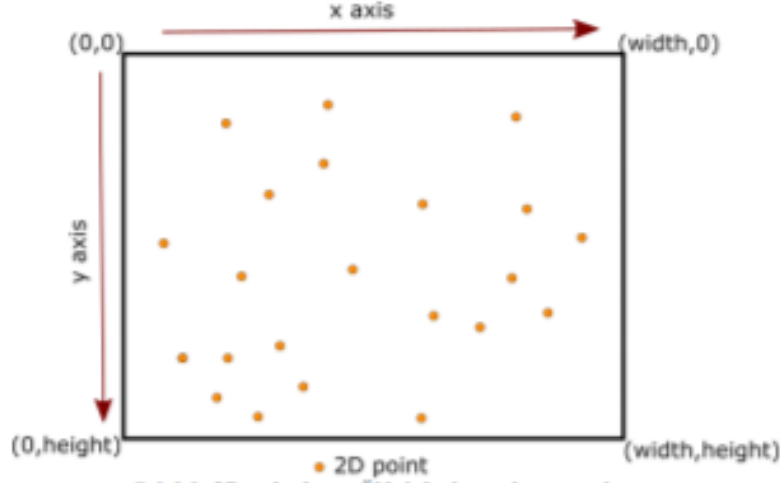


1) GENERATING DISTANCE MATRIX FROM POINTS IN A 2D PLANE

Projenin bu bölümünde 2 boyutlu (2D) Öklid uzayında noktalar üretmeniz ve bu noktalar üzerinde bazı hesaplamalar gerçekleştirmeniz istenmektedir. Şekil 1’de örnek bir 2D Öklid düzlemi verilmiş olup düzlemin başlangıç noktası (0, 0) sol üst köşe olarak belirlenmiştir.



Şekil 1. 2D noktaların Öklid düzleminde gösterilmesi.

2D Öklid uzayında herhangi iki nokta arasındaki uzaklık aşağıdaki formüle göre hesaplanmaktadır:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Örneğin, $P1(x_1=10.5, y_1=20.7)$ ve $P2(x_2=3.1, y_2=19.9)$ gibi iki adet nokta koordinatları için aralarındaki mesafe şu şekilde bulunacaktır:

$$d = \sqrt{(3.1 - 10.5)^2 + (19.9 - 20.7)^2} \cong 7.44$$

- a) **Rastgele Nokta Üretimi:** Genişliği (width) ve yüksekliği (height) verilen 2 boyutlu alan içerisinde n adet rastgele nokta üreten ve döndüren metodu yazınız. Üretilen noktalar **$n \times 2$ matris** içerisinde; her bir satır bir noktaya ve her bir sütun da sırasıyla x ve y koordinat değerlerine karşılık gelecek biçimde saklanacak ve döndürülecektir. Üretilen koordinatlar **double** tipinde olmalıdır.

Bu metodu aşağıdaki parametreler ile ayrı ayrı test ediniz: Test sonucu döndürülen matrisin bilgilerini (her bir nokta için x ve y koordinat değerleri) konsola yazdırınız.

1. **n=10, width=100, height=100**
2. **n=100, width=100, height=100**

b) Uzaklık Matrisi (Distance Matrix-DM) Üretimi: Kendisine verilen **$n \times 2$** noktalar matrisini (bir önceki maddede istenen metot kullanılarak üretilmiş) **$n \times n$** 'lik uzaklık matrisine çeviren ve döndüren metodu yazınız. Uzaklık matrisi (DM) her bir nokta çifti arasındaki uzaklık bilgisini içermektedir. Örneğin, $DM[i,j]$ i ve j noktaları arasındaki mesafeyi verecektir. Uzaklıklar simetrik olduğundan $DM[i,j]=DM[j,i]$ eşitliği sağlanacaktır (i'den j'ye uzaklık ile j'den i'ye uzaklık aynıdır). Şekil 2'de örnek bir uzaklık matrisi yer almaktadır.

Bu metodu n=10, width=100, height=100 parametreleri ile test ediniz. Üretilen DM'yi konsola yazdırınız.

Distance Matrix						
	0	1	2	3	4	5
0	0.0	1.2	0.5	4.7	5.6	4.9
1	1.2	0.0	3.1	2.0	1.4	4.0
2	0.5	3.1	0.0	6.1	2.8	1.9
3	4.7	2.0	6.1	0.0	2.1	3.5
4	5.6	1.4	2.8	2.1	0.0	3.3
5	4.9	4.0	1.9	3.5	3.3	0.0

Şekil 2. Altı adet nokta için örnek bir DM.

2 CLASSIFICATION USING K-NEAREST NEIGHBORS (KNN) ALGORITHM K EN YAKIN KOMŞU YÖNTEMİ İLE SINIFLANDIRMA

Gerçek ve sahte banknot görüntü örneklerinden çıkarılan çeşitli öznelilikler bulunmaktadır. Bu bilgiler aracılığı ile verilen görüntü gerçek/sahte olarak sınıflandırılabilir. Her bir örnek için 4'er adet özellik (varyans, çarpıklık, basıklık, entropi) bilgisi ve gerçek para olup/olmadığı (tür) hazır olarak verilmektedir. Tablo 1'de 6 tanesine yer verilmiştir. Elimizde toplam 1372 adet örneğe ilişkin veriler bulunmaktadır. Bu verileri kullanarak, görüntünün iki farklı türden hangisine ait olduğunu bulduran bir algoritmanın yazılması istenmektedir. Görüntü yerine görüntü öznelilikleri kullanılarak işlemler gerçekleştirilecektir.

Veriseti: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>

Tablo 1: Banknot verisetinden alınmış 6 adet banknot örneğine ilişkin bilgiler

Örnek No	Varyans Değeri	Çarpıklık Değeri	Basıklık Değeri	Entropi Değeri	Tür
0	3.6216	8.6661	-2.8073	-0.44699	0
1	4.5459	8.1674	-2.4586	-1.4621	0
2	3.866	-2.6383	1.9242	0.10645	0
3	-1.3971	3.3191	-1.3927	-1.9948	1
4	0.39012	-0.14279	-0.031994	0.35084	1
5	-1.6677	-7.1535	7.8929	0.96765	1
...

a) **kNN ile sınıflandırma:** Bulduğumuz ancak türünü bilmediğimiz bir banknotun hangi türe ait olduğunu tespit eden algoritmayı (**k en yakın komşu yöntemi**) yazınız (hazır kNN kullanmayınız). k değerini, kullanıcı tarafından girilebilen bir banknotun **tüm** özellik(ler)ini girdi olarak alarak bu yöntemle hangi sınıftan (*gerçek (1) / sahte (0)*) olduğunu bulduran kNN algoritmasını kendiniz yazınız.

kNN Yöntemi:

Elinizdeki türü bilinmeyen banknotun özelliklerini, verisetindeki tüm kayıtlarla karşılaştırarak özellikleri uzaklık d (distance) formülüne göre en yakın olan k tane banknotu bulmalısınız. Bulduğunuz bu k tane banknotun türlerine bakarak en çok sayıda hangi türden banknot varsa banknotunuzu o türden sayacak ve sınıflandıracaksınız.

$A = (x_1, x_2, \dots, x_m)$ ve $B = (y_1, y_2, \dots, y_m)$ özellik vektörleri, m özellik sayısı olmak üzere iki banknot (A ve B) arasındaki uzaklığı (distance) hesaplayan d(A,B) Formülü: $\sqrt{\sum_{i=1}^m (x_i - y_i)^2}$
Tablo 1'deki ikinci yani 1 numaralı banknotun özellikleri sırası ile 4.5459, 8.1674, -2.4586 ve -1.4621'dir.

Örnek olarak K değerini kullanıcı 3 girdiyse, verdiğiniz öznitelik dizisine uzaklığı en yakın (az) olan 3 banknotu tespit etmelisiniz. İki tanesi *gerçek (1)*, bir tanesi de *sahte (0)* ise oy çokluğu ile banknotu *gerçek* olarak sınıflandıracaksınız. Eğer oy çokluğu konusunda 1'den fazla banknot arasında eşitlik olursa en yakın banknotun türünde sınıflandırabilirsiniz (k=1 için).

* kNN tutorial: [KNN Algorithm - Finding Nearest Neighbors - Tutorialspoint](https://www.tutorialspoint.com/knn/knn_algorithm.htm)

* kNN demo: vision.stanford.edu/teaching/cs231n-demos/knn/

b) **Banknot sınıflandırma:** Yazdığınız kNN algoritmasının k değerini, kullanıcı tarafından girilebilen bir banknotun 4 adet özelliğini girdi olarak alarak, en yakın k adet banknotun **özelliklerini, uzaklıklarını ve hangi sınıflardan olduklarını** bir tablo olarak **ekrana listeleyiniz**. Bağlantısı önceki sayfada verilen verisetini kullanarak kNN yöntemi ile banknotun da türünü **tahminleyiniz ve ekrana yazdırınız**.

Dileyenler Tablo 2’de verilen örnek veriler ve tahminlenecek tür bilgisi için programını test edebilir.

Tablo 2: Banknot veriseti için verilen 10 adet yeni örnek ve k değeri için tahminlenecek türe ilişkin değerler

Örnek Veri No	Varyans	Çarpıklık	Basıklık	Entropi	K Değeri	Tahminlenecek Tür
1	1.89	-2.05	0.93	1.24	5	1
2	2.43	2.82	-2.79	-2.81	3	1
3	-2.24	2.74	2.09	-1.34	1	1
4	2.48	-0.09	2.60	-2.72	1	0
5	0.79	1.80	1.07	-2.42	3	0
6	-2.41	-2.15	1.55	1.94	5	1
7	-1.33	-0.47	1.46	1.17	3	1
8	0.28	2.49	-0.65	-1.10	3	1
9	2.75	1.75	0.93	2.70	1	0
10	2.79	2.76	-1.97	-2.79	5	1

c) **Başarı Ölçümü:** Verisetinde her bir tür banknot örneğinin sonunda yer alan 100’er veriyi test verisi olarak ayırınız. k değerini kullanıcından alarak, test verilerinden herbirini, a maddesindeki yöntemi ve 4 özelliğin tümünü kullanarak kalan 1172 adet örnek veri üzerinden sınıflandırınız, b’deki listelemeleri yapınız. Test verilerinin gerçek sınıfları ile, kNN ile tahminlediğiniz sınıflarını karşılaştırınız (gerçek ve tahminlenen türlerin / sınıfların her ikisini de yazdırınız). **Başarı oranını:**

doğru sınıflandırılan banknot sayısı / verisetinde test amaçlı kullandığınız toplam banknot sayısı

olarak hesaplayarak yazdırınız.

d) **Listeleme:** Bellekteki verisetindeki değerleri görüntüleyen kodu yazınız.

- İlk proje olduğundan her projede tek sınıf kullanılması yeterlidir. Dileyenler, her bir banknotun bilgilerini birer nesnede tutup işlemlerini yürütebilirler. Listeler için dizi, matris, ArrayList gibi önereceğiniz veri yapılarını kullanabilirsiniz. Dosya kullanımı anlatılmadığı için dileyen öğrenciler, veriyi programa yapıştırarak ayrıştırabilirler veya <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/file-system/how-to-read-from-a-text-file>.
- Programınız farklı değerler ile (örnek olarak k=1 ile 5 arasında değerler verilerek) test edilebilmelidir. Varsayımlarınızı raporda belirtmelisiniz. Sınıf (tür) sayısının 2 olarak sabit alındığını varsayabilirsiniz.