
Choose normalisation method														
Max-Min Normalisation														
Normalize														
Data Set Name :														
Save As														
Kolon 0	Kolon 1	Kolon 2	Kolon 3	Kolon 4	Kolon 5	Kolon 6	Kolon 7	Kolon 8	Kolon 9	Kolon 10	Kolon 11	Kolon 12	Kolon 13	Kolon 14
0.0	0.0	0.7567...	0.6231...	0.9719...	0.1500...	0.1195...	0.1369...	0.1817...	0.6085...	0.3454...	0.3514...	0.1398...	0.0	0.0
0.0	0.0	0.5405...	0.5072...	0.2056...	0.0366...	0.0370...	0.0058...	0.0169...	0.0597...	0.0282...	0.0559...	0.0462...	0.0	0.0
0.0	0.0	0.6351...	0.6956...	0.5584...	0.3768...	0.5060...	0.8198...	0.4704...	0.7902...	1.0	0.6054...	0.4275...	0.0	0.0
0.0	0.0	0.6351...	0.8260...	0.3107...	1.0	0.1705...	0.2485...	0.1294...	0.2501...	0.4785...	0.5284...	0.5721...	0.0	0.0
0.0	0.0	0.5945...	0.7681...	0.3037...	0.4995...	0.4672...	0.1505...	0.6732...	0.9865...	0.7630...	0.4737...	0.5016...	0.0	0.0
0.0	0.0	0.5945...	0.9565...	0.4929...	0.8877...	0.2331...	0.2408...	0.1981...	0.3277...	0.2579...	0.3573...	0.5353...	0.0	0.0
0.0	0.0	0.5810...	1.0	0.6261...	0.3156...	0.1614...	0.3657...	0.4929...	0.7888...	0.4061...	0.6657...	1.0	0.0	0.0
0.0	0.0	0.5405...	0.8840...	0.3177...	0.0800...	0.0159...	0.0079...	0.0061...	0.0278...	0.0244...	0.0345...	0.0424...	0.0	0.0
0.0	0.0	0.5810...	1.0	0.3621...	0.2448...	0.1595...	0.2585...	0.5169...	0.2344...	0.6772...	0.7801...	0.3823...	0.0	0.0
0.0	0.0	0.6351...	1.0	0.1845...	0.0407...	0.1531...	0.1920...	0.1984...	0.9067...	0.2211...	0.3443...	0.7049...	0.0	0.0
0.0	0.0	0.6486...	0.5507...	0.2897...	0.3762...	0.2455...	0.0131...	0.0290...	0.4105...	0.3295...	0.4950...	0.2011...	0.0	0.0
0.0	0.0	0.5945...	0.6956...	0.4906...	0.3317...	0.1870...	0.3046...	0.1229...	1.0	0.2642...	0.7919...	0.5368...	0.0	0.0
0.0	0.0	0.6351...	0.4927...	0.6051...	0.0821...	0.0555...	0.0128...	0.0473...	0.1582...	0.1490...	0.1161...	0.1373...	0.0	0.0
0.0	0.0	0.6486...	0.4347...	0.2009...	0.3665...	0.1111...	0.0090...	0.2935...	0.4726...	0.2171...	0.8559...	0.4170...	0.0	0.0
0.0	0.0	0.6351...	0.6956...	0.4836...	0.2867...	0.2367...	0.1052...	0.1636...	0.9534...	0.3403...	0.7319...	0.4314...	0.0	0.0
0.0	0.0	0.7702...	0.6231...	0.4252...	0.1167...	0.1067...	0.0324...	0.2466...	0.2575...	0.4878...	0.4338...	0.3059...	0.0	0.0
0.0	0.0	0.7702...	0.5797...	0.6588...	0.3336...	0.1757...	0.0289...	0.1083...	0.5089...	0.5024...	0.5062...	0.2782...	0.0	0.0
0.0	0.0	0.9459...	0.5797...	0.3808...	0.0165...	0.0194...	0.0445...	0.0788...	0.3117...	0.2843...	0.6386...	0.6121...	0.0	0.0
0.0	0.0	1.0	0.6376...	0.0	0.5095...	0.3536...	0.2730...	0.2802...	0.2195...	0.6191...	0.9556...	0.4442...	0.0	0.0
0.0	0.0	0.9054...	0.7681...	0.3808...	0.3759...	0.1284...	0.2426...	0.2442...	0.3510...	0.4072...	0.4717...	0.2416...	0.0	0.0
0.0	0.0	0.9459...	0.9565...	0.6121...	0.0432...	0.0094...	0.0202...	0.0213...	0.0463...	0.0874...	0.0996...	0.0829...	0.0	0.0

Resim 3 : Seçilen metoda göre normalize edilmiş datayı görüntüleme.

## Program Altyapısı

```
public HashMap<Integer, ArrayList> getData(String url, String separator) {  
    try {  
        br = new BufferedReader(new FileReader(url));  
        String line;  
        int iteration;  
        iteration = 0;  
        while ((line = br.readLine()) != null) {  
            ArrayList<Double> satirList = new ArrayList<>();  
            String[] elements = line.split(separator); // line'ı separator'e göre ayırır  
            System.out.println("\t\t\tsatır no -> " + iteration + "\t");  
  
            for (int i = 0; i < elements.length; i++) { // i = kolon no  
                satirList.add(Double.valueOf(elements[i])); // line'in seçilen kolonlarını(i) arrayList'te double formatında tutar  
                System.out.println("Sutun no -> " + i + " => " + Double.valueOf(elements[i]));  
            }  
            data.put(iteration, satirList); // format -> key(1)/value(satirList) -> 1. satır/o satırdaki elemanların listesi  
            iteration++;  
        }  
    } catch (FileNotFoundException e) {}  
    } catch (IOException e) {}  
    finally {  
        if (br != null) {  
            try {  
                br.close();  
            } catch (IOException e) {}  
        }  
    }  
}  
  
return data;  
}
```

**getData** metodu ile datayı belirlenen path'ten okuyor ve belirlenen ayırma yöntemi kullanılarak her bir veri bir ArrayList'in içinde saklanıyor. Sonra datanın satırlarını tutan her ArrayList; index numarası, satır numarasını belli edecek şekilde bir HashMap'te saklanıyor.

```
public void fillTable(HashMap<Integer, ArrayList> data) {
    dtm.setRowCount(0);
    dtm.setColumnCount(0);
    Object[] row = new Object[data.get(0).size()];
    Object[] kolon = new Object[data.size()];

    for (int i = 0; i < data.get(0).size(); i++) {
        for (int j = 0; j < data.size(); j++) {
            kolon[j] = data.get(j).get(i);
        }
        dtm.addColumn("Kolom " + i, kolon);
    }
    jTable.setModel(dtm);
}
```

**fillTable** metodu ile, `getData`'dan elde edilen `HashMap` gezilerek tablo dolduruluyor.

```

public void maxMinNormalization(HashMap<Integer, ArrayList> data) {
    for (int i = 0; i < data.get(0).size(); i++) {
        double minValue = (double) data.get(i).get(0);
        double maxValue = (double) data.get(i).get(0);
        Double normalizedValue = null;
        for (int j = 0; j < data.size(); j++) {
            if ((double) data.get(j).get(i) < minValue) {
                minValue = (double) data.get(j).get(i);
            }
            if ((double) data.get(j).get(i) > maxValue) {
                maxValue = (double) data.get(j).get(i);
            }
        }
        for (int j = 0; j < data.size(); j++) {
            double currentValue = (double) data.get(j).get(i);
            System.out.println("currentValue = " + (double) currentValue);
            if (maxValue - minValue == 0) {
                normalizedValue = 0.0;
            } else {
                normalizedValue = ((double) (currentValue - minValue)) / (maxValue - minValue);
            }
            System.out.println("Normalized value is -> " + (double) normalizedValue);
            data.get(j).remove(i);
            data.get(j).add(i, normalizedValue);
        }
    }
}

```

**maxMinNormalization** metodu ile, her kolon'un en büyük ve en küçük verileri bulunup; bulunan verilere göre bütün kolon 0 ve 1 arası bir sayıya dönüştürülüyor. HashMap'in üzerine yazılıyor.

```

public void writeDataset(HashMap<Integer, ArrayList> data, String fileName) {
    PrintWriter writer = null;
    try {
        File desktopDir = new File(System.getProperty("user.home"), "Desktop");
        System.out.println(desktopDir.getPath());

        writer = new PrintWriter(desktopDir.getPath().toString() + "/" + fileName + ".csv", "UTF-8");

        for (int i = 0; i < data.size(); i++) {
            for (int j = 0; j < data.get(i).size(); j++) {
                writer.print(data.get(i).get(j));
                writer.append(",");
            }
            System.out.println("");
            writer.println();
        }
    } catch (FileNotFoundException ex) {
        Logger.getLogger(GetDataset.class.getName()).log(Level.SEVERE, null, ex);
    } catch (UnsupportedEncodingException ex) {
        Logger.getLogger(GetDataset.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        writer.close();
    }
}

```

**writeDataset** metodu ile, kullanıcının kaydedilecek dosya için belirlediği isimle, kullanıcının masaüstü klasörüne kayıt yapılıyor. Kayıt, csv(Comma Separated Values) formatında gerçekleştiriliyor.