

**EGE UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

**204 DATA STRUCTURES (3+1)**

**2020–2021 FALL SEMESTER**

**PROJECT-1 REPORT**

**(Arrays, Matrices, Methods, Random Numbers)**

**DELIVERY DATE**

12/12/2020

**PREPARED BY**

05190000841, Veli Yaşar

İçindekiler

[1) POINTS IN A 2D PLANE 2](#_Toc57390051)

[1.a Rastgele Nokta Üretimi 2](#_Toc57390052)

[1.a.1 Kodlar 2](#_Toc57390053)

[1.a.2 Ekran görüntüleri 2](#_Toc57390054)

[1.a.3 Açıklama 2](#_Toc57390055)

[1.b Uzaklık Matrisi 2](#_Toc57390056)

[1.b.1 Kodlar 2](#_Toc57390057)

[1.b.2 Ekran görüntüleri 2](#_Toc57390058)

[1.b.3 Açıklama 2](#_Toc57390059)

[2) CLASSIFICATION USING K-NEAREST NEIGHBORS (KNN) ALGORITHM 2](#_Toc57390060)

[2.a KNN ile sınıflandırma 2](#_Toc57390061)

[2.a.1 Algoritma sözde-kodu 2](#_Toc57390062)

[2.b Banknot sınıflandırma 2](#_Toc57390063)

[2.b.1 Kodlar 2](#_Toc57390064)

[2.b.2 Ekran görüntüleri 2](#_Toc57390065)

[2.b.3 Açıklama 2](#_Toc57390066)

[2.c Başarı ölçümü 2](#_Toc57390067)

[2.c.1 Kodlar 2](#_Toc57390068)

[2.c.2 Ekran görüntüleri 2](#_Toc57390069)

[2.c.3 Açıklama 3](#_Toc57390070)

[2.d Listeleme 3](#_Toc57390071)

[2.d.1 Kodlar 3](#_Toc57390072)

[2.d.2 Ekran görüntüleri 3](#_Toc57390073)

[2.d.3 Açıklama 3](#_Toc57390074)

[Özdeğerlendirme Tablosu 3](#_Toc57390075)

# 1) POINTS IN A 2D PLANE

C#, Visual Studio Community16.8.3, .NET Framework 4.8

## 1.a Rastgele Nokta Üretimi

### 1.a.1 Kodlar

var random = new Random();

var coordinates = new double[2, n];

for (var i = 0; i < n; i++)

{

double randomRow = random.Next(height); //value of "y"

double randomColumn = random.Next(width); //value of "x"

coordinates[0, i] = randomColumn;

coordinates[1, i] = randomRow;

}

Console.WriteLine("Printing out the coordinates of {0} random points: ", n);

for (var i = 0; i < coordinates.GetLength(0); i++)

{

for (var j = 0; j < coordinates.GetLength(1); j++)

{

if (j == 0 && i == 0) Console.Write("x: ");

if (j == 0 && i == 1) Console.Write("y: ");

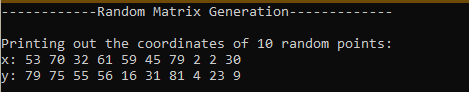
Console.Write(coordinates[i, j] + " ");

}

Console.WriteLine();

}

### 1.a.2 Ekran görüntüleri



### 1.a.3 Açıklama

Random sınıfı kullanılarak istenilen aralıkta koordinatlar oluşturulup yazdırıldı. İki boyutlu dizi yapısı kullanıldı.

## 1.b Uzaklık Matrisi

### 1.b.1 Kodlar

var distanceMatrix = new double[n, n];

for (var i = 0; i < n; i++)

{

for (var j = 0; j < n; j++)

distanceMatrix[i, j] =

Distance(coordinates[0, i], coordinates[1, i],

coordinates[0, j], coordinates[1, j]);

}

Console.WriteLine("\n" + "Printing out the symmetric Distance Matrix of {0} points: ", n);

for (var i = 0; i < distanceMatrix.GetLength(0); i++)

{

for (var j = 0; j < distanceMatrix.GetLength(1); j++)

{

Console.Write(Math.Round(distanceMatrix[i, j], 1) + " ");

}

Console.WriteLine();

}

Console.WriteLine();

double Distance(double x1, double y1, double x2, double y2)

{

return Math.Sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));

}

### 1.b.2 Ekran görüntüleri

### 

### 1.b.3 Açıklama

# Uzaklık matrisi için bütün noktaların birbirleriyle uzaklığı alınarak simetrik matrisi oluşturuldu ve bu uzaklıklar yuvarlanarak yazdırıldı. Yerel noktalama işaretlerine dikkat edildi. İki boyutlu dizi yapısı kullanıldı.

# 2) CLASSIFICATION USING K-NEAREST NEIGHBORS (KNN) ALGORITHM

C#, Visual Studio Community16.8.3, .NET Framework 4.8

## 2.a KNN ile sınıflandırma

### 2.a.1 Algoritma

1. Load the training and test data   
2. Choose the value of K   
3. For each point in test data:  
- find the Euclidean distance to all training data points  
- store the Euclidean distances in a list and sort it   
- choose the first k points   
- assign a class to the test point based on the majority of classes present in the chosen points  
4. End

## 2.b Banknot sınıflandırma

### 2.b.1 Kodlar

static void KNN(int k, double V1\_varyans, double V2\_çarpıklık, double V3\_basıklık, double V4\_entropi)

{

var dataSize = System.IO.File.ReadAllLines(@"data\_banknote\_authentication.txt").Length;

var text = System.IO.File.ReadAllText(@"data\_banknote\_authentication.txt");

var lines = System.IO.File.ReadAllLines(@"data\_banknote\_authentication.txt");

var dataMatrix = new double[dataSize, dataSize];

var i = 0;

foreach (var row in text.Split('\n'))

{

var j = 0;

foreach (var column in row.Split(','))

{

dataMatrix[i, j] = double.Parse(column, CultureInfo.InvariantCulture);

j++;

}

i++;

}

var margin = new double[dataSize];

var counterfeit = new int[dataSize];

for (var j = 0; j < dataMatrix.GetLength(0); j++)

{

var eucledianDistance =

Math.Sqrt(Math.Pow(V1\_varyans - dataMatrix[j, 0], 2) +

Math.Pow(V2\_çarpıklık - dataMatrix[j, 1], 2) +

Math.Pow(V3\_basıklık - dataMatrix[j, 2], 2) +

Math.Pow(V4\_entropi - dataMatrix[j, 3], 2));

margin[j] = eucledianDistance;

counterfeit[j] = (int)dataMatrix[j, 4];

}

Array.Sort(margin, counterfeit); //Sorting by keys and items

var notCounterfeit = 0;

var isCounterfeit = 0;

for (var j = 0; j < k; j++)

{

switch (counterfeit[j])

{

case 0:

isCounterfeit++;

break;

case 1:

notCounterfeit++;

break;

}

}

for (var j = 0; j < k; j++)

{

Console.Write("k{0}: ", j + 1);

Console.Write("Distance: " + Math.Round(margin[j], 4) + " ");

Console.Write("Class: " + counterfeit[j] + "\n");

}

if (isCounterfeit > notCounterfeit)

Console.WriteLine("\nSpecimen is counterfeit.\n");

else

### Console.WriteLine("\nSpecimen is genuine.\n");

### 2.b.2 Ekran görüntüleri

### 

### 2.b.3 Açıklama

data\_banknote\_authentication.txt dosyasından veriler çekildi. Bu veriler dizi, string ve satır sayısı olarak kaydedildi. Verinin her satırındaki virgülle ayrılmış özellikler iki boyutlu dizi içinde saklandı. Her bir örnek için kullanıcıdan alınan k örneği karşılaştırıldı. Verilen sahte/gerçek değeri ve uzaklık (hata payı) için diziler oluşturuldu. k kadar en yakın örneğin bilgileri ekrana yazdırıldı.

## 2.c Başarı ölçümü

### 2.c.1 Kodlar

var testListOfOnes = new ArrayList();

var testListOfZeros = new ArrayList();

foreach (var line in lines)

{

if (line.EndsWith("0"))

testListOfZeros.Add(line);

if (line.EndsWith("1"))

testListOfOnes.Add(line);

}

testListOfOnes.RemoveRange(0, testListOfOnes.Count - 100); //the last 100 genuine specimens

testListOfZeros.RemoveRange(0, testListOfZeros.Count - 100); //the last 100 counterfeit specimens

### 2.c.2 Ekran görüntüleri

-

### 2.c.3 Açıklama

2 çeşit test verisi için iki ArrayList oluşturuldu. Satırın sonundaki 0 veya 1 değerine göre veriler gruplara ayrıldı. Son 100 örnek seçildi.

## 2.d Listeleme

### 2.d.1 Kodlar

Console.OutputEncoding = Encoding.UTF8; //to include Turkish phonetics

Console.WriteLine("\n-------------------Veriseti-------------------\n");

Console.WriteLine("Varyans - Çarpıklık - Basıklık - Entropi - Sınıf\n");

foreach (var line in lines)

### Console.WriteLine(line);

### 2.d.2 Ekran görüntüleri

### 

…

### 2.d.3 Açıklama

Önceden ayrıştırılan veri satırları foreach döngüsüyle yazdırıldı.

# Özdeğerlendirme Tablosu

|  |  |  |  |
| --- | --- | --- | --- |
| **Proje 1 Maddeleri** | **Not** | **Tahmini Not** | **Açıklama** |
| 1.a | 15 | 15 | Yapıldı. |
| 1.b | 15 | 12 | Örneklerin bazı özellikleri yazdırılmadı. |
| Bölüm 1. Rapor | 10 | 10 | Yapıldı. |
| 2.a | 10 | 10 | Yapıldı. Farklı kaynakları araştırarak bilgi edindim. |
| 2.b | 10 | 10 | Yapıldı. Forumlardan yardım aldım. Yararlı kod parçalarını uyarladım. |
| 2.c | 10 | 2 | Tamamlanmadı. |
| 2.d | 10 | 10 | Yapıldı. |
| Bölüm 2. Rapor | 10 | 10 | Yapıldı. |
| Özdeğerlendirme Tablosu | 10 | 10 | Yapıldı. |

**Açıklama kısmında yapıldı, yapılmadı bilgisi veya hangi maddelerin nasıl yapıldığı kısaca yazılabilecektir.**