

DATA STRUCTURES

PROJE 1

Sercan Bayram

05170000038

Oğuz Türk

05170000079

1.1.1. Gerçekleştirilen Platform ve ve Sürüm Adı:

C# (Microsoft Visual Studio 2019)

1.1.2. Program 1.1'in kısa tanımı:

Doğum Günü Paradoksu:

Doğum günü paradoksu, bir odadaki iki insanın aynı doğum gününe sahip olma olasılığının, odadaki insan sayısının n'den fazla olması kaydıyla yarısından fazla olduğunu söylüyor. Bu özellik gerçekten bir paradoks değil, birçok insan bunu şaşırtıcı buluyor.

Programda kullanılan çakışma ise şu şekilde açıklanabilir:

Çakışma: 2 kişinin doğum tarihi haftanın aynı gününe geliyorsa 1 çakışma vardır. Bunun için doğulan gün adının aynı olması yeterlidir. Doğum yılının, ayının ve gün numarasının eşit olmasına gerek yoktur. 3 kişi aynı günde doğdularsa 2 çakışma vardır. 7 kişinin aynı doğum gününe sahip olmaları durumunda 6 çakışma vardır.

Programda bizden istenen ise şu şekilde:

Programımız aşağıdaki iki listelemeyi de içermelidir:

1-) Simülasyonlar yapılarak 15 deney için sadece istatistiksel değerler (çakışma sayıları) 16 x 5'lik ayrı bir tablo olarak ekrana verilmelidir. n'in farklı değerleri (n = 2, 3, 5, 10, 20 kişi) için 5 sütun, her deneydeki ortalama çakışma sayıları için 15 satır ve bu 15 değer için 16. satır yazdırılmalıdır.

2-) Tek tek denemelerdeki çakışmalar gösterilmelidir. Her deneme için 7 elemanlı bir dizide gösterilen haftanın günlerindeki çakışma sayıları yazdırılmalıdır. n'in her bir değeri için de bu işlem 15 kere tekrarlanır. Ardından i)'deki 16x5'lik özet liste de verilmelidir.

1.1.3. Program 1.1 için kullanılan bileşik veri tiplerinin ve metotların kısa açıklamaları:

```
string[] sayilar = { " 2", " 3", " 5", "10", "20" };  
// Kişi Sayılarını tutan String tipinde array.
```

```
Random gen = new Random();  
// Randon gün bulmak için kullanılan random metodu.
```

```
double[] cakisma = { 0, 0, 0, 0, 0 };  
// Kişi Sayısı İçin Ortalama Tutan array.
```

```
int[,] cakismalar = new int[5, 15];  
// Çakışma sayılarını tutan 2 boyutlu array.
```

```
int randomgun = gen.Next(0, 7);  
//Günleri temsil eden random sayılar
```

1.1.4. Elde edilen örnek sonuç:

```
Deney Sayısı  1---2---3---4---5---6---7---8---9---10--11--12--13--14--15--Ort  
2 kişi için: 0 - 0 - 0 - 0 - 0 - 0 - 1 - 0 - 1 - 1 - 1 - 0 - 0 - 0 - 0 - 0,27  
3 kişi için: 1 - 0 - 0 - 0 - 0 - 1 - 1 - 1 - 0 - 0 - 0 - 1 - 0 - 0 - 0 - 0,33  
5 kişi için: 1 - 2 - 3 - 1 - 2 - 1 - 1 - 0 - 1 - 1 - 3 - 0 - 2 - 1 - 2 - 1,4  
10 kişi için: 4 - 5 - 4 - 4 - 3 - 4 - 3 - 4 - 4 - 5 - 4 - 4 - 4 - 6 - 5 - 4,2  
20 kişi için: 13- 15- 13- 13- 14- 13- 13- 14- 13- 13- 13- 13- 14- 14- 13- 13,4
```

1.1.5. Proje 1.1 Yazılım Geliştirme İçin Harcanan Süreler:

Sercan Bayram: 4 Saat

Oğuz Türk: 4 Saat

1.2.1. Gerçekleştirilen Platform ve Dil ve Sürüm Adı:

C# (Microsoft Visual Studio 2019)

1.2.2. Program 2'nin Kısa Tanımı:

Doğada gördüğümüz bir çiçeğin, Zambak (veya süsen) bitkisi olduğu biliniyor. Üç farklı türünden hangisine ait olduğunu bulduran bir algoritmanın yazılması istenmektedir. Elimizde her bir çiçek türünden 150 örnek üzerinden ölçülerek alınan veriler bulunmaktadır. Her bir örnek için 4'er adet özellik (çanak yaprak uzunluğu, çanak yaprak genişliği, taç yaprak uzunluğu, taç yaprak genişliği) ve hangi sınıfta (tür) olduğu bilgisi hazır olarak verilmektedir.

Bizden istenenler şu şekildedir:

1-) Bulduğumuz ancak türünü bilmediğimiz bir çiçeğin hangi türe ait olduğunu tespit eden algoritmayı (k en yakın komşu yöntemi) yazmak.

2-) **Bitki sınıflandırma:** Yazdığımız kNN algoritmasının k değerini:

- (i) yeni çiçeğin 4 adet özelliğini girdi olarak aldırarak,
 - (ii) yeni çiçeğin sadece 2 adet çanak yaprak özelliğini (uzunluk ve genişlik) girdi olarak aldırarak,
 - (iii) yeni çiçeğin sadece 2 adet taç yaprak özelliğini (uzunluk ve genişlik) girdi olarak aldırarak,
- en yakın k adet bitkinin özelliklerini, uzaklıklarını ve hangi sınıflardan olduklarını bir tablo olarak ekrana listelemek. 150 satırlı verisetini kullanarak kNN yöntemi ile bitkimizin de türünü tahminleyip, ekrana yazdırmak.

3-) Başarı Ölçümü: Verisetinde her bir tür bitki örneğinin sonunda yer alan 10'ar veriyi test verisi olarak ayırıp, k değerini kullanıcıdan aldırarak, test verilerinden her birini, 4 özelliğin tümünü kullanarak kalan 80% veri üzerinden sınıflandırıp, b'deki listelemeleri yapmak. Test verilerinin gerçek sınıfları ile, kNN ile tahminlediğimiz sınıflarını karşılaştırınız (gerçek ve tahminlenen türlerin / sınıfların her ikisini de yazdırmak). Başarı oranını, doğru sınıflandırılan (bitki sayısı / verisetinde test amaçlı kullandığınız toplam bitki sayısı) olarak hesaplayıp yazdırmak.

4-) Ekleme ve Silme İşlemleri: Klavyeden bellekteki verisetine yeni örnek çiçek verisi (öznitelikler ve sınıf) ekleyen metodu yazmak. Sadece indisi verilen veriyi ve tüm verileri silen metotları oluşturmak.

5-) Listeleme: Bellekteki verisetindeki değerleri görüntüleyen kodu yazınız.

1.2.3. Program 2 için kullanılan bileşik veri tiplerinin, sınıf ve metotların kısa açıklamaları:

```
string[] liste = System.IO.File.ReadAllLines(@"iris.data");
//Verisetindeki bilgileri tutan array.

ArrayList array = new ArrayList();
//Ekleme ve silme işlemlerinde kullanılan arraylist.

int menu;
//Menü işlemlerindeki input değeri.

int k;
//Tahminlemede kullanılan k değeri(Örnek olarak k değerini kullanıcı 3 girdiyse, öznitelik dizisine uzaklığı en yakın (az) olan 3 çiçeği tespit edilmeli).

int giris;
//Tahminlemede kaç bilgi girişi olacağını belirleyen değişken (2 veya 4 olarak belirtilmiştir)

String yaprak;
//Eğer kullanıcı 2 giriş yaparsa bilgilerin taç yaprak için mi çanak yaprak için mi tutulacağını belirten değişken
```

```

double[] ozellik = new double[giris];
//2 veya 4 giriş için girilen çiçeğin özellik input'larını tutan her
seferinde yeniden açılan array

double[,] listeCopy = new double[liste.Length,2];//Distance
değerlerini ve indexleri tutan 2 boyutlu array

string[] temp = liste[i].Split(',');
//listede bulunan elemanların içine ayrı ayrı erişip tutan geçici
bir array

int Setosa = 0;
int Versicolor = 0;
int Virginica = 0;
//k değerine bağlı çıkan çiçeklerin hangisinden kaç tane olduğunu
belirten değişkenler

int kucukindex = 0;
int flag = 0;
//çıkan eşitlik durumunda ilk gelen çiçeği bulmak için kullanılan
bir değişkenler

double enkucuk = 9998;
double index = 0;
//girilen çiçek bilgilerini tahminlemek için verisetindeki diğer
çiçeklerle karşılaştırmak için kullanılan değişkenler

```

1.2.4. Elde edilen örnek sonuçlar:

```

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-)Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz :

```

Program çalıştığında karşımıza böyle bir menü geliyor ve buraya hangi işlemi yapmak istediğimizi giriyoruz.

```

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 1
k değerini giriniz: 3
Kaç giriş yapmak istiyorsunuz (2-4)? : 2
Taç Yaprak mı Çanak Yaprak mı ?(Taç ise Taç,Çanak ise Çanak giriniz) Taç
Taç Yaprak Uzunluğunu giriniz(Double bir değer girmek isterseniz virgülle giriniz): 2,3
Taç Yaprak Genişliğini giriniz(Double bir değer girmek isterseniz virgülle giriniz): 4

5.1,2.5,3.0,1.1,Iris-versicolor 2,98328677803526

5.6,2.9,3.6,1.3,Iris-versicolor 2,99666481275434

5.2,2.7,3.9,1.4,Iris-versicolor 3,05286750449475

**Tahminlenen çiçek türü: Iris-versicolor**

```

Giriş sayısı 2 olduğunda ve girilen değerler **taç** yaprağa ait olduğunda elde edilen örnek görüntü

```

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 1
k değerini giriniz: 3
Kaç giriş yapmak istiyorsunuz (2-4)? : 2
Taç Yaprak mı Çanak Yaprak mı ?(Taç ise Taç,Çanak ise Çanak giriniz) Çanak
Çanak Yaprak Uzunluğunu giriniz(Double bir değer girmek isterseniz virgülle giriniz): 3,4
Çanak Yaprak Genişliğini giriniz(Double bir değer girmek isterseniz virgülle giriniz): 1,5

4.5,2.3,1.3,0.3,Iris-setosa 1,36014705087354

5.0,2.0,3.5,1.0,Iris-versicolor 1,67630546142402

4.4,2.9,1.4,0.2,Iris-setosa 1,72046505340853

**Tahminlenen çiçek türü: Iris-setosa**

```

Giriş sayısı 2 olduğunda ve girilen değerler **çanak** yaprağa ait olduğunda elde edilen örnek görüntü

```

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-)Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 1
k değerini giriniz: 3
Kaç giriş yapmak istiyorsunuz (2-4)? : 4
Çanak Yaprak Uzunluğunu giriniz(Double bir değer girmek isterseniz virgülle giriniz): 2,1
Çanak Yaprak Genişliğini giriniz(Double bir değer girmek isterseniz virgülle giriniz): 3,4
Taç Yaprak Uzunluğunu giriniz(Double bir değer girmek isterseniz virgülle giriniz): 5
Taç Yaprak Genişliğini giriniz(Double bir değer girmek isterseniz virgülle giriniz): 1,3

4.9,2.5,4.5,1.7,Iris-virginica 3,00998338865848

5.2,2.7,3.9,1.4,Iris-versicolor 3,36452076825214

5.4,3.0,4.5,1.5,Iris-versicolor 3,36749164809655

**Tahminlenen çiçek türü: Iris-versicolor**

```

Giriş sayısı 4 olduğunda elde edilen örnek görüntü

```

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-)Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 2
Tür ismini giriniz : Iris-setosa
Çanak Yaprak uzunluğu (Değerleri (2.0 , 5.5 , 3.9 , 3.6 gibi noktalı bir şekilde giriniz.):
2.5
Çanak Yaprak Genişliği Değerleri (2.0 , 5.5 , 3.9 , 3.6 gibi noktalı bir şekilde giriniz.):
3.2
Taç Yaprak Uzunluğu Değerleri (2.0 , 5.5 , 3.9 , 3.6 gibi noktalı bir şekilde giriniz.):
4.0
Taç Yaprak Genişliği Değerleri (2.0 , 5.5 , 3.9 , 3.6 gibi noktalı bir şekilde giriniz.):
1.2

Ekleme başarıyla yapıldı.

6.8,3.2,5.9,2.3,Iris-virginica
6.7,3.3,5.7,2.5,Iris-virginica
6.7,3.0,5.2,2.3,Iris-virginica
6.3,2.5,5.0,1.9,Iris-virginica
6.5,3.0,5.2,2.0,Iris-virginica
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica
2.5,3.2,4.0,1.2,Iris-setosa
Devam etmek için herhangi bir tuşa basınız..

```

Verisetine eleman ekleyip, listeleme işlemini kullanarak elemanın eklendiğini kontrol eden bir örnek görüntü


```

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-)Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 3
Kaçıncı indisi silmek istiyorsunuz? : 1

Silme işlemi başarıyla yapıldı...

1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-)Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 5
5.1,3.5,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa

```

Verisetinden indise bağlı silme işlemi yaptıktan sonra, listeleme işlemini kullanarak elemanın silindiğini kontrol eden örnek görüntü(1.indisteki eleman 4.9,3.0,1.4,0.2,Iris-setosa olmakta olup görüldüğü üzere verisetinden silinmiştir.)

```
1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 4
```

Silme işlemi başarıyla yapıldı...

```
1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 5
```

Liste boş...

Devam etmek için herhangi bir tuşa basınız..

Tüm listeyi silme işleminden sonra kontrolünü de listeleme işlemiyle yapan örnek görüntü

```
1-)Tahminleme İşlemi(Bitki Sınıflandırma)
2-)Ekleme İşlemi
3-)İndise Bağlı Silme İşlemi
4-)Tüm Listeyi Silme İşlemi
5-Listeleme İşlemi
0-)Çıkış
Hangi işlemi yapacağınızı seçiniz : 0
Çıkış yapılıyor.....
Press any key to continue . . .
```

Çıkış ekranına ait örnek görüntü

1.2.5. Proje 1.2 Yazılım Geliştirme İçin Harcanan Süreler (kişi ve saat bazında):

Sercan Bayram:10 Saat

Oğuz Türk:10 Saat

Ek 1: Program 1.2 a seçeneği: Sözdekod / Algoritma / Yöntem (Mantık) Anlatımı:

Öncelikle verisetindeki dataları tek boyutlu bir listeye aktardık. Ekleme ve silme işlemlerinde array bize sıkıntı çıkarabileceği için bu işlemler için de bir arraylist açtık. Tahinleme işlemi için öncelikle gerekli koşullu ifadelerle menüyü düzenleyip gerekli kullanıcıdan girdileri aldırarak.

Kullanıcıdan aldığımız bu değerleri girilen değer kadar eleman sayısı olan bir listeye atıp bu listedeki inputlar ile verisetindeki tüm gerekli değerleri karşılaştırıp distance(mesafe) değerini bulduk. Bu işlemi yaparken listedeki elemanların içine ayrı ayrı erişip tutan geçici bir array açtık. Distance değerlerini daha rahat kullanabilmek ve yazdırabilmek için distance(mesafe) değerlerini ve Index'leri tutan 2 boyutlu bir array açtık. Ardından k değeri kadar bu elde ettiğimiz mesafeler arasından en küçük k tanesini bulup yazdırdık. Sonrasında ise çıkan bu k tane değer arasından en fazla olan tür sayısını bulup yazdırdık. Eşitlik durudumunda da gerekli düzenlemeyi yaptık. Bu işlemleri 2 bilgi girişi veya 4 bilgi girişi için genelledik. Ardından ekleme için eklenecek girdi bilgilerini kullanıcıdan aldrıp arrayliste ekledik. Yani listede var olan bilgileri koruyarak eklenen bilgileri arraye aktarmış olduk. İndisten silme işleminde de çok benzer bir işlemle yine arrayden seçilen indisteki elemanı sildirdik. Tüm verisetini silmek için de var olan listemizi 0 elemanlı bir listeye çevirdik. Listelemek için de foreach ile listemizin içinde gezerek var olan elemanları yazdırdık. Son olarak da çıkış için break kullandık.

Özdeğerlendirme Tablosu

Proje 1 Maddeleri	Not	Tahmini Not	Açıklama
1.1	15	15	Yapıldı
1.1 Rapor	10	10	Yapıldı
1.2.a	10	5	Bu işlemi metotlaştıramadığımız için birden fazla tekrar etmek zorunda kaldık.
1.2.b	25	20	Sınıflandırma yaparken a'daki kodu kullanmamız gerektiği için burada da 3 farklı durum için çok benzer kodları tekrar etmek durumunda kaldık.
1.2.c	10	0	Yapılamadı
1.2.d.e	10	10	Yapıldı
1.2 Rapor	10	10	Yapıldı
Özdeğerlendirme Tablosu	10	10	Yapıldı
Toplam	100	80	