



**YAPAY ZEKA
İLE KANSER TESPİTİ**



Projenin amacı:

C dilinde bir kod yazarak yapay zeka ile kanser tespiti yapmak.

Kullandığımız Araç:

Proje kapsamında, ödev ve LAB'larda kullanılan Microsoft'un derleyicisi olan Visual Studio Community kullanılacaktır.

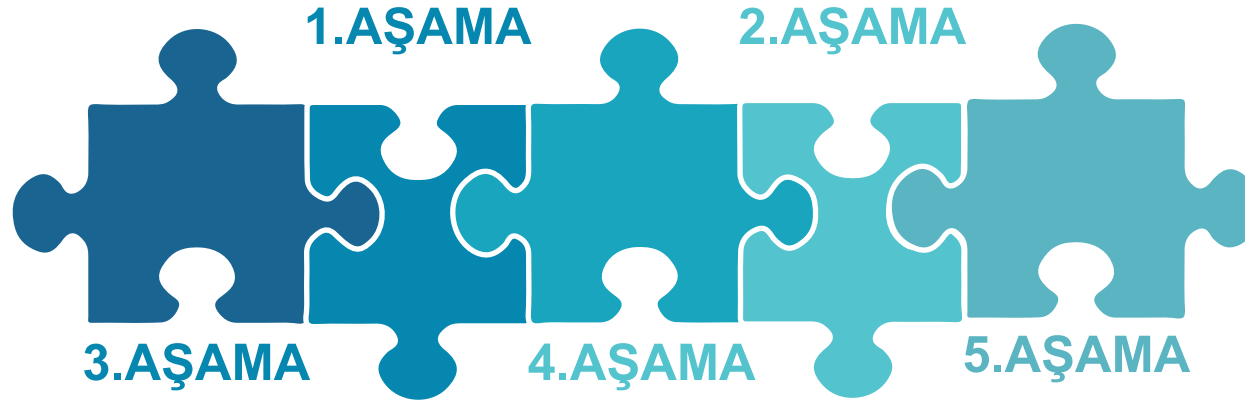


Peki nelerden yararlanılacak?

Bu proje yapay zeka uygulamalarında kullanılan bir algoritma olan kNN (k Nearest Neighborhood, En Yakın k Komşu) algoritması gerçekleştirilecektir. Gerçeklenen algoritma ile, UC Irvine Üniversitesi'nin sağlamış olduğu Göğüs Kanseri verileri işlenecektir. Hasta olup olmadığı belli olmayan bir kişinin verileri sisteme beslenerek, hastalık tahmini yapılacaktır.

En yakın k komşu sayısı için bir k değeri belirle, tek sayı olmalıdır. (3, 5, 7 ... gibi)

Test edilecek veriyi, verisetindeki tüm verilere göre uzaklığını bul



Uzaklıkları küçükten büyüğe doğru sırala

Sıralanmış örneklerin en küçük uzaklığa sahip k örneği al

K örneğin içindeki sınıf değeri en çok ne ise, test edilen verinin sınıf değeri olarak belirle

Elimizde 4 örnek bulunan bir kağıt mendilin kaliteli olup olmadığının ölçümü yapılmış bir veriseti vardır. Kağıtların kaliteli olup olmadığı, bu işin uzmanları tarafından belirlenmiştir.

X1=AsitDayanıklılığı (saniye)	X2=Esname(kg/m2)	Sınıflandırılma
7	7	Kötü
7	4	Kötü
3	4	İyi
1	4	İyi

Elimize yeni bir kağıt geldi. Bu kağıtın asit dayanıklılığı (X1) 3 ve esneme değeri (X2) 7 olarak geldi. Bu kağıtı bir uzmana göstermeden, mevcut versetini ve kNN yapay zeka algoritmasını kullanarak, kağıtın kalitesini nasıl belirlenebilir?

1. İlk aşama k değerini belirlemektir. K değeri 3 seçilmiştir (Örnek sayımız az olduğu için 3 seçildi, genellikle uygulamalarda 3 veya 5 seçilmektedir).
2. Eğitim verisi ile test edilen (3, 7) arasındaki uzaklıkları bul, uzaklık için Öklid yöntemi kullanılacaktır.

X1=AsitDayanıklılığı (saniye)	X2=Esname(kg/m2)	Uzaklıklar
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 9$
1	4	$(1-3)^2 + (4-7)^2 = 13$

3. Uzaklığa göre en küçükten en büyüğe doğru sırala

X1=AsitDayanıklılı ğı (saniye)	X2=Esname(kg/ m2)	Uzaklıklar	Sınıflar
7	7	$(3-3)^2 + (4-7)^2 = 9$	İyi
7	4	$(1-3)^2 + (4-7)^2 = 13$	İyi
3	4	$(7-3)^2 + (7-7)^2 = 16$	Kötü
1	4	$(7-3)^2 + (4-7)^2 = 25$	Kötü

4. k örneđi al, k sayısı 3 olduđu için ilk 3 örnek alınacaktır.

X1=AsitDayanıklılıđı (saniye)	X2=Esneme(kg/m ²)	Uzaklıklar	Sınıflar
7	7	$(3-3)^2 + (4-7)^2 = 9$	İyi
7	4	$(1-3)^2 + (4-7)^2 = 13$	İyi
3	4	$(7-3)^2 + (7-7)^2 = 16$	Kötü

5. k örneđin içindeki en çok sınıf değeri ne ise, sonuç olarak belirle. 3 örneđin içerisinde, iyi, iyi ve kötü sınıf değeri vardır. İyi değeri daha çok olduđu için, verilmiş test örneđi iyi olarak tespit edilmiştir.

ŞİMDİ KODUMUZU İNCELEYELİM

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>

int kullanicigirisihesapla(int diziegitim[600][10]);
int sonucbul(int diziegitim[600][10],int dizitest[83][10]);

int main()
{
    FILE* dosya;
    FILE* dosya2;
    int i;
    int istenilen;
    int a;
    int diziegitim[600][10];
    int dizitest[83][10];
    int sayi1,sayi2,sayi3,sayi4,sayi5,sayi6,sayi7,sayi8,sayi9,sayi10;
    int sayi11,sayi12,sayi13,sayi14,sayi15,sayi16,sayi17,sayi18,sayi19,sayi20;
```

- Öncelikle yaptığımız kod, fonksiyon şeklinde olduğu için bu fonksiyonları tanımladık. Bize verilen eğitim ve test verilerini dosyadan okumak için gerekli tanımlamaları yaptık. Her bir verisetinde 11 tane sayı bulunmaktadır. Bu 11 sayıdan ilki hastadan alınan örneğin numarası iken, son sayı sınıf bilgisi yani hasta olup olmadığını bildirmektedir. Geriye kalan 9 sayı hastanın değerleridir. Eğitim verisetinde 600 örnek varken test verisetinde 83 tane örnek olduğu için dizi halinde belirttik


```
fopen_s(&dosya, "egitim.txt", "r");

for(i=0;i<600;i++) {

fscanf_s(dosya, "%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d", &sayi1, &sayi2, &sayi3, &sayi4, &sayi5, &sayi6, &sayi7, &sayi8, &sayi9, &sayi10);

dizi[egitim[i]][0]=sayi1;
dizi[egitim[i]][1]=sayi2;
dizi[egitim[i]][1]=sayi2;
dizi[egitim[i]][2]=sayi3;
dizi[egitim[i]][3]=sayi4;
dizi[egitim[i]][4]=sayi5;
dizi[egitim[i]][5]=sayi6;
dizi[egitim[i]][6]=sayi7;
dizi[egitim[i]][7]=sayi8;
dizi[egitim[i]][8]=sayi9;
dizi[egitim[i]][9]=sayi10;

}

fclose(dosya);
```

Dizi halinde eğitim veriseti dosyadan okutulmaktadır.

```
fopen_s(&dosya2, "test.txt", "r");
for(a=0;a<83;a++) {

fscanf_s(dosya, "%d,%d,%d,%d,%d,%d,%d,%d,%d,%d", &sayi11, &sayi12, &sayi13, &sayi14, &sayi15, &sayi16, &sayi17, &sayi18, &sayi19, &sayi20);

dizi[test[a]][0]=sayi11;
dizi[test[a]][1]=sayi12;
dizi[test[a]][2]=sayi13;
dizi[test[a]][3]=sayi14;
dizi[test[a]][4]=sayi15;
dizi[test[a]][5]=sayi16;
dizi[test[a]][6]=sayi17;
dizi[test[a]][7]=sayi18;
dizi[test[a]][8]=sayi19;
dizi[test[a]][9]=sayi20;

}

fclose(dosya2);
```

Dizi halinde test veriseti dosyadan okutulmaktadır.

```
home:
printf("Test dosyasındaki verileri hesaplamak için 1 , Kullanıcıdan parametre alıp verileri hesaplamak için 2 giriniz:");
scanf_s("%d",&istenilen);
if(istenilen==1){

sonucbul(diziegitim,dizitest);

}

else if(istenilen==2){

kullanicigirisihesapla(diziegitim);

}
else {
printf("HATALI GİRİŞ TEKRAR GİRİN:");
goto home;
}
```

Kod, fonksiyon şeklinde olduğu için öncelikle kullanıcıya ne yapmak istediği sorulmaktadır.


```
int sonucbul(int diziegitim[600][10],int dizitest[83][10]) {
    int x,i,j;
    int m;
    int z;
    int sonuc=0,hastalikdegeri,enkucuk,gecicibellek,gecicibellek2;
    int hasta;
    int hastadegil;
    int dizitestsonucu[600][2];
    int dizitestsonucuDY[6];
    char dizitestsonucuDYsay[83];
    int k=5;
    float dogrusayisi=0;
    float yanlissayisi=0;
```

Kullanıcı 1'e bastıysa sonucbul fonksiyonuna gidilmektedir. Eğitim ve test dizileri,bu kısmı hesaplamada gerekli olan diğer argümanlar tanımlanıyor.


```
for(x=0;x<83;x++)
{
    for (j=0;j<600;j++)
    {
        sonuc=0;
        for(z=0;z<9;z++)
        {
            int diff = (diziegitim[j][z]-dizitest[x][z]);

            sonuc=sonuc+ diff * diff;

            dizitestsonucu [j][0]=sonuc;
        }

        hastalikdegeri=diziegitim[j][9];
        dizitestsonucu[j][1]=hastalikdegeri;
    }
}
```

Test ve eğitim verisetinin hasta numarası ve sınıf bilgisi hariç diğer 9 değerler arasındaki uzaklıkların bulunması için bir döngü kurduk. Test verisetindeki her bir hasta eğitim verisetinde 600 hastaya olan uzaklıkları bulunuyor. Sonuçlar iki boyutlu dizide kaydedilmektedir. Dizinin 0. İndeksine uzaklık değeri , 1. indeksine sınıf bilgisi kaydedilmektedir.


```
for (i = 0; i < 599; i++)
{
enkucuk= i;
for (j = i + 1; j < 600; j++){
if (dizitestsonucu[j][0] <dizitestsonucu[enkucuk][0])
    enkucuk= j;
}
gecicibellek=dizitestsonucu [enkucuk][0];
dizitestsonucu[enkucuk][0] = dizitestsonucu[i][0];
dizitestsonucu[i][0] = gecicibellek;

gecicibellek=dizitestsonucu [enkucuk][1];
dizitestsonucu[enkucuk][1] = dizitestsonucu[i][1];
dizitestsonucu[i][1] = gecicibellek;
}
```

Selection sort algoritması ile bulduğumuz uzaklıkları sıraladık.Sonuçları dizitestsonuc dizisinin 0.ineksine bağlı olarak sıraladık


```
hasta=0;
hastadegil=0;
for (i=0;i<k;i++)
{
    gecicibellek2=dizitestsonucu[i][1];
    if (gecicibellek2==2) {
        hastadegil++;
    }
    else {
        hasta++;
    }
}

if (hastadegil>hasta) {
    dizitestsonucuDY[i]=2;
}
else {
    dizitestsonucuDY[i]=4;
}

if (dizitestsonucuDY[i]==dizitest[x][9]) {
    dizitestsonucuDYSay[x]='D';
}

else {
    dizitestsonucuDYSay[x]='Y';
}
}
```

En yakın seçtiğimiz k tane dizitestsonucu değerinin 1. indeksine göre hasta ise 4'ü, hasta değil ise 2'yi arttırdık. Eğer hasta değiller fazla ise dizitestsonucuDY dizisine 2, eğer hastalar fazla ise 4 yazdırdık. Bu diziyi en son dizitest ile karşılaştırdık. Eğer eşleşiyorlarsa dizitestsonucuDYSay'a 'D', eşleşmiyorlarsa 'Y' yazdırdık.


```
for(m=0;m<83;m++)
{
    printf("%d.veri== %c\n",m+1,dizitestsonucuDYsay[m]);
    if (dizitestsonucuDYsay[m]=='D') {
        dogrusayisi++;
    }
    else {
        yanlissayisi++;
    }
}

printf("\n DOĞRU SAYISI:%f\n YANLIŞ SAYISI:%f\n",dogrusayisi,yanlissayisi);

printf("BAŞARI ORANI YÜZDE:%.3f\n",((dogrusayisi*100)/83));
getchar();
getchar();
return 0;
}
```

Yaptığımız 83 tane test verisetinde kaç tane doğru kaç tane Y olduğunu hesaplattırdık.Son olarak başarı oranını hesapladık


```
int kullanicigirisihesapla(int diziegitim[600][10])
{
    int dizikullanicigirisi[9];
    int dizikullanicigirisiuzaklik[600][2];
    int gecicibellek,gecicibellek2,sonuc,hastalikdegeri,i,j,enkucuk;
    int k=5,hasta=0,hastadegil=0;
    int z;

    for(i=0;i<9;i++){
        printf("%d.degeri giriniz:",i+1);
        scanf_s("%d",&gecicibellek);
        dizikullanicigirisi[i]=gecicibellek;
    }
}
```

Kullanıcıdan parametre alıp hastanın kanser olup olmadığını hesaplamak için gerekli argümanları tanımladık.dizikullanicigirisi adında bir dizi tanımlayarak 9 değeri kullanıcıdan aldık


```
for (j=0;j<600;j++) {  
  
    sonuc=0;  
    for(z=0;z<9;z++) {  
        int diff = (diziegitim[j][z]-dizikullanigirisi[z]);  
  
        sonuc=sonuc+ diff * diff;  
        hastalikdegeri=diziegitim[j][9];  
  
        dizikullanigirisiuzaklik[j][0]=sonuc;  
        dizikullanigirisiuzaklik[j][1]=hastalikdegeri;  
    }  
}
```

Aldığımız 9 değeri eğitim verisetindeki değerlere uzaklığını bulup ,bulunan sonucu dizikullanigirisiuzaklik adındaki dizinin 0.ineksine kaydettik


```
for (i=0;i<599;i++) {
    enkucuk=i;
    for(j=i+1;j<600;j++)
    {
        if (dizikullanicigirisiuzaklik[j][0]<dizikullanicigirisiuzaklik[enkucuk][0])
            enkucuk=j;

        gecicibellek=dizikullanicigirisiuzaklik[enkucuk][0];
        dizikullanicigirisiuzaklik[enkucuk][0]=dizikullanicigirisiuzaklik[i][0];
        dizikullanicigirisiuzaklik[i][0]=gecicibellek;

        gecicibellek=dizikullanicigirisiuzaklik[enkucuk][1];
        dizikullanicigirisiuzaklik[enkucuk][1]=dizikullanicigirisiuzaklik[i][1];
        dizikullanicigirisiuzaklik[i][1]=gecicibellek;
    }
}
```

Selection sort algoritması ile bulduğumuz uzaklık değerlerini sıraladık.


```
for(i=0;i<k;i++) {
    gecicibellek2=dizikullanigirisiuzaklik[i][1];
    if(gecicibellek2==2) {
        hastadegil++;
    }
    else {
        hasta++;
    }
}

if(hastadegil>hasta){
    printf("\n HASTA DEGİLSİNİZ");
}
else {
    printf("\n HASTASİNİZ");
}

getchar();
getchar();
return 0;
}
```

En yakın k tane değere göre hasta değil ve hasta değerlerini hesapladık. Bu değerlere göre eğer hasta sayısı hasta değil sayısından fazla ise ekrana HASTASINIZ yazısını, hasta değil sayısı fazla ise HASTA DEĞİLSİNİZ yazısını yazdırdık

HAZIRLAYANLAR



Deniz Uzun

Student

Fenerbahçe Üniversitesi
Bilgisayar Mühendisi
Öğrencisi
(190301015)



Özlem Çalı

Student

Fenerbahçe Üniversitesi
Bilgisayar Mühendisi
Öğrencisi
(190301002)



TEŐEKKÜR EDERİZ.