

**EGE UNIVERSITY**  
**FACULTY of ENGINEERING**  
**COMPUTER ENGINEERING DEPARTMENT**  
**204 DATA STRUCTURES (3+1)**  
**2021-2022 FALL SEMESTER**

**PROJECT 1 : (Arrays, Matrices, Methods, Classes, Random Numbers)**

**1. GENERATING DISTANCE MATRIX FROM POINTS IN A 2D PLANE**  
**(İKİ BOYUTLU DÜZLEMDE NOKTALARDAN UZAKLIK MATRİSİ**  
**ÜRETİLMESİ) and**  
**TRAVELING POINTS USING NEAREST NEIGHBOR ALGORITHM**  
**(EN YAKIN KOMŞU YÖNTEMİ İLE TÜM NOKTALARIN DOLAŞILMASI)**  
**2. DEVELOPING A PERCEPTRON MODEL and IMPLEMENTATION of**  
**CLASSIFICATION (Algılayıcı Modeli Üretilmesi ve Sınıflandırma Gerçekleştirimi)**

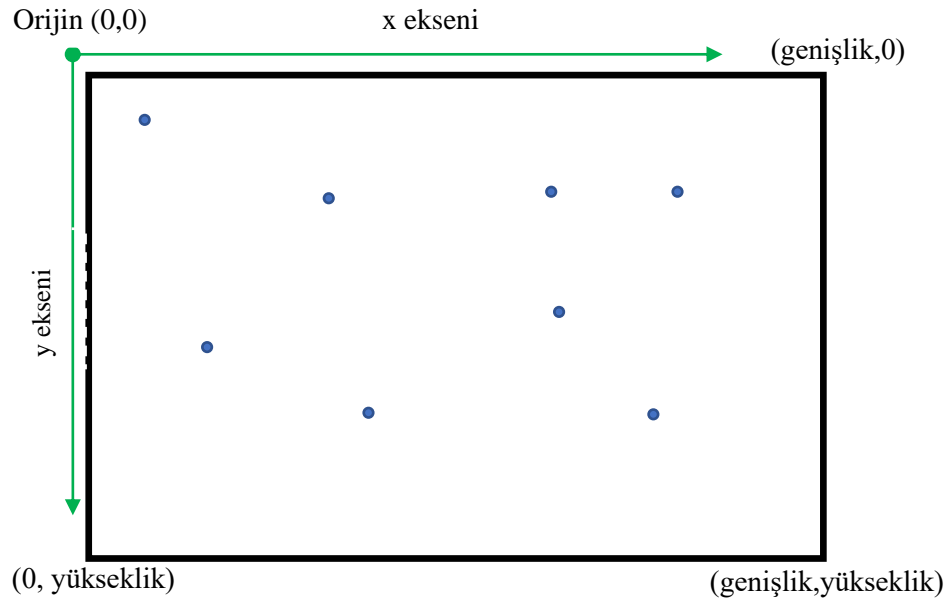
**Date Given : 09.12.2021**

**Submission Date : 23.12.2021**

\*Bu proje tek kişi olarak hazırlanabileceği gibi, iki veya en fazla üç kişilik gruplar oluşturularak da yapılabilir.

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

Projenin ilk 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: 2 boyutlu noktaların Öklid düzleminde gösterimi**

İki boyutlu uzayda iki nokta arasındaki Öklid uzaklığı aşağıdaki formüle göre hesaplanmaktadır:

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

Örnek olarak P1 ( $x_1=10.2$ ,  $y_1=20.7$ ) ve P2 ( $x_2=3.5$ ,  $y_2=19.9$ ) noktaları arasındaki uzaklık

$$d = \sqrt{(3.5 - 10.2)^2 + (19.9 - 20.7)^2} \cong 6.75 \text{ olarak bulunacaktır.}$$

10 p

- 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 x 2 'lik bir 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=20, width=100, height=100**
2. **n=50, width=100, height=100**

10 p

- b) **Uzaklık Matrisi (Distance Matrix-DM) Üretimi:** Kendisine verilen nx2 noktalar matrisini (bir önceki maddede istenen metot kullanılarak üretilmiş) nxn'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). Tablo 1'de örnek bir uzaklık matrisi yer almaktadır.

**Bu metodu n=20, width=100, height=100 parametreleri ile test ediniz.** Üretilen DM'yi konsola yazdırınız.

**Tablo 1: 6 adet nokta için örnek bir Uzaklık Matrisi (Distance Matrix – DM)**

Uzaklık Matrisi (DM)						
	0	1	2	3	4	5
0	0	1,2	0,5	4,7	5,6	4,9
1	1,2	0	3,1	2	1,4	4
2	0,5	3,1	0	6,1	2,8	1,9
3	4,7	2	6,1	0	2,1	3,5
4	5,6	1,4	2,8	2,1	0	3,3
5	4,9	4	1,9	3,5	3,3	0

15 p

- c) **n = 20 için rastgele bir noktadan başlayarak tüm noktaları en yakın komşu yöntemine (nearest neighbor) göre dolaşan metodu yazınız** (En yakın komşu yöntemi, Öklid uzaklığına göre başlangıç noktasına en yakın noktayı bularak devam eder. Sonra, dolaşılmamış noktalar içinde bu yeni noktaya en yakın noktaya gider. Tüm noktalar dolaşılana kadar bu işlem devam eder ve turu tamamlar). Toplam yol uzunluğunu hesaplayınız. Bunu 10 farklı rastgele başlangıç noktası için tekrarlayınız ve 10 farklı tur için aşağıdaki bilgileri konsolda listeleyiniz:

- DM (uzaklık matrisi),

Her bir tur için:

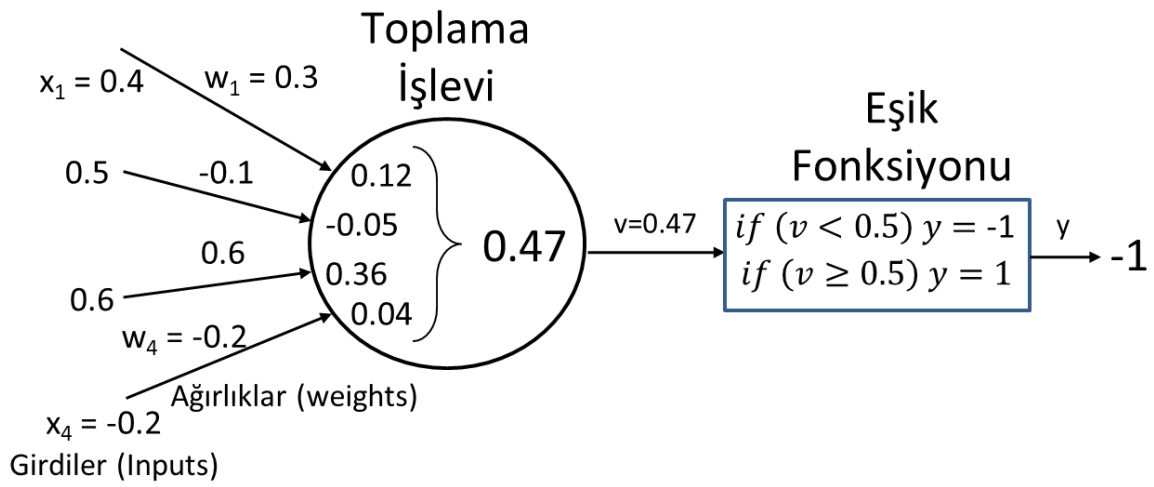
- Tur numarası (1'den 10'a kadar),
- İlgili turda sırayla hangi numaralı noktalara uğradığı, (5-2-0-1-3-4 gibi),
- Turun toplam yol uzunluğu (5-2-0-1-3-4 turu çıktıysa uzaklıklar toplamı).

## 2. DEVELOPING A PERCEPTRON MODEL and IMPLEMENTATION of CLASSIFICATION

**Bir yapay sinir hücresi üreterek (sınıfı oluşturarak) verilen sınıflandırma probleminin çözümünde kullanınız.**

Makine Öğrenmesi alanında kullanılan ve derin öğrenme alanının temelini de oluşturan Yapay Sinir Ağları (Artificial Neural Networks - ANN) konusundaki en temel yapılar Yapay Sinir Hücreleridir (Artificial Neuron). ANN'ler sınıflandırma, kümeleme ve tahminleme gibi birçok problemin çözümünde kullanılırlar.

Yapay sinir hücresinin yapısı ve örnek bir hesaplama işlemi Şekil 3'te gösterilmektedir. Şekildeki Algılayıcı (Perceptron) modelinin veya nöronun 4 adet girdisi (x) ve 1 adet çıktısı (y) bulunmaktadır.



Şekil 3: Perceptron Modeli ve İşleyişi

Toplama İşlevi, girdilerle ağırlıkların çarpımları toplamının alınması şeklinde gerçekleştirilir:

$$v_k = \sum_{i=1}^n w_i x_i = w_1 x_1 + w_2 x_2 + \dots + w_n x_n = [w_1 \ w_2 \ \dots \ w_n] \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Gözetimli Öğrenmede (Supervised Learning), girdilerle beraber olması gereken çıktı değerleri (target) verilir / sistem tarafından sağlanır. Elimizde, toplamaları pozitif olan iki adet sayıyı  $1$ , negatif olanları ise  $-1$  olarak sınıflandırmamızı gerektiren bir problemimiz olsun. Toplamaları  $0$  olan sayıları hariç tutuyoruz. Ağın eğitiminde kullanılmak üzere Eğitim (training) verileri üretelim (Tablo 2):

Tablo 2 : Veri Seti

	Veri 1	Veri 2	Veri 3	Veri 4	Veri 5	Veri 6	Veri 7	Veri 8
$x_1$	6	2	-3	-1	1	-2	-4	-6
$x_2$	5	4	-5	-1	1	7	-2	3
Target	1	1	-1	-1	1	1	-1	-1

**Ağın eğitimi işleminde kullanılan Öğrenme Kuralının aşağıdaki gibi verildiğini varsayın:**

- Ağın ürettiği çıktı -1 ve olması gereken değer (beklenen) değer yani target 1 ise, ağırlıkları öğrenme katsayısı  $\lambda \cdot (t-o) \cdot x_i$  kadar arttır (t:target, o:output olmak üzere).
- Ağın ürettiği çıktı 1 ve olması gereken değer (beklenen) değer yani target -1 ise, ağırlıkları öğrenme katsayısı  $\lambda \cdot (t-o) \cdot x_i$  kadar azalt.
- Output ve target değerleri aynı ise ağırlıklarda değişiklik yapma.

15 p

a) **Bir Neuron (Sinir Hücresi) sınıfı** oluşturunuz. Girdiler ve ağırlıkları tutmak için **uygun veri yapılarını** tercih ediniz. **Hesaplamaları ve gerekli işlemleri yapan metotları yazınız.** Ağırlıklar için en başta [-1, 1] aralığında rastgele double değerler üretiniz.

10 p

b) **Eğitim:** Tablo 1'deki tüm girdi değerlerini 10'a bölerek oluşan veri seti üzerinde **eğitim işleminizi gerçekleştiriniz.**  $\lambda = 0.05$  olarak alabilirsiniz. 10 epok ve 100 epok (epoch) sonunda yönteminizin veri seti üzerindeki doğruluk (accuracy) değerini hesaplayıp yazdırınız. Bir epok, tüm eğitim verilerinin sisteme bir kere sıra ile verilerek ağırlıkların değiştirilmesi işlemidir. Doğruluk değeri Doğru olarak sınıflandırılan örnek (veri) sayısı / toplam örnek sayısıdır. Elinizdeki 8 verinin 5 tanesi doğru olarak sınıflandırıldıysa doğruluk değeri  $acc = 5/8 = \%62.5$ 'tir.

10 p

c) **Test:** Eğitim verileri dışında farklı 5 tane test verisi oluşturarak yönteminizin doğruluğunu test ediniz. Test verilerini ve başarı değerini rapora ekleyiniz. Başarı değerini artırabilmek için neler yapabileceğinizi araştırınız ve rapora yazınız.

## PROJE TESLİMİNE İLİŞKİN BİLGİLER

- 1) Data Structures dersinin bu projesi için (ortak çalışma imkânınız bulunan kişilerle) **2 veya 3 kişilik çalışma grupları** oluşturunuz önerilir. Dileyen öğrenciler projeyi **tek** kişi olarak da hazırlayabilir. İkinci ve daha sonraki projeler ise sadece tek kişi olarak hazırlanacaktır, grup olarak teslim edilmeyeceklerdir.
- 2) **Rapor (20 p) hazırlanırken Egeders'te yer alan “DS\_21\_P1 Report Template.docx” şablonunu kullanınız. Sonda yer alan özdeğerlendirme tablosunu (10 p) doldurmayı ve rapor yazımı sonrası içindekiler bölümündeki sayfa numaralarını güncellemeyi unutmayınız.**
- 3) Projenizin işleyişini anlatan maksimum **5 dk'lık bir demo** videosu hazırlayınız. Video hazırlama için:
  1. Teams içerisinde **takvim** (calendar) sayfasına gidip “**şimdi toplantı yapın** (meet now)” seçeneği ile bir toplantı başlatın.
  2. Eğer grup çalışması yapıyorsanız diğer grup üyelerini de bu toplantıya dahil edin.
  3. Toplantı içerisinde kaydı başlatın.
  4. Ekran paylaşımı ile kodunuzu ve nasıl çalıştırıldığını, projede istenenleri dikkate alarak anlatın. (Grup çalışmasında her bir grup üyesi anlatımda görev almalıdır).
  5. Kaydı durdurun.
  - 6 **Toplantıdan ayrılmadan, sohbet** (chat) bölümünü açın. Videonuz burada oluşturulacaktır. Videonuza tıklayarak **indirin ve saklayın.**
    - Mikrofon gibi ekipman sıkıntısı yaşayanlar alternatif olarak demoyu ekran görüntüsü destekli yazılı doküman olarak da hazırlayabilirler. Demo dokümanı rapordan farklı olarak adım adım ekran görüntüleri ve ayrıntılı açıklama ile proje kodlarının nasıl

alıřtırılacađını anlatmalıdır (raporda sadece ilgili maddenin konsol ıktısı istenmekteydi). Bir bařka deyiřle video demosunun yazılı bir versiyonudur.

- 4) **Rapor, demo ve her iki programın aıklama satırları destekli kaynak kodları (.cs uzantılı), alıřma grubundan bir ğrenci tarafından (dersin duyurularında belirtilen formata uygun olarak) son teslim tarihine kadar ilgili seenekten sisteme **tek bir zip dosyası** halinde yklenmelidir.** Oluřturulan “zip” veya “rar” dosyasının adı 8 haneli ğrenci numarası ve ğrenci ad ve soyadının alt izgi ( \_ ) ile ayrılmasıyla oluřturulacaktır. Birden fazla ğrenci olması durumunda, proje yeleri birbirinden yine alt izgi kullanılarak ayrılacaktır.

**rneđin; “05146924\_AliYılmaz\_05130084\_Ayřeztrk.zip”**

- 5) **MOSS zerinden belli lde kod benzerlikleri grlen gruplara proje notu olarak 0 atanacaktır.**