

T.C.
İNÖNÜ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

EKG SİNYALİ İLE BELİRLİ HASTALIKLARIN TEŞHİSİ

Hazırlayanlar

Ayberk SOYVAR

Ayşe ÖZAVCI

Bekir DEMİR

Danışman

Doç.Dr. BİLAL ŞENOL

İÇİNDEKİLER TABLOSU

İÇİNDEKİLER TABLOSU	i
TEŞEKKÜR	ii
ŞEKİLLER LİSTESİ.....	iii
1. GİRİŞ	1
1.1 Projenin Özeti.....	2
1.2 EKG Sinyalinin Teknik Analizi	3
2. PROJENİN TEKNİK ALTYAPISI	4
2.1 Kullandığımız Makine Öğrenmesi Olan K-NN Algoritması Nedir ?	4
2.2 Neden K-NN Algoritmasını Kullanıyoruz ?	5
2.3 Neden EKG Sinyalini Gürültüden Arındırma İhtiyacı Duyuyoruz ?	5
3. PROJENİN KODLARI VE AÇIKLAMASI	5
3.1 Projenin Temel İşleyiş Mantığı ve Amacı	6
3.2 Projedeki Temel Sorunlar ve Yapılanlar.....	7
3.3 Projenin Kodları ve Açıklamaları	8
3.3.1 KNN_Hesapla Fonksiyonu	9
3.3.2 KNN_Siniflandir Fonksiyonu.....	10
3.3.3 Uygulamanın Main Kodları.....	11
3.3.4 EKG Sinyalinin Grafik Olarak Çizdirilmesi	12
3.3.5 EKG Sinyalini Yumuşatma ve Gürültü Giderme İşlemi	13
4. SONUÇ VE KAZANIMLAR.....	15
5. KAYNAKLAR	16

TEŞEKKÜR

Bu bitirme projemizin her aşamasında bizden yardım ve desteğini esirgemedi bizi her konuda yönlendiren danışman hocamız sayın Doç.Dr. BİLAL ŞENOL'a ;

Ayrıca bugüne kadar benden desteklerini esirgemeyen çok değerli AİLEM'e teşekkür ederim.

ŞEKİLLER LİSTESİ

Şekil 1.1	Projenin Özet Diyagramı.....	2
Şekil 1.2	EKG Sinyali Teknik Analizi	3
Şekil 2.1	K-NN Algoritması Görseli	4
Şekil 2.2	Sınıflandırma Algoritmaları ve Sınıflandırma Başarımları	5
Şekil 3.1	KisiSinyali Class Yapısı Görseli	8
Şekil 3.2	KNN_Hesapla Fonksiyonu Görseli	9
Şekil 3.3	KNN_Siniflandir Fonksiyonu Görseli	10
Şekil 3.4	Main Scripti Kodlarının Görseli	11
Şekil 3.5	Projenin Çıktı Görseli	11
Şekil 3.6	EKG Sinyalini Çizdirme Kodu	12
Şekil 3.7	EKG Sinyalinin Çizdirilmiş Hali	12
Şekil 3.8	EKG Sinyalinin Peek Noktalarını Bulma Kodu	13
Şekil 3.9	EKG Sinyalini Butterworth Filtresinden Geçirme Kodları	14
Şekil 3.10	EKG Sinyalinin Filtre İşlemi Uygulanmış ve Uygulanmamış halinin karşılaştırılması	14

1. GİRİŞ

Bitirme projemizin konusu , bizim belirlediğimiz bir makine öğrenmesi algoritmasını kullanarak ve çeşitli yöntemlerin de yardımıyla EKG sinyallerini sınıflandırmaktı. En başta bir kişinin EKG sinyaline bakarak ;

- Aritmi Hastası mı ?
- Sağlıklı mı ?
- Risk Grubunda mı ?

Sorularına cevap vermek için **Sağlıklı** , **Hasta** veya **Risk Grubunda** olmak üzere 3 farklı sınıf ile sınıflandırma yapmak gibi bir hedefimiz vardı. Fakat sonrasında , veri setini alacağımız Veritabanında , Sağlıklı ve Risk Grubunda olan kişilerin EKG sinyalleri bulunmadığı için , bu sınıflandırmaya **Sağlıklı** ve **Risk Grubunda** kısımlarını dahil edemeyeceğimizi anladık. Kullanacağımız veri setindeki veriler , Ayakta Tedavi Gören (Hafif Hasta) ve Yatarak Tedavi Gören (Ağır hasta) olarak 2 sınıfa ayrılıyordu.

Bizim bu projede kullandığımız makine öğrenmesi algoritması olan K-NN algoritması gereği , bir sınıf ile bir veriyi sınıflandırmak istiyorsak , kullandığımız veri setinde o sınıftan bulunmak zorundaydı. Bu yüzden de projenin hedefini **Ağır** ve **Hafif** aritmi hastalarının sınıflandırılması olarak değiştirdik.

Bitirme tezinin sonlarında nedenlerini açıklayacağım sebepten ötürü proje tamamlanmamış olsa da , yapacağımız şeylerin çoğunu yapmış ve bu projenin bize katması gereken tecrübeleri edinmiş bulunuyoruz.

Bitirme projesinde kullandığımız makine öğrenmesi algoritmasını , kendimizin verdiği temsili değerler ve sınıflar ile çalıştırdık. **MIT-BIH Arrhythmia Database**'den aldığımız .mat uzantılı EKG sinyallerini , gürültüden arındırdık. Araştırmalarımız sonucu hangi EKG sinyali özelliklerinin Aritmi hastalığının teşhisinde kullanıldığını öğrendik. Bütün bu süreçler , hiç fikrimizin olmadığı bu konularda bile , nasıl araştırma yapıp nasıl bir yol izlememiz gerektiğini öğretti.

Bir sonraki sayfada , bizim şu ana kadar bitirme projesi için neler yaptığımızın bir özeti şeklinde oluşturduğumuz diyagram bulunuyor.

1.1 Projenin Özeti

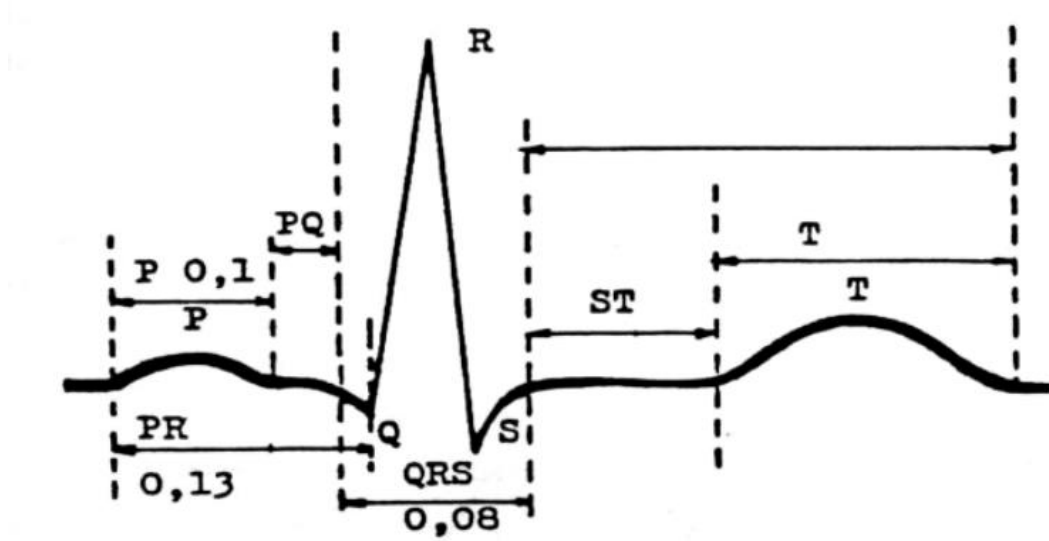
Şekil 1.1 Projenin Özet Diyagramı



1.2 EKG Sinyalinin Teknik Analizi

İşleyeceğimiz sinyal olan EKG sinyalinin , öznitelik çıkarma ve sınıflandırma aşamasında sinyalin içerdiği dalgaları ve bunların anlamlarını anlatarak bu kısma başlamak istiyoruz.

Şekil 1.2 EKG Sinyali Teknik Analizi



Normal EKG sinyali , kalbin dinlenme durumundaki taban seviyesi üzerine sıralanan belli başlı P, Q, R, S ve T adları verilen dalgalardan oluşur. Bazen T dalgasını takiben küçük genlikli bir U dalgası da olabilir.

P dalgası olarak isimlendirilen kısım atriumların kasılması sonucu oluşur. PQ aralığı his demeti iletim zamanını gösterir. QRST dalgası, ventriküler kompleks olarak isimlendirilir. QRS, ventriküler depolarize olması anlamına gelir. His demeti ve kollarındaki iletim bozuklukları QRS dalgasında değişikliklere yol açar.

EKG Sinyalindeki QRS dalgası , kalbin kasıldığı anı yani sol ventrikül depolarizasyonunu gösterir. Bizim projemiz için EKG sinyalini işlerken en kritik kısım , sinyalin QRS dalgası ve iki QRS dalgasının R yani sinyalin tepe noktaları arasındaki **RR** aralığıdır.

2. PROJENİN TEKNİK ALTYAPISI

Bu kısımda , projemizin bütün teknik sürecinden bahsedeceğiz. Seçtiğimiz yöntemlerin sebep ve sonuçlarını açıklayacağız. Tamamen bizim yazdığımız Makine Öğrenmesi algoritmasının kodlarından bahsedeceğiz.

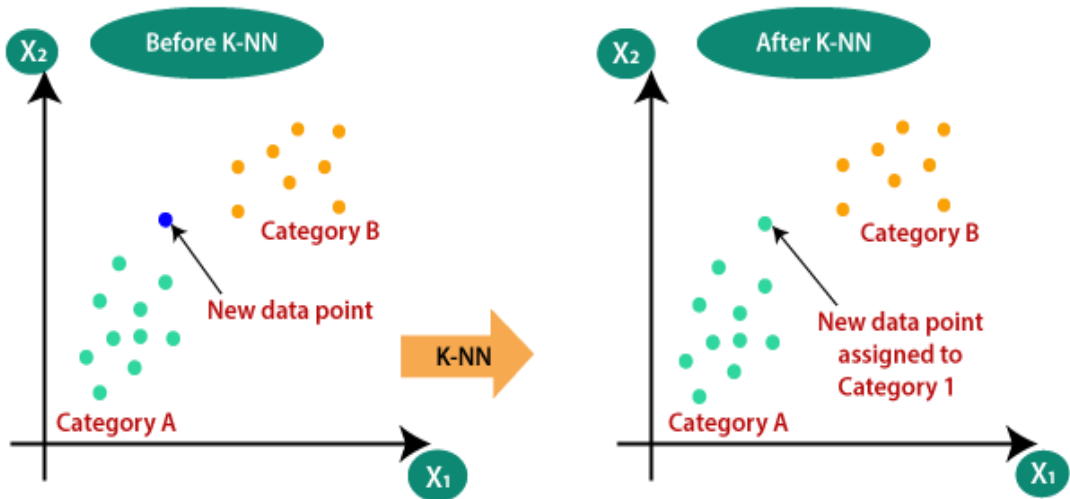
2.1 Kullandığımız Makine Öğrenmesi Olan K-NN Algoritması Nedir ?

KNN algoritması sınıflandırılmak istenen bir veriyi daha önceki verilerle olan yakınlık ilişkisine göre sınıflandıran bir algoritmadır. K-NN (K-Nearest Neighbor) algoritması en basit ve en çok kullanılan sınıflandırma algoritmasından biridir. K-NN non-parametric (parametrik olmayan) , lazy (tembel) bir öğrenme algoritmasıdır. Lazy(Tembel) kavramını anlamaya çalışırsak Eager(İstekli) Learning'in aksine Lazy Learning'in bir eğitim aşaması yoktur. Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini “ezberler”. Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar. En yakın komşuları bulmak için de veriler arasında uzaklıkları hesaplar ve genelde 3 tip uzaklık fonksiyonu kullanılmaktadır. Bunlar :

- **Euclidean (Öklid) Uzaklığı**
- **Manhattan Uzaklığı**
- **Minkowski Uzaklığı**

Biz projemizde **Euclidean (Öklid)** uzaklığını kullandık.

Şekil 2.1 K-NN Algoritması Görseli



2.2 Neden K-NN Algoritmasını Kullanıyoruz ?

Araştırmalarımıza göre , bu tip sinyal işleme projelerinde K-NN Makine Öğrenmesi Algoritmasının uygulanması hem daha basit hem de makine öğrenmesi algoritmalarını karşılaştırmak için kullanılan doğruluk , duyarlılık ve özgüllük oranı en yüksek algoritma olduğunu gördük.

Yapılan bir projede EKG Sinyalinin sınıflandırılmasında aşağıdaki tabloda yer alan makine öğrenmesi algoritmaları kullanılmış ve sonuçlara bakınca en başarılı algoritmanın K-NN algoritması olduğu görülmektedir. Projenin çeşidine , projede kullanılacak veri tipine ve sayısına bağlı olarak bu algoritmaların başarımları farklılık gösterebiliyor olsa da , bizim yapacağımız projede K-NN algoritmasının en iyi sonuç verecek makine öğrenmesi algoritması olduğunu düşünüyoruz.

Şekil 2.2 Sınıflandırma Algoritmaları ve Sınıflandırma Başarımları

Sınıflandırma algoritmaları ve sınıflandırma başarımları (%)			
Algoritma - Başarım	Doğruluk	Duyarlılık	Özgüllük
YSA-1	98,67	99,33	90,87
YSA-2	98,61	99,21	91,56
YSA-3	98,85	99,49	91,31
KNN-1	99,31	99,76	94,09
KNN-2	99,11	99,79	91,05
NAIVE BAYES	92,28	92,59	88,28
DT	98,76	99,43	90,93
DT - ID3	98,65	99,1	93,31
SVM	95,91	99,52	53,34

2.3 Neden EKG Sinyalini Gürültüden Arındırma İhtiyacı Duyuyoruz ?

Bu aşamanın diğer bir adı da “Normalizasyon ve Ön işleme”dir. EKG sinyallerinin genliklerindeki değişim, elde edilecek öznitelikleri olumsuz yönde etkilemektedir. Farklı hastalardan alınan aynı tür EKG sinyallerinde dahi dikkate değer bir değişim gözlemlenebilmektedir.

Normalizasyon ve ön işleme yapılarak EKG sinyalindeki değişiklikler (gürültü ve taban hattından sapma) en aza indirilir. İşlem yükünü azaltmak için sinyalin ortalaması sıfırlanır. Böylece EKG özniteliklerinin hastanın yaşına, cinsiyetine ve ölçüm sisteminin parametrelerine olan bağımlılık minimum düzeye indirilmiş olur.

3. PROJENİN KODLARI VE AÇIKLAMASI

Bu bölümde projemizin kodları ile alakalı bütün açıklamaları yapacağız. Kodlarımızda verilen değerlerin hangi mantığa göre verildiği veya Makine Öğrenmesi algoritmamız olan K-NN algoritmasını kodlarken yazdığımız kodların içeriği de açıklanacaktır.

3.1 Projenin Temel İşleyiş Mantığı ve Amacı

Uygulamamız , veri setindeki belirli iki parametreye mantıksal değerler ve mantıksal durumlar verip , K-NN algoritması ile bundan bir sonuç çıkarımı yapıyor. Uygulamamızdaki örnek veri setini biz oluşturduk ve bu örnek veri seti kişilerin EKG sinyallerinden oluşuyor. Bu veri setindeki sinyallerin 2 parametresi ve bunun sonucunda olabileceği 3 tane durumu var. Bu parametreler **X** ve **Y** değerleridir. Bu parametrelerin anlamsal olarak bir karşılığı gözükme de veri setini oluştururken verdiğimiz değerler , bu parametreleri anlamlı kılıyor.

Bu parametrelerin değerlerini belirlerken şöyle bir yol izledik.

- ➔ **X** parametresi 0-1 arası bir değer alır. Bu değer ne kadar artar ise , kişinin hasta olma ihtimali artar.
 - Veri setinde halihazırda bulunan kişi sinyallerinin **X** parametresi ne kadar 0 değerine yakın ise , kişi o kadar **SAĞLIKLI** olabilir demektir.
 - Veri setinde halihazırda bulunan kişi sinyallerinin **X** parametresi ne kadar 1 değerine yakın ise , kişi o kadar **HASTA** olabilir demektir.
 - Veri setinde halihazırda bulunan kişi sinyallerinin **X** parametresi ne kadar ortalamaya yakın yani 0.5 değerine yakın ise , kişi o kadar **RİSK GRUBUNDA** olabilir demektir.
- ➔ **Y** parametresi 0-1 arası bir değer alır. Bu değer ne kadar çok artar ise ,_kişinin hasta olma ihtimali azalır.
 - Veri setinde halihazırda bulunan kişi sinyallerinin **Y** parametresi ne kadar 0 değerine yakın ise , kişi o kadar **HASTA** olabilir demektir.
 - Veri setinde halihazırda bulunan kişi sinyallerinin **Y** parametresi ne kadar 1 değerine yakın ise , kişi o kadar **SAĞLIKLI** olabilir demektir.
 - Veri setinde halihazırda bulunan kişi sinyallerinin **Y** parametresi ne kadar ortalamaya yakın yani 0.5 değerine yakın ise , kişi o kadar **RİSK GRUBUNDA** olabilir demektir.

Yani , ilgili kişi sinyalinin belirlediğimiz 3 durum olan **HASTA** , **SAĞLIKLI** ve **RİSK GRUBUNDA** sınıflarında olması , kişi sinyalinin bu iki parametresine bağlı olup , durumu belirlenmesi yani sınıflandırılması istenen kişi sinyalinin, veri setindeki en yakın olduğu kişi sinyallerinin durumlarına bağlıdır. Bu söylediğim işleyiş de zaten K-NN algoritmasının getirdiği bir durumdur ve uygulamamızda K-NN algoritmasının kullanıldığının bir göstergesidir.

3.2 Projedeki Temel Sorunlar ve Yapılanlar

Şu ana kadar ki anlattıklarımıza bakacak olursak bir sorun yok gibi duruyor. Kaldı ki , belli bir zamana gelene kadar durum böyleydi. Fakat projemizde , planlandığı gibi yukarıda açıkladığım temsili değerler yerine gerçek EKG Öznitelikleri kullanılamadı. Projemizdeki yazdığımız K-NN algoritması , en başta verdiğimiz bu temsili değerleri ve bu temsili veri setini kullanıyor. Yani özet olarak , proje planlandığı gibi sonlanmadı. Bunun sebepleri olarak ise ;

- Hasta , Sağlıklı ve Risk Grubunda olan kişilerin EKG sinyallerini elde edebileceğimiz bir veri setini çok araştırsak da bulamadık.
- Başta söylediğimiz gibi , sonrasında bulduğumuz veritabanı sadece 2 tip EKG sinyali içeriyordu. Bunlar **Ağır Hasta** ve **Hafif Hasta**'ydı. Elimizde sadece bu veri seti olduğundan dolayı projeyi 3 sınıf ile sınıflandırmak yerine **Ağır Hasta** ve **Hafif Hasta** olmak üzere 2 sınıf ile sınıflandırmaya karar verdik.
- Veri setindeki bütün EKG sinyallerini indirip MATLAB ortamında açabildik ve çizdirebildik. Bu konuda bir sıkıntı yoktu. Fakat hangi sinyalin hangi hasta tipinden geldiği veri setinde belirtilmediği için biz de bu projeyi tam olarak sonlandıramadık. Çünkü K-NN algoritması gereği , veri setinde bulunan değerlerin hangi sınıfa ait olduğunun önceden bilinmesi gerekiyordu. Çünkü Sınıflandırma işlemi bu verilere göre yapılıyordu.

Alt tarafta da görüleceği üzere , biz projemizin kodlarını yazarken , 3 sınıflı bir Sınıflandırma Algoritmasına göre yazdık. Yani oluşturduğumuz örnek veri seti ve verdiğimiz temsili parametreler , 3 tane sınıfa göre sınıflandırma yapabiliyor. Normalde K-NN algoritması doğası gereği sadece 2 sınıf için sınıflandırma yapabiliyorken , bizim araştırmalarımız sonucu bulduğumuz **Ağırlık Oylama** yöntemi ile bunu başarabildik. Bu sebepten ötürü aslında , baştaki hedefimize ulaştık diyebiliriz. Çünkü parametre sayısı değişebilir ve temsili parametreler yerine gerçek EKG Öznitelikleri gelebilir. Böylece proje , asıl amacına ulaşabilir. Biz bunu yapmak için gereken bütün kodları yazdık. Fakat ulaşamadığımız veriler ve kaynaklar , projenin nihai sona ulaşamamasına neden oldu.

3.3 Projenin Kodları ve Açıklamaları

Kodlarımızı yazmadan önce MATLAB ortamını öğrenirken , MATLAB’ın class yapısını desteklediği Nesne Tabanlı Konsepti ile kod yazılabildiğini öğrendik. Halihazırda Java daki yazdığımız kodun da Nesne Tabanlı konsepti ile yazılması ve bu konsepti aşına olmamız , MATLAB’da da aynı yöntemi uygulamamıza imkan sağladı.

Şekil 3.1 KisiSinyali Class Yapısı Görseli

```
classdef KisiSinyali

    properties
        x_degeri
        y_degeri
        durum
        ilgili_noktaya_uzaklik
    end

    methods
        function obj = KisiSinyali(x_degeri,y_degeri,durum)
            if nargin < 3
                obj.x_degeri = x_degeri;
                obj.y_degeri = y_degeri;
            else
                obj.x_degeri = x_degeri;
                obj.y_degeri = y_degeri;
                obj.durum = durum;
            end
        end
    end
end
```

KisiSinyali.m classındaki bu kod , Java da ki Entity (Varlık) sınıfı gibi düşünülebilir. Sinyal’in bütün özellikleri ile beraber tanımlandığı bir class yapısıdır. x_degeri ve y_degeri nitelikleri temsili değerler olup ilerleyen zamanlarda bu değerlerin yerinde Aritmi hastalığı için EKG sinyallerinde ayırt edici özellikler olan **QRS genişliği, QRS genliği, RR aralığı** nitelikleri olacaktır.

3.3.1 KNN_Hesapla Fonksiyonu

KNN_Hesapla adında yazdığımız fonksiyon ise , sınıflandırılmak istenen sinyali , veri setindeki bütün sinyaller ile belirlediğimiz öznitelikler kapsamında aralarındaki Öklid uzaklığını bulup **ilgili_noktaya_uzaklik** adındaki sinyal değişkenine atar.

Şekil 3.2 KNN_Hesapla Fonksiyonu Görseli

```
function dizi = KNN_Hesapla(dizi, siniflandirilacak_sinyal)

    for k = 1 : length(dizi)
        a = [dizi(k).x_degeri dizi(k).y_degeri];
        b = [siniflandirilacak_sinyal.x_degeri siniflandirilacak_sinyal.y_degeri];
        dizi(k).ilgili_noktaya_uzaklik = norm(a-b);
    end

    [~, ind] = sort([dizi.ilgili_noktaya_uzaklik]);
    dizi = dizi(ind);

end
```

Ondan sonra ise , **ilgili_noktaya_uzaklik** değişkenine göre küçükten büyüğe doğru veri setindeki sinyalleri sıralar. K-NN Algoritmasına göre en küçük mesafeler , sınıflandırılmak istenen sinyale en yakın değerler olduğu için bu işlemi yapıyoruz. Bu sıralanan dizi ise , fonksiyona parametre olarak gelen diziye yani veri setine eşitlenip kullanılmak üzere fonksiyondan geri döndürülür.

3.3.2 KNN_Siniflandir Fonksiyonu

KNN_Siniflandir fonksiyonu ise , veri setinde sınıflandırılacak sinyale en yakın sinyaller sıralandıktan sonra , en yakın **k** tane sinyali alıp **Ağırlıklı Oylama** yöntemi ile **SAĞLIKLI** , **HASTA** veya **RİSK GRUBUNDA** şeklinde sınıflandırır. Eğer 2 tane sınıf olsaydı , Ağırlıklı Oylama yöntemini kullanmaya gerek kalmazdı fakat bizim projemizde 3 tane sınıfımız olduğu için , Ağırlıklı Oylama yöntemini kullanmak zorundaydık. Çünkü , seçilen her **k** değerinde , 3 sınıftan en az 2 sinin sayısının eşit gelme olasılığı vardı ve eşit gelme durumunda da sınıflandırma yapılamazdı. Biz de , kodda görüleceği üzere veri setindeki sınıflandırılacak sinyale en yakın sinyalleri Ağırlıklı Oylama yöntemiyle birbirine eşit olma durumu nerdeyse hiç olmayan bir işlem uyguluyoruz ve sınıflandırma her zaman yapılmış oluyor.

Şekil 3.3 KNN_Siniflandir Fonksiyonu Görseli

```
function siniflandirilacak_sinyal = KNN_Siniflandir(dizi,siniflandirilacak_sinyal,k_degeri)

hastaAgirlikliOylama = 0;
saglikliAgirlikliOylama = 0;
riskGrubuAgirlikliOylama = 0;

for k = 1:k_degeri
    if dizi(k).durum == "HASTA"
        hastaAgirlikliOylama = hastaAgirlikliOylama + 1 / (power(dizi(k).ilgili_noktaya_uzaklik,2));
    elseif dizi(k).durum == "SAĞLIKLI"
        saglikliAgirlikliOylama = saglikliAgirlikliOylama + 1 / (power(dizi(k).ilgili_noktaya_uzaklik,2));
    else
        riskGrubuAgirlikliOylama = riskGrubuAgirlikliOylama + 1 / (power(dizi(k).ilgili_noktaya_uzaklik,2));
    end
    disp(dizi(k).durum);
end
disp("Hasta Ağırlıklı Oylama --> "+hastaAgirlikliOylama);
disp("Sağlıklı Ağırlıklı Oylama --> "+saglikliAgirlikliOylama);
disp("Risk Grubunda Ağırlıklı Oylama --> "+riskGrubuAgirlikliOylama);

maks_agirlikli_oylama = max([hastaAgirlikliOylama,saglikliAgirlikliOylama,riskGrubuAgirlikliOylama]);

if hastaAgirlikliOylama == maks_agirlikli_oylama
    siniflandirilacak_sinyal.durum = "HASTA";
elseif saglikliAgirlikliOylama == maks_agirlikli_oylama
    siniflandirilacak_sinyal.durum = "SAĞLIKLI";
else
    siniflandirilacak_sinyal.durum = "RİSK GRUBU";
end

end
```

En yakın k tane sinyalin **ilgili_noktaya_uzaklik** değerine Ağırlıklı Oylama yöntemi formülünü uygulayıp , sinyalin sınıfına bağlı olarak bu sınıf çerçevesinde bu değerleri topluyoruz ve en yüksek toplama sahip sınıfı da , sınıflandırılmak istenen sinyalin sınıfı olarak buluyoruz.

3.3.3 Uygulamanın Main Kodları

Main.mlx scripti de , uygulamamızın başladığı yer olarak düşünülebilir. Başlangıçta , biz projemizi Nesne Tabanlı Konseptine uygun olarak yazdığımızı söylemiştik. Bu sebepten ötürü , sinyalin varlık (Entity) classı olan **KisiSinyali.m** 'den veri setini oluşturup tutabileceğimiz bir Nesne dizisi oluşturduk. Bu diziyi **sinyal_dizisi** adını verdik. Böylece , kişi sinyalleri çok daha düzenli bir şekilde uygulamada tutulacaktı. Bizim kendi mantığımızca oluşturduğumuz ve sonrasında yerine gerçek EKG verilerinin geleceği örnek veri setini de burada oluşturduk.

Ardından yine kendi mantığımızca , **sınıflandırılacak_sinyal** adında yine KisiSinyali tipinde sınıflandırmak istenen sinyali oluşturduk. Sonrasında da , şu ana kadar kodlarını ve işlevlerini anlattığımız fonksiyonları burada çağırıp ve bu değerleri parametre olarak gönderip sonucunda sınıflandırma işlemi başarıyla yapabildik.

Şekil 3.4 Main Scripti Kodlarının Görseli

```
clc;
clear all;

k_degeri = 3; %Uygulamamız için kritik olan değer

sinyal_dizisi = KisiSinyali.empty;

sinyal_dizisi(1) = KisiSinyali(0.9, 0.2, "HASTA");
sinyal_dizisi(2) = KisiSinyali(0.4, 0.6, "RİSK GRUBU");
sinyal_dizisi(3) = KisiSinyali(0.2, 0.7, "SAĞLIKLI");
sinyal_dizisi(4) = KisiSinyali(0.9, 0.2, "HASTA");
sinyal_dizisi(5) = KisiSinyali(0.6, 0.5, "RİSK GRUBU");
sinyal_dizisi(6) = KisiSinyali(0.8, 0.3, "HASTA");
sinyal_dizisi(7) = KisiSinyali(0.2, 0.8, "SAĞLIKLI");

sınıflandırılacak_sinyal = KisiSinyali(0.8,0.35);

sinyal_dizisi = KNN_Hesapla(sinyal_dizisi,sınıflandırılacak_sinyal);

sınıflandırılacak_sinyal = KNN_Sınıflandır(sinyal_dizisi,sınıflandırılacak_sinyal,k_degeri);

disp("Kişi sinyalinin durumu "+sınıflandırılacak_sinyal.durum+" olarak sınıflandırıldı");
```

Mesela bu örnek verileri ele alalım. Bizim oluşturduğumuz bu verilerin mantığı şudur. Sinyal nesnelerinin ilk parametresi olan **x_degeri** ne kadar büyük ise kişi hasta olmaya yakındır. Aynı şekilde ikinci parametre olan **y_degeri** ne kadar küçükse , kişi hasta olmaya yine o kadar yakındır. Bunun tersi de yani **x_degeri** ne kadar küçülüp **y_degeri** ne kadar büyürse kişi o kadar sağlıklı olur.

Bu mantık çerçevesinde oluşturduğumuz örnek veriler ile çalıştırılan bu sınıflandırma algoritması aşağıdaki çıktıyı vermiştir.

Şekil 3.5 Projenin Çıktı Görseli

```
HASTA
HASTA
HASTA
Hasta Ağırlıklı Oylama --> 461.5385
Sağlıklı Ağırlıklı Oylama --> 0
Risk Grubunda Ağırlıklı Oylama --> 0
Kişi sinyalinin durumu HASTA olarak sınıflandırıldı
```

3.3.4 EKG Sinyalinin Grafik Olarak Çizdirilmesi

Yazdığımız projenin kodlarını ve işleyişini anlattık. Son olarak da , yapılan aşamalarda da bahsettiğimiz **MIT-BIH Arrhythmia Database**'i nasıl kullandığımızı anlatalım.

MIT-BIH Arrhythmia Database projemizin konusu olan Aritmi hastalığı yaşayan kişilerin EKG sinyallerini bize veren bir veritabanıdır. Fakat bu veritabanını başlangıçta kullanamadık. Çünkü **.dat** uzantılı sinyal dosyalarını MATLAB'da kullanamadık. MATLAB da bu dosya formatını okuyabilmek için ne kadar araştırma yapsak da sağlıklı sonuçlar elde edemedik. Araştırmalarımız sonucu ise , **.dat** uzantılı bu EKG sinyal dosyalarının , MATLAB'ın direk okuyabileceği ve kolaylıkla kullanabileceğimiz **.mat** uzantılı bir formata çevirilebildiğini öğrendik. Sonrasında bunu yaptık ve MATLAB ortamında sadece **load()** fonksiyonunu kullanarak sinyali MATLAB ortamına aktardık.

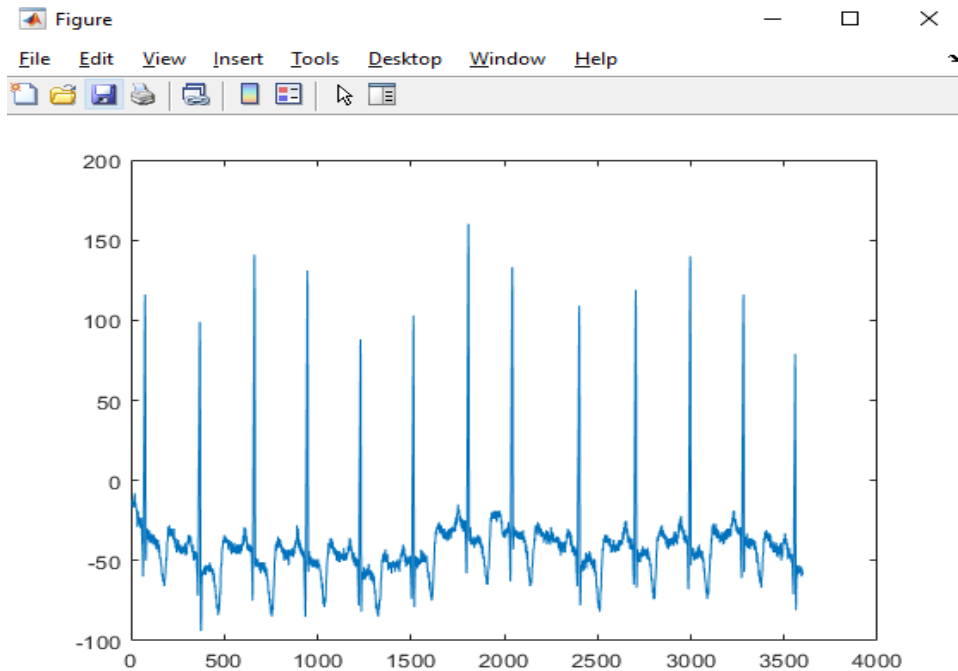
Deneme olarak **MIT-BIH Arrhythmia Database**'den 100. kaydı ele aldık.. **100m.mat** adındaki sinyal dosyamızı indirdik ve MATLAB 'da ki kod dizinimize aktardık. Sadece 2 satırlık bu kodla , 1 boyutlu 3600 değer uzunluğundaki diziden oluşan EKG sinyalini aldık.

Şekil 3.6 EKG Sinyalini Çizdirme Kodu

```
sinyal = load("100m.mat").val  
  
plot(sinyal)
```

Ve sadece **plot()** komutu ile yazdırabildik.

Şekil 3.7 EKG Sinyalinin Çizdirilmiş Hali



3.3.5 EKG Sinyalini Yumuşatma ve Gürültü Giderme işlemi

İleriki aşamalarda sinyalden daha fazla verim alıp, daha kolay işleyebilmemiz için sinyalden gürültünün giderilmesi gerekiyordu. Şu an bir filtre uyguladık. Fakat gürültüyü daha fazla giderebilmek için birkaç filtre daha uygulamamız gerekebilir.

Öncelikle F_s 'i yani sampling rate(örnekleme oranı)'ı bulmamız gerekiyordu. Araştırmalarımıza göre sinyalin peek noktaları bu değeri veriyor. Bu sebeble sinyal dizisinin üzerinden döngü ile geçerek peek sayısını bulduk. Yazılan kod aşağıdaki gibidir

Şekil 3.8 EKG Sinyalinin Peek Noktalarını Bulma Kodu

```
sayac = 0;
%Sinyalin en yüksek peek noktaları
for k = 2 : length(sig)-1
    if((sig(k) > sig(k-1)) && (sig(k) > sig(k+1)) && (sig(k) > 1))
        disp("Sinyalin en yüksek peek noktaları")
        sayac=sayac+1;
        disp(sig(k));
    end
end
%Sinyalin en düşük peek noktaları
for k = 2 : length(sig)-1
    if((sig(k) < sig(k-1)) && (sig(k) < sig(k+1)) && (sig(k) < 0))
        disp("Sinyalin en düşük peek noktaları")
        sayac = sayac+1;
        disp(sig(k));
    end
end
```

Burada sayaç değişkeni peek sayısını veriyor. Bizim sinyalimiz için bu değer ; 515 olarak bulunmuştur. Daha sonra filtreyi sinyalimize uyguladık. Projemizde **Alçak Geçirgen Filtre** olan **Butterworth** filtresini kullandık.Sinyalin filtre uygulanmadan önce ve filtre uygulandıktan sonra olan görünümünü çizdirmek amacıyla subplot komutunu kullandık. Böylece filtrenin etkisini daha net gözlemlemiş olduk. Öncelikle sinyalin orijinal halini çizdirdik. Daha sonra filtre uygulanmış halini çizdirdik. Matlab'in kendi dökümantasyonun da anlattığı üzere parametreleri yerine koyup öncelikle filtreyi tasarladık.

Bir filtrenin sinyale uygulanabilmesi için önce filtrenin oluşturulması gerekir. O yüzden öncelikle filtreyi oluşturduk. Daha sonra filtreyi **filter()** fonksiyonuyla sinyale uyguladık.

Filtre için yazılan kod aşağıdaki gibidir.

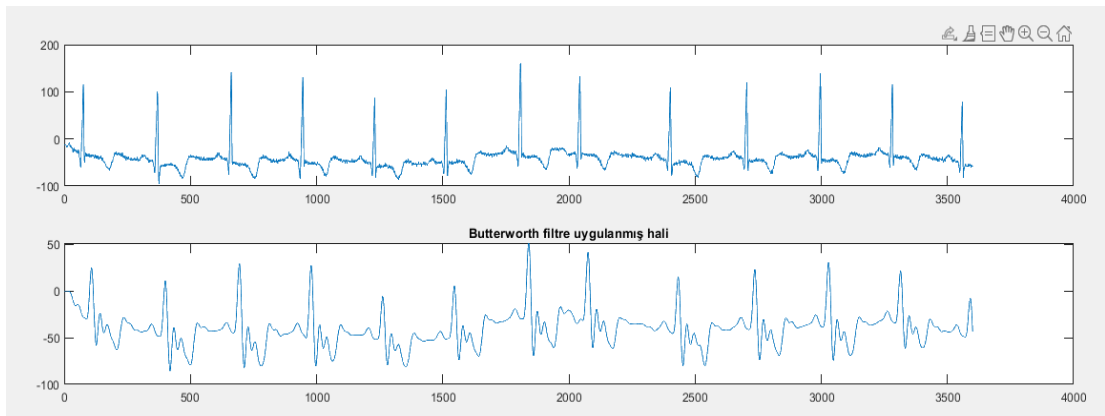
Şekil 3.9 EKG Sinyalini Butterworth Filtresinden Geçirme Kodları

```
%Sinyalin yüklenmesi ve çizdirilmesi
sinyal = load("100m.mat").val
plot(sinyal)
figure(1)
subplot(3,1,1)
plot(sinyal)

%Alçak geçişgen filtrenin tasarımı
fs = 515;
filterOrder = 10; %Filtre dizi sabiti
cutOffFreq = 18;
[b,a] = butter(filterOrder, cutOffFreq/(fs/2),'low'); %Parametreler yerine kondu.
%Filtrenin uygulanması ve çizdirilmesi
ECG_highpass = filter(b,a,sinyal);
figure(1)
subplot(3,1,2)
plot(ECG_highpass)
title('Butterworth filtre uygulanmış hali');
```

Filtreyi uyguladıktan sonra sinyalin orijinal haliyle karşılaştırdık. Karşılaştırma sonucu

Şekil 3.10 EKG Sinyaline Filtre İşlemi Uygulanmış ve Uygulanmamış halinin karşılaştırılması



4. SONUÇ VE KAZANIMLAR

Bitirme Projesinin raporunun bu sonuç kısmına gelindiğinde görülüyor ki biz , projemizi sonuçlandırma ve nihai çıktıyı önceden bahsettiğimiz sebeplerden ötürü yapamadık fakat bunun dışındaki nerdeyse bütün süreçleri tamamlamış bulunuyoruz.

Bir Makine Öğrenmesi algoritması nedir ? Neden bu algoritmaları kullanıyoruz ? Bu algoritmaların sağlık alanı ile ilişkisi nedir ? Bu sorular hakkında bilgimizi bu bitirme projesi kapsamında çok fazla arttırdık. Seçtiğimiz Makine Öğrenmesi Algoritması olan K-NN Algoritmasını derinlemesine öğrendik. K-NN Algoritmasının da yeterli olmadığı durumlar oldu. Biz de araştırmaya devam ederek ve K-NN Algoritması ile beraber çalışan (Ağırlıklı Oylama) yeni yöntemler bulduk.

Bir sinyali elde etsek bile o sinyalin direk kullanılmaması gerektiğini de öğrendik. Bir sinyal , bizim işimize yarayan istediğimiz amaç doğrultusundaki bilgileri bize verebilirken , bizim ihtiyacımız olmayan ve sadece işimizi zorlaştıran sinyal gürültüsü gibi şeyleri de içerebilir. Bu sebepten ötürü EKG sinyali gibi sinyalleri belirli filtrelerden geçirmek ve gürültüyü temizlemek için farklı yöntemler uygulamak gerekir. Biz de bu projemizde bir Alçak Geçirgen Filtre olan Butterworth filtresini kullanarak , bu bahsettiğimiz işlemleri gerçekleştirdik.

Bölümümüzün gereği olan konulara hakim olduğumuz gibi , bölümümüz dışındaki konular hakkında da bilgi sahibi olduk. Örneğin , projemizin konusu gereği EKG sinyalinin yapısını , Aritmi Hastalığını , EKG sinyalindeki hangi özelliklerin Aritmi Hastalığını teşhis etmede kullanıldığı gibi bilgileri de öğrenmiş olduk. Makine Öğrenmes , MATLAB ve sinyali filtreleme gibi bizim bölümümüz gereği teknik konular haricinde böyle tıbbi bilgiler de öğrenmemiz bizim kazanımlarımızdan biri oldu diyebiliriz.

Özetlemek gerekirse bütün sene boyunca uğraştığımız bu bitirme projesi , içeriği hakkında önceden çok fazla bilgimiz olmamasına rağmen araştırmalar yapıp bahsettiğimiz bütün bu bilgileri öğrenmek , mezun olma aşamasında bize çok değerli bilgi birikim ve tecrübeler kazandırdı. Farklı disiplinleri tanıma ve farklı bakış açıları ile bakma şansını elde ettik.

Tüm bu güzel ve verimli kazanımlar için başta sayın danışman hocamız Doç.Dr. BİLAL ŞENOL'a ve ekip arkadaşlarıma teşekkürlerimi sunuyorum.

5. KAYNAKLAR

- [1] SET Turan, “Birinci Basamak İçin Temel EKG Okuma Becerisi” , Trabzon
- [2] MOODY George, MARK Roger, <https://www.physionet.org/content/mitdb/1.0.0/>, 2015
- [3] MOODY George, Records In The MIT-BIH Arrhythmia Database ,2010
KAYA Yasin, PEHLİVAN Hüseyin , “KNN, NN, BAYES, DT ve SVM kullanılarak EKG vurularının sınıflandırılması”, Research Gate , 2014
- [4] TAŞCI Erdal, ONAN Aytuğ , “K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi”, İzmir
- [5] S. Jayalalitha, D. Susan, Shalini Kumari and B. Archana, 2014. K-nearest Neighbour Method of Analysing the ECG Signal (To Find out the Different Disorders Related to Heart). *Journal of Applied Sciences*, 14: 1628-1632.
- [6] H. Kodal Sevindir , S. Çetinkaya ve C. Yazıcı , "Makine öğrenmesi algoritmaları ve dalgacık dönüşümü ile EKG sinyalinin özellik çıkarımı", *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, c. 20, sayı. 1, ss. 94-109, Nis. 2018, doi:10.25092/baunfbed.413705
- [7] TEPE Cengiz, SEZGİN Hatice, “EKG Sinyallerinde Gürültü Gidermede Ayırık Dalgacık Dönüşümünde Farklı Ana Dalgacıkların Ve Ayırıştırma Seviyelerinin Karşılaştırılması” , Samsun
- [8] E. Erçelebi, ``Electrocardiogram signals de-noising using lifting-based discrete wavelet transform", *Computers in Biology and Medicine* 34(2004) 479-493.
- [9] LStephan Mallat, A Wavelet Tour of Signal Processing, Acedemic Press, London, 1999.
- [10] Yakut Ö., Solak S. ve Bolat E.D., “EKG işaretindeki gürültülerin temizlenmesi için IIR tabanlı sayısal filtre tasarımı”, *Politeknik Dergisi*, 21(1): 173-181, (2018).