

**EGE UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

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

**2022–2023 FALL SEMESTER**

**PROJECT-1 REPORT**

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

**DELIVERY DATE**

14/11/2022

**PREPARED BY**

Yiğit KİRAZ

İçindekiler

[1) YOLCULARI KOLTUKLARA ATAMA 2](#_Toc117859067)

[1.a Uzaklık Matrisi Oluşturma 2](#_Toc117859068)

[1.a.1 Kodlar 2](#_Toc117859069)

[1.a.2 Ekran görüntüleri 2](#_Toc117859070)

[1.b Yolcuları Koltuklara Yerleştirme 2](#_Toc117859071)

[1.b.1 Kodlar 2](#_Toc117859072)

[1.b.2 Açıklama 2](#_Toc117859073)

[1.c Yolcuları Listeleme 2](#_Toc117859074)

[1.c.1 Kodlar 2](#_Toc117859075)

[1.c.2 Ekran görüntüleri 2](#_Toc117859076)

[1.d Her Bir Yolcunun Toplam Uzaklıklarını Listeleme 2](#_Toc117859077)

[1.d.1 Kodlar 2](#_Toc117859078)

[1.d.2 Ekran görüntüleri 2](#_Toc117859079)

[1.e Tüm Yolcuların Uzaklıklar Toplamı 2](#_Toc117859080)

[1.e.1 Kodlar 2](#_Toc117859081)

[1.e.2 Ekran görüntüleri 2](#_Toc117859082)

[2) FLOWER CLASSIFICATION USING A SIMPLE NEURAL NETWORK 3](#_Toc117859083)

[2.a Neuron (Sinir Hücresi) Sınıfı 3](#_Toc117859084)

[2.a.1 Kaynak Kod 3](#_Toc117859085)

[2.a.2 Açıklama 3](#_Toc117859086)

[2.b Neural Network (Yapay Sinir Ağı) Sınıfı 3](#_Toc117859087)

[2.b.1 Kaynak Kod 3](#_Toc117859088)

[2.b.2 Açıklama 3](#_Toc117859089)

[2.c Eğitim 3](#_Toc117859090)

[2.c.1 Kaynak Kod 3](#_Toc117859091)

[2.c.2 Sonuçlar/Ekran görüntüleri 3](#_Toc117859092)

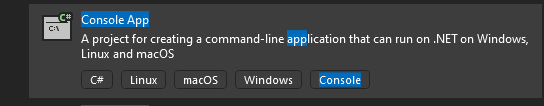
[2.d Deneyler 3](#_Toc117859093)

[2.d.1 Kaynak Kod 3](#_Toc117859094)

[2.d.2 Sonuçlar 3](#_Toc117859095)

[Öz değerlendirme Tablosu 3](#_Toc117859096)

# 1) YOLCULARI KOLTUKLARA ATAMA

// C# .NET 6.0 Console App 

## 1.a Uzaklık Matrisi Oluşturma

### 1.a.1 Kodlar

public static double[,] DistanceMatrixCreator(int matrixRows, int matrixColumns)

{

var matrixDistance = new double[matrixRows, matrixColumns];

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

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

if (i != j)

matrixDistance[i, j] = RandomReturner();

else

matrixDistance[i, j] = 0.0;

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

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

matrixDistance[j, i] = matrixDistance[i, j];

return matrixDistance;

### }1.a.2 Ekran görüntüleri

## 1.b Yolcuları Koltuklara Yerleştirme

### 1.b.1 Kodlar

public static void CreateSeatsForPassengers(Passengers[] \_passengersArray, double[,] \_distanceMatrix)

{

Random r = new Random();

int luckyNumber = r.Next(\_distanceMatrix.GetLength(0));

\_passengersArray[luckyNumber].seatNumber = 1;

\_passengersArray[luckyNumber].seated = 1;

\_passengersArray[luckyNumber].type = +2;

\_passengersArray[luckyNumber].distance = 0;

Passengers seatedPassenger;

for (int i = 1; i < \_distanceMatrix.GetLength(0); i++) // i means the seat numbers. Start point

{

seatedPassenger = FindLowDistance(\_passengersArray, \_distanceMatrix, i+1);

seatedPassenger.seatNumber = i + 1;

seatedPassenger.seated = 1;

seatedPassenger.type = TypeFinder(i + 1);

}

### }

### 1.b.2 Açıklama

Sıradaki boştaki koltuğa oturacak olan yolcuyu atar (passengers sınıfından olan yolcuyu).

Koltuğa oturacak kişinin uzaklığı diğer koltuklara minimum olması gerekmektedir.

Minimum uzaklığa sahip kişinin bulunması koltuktan koltuğa değimektedir.

Bu yüzden koltuklar için tip değerleri tanımlanmaktadır.

//type = -3 // it means seat number not known yet.

//type = -2; // it means this is the first row. (First row seats.)

//type = -1; // it means this is the first column. (First column seats.)

//type = 0; // it means this the last column. (Last column seats.)

//type = +1; // it means this is the other columns. (Other column seats)

//type = +2 // it means this is the first seat.

Type değerlerinin amacı try except ‘ ten kaçmaktadır tüm koltuk indexlerini kontrol edip try except yapmak tam bir optimizasyon düşmanıdır.

Bunu farkettiğimizden ötürü try-except yerine type değerleri ile toplam yapılmıştır.

Oturulacak koltuğun tipine göre switch case yapısı ile koltuğa oturan yolcunun 135 derecelik etrafındaki yolculara olan uzaklığı otomatik olarak toplanır.

FindLowDistance() fonksiyonu ile de bu toplama en küçük olan yolcu nesnesini döndürür.

Ve son olarakta en küçük uzaklık değerine sahip olan nesne koltuğa oturur ve bu işlem yolcu sayısı kadar devam eder.

Ve yolcuların hepsi düzgün bir şekilde koltuklara oturduktan sonra işlem biter ve ekrana printlenir.

## 1.c Yolcuları Listeleme

### 1.c.1 Kodlar

public static void WritePassengerMatrix(Passengers[,] \_passengerMatrix)

{

int toStringLength;

int maxStringLengthFor1Passenger = 50;

for (int i = 0; i < \_passengerMatrix.GetLength(1); i++)

{

for (int j = 0; j < \_passengerMatrix.GetLength(0); j++)

{

toStringLength = ToString(\_passengerMatrix[j, i]).Length;

Console.Write(ToString(\_passengerMatrix[j, i]) );

for (int k = 0; k < maxStringLengthFor1Passenger - toStringLength; k++) {

Console.Write(" ");

}

if (j != busRowsLength - 1) {

Console.Write("|");

}

if (j % busRowsLength == 0 && j != 0)

{

Console.WriteLine();

}

}

Console.WriteLine();

for (int l = 0; l < busRowsLength \* maxStringLengthFor1Passenger; l++)

{

Console.Write("\_");

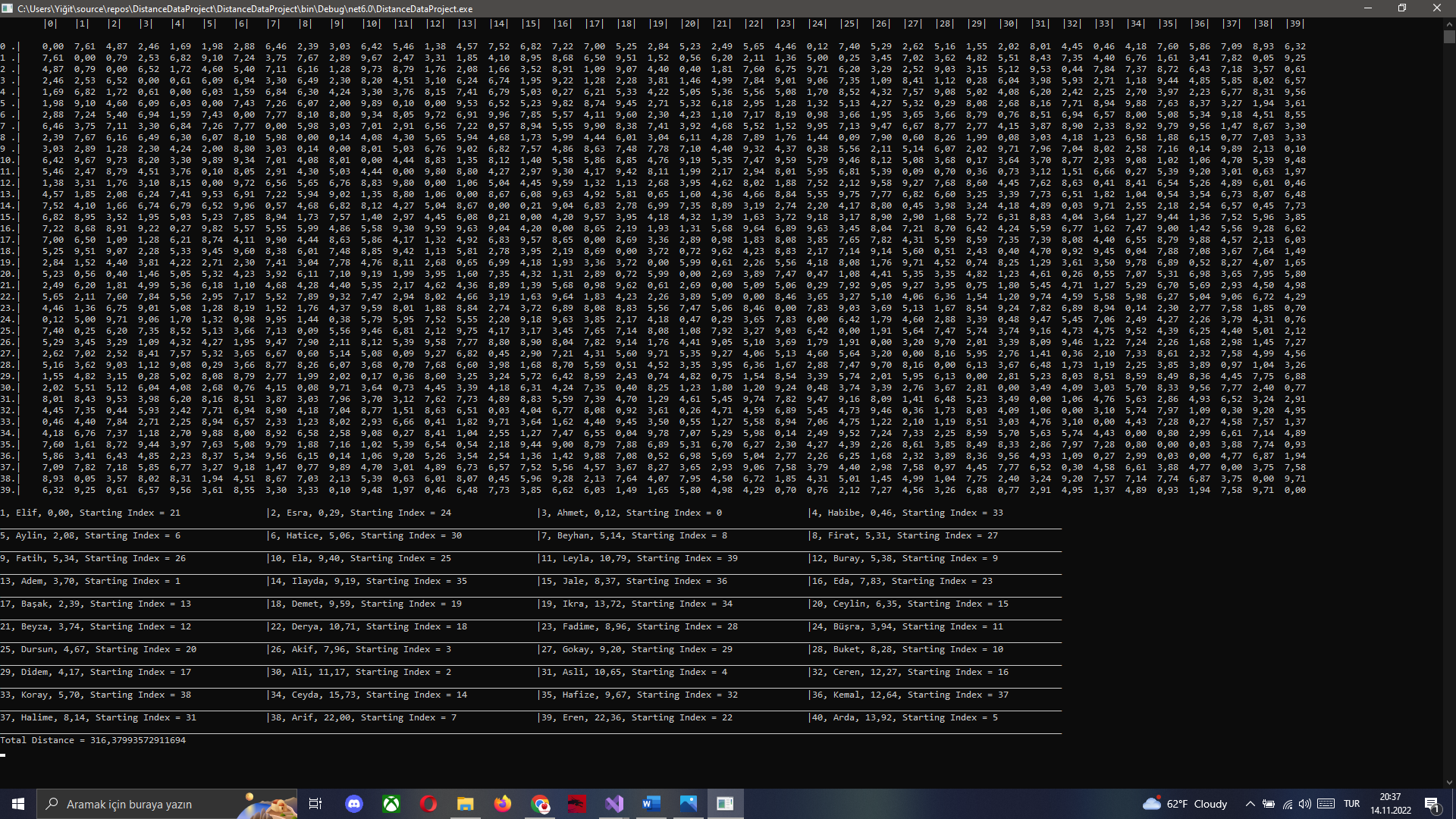
}

Console.WriteLine();

}

### }1.c.2 Ekran görüntüleri

//Konsol çıktısına ait ekran görüntülerini buraya ekleyiniz



## 1.d Her Bir Yolcunun Toplam Uzaklıklarını Listeleme

Uzaklıklar her yolcu nesnesinin içinde variable olarak saklandığından ötürü passengers arrayi sayesinde çok rahatlıkla isim uzaklık ve ilk isim sıralaması listesinde kaçıncı sıralamada başladıkları yazdırılmıştır.(Başlangıç sıralaması test ve doğruluk için gerekmektedir bunun yararlı olacağını düşünüp kodumuza ekledik.)

### 1.d.1 Kodlar

public static void WritePassengersDistance(Passengers[] passengers)

{

for(int i = 0; i < passengers.Length; i++ )

Console.WriteLine("Koltuk No = " + passengers[i].seatNumber + ", Distance = " + passengers[i].distance);

### }1.d.2 Ekran görüntüleri

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

ÖNEMLİ:

KoltukNo/Uzaklık/İsim Listesindeki başlangıç sıralaması şeklinde printlenmiştir.

## 1.e Tüm Yolcuların Uzaklıklar Toplamı

### 1.e.1 Kodlar

public static double TotalDistance(Passengers[] pList)

{

double totalDistance = 0;

for (int i = 0; i < pList.Length; i++)

{

totalDistance += pList[i].distance;

}

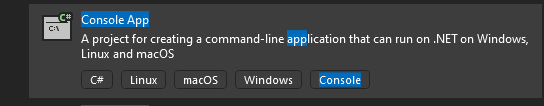
return totalDistance;

### }1.e.2 Ekran görüntüleri



# 

# 2) FLOWER CLASSIFICATION USING A SIMPLE NEURAL NETWORK

// C# .NET 6.0 Console App 

## 2.a Neuron (Sinir Hücresi) Sınıfı

### 2.a.1 Kaynak Kod

public class Neuron

{

public double w1;

public double w2;

public double w3;

public double w4;

public int target;

public string flowerOfInterest;

public Neuron(string flowerOfInterest)

{

this.flowerOfInterest = flowerOfInterest;

}

public void SetWeights(double w1, double w2, double w3, double w4)

{

this.w1 = w1;

this.w2 = w2;

this.w3 = w3;

this.w4 = w4;

}

public void SetTarget(int target)

{

this.target = target;

}

}

### }2.a.2 Açıklama

Neuron sınıfında Neuron nesneleri için 4 ağırlık variable dtutulmaktadır.

Ayrıca eğitilmek istenilen türün string isimleri ve target değerleri belirtilmiştir. Target değerler defult olarak başta neuron nesneleri için 0’dır çünkü henüz veri okunmamıştır okunan verilere göre target değerleri ilgili neuron’un 1 ve diğer neuronalrın 0 olmaktadır bir sonraki veriye geçildiği zaman neuronların target değerleri tekrar 0lanır ve ilgili işleme göre tekrar atanır.

Bu optimizasyon ve tekrar tekrar yapılması gerekebilen comparison işlemlerinden bizi kurtarmaktadır.

Kalıcı Herhangi bir veri yapısı tutulmamıştır. Kod optimize yazılmıştır. RAM’de tutulan tek veri neurons[] arrayidir bu array neuron sınıfından 3 nesne tutar ve bu nesnelerin ağırlık değerleri target değerleri gibi vs değerler tutulur.

Herhangi bir veri taşıması yapılmamıştır.

Dışardan alınan veriler sonsuz kabul edilmiş ve kesinlikle tüm verileri bir array’e eklenmemiştir. Veriler satır satır okunup split edilip işlenip makine öğrenmesi yapılmaktadır.

Ve okunan satırlar Main methodunda nöron katsayıları değiştirilip epokla eğitilmiştir.

Verileri okurken “tr-TR” olarak algıladığı için

CultureInfo usCulture = new CultureInfo("en-US");

NumberFormatInfo dataNumberFormat = usCulture.NumberFormat;

Hata giderilmiş ve veriler transpoze edilmiştir.

Bunun sebebi ise kodun tüm ülkelerde uygun bir şekilde çalışmasıdır. Default double değer okuma şekli “x.xxxxxxxxxxxx” şeklinde tanımlanmıştır.

Veri okuma evrenseldir. Tüm ülkeler tarafından ilgili veri okunur ve ekrana yansıtır sorunsuz çalışmaktadır.

## 2.b Neural Network (Yapay Sinir Ağı) Sınıfı

### 2.b.1 Kaynak Kod

public class ArtificialNeuralNetwork

{

public Neuron[] neurons;

public int epoch;

public double lambda;

public ArtificialNeuralNetwork(Neuron[] neurons, int epoch, double lambda)

{

this.neurons = neurons;

this.epoch = epoch;

this.lambda = lambda;

}

public ArtificialNeuralNetwork(Neuron[] neurons)

{

this.neurons = neurons;

}

public void SetANN( int epoch, double lambda)

{

this.epoch = epoch;

this.lambda = lambda;

}

### }2.b.2 Açıklama

## 

## ArtificialNeuralNetwork class’ımzıda

public Neuron[] neurons;

public int epoch;

public double lambda;

değerleri tutulmaktadır.

## Neuron sınıfından nesneler array olarak tutulmaktadır. 3 çeşit iris verisi değil de birden fazla veri çeşidi girilirse nöron sayısı artacağından dolayı nöronlar array içerisinde saklanır ve ANN sınıfı oluştururken constructor ile bu neurons arrayleri ANN nesnesine atanır.

public ArtificialNeuralNetwork(Neuron[] neurons, int epoch, double lambda)

{

this.neurons = neurons;

this.epoch = epoch;

this.lambda = lambda;

}

Constructor ile ANN nesnesi oluşturulur.

## 2.c Eğitim

### 2.c.1 Kaynak Kod

public static void TrainingNetworkFromData(ArtificialNeuralNetwork ANN, string[] flowerNames, string fileName)

{

string[] strDatas = new string[NUMBEROFSPLITDATA];

double[] dataDoubles = new double[NUMBEROFSPLITDATA - 1];

string flowerName;

Neuron[] neurons = ANN.neurons;

SetFirstWeights(neurons);

string line;

Neuron neuronTarget;

Neuron maxNeuron;

for (int timesOfOpech = 0; timesOfOpech < ANN.epoch ; timesOfOpech++)

{

using (StreamReader reader = new StreamReader(fileName))

{

while ((line = reader.ReadLine()) != null )

{

strDatas = LineToData(line);

dataDoubles = GetNormalizeDoublesFromLineStr(strDatas);

flowerName = strDatas[NUMBEROFSPLITDATA - 1];

neuronTarget = setTargetFromFlowerName(neurons, flowerName);

maxNeuron = GetMaxOutputNeuron(neurons, dataDoubles);

if (maxNeuron.target != 1)

{

DecreaseWeights(maxNeuron, dataDoubles, ANN.lambda);

IncreaseWeights(neuronTarget, dataDoubles, ANN.lambda);

}

}

reader.Close();

}

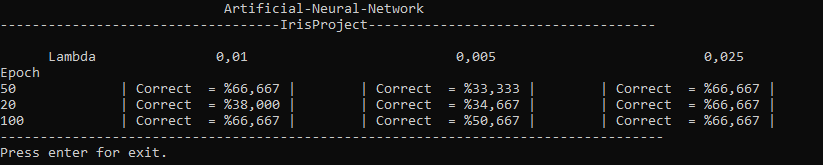
}

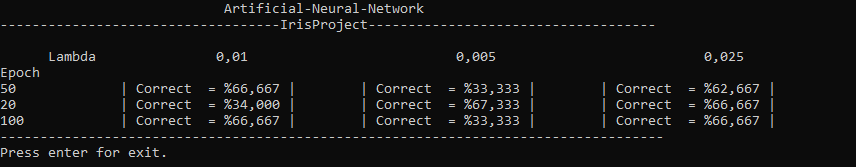
}

### 2.c.2 Sonuçlar/Ekran görüntüleri

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu





## 

## 2.d Deneyler

### 2.d.1 Kaynak Kod

public static string testANN(string fileName, Neuron[] neurons )

{

string testString;

using (StreamReader sr = new StreamReader(fileName))

{

double correctTests = 0;

double inCorrectTests = 0;

string[] strDatas;

double[] dataDoubles;

string flowerName;

string line;

while ((line = sr.ReadLine()) != null)

{

strDatas = LineToData(line);

dataDoubles = GetNormalizeDoublesFromLineStr(strDatas);

flowerName = strDatas[NUMBEROFSPLITDATA - 1];

setTargetFromFlowerName(neurons, flowerName);

if (GetMaxOutputNeuron(neurons, dataDoubles).target == 1)

{

correctTests++;

}

else

{

inCorrectTests++;

}

}

sr.Close();

double correctPercent = correctTests \* 100.0 / (inCorrectTests + correctTests);

testString = "| Correct = %" + String.Format("{0:0.000}", correctPercent) + " |";

}

return testString;

### }2.d.2 Sonuçlar

|  |  |  |  |
| --- | --- | --- | --- |
| Deney 1 | 20 Epok | 50 Epok | 100 Epok |
| λ = 0.005 | %60,667 | %55,333 | %69,333 |
| λ = 0.01 | %33,333 | %66,668 | %69,667 |
| λ = 0.025 | %66,667 | %66,667 | %66,667 |

|  |  |  |  |
| --- | --- | --- | --- |
| Deney 2 | 20 Epok | 50 Epok | 100 Epok |
| λ = 0.005 | %34,667 | %33,333 | %50.667 |
| λ = 0.01 | %38,000 | %66,667 | %66,667 |
| λ = 0.025 | %66,667 | %66,667 | %66,667 |

//Lambdayı çok küçülttüğümüz zaman istikrarsız ve daha düşük doğruluk verileri elde ediyoruz.

Epoku arttırdıkça doğruluk oranlarının ortalaması %66,667 ye yaklaşıyor.

|  |  |  |  |
| --- | --- | --- | --- |
| Deney 3 | 20 Epok | 50 Epok | 100 Epok |
| λ = 0.005 | %67,333 | %33,333 | %33,333 |
| λ = 0.01 | %34,000 | %66,667 | %66,667 |
| λ = 0.025 | %66,667 | %62,667 | %66,667 |

# Öz değerlendirme Tablosu

|  |  |  |  |
| --- | --- | --- | --- |
| **Proje 1 Maddeleri** | **Not** | **Tahmini Not** | **Açıklama** |
| 1.a | 10 | 10 | Matrix flexible dır sadece 40 değil kullanıcı isterse 40\*50 veya 50\*60 oluşturabilir.  Kod otobüsteki koltukların dizilişinden tutun, koltuk sayısı matrixin uzunluğu da değişse her harikularda çalışmaktadır.  Fakat istenilen değer 40 olduğu için  CreateDistanceMatrix fonksiyonunda paramatrelere (40, 40) değerleri gönderilerek 40\*40 lık bir matrix oluşturduk. |
| 1.b | 10 | 10 | Yolcular sorunsuz bir şekilde koltuklara istenilen koşullarda yerleştiriliyor. |
| 1.c | 10 | 10 | Yolcular sorunsuz şekilde listeneleniyor ve rapordada belirtildi. |
| 1.d | 5 | 10 | Her yolcunun toplam uzaklıkları listeleniyor.  Ayrıca Passengers nesneleri koltuk numarası distance vs gibi değerleri tuttuğu için passengers paramatresi alan toString methodu yaparak orada da ayrıca listeledik.  Raporda da belirtilmiştir. |
| 1.e | 5 | 10 | Sorunsuz çalışmaktadır ilgili açıklama ve kod yukarıda belirtilmiştir. |
| 2.a | 10 | 10 | Sorunsuz çalışmaktadır ilgili açıklama ve kod yukarıda belirtilmiştir. |
| 2.b | 10 | 10 | Sorunsuz çalışmaktadır ilgili açıklama ve kod yukarıda belirtilmiştir. |
| 2.c | 10 | 10 | Sorunsuz çalışmaktadır ilgili açıklama ve kod yukarıda belirtilmiştir. |
| 2.d | 10 | 10 | Sorunsuz çalışmaktadır ilgili açıklama ve kod yukarıda belirtilmiştir. |
| Rapor | 10 | 10 | Sorunsuz çalışmaktadır ilgili açıklama ve kod yukarıda belirtilmiştir. |
| Öz değerlendirme Tablosu | 10 | 10 | Düzgünce yapıldı kodda yazılan açıklamalara ek olarak rapor ile program pekiştirildi. |
| **Toplam** | **100** | **100** | **100** |

**Açıklama kısmında yapıldı, yapılmadı bilgisi ve hangi maddelerin nasıl yapıldığı veya neden yapılamadığı kısaca yazılmalıdır.**

**Not: Raporu teslim edilmeyen projeler değerlendirmeye alınmayacaktır.**