



T.C.

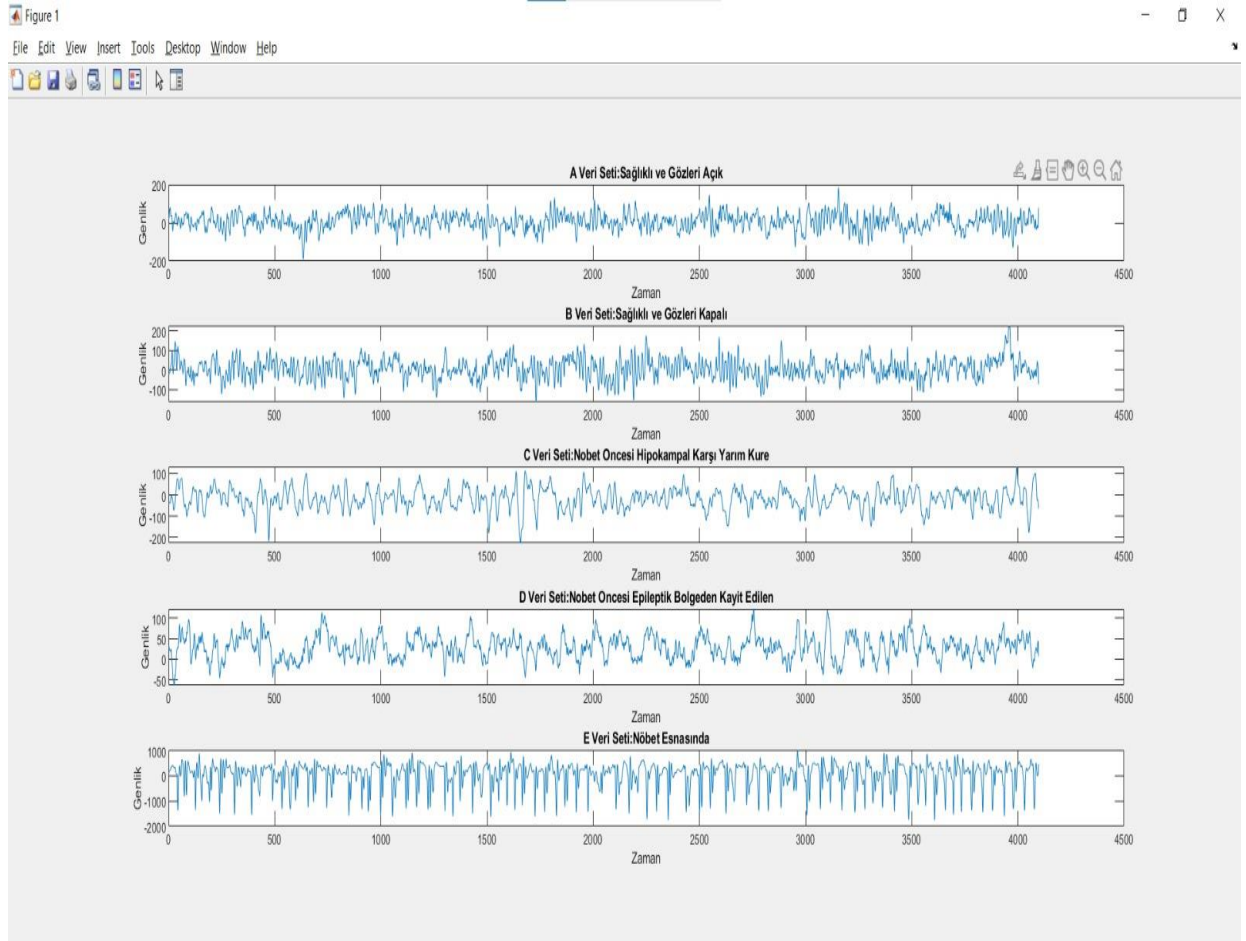
İNÖNÜ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BİTİRME PROJESİ ANALİZ RAPORU

Ders Sorumlusu	Dr. Öğr. Üyesi Bilal ŞENOL
Öğrenciler	Tuğba GÜNAÇGÜN, Doğuş İPEKSAÇ, Bünyamin ERTAŞ
Proje İsmi	EEG Sinyallerini Sınıflandırarak Epilepsi Hastalıklarını Teşhis Etmek
Aşama Tanımı	Proje Süresince Yapılanlar

Projemizin ilk aşamasında veri seti bulduk. Bu çalışmada kullanılan EEG verileri Bonn Üniversitesi veri tabanından alınmıştır. Veri seti A,B,C,D,E olmak üzere beş kümeden oluşmaktadır. A veri seti sağlıklı, gözler açık ve B veri seti sağlıklı, gözler kapalı C veri seti nöbet öncesi hipokampal karşı yarım küre ve D veri seti nöbet öncesi epileptik bölgeden kayıt edilen EEG sinyalleridir. E veri seti EEG sinyali ise nöbet esnasında ve elektrotlar epileptik bölgede konumlandırılarak kayıt edilmiştir. Her bir küme tek kanallı 100 bölütten oluşmaktadır. Aşağıda her bir veri setinden bir kişinin EEG sinyali verilmiştir.

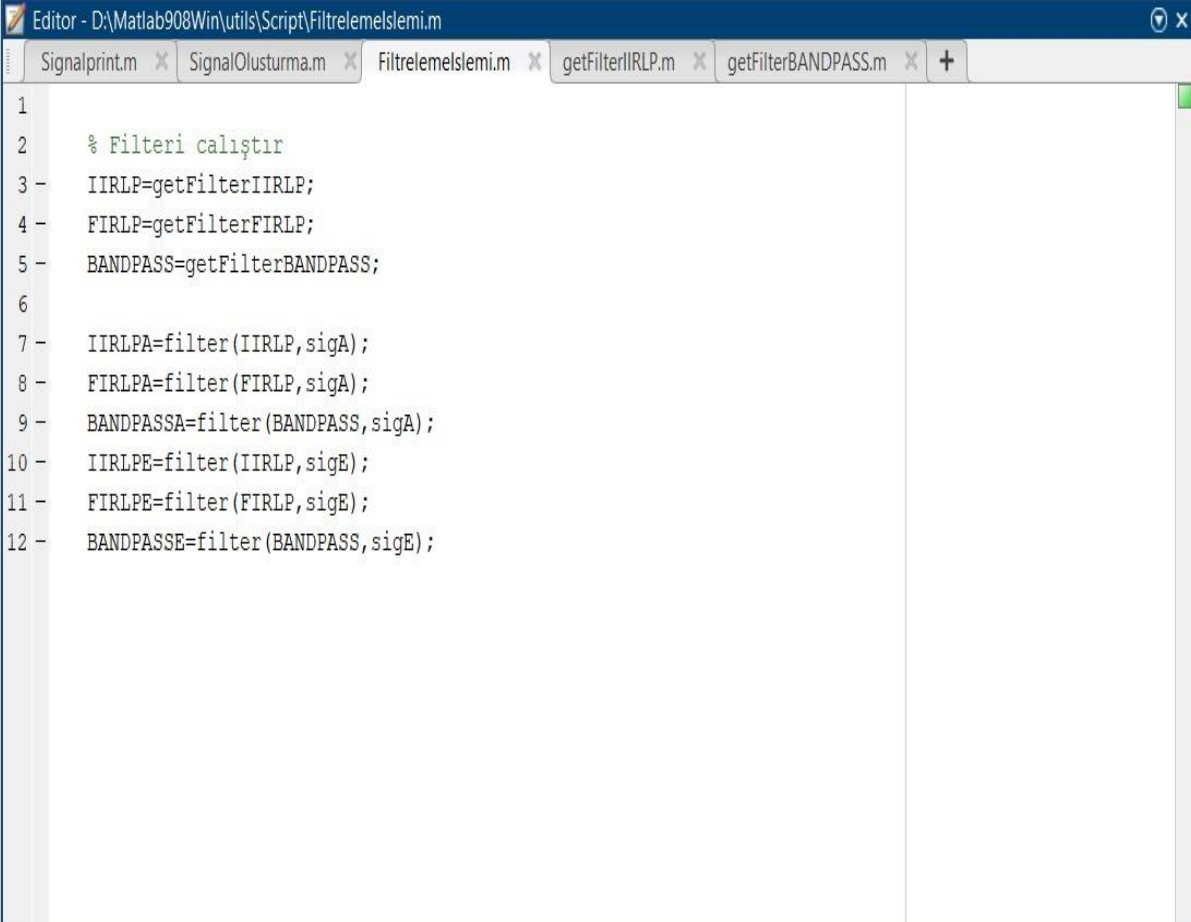


Sonrasında filtreleme işlemini gerçekleştirdik. Elimizdeki EEG veri seti tek kanallı bir sinyal olduğundan dolayı iyi bir filtreleme işlemi yapmamız gerekiyordu. Üç çeşit filtre ile filtreledik. Kullanılan EEG sinyallerini 0.53-40Hz arasında sırayla IIR lowpass ve FIR lowpass ile filtreledik.

Araştırmalarımız sonucunda BANDPASS için EEG sinyallerinin en ideal değerleri $F_{stop1} = 1$, $F_{pass1} = 3$, $F_{pass2} = 40$, $F_{stop2} = 85$ olduğunu bulduk ve bu değerler ile filtreledik. IIR ve FIR ile filtrelenen EEG sinyallerinin örnekleme hızı 173.61 Hz'dir. BANDPASS ile filtrelenen EEG sinyalinin örnekleme hızı 178.61 Hz'dir.

Aşağıda filtreleme işlemlerinin kodları ve ekran çıktıları verilmiştir.

Filtreleme İşlemi

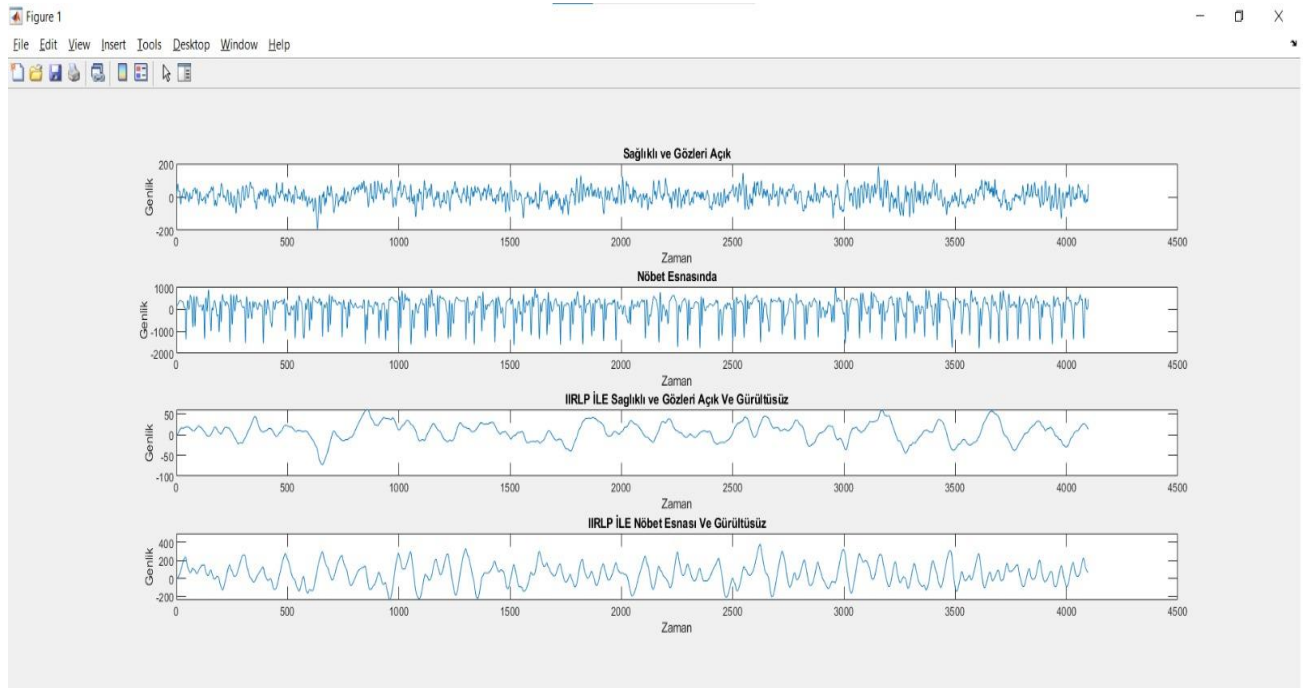


```
1  
2 % Filtreleri çalıştır  
3 IIRLP=getFilterIIRLP;  
4 FIRLP=getFilterFIRLP;  
5 BANDPASS=getFilterBANDPASS;  
6  
7 IIRLPA=filter(IIRLP,sigA);  
8 FIRLPA=filter(FIRLP,sigA);  
9 BANDPASSA=filter(BANDPASS,sigA);  
10 IIRLPE=filter(IIRLP,sigE);  
11 FIRLPE=filter(FIRLP,sigE);  
12 BANDPASSE=filter(BANDPASS,sigE);
```

IIR FİLTRE

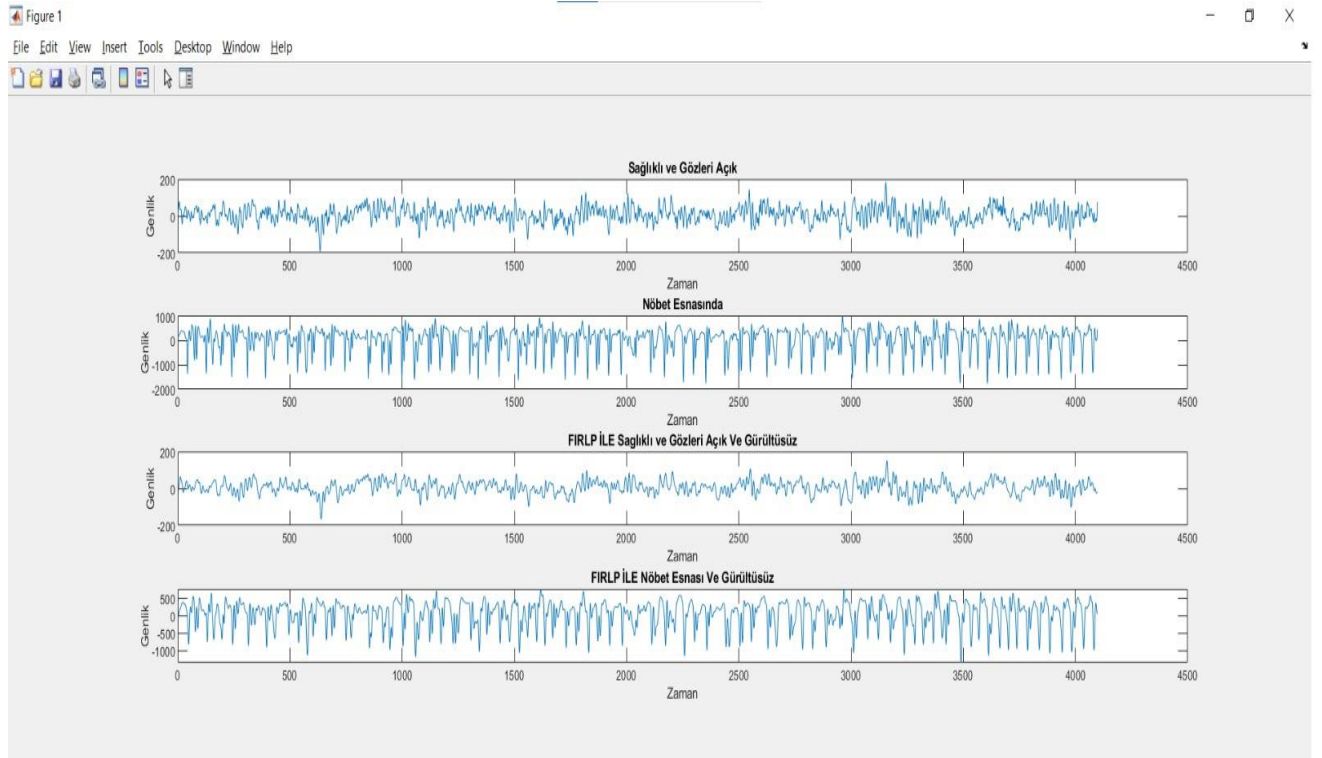
```
Editor - D:\Matlab908Win\utils\Script\getFilterIIRLP.m
Signalprint.m x SignalOlusturma.m x FiltrelemeIslemi.m x getFilterIIRLP.m x +

1
2 function Hd = getFilterIIRLP
3 %GETFILTER Returns a discrete-time filter object.
4
5 % MATLAB Code
6 % Generated by MATLAB(R) 9.8 and DSP System Toolbox 9.10.
7 % Generated on: 31-Dec-2020 17:08:33
8
9 Fpass = 0.53; % Passband Frequency
10 Fstop = 40; % Stopband Frequency
11 Apass = 1; % Passband Ripple (dB)
12 Astop = 60; % Stopband Attenuation (dB)
13 Fs = 173.61; % Sampling Frequency
14
15 h = fdesign.lowpass('fp,fst,ap,ast', Fpass, Fstop, Apass, Astop, Fs);
16
17 Hd = design(h, 'butter', ...
18 'MatchExactly', 'stopband', ...
19 'SOSScaleNorm', 'Linf');
20
21
```



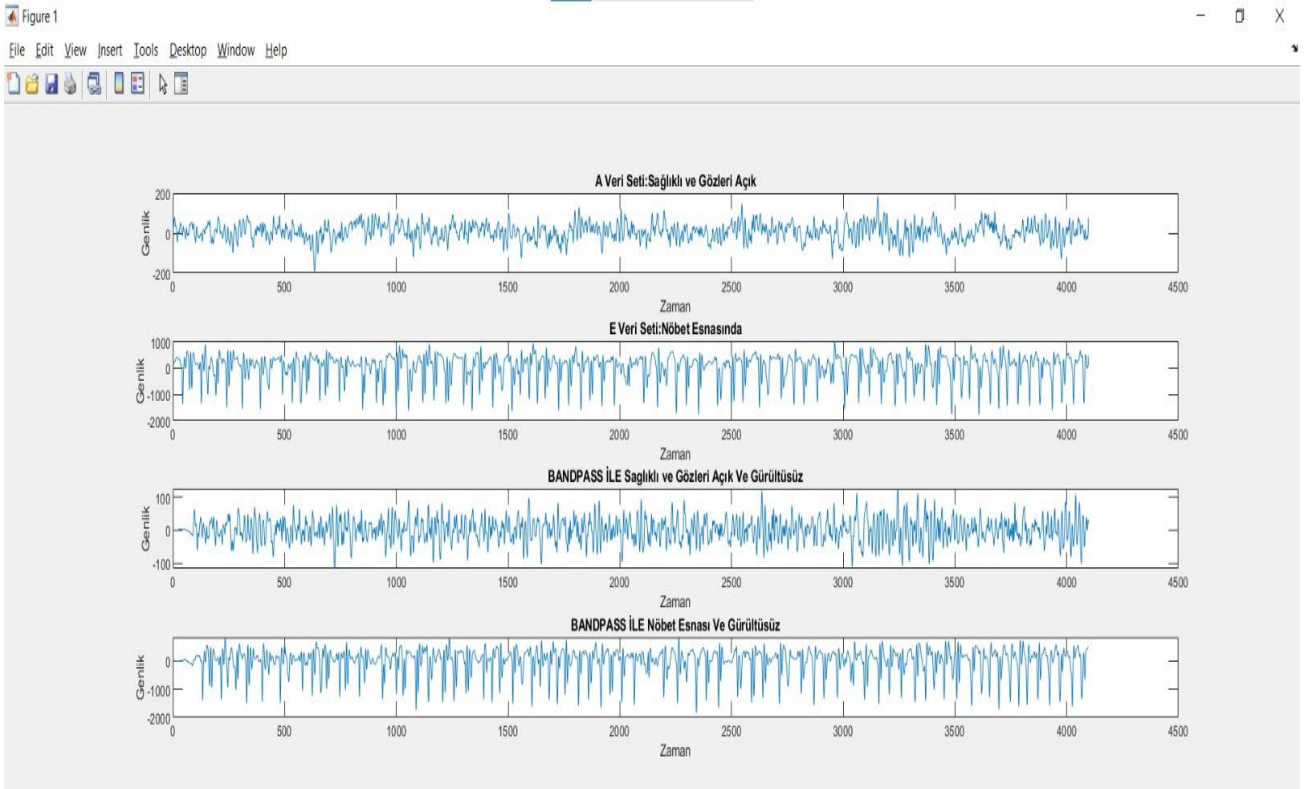
FIR FİLTRE

```
Editor - D:\Matlab908Win\utils\Script\getFilterFIRLP.m*
Signalprint.m x SignalOlusturma.m x FiltrelemeIslemi.m x getFilterIIRLP.m x getFilterBANDPASS.m x getFilterFIRLP.m* x +
1 function Hd = getFilterFIRLP
2 %GETFILTER Returns a discrete-time filter object.
3
4 % MATLAB Code
5 % Generated by MATLAB(R) 9.8 and DSP System Toolbox 9.10.
6 % Generated on: 01-Jan-2021 19:12:45
7
8 Fpass = 0.53; % Passband Frequency
9 Fstop = 40; % Stopband Frequency
10 Apass = 1; % Passband Ripple (dB)
11 Astop = 60; % Stopband Attenuation (dB)
12 Fs = 173.61; % Sampling Frequency
13
14 h = fdesign.lowpass('fp,fst,ap,ast', Fpass, Fstop, Apass, Astop, Fs);
15
16 Hd = design(h, 'equiripple', ...
17 'MinOrder', 'any', ...
18 'StopbandShape', 'flat');
19
20
21
```



BANDPASS FİLTRE

```
Editor - D:\Matlab908Win\utils\Script\getFilterBANDPASS.m
Signalprint.m x SignalOlusturma.m x FiltrelemeIslemi.m x getFilterIIRLP.m x getFilterBANDPASS.m x getFilterFIRLP.m x +
1 function Hd = getFilterBANDPASS
2 %GETFILTER Returns a discrete-time filter object.
3
4 % MATLAB Code
5 % Generated by MATLAB(R) 9.8 and DSP System Toolbox 9.10.
6 % Generated on: 02-Jan-2021 20:40:33
7
8 Fstop1 = 1; % First Stopband Frequency
9 Fpass1 = 3; % First Passband Frequency
10 Fpass2 = 40; % Second Passband Frequency
11 Fstop2 = 85; % Second Stopband Frequency
12 Astop1 = 60; % First Stopband Attenuation (dB)
13 Apass = 1; % Passband Ripple (dB)
14 Astop2 = 60; % Second Stopband Attenuation (dB)
15 Fs = 178.61; % Sampling Frequency
16
17 h = fdesign.bandpass('fst1,fp1,fp2,fst2,ast1,ap,ast2', Fstop1, Fpass1, ...
18 Fpass2, Fstop2, Astop1, Apass, Astop2, Fs);
19
20 Hd = design(h, 'equiripple', ...
21 'MinOrder', 'any');
```



ÇIKARIMLARIMIZ

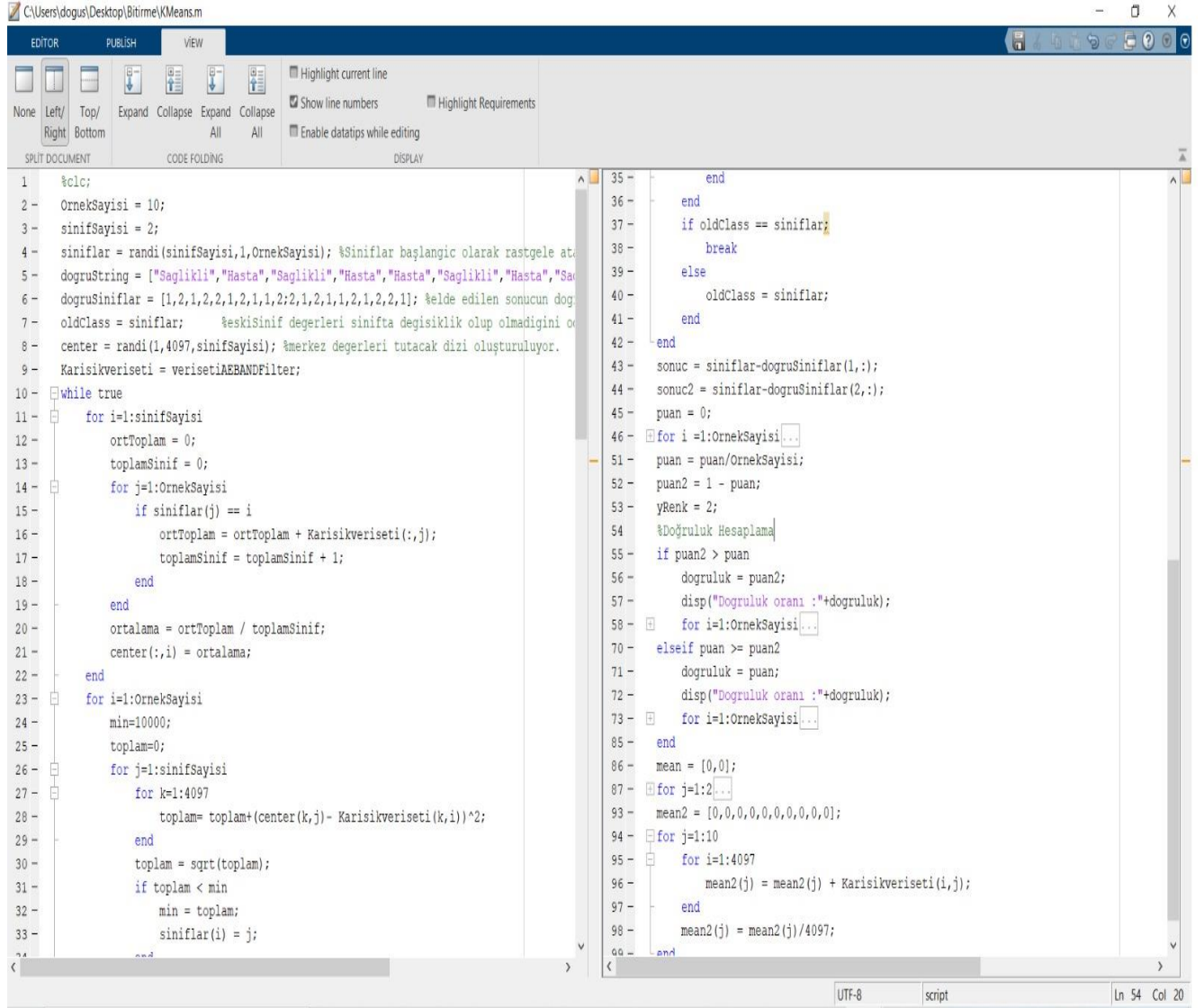
Biz projemizin ilk aşamasında veri setini elde ettik. İkinci aşamada kullandığımız veri seti için en uygun filtreyi bulmak istedik. Projemizin üçüncü aşamasında dalgacık dönüşümü algoritması ile bu filtreleme algoritmalarını kullanarak öznelik çıkartmamız gerekiyordu. Bu aşamayı yapabilmek için filtreleme algoritmalarını öğrendik ve projemizin ilk iki adımını yaptıktan sonra dalgacık dönüşümünü diğer döneme bıraktık ve bu filtrelediğimiz sinyalleri K-Means algoritması ile sınıflayarak doğruluk değerlerini elde ettik. Burada gerçekleştirdiğimiz aşamaları sarı renkle gösterdik.



K-MEANS

K-Means kümeleme yöntemi kümeleme problemlerinde kullanılan en basit gözetimsiz öğrenme algoritmalarından biridir. Bu yöntem k adet küme için orta nokta belirleyip n adet gözlemi bu k adet kümeye dağıtmaya çalışan bir yöntemdir. Kümeleme işlemi sonunda amaç küme içi benzerliğin en yüksek kümeler arası benzerliğin ise en düşük olmasını sağlamaktadır.

Biz burada yukarıda filtrelediğimiz sinyalleri K-Means ile değerlendirdik ve K-Means kodunu kendimiz yazdık.



```
1 %clc;
2 OrnekSayisi = 10;
3 sinifSayisi = 2;
4 siniflar = randi(sinifSayisi,1,OrnekSayisi); %Siniflar başlangic olarak rastgele at
5 dogruString = ["Saglikli","Hasta","Saglikli","Hasta","Hasta","Saglikli","Hasta","Sag
6 dogruSiniflar = [1,2,1,2,2,1,2,1,1,2,2,1]; %elde edilen sonucun dog
7 oldClass = siniflar; %eskiSinif degerleri sinifta degisiklik olup olmadigini o
8 center = randi(1,4097,sinifSayisi); %merkez degerleri tutacak dizi olusturuluyor.
9 Karisikveriseti = verisetiAEBANDFilter;
10 while true
11     for i=1:sinifSayisi
12         ortToplam = 0;
13         toplamSinif = 0;
14         for j=1:OrnekSayisi
15             if siniflar(j) == i
16                 ortToplam = ortToplam + Karisikveriseti(:,j);
17                 toplamSinif = toplamSinif + 1;
18             end
19         end
20         ortalama = ortToplam / toplamSinif;
21         center(:,i) = ortalama;
22     end
23     for i=1:OrnekSayisi
24         min=10000;
25         toplam=0;
26         for j=1:sinifSayisi
27             for k=1:4097
28                 toplam= toplam+(center(k,j)- Karisikveriseti(k,i))^2;
29             end
30             toplam = sqrt(toplam);
31             if toplam < min
32                 min = toplam;
33                 siniflar(i) = j;
34         end
35     end
36     end
37     if oldClass == siniflar;
38         break
39     else
40         oldClass = siniflar;
41     end
42 end
43 sonuc = siniflar-dogruSiniflar(1,:);
44 sonuc2 = siniflar-dogruSiniflar(2,:);
45 puan = 0;
46 for i =1:OrnekSayisi
47     puan = puan/OrnekSayisi;
48     puan2 = 1 - puan;
49     yBerk = 2;
50     %Dogruluk Hesaplama
51     if puan2 > puan
52         dogruluk = puan2;
53         disp("Dogruluk oranı :"+dogruluk);
54         for i=1:OrnekSayisi
55             elseif puan >= puan2
56                 dogruluk = puan;
57                 disp("Dogruluk oranı :"+dogruluk);
58                 for i=1:OrnekSayisi
59                     end
60                 mean = [0,0];
61                 for j=1:2
62                     mean2 = [0,0,0,0,0,0,0,0,0,0];
63                     for j=1:10
64                         for i=1:4097
65                             mean2(j) = mean2(j) + Karisikveriseti(i,j);
66                         end
67                         mean2(j) = mean2(j)/4097;
68                     end
69                 end
70                 end
71                 end
72                 end
73                 end
74                 end
75                 end
76                 end
77                 end
78                 end
79                 end
80                 end
81                 end
82                 end
83                 end
84                 end
85                 end
86                 end
87                 end
88                 end
89                 end
90                 end
91                 end
92                 end
93                 end
94                 end
95                 end
96                 end
97                 end
98                 end
99                 end
100                end
```


Burada BANDPASS, FIRLP ve IIRLP filtreleriyle filtrelediğimiz verilerin ve orijinal halinin K-Means algoritması ile 4 kez ayrı ayrı sınıflandırmalar yapıldıktan sonraki doğruluk oranı ve hatayı bulduğumuz kısımlar gösterilmiştir.

BANDPASS FİLTRE

The screenshot displays the MATLAB R2020a environment with the K-Means clustering algorithm applied to data filtered by a BANDPASS filter. The main editor window shows the script 'KMeans.m' with the following code:

```
1 %clc;
2 OrnekSayisi = 10;
3 sinifSayisi = 2;
4 siniflar = randi(sinifSayisi,1,OrnekSayisi); %siniflar başlangic olarak rastgele ataniyor.
5 dogruString = ["Saglikli","Hasta","Saglikli","Hasta","Hasta","Saglikli","Hasta","Saglikli","Saglikli","Hasta"];
6 dogruSiniflar = [1,2,1,2,2,1,2,1,2,2]; %elde edilen sonucun dogruluğu hesaplanıyor.
7 oldClass = siniflar; %teskisinif degerleri sinifta degisiklik olup olmadigini ogrenmek icin kullaniliyor.
8 %siniflar = randi(2,1000,OrnekSayisi); %siniflar degerleri tutacak dizi olusturuluyor.
9 Karisikveriseti = verisetiAEBANDFilter;
10 %siniflar = siniflar;
11 for i=1:sinifSayisi
12     ortalama = 0;
13     toplamSinif = 0;
14     for j=1:OrnekSayisi
15         if siniflar(j) == i
16             ortalama = ortalama + Karisikveriseti(:,j);
17             toplamSinif = toplamSinif + 1;
18         end
19     end
```

The Command Window shows the execution of the K-Means algorithm, displaying the following output:

```
>> KMeans
Dogruluk oranı :1
Saglikli Hasta Saglikli Hasta Hasta Saglikli Hasta Saglikli Saglikli Hasta >> KMeans
Dogruluk oranı :0.7
Saglikli Saglikli Saglikli Hasta Saglikli Saglikli Saglikli Saglikli Saglikli Hasta >>
Dogruluk oranı :0.7
Saglikli Saglikli Saglikli Hasta Saglikli Saglikli Hasta Saglikli Saglikli Saglikli >>
Dogruluk oranı :0.9
Saglikli Hasta Saglikli Hasta Saglikli Saglikli Hasta Saglikli Saglikli Hasta >>
```

The Workspace window shows the following variables and their values:

Name	Value
BANDPASS	1x1 dfir
center	4097x2 double
dogruluk	0.9000
dogrulukTekrarla...	10
dogruSiniflar	2x10 double
dogruString	1x10 string
FIRLP	1x1 dfir
i	4097
IIRLP	1x1 dfir
j	10
k	4097
Karisikveriseti	4097x10 double
KarisikverisetiAE	4097x10 double
mean	[-0.0181,-0.0202]
mean2	[0.0586,-0.5910,0.0662,-0...
min	10000
oldClass	[1,2,1,2,1,2,1,2]
OrnekSayisi	10
orta	7.6400
ortalama	4097x1 double
ortalamaSonuc	[0.7290,0.7620,0.7600,0.76...
ortToplam	4097x1 double
p	100
puan	0.9000
puan2	0.1000
renkler	'rbgc'
siniflamaTekrarla...	100
siniflar	[1,2,1,2,1,2,1,2]
sinifSayisi	2
sonuc	[0.0,0.0,1.0,0.0,0.0]
sonuc2	[-1,-1,1,0,-1,-1,-1,1]
titles	1x4 string
toplam	2.4545e+04
toplam1	77.7000
toplamSinif	4

FIRLP FİLTRE

The MATLAB R2020a interface shows the following code in the Editor window:

```
1 %clc;
2 OrnekSayisi = 10;
3 sinifSayisi = 2;
4 siniflar = randi(sinifSayisi,1,OrnekSayisi); %Siniflar baglangic olarak rastgele atiliyor.
5 dogruString = ["Saglikli","Hasta","Saglikli","Hasta","Hasta","Saglikli","Hasta","Saglikli","Saglikli","Saglikli"];
6 dogruSiniflar = [1,2,1,2,2,1,2,1,2,2,1]; %elde edilen sonucun dogruluğu hesaplanıyor.
7 oldClass = siniflar; %eskiSinif degerleri sinifta degisiklik olup olmadigini ogrenmek icin
8 center = randi(1,4097,sinifSayisi); %merkez degerleri tutacak dizi olusturuluyor.
9 Karisikveriseti = verisetiAEFIRFilter;
10 while true
11     for i=1:sinifSayisi
12         ortToplam = 0;
13         toplamSinif = 0;
14         for j=1:OrnekSayisi
15             if siniflar(j) == i
16                 ortToplam = ortToplam + Karisikveriseti(:,j);
17                 toplamSinif = toplamSinif + 1;
18             end
19         end
20     end
```

The Command Window shows the results of the KMeans clustering:

```
>> KMeans
Dogruluk oranı :0.9
Saglikli Hasta Saglikli Saglikli Hasta Saglikli Hasta Saglikli Saglikli Hasta >> KMeans
Dogruluk oranı :0.7
Saglikli Hasta Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Hasta >> KMeans
Dogruluk oranı :0.9
Saglikli Saglikli Saglikli Hasta Hasta Saglikli Hasta Saglikli Saglikli Hasta >> KMeans
Dogruluk oranı :0.8
Saglikli Saglikli Saglikli Saglikli Hasta Hasta Saglikli Hasta Saglikli Saglikli Saglikli >> KMeans
Saglikli Hasta Saglikli Saglikli Hasta Saglikli Hasta Saglikli Saglikli Saglikli >>
```

The Workspace window shows the following variables:

Name	Value
i	4097
iIRLP	1x1 d2xos
j	10
k	4097
Karisikveriseti	4097x10 double
KarisikverisetiAE	4097x10 double
mean	[7.8375,10.6350]
mean2	[54.1901,-7.2037,54.5460,-...
min	10000
oldClass	[2,1,2,2,1,2,2,2]
OrnekSayisi	10
orta	7.6400
ortalama	4097x1 double
ortalamaSonuc	[0.7290,0.7620,0.7600,0.76...
ortToplam	4097x1 double
p	100
puan	0.2000
puan2	0.8000
renkler	ybgc'
siniflamaTekrarla...	100
siniflar	[2,1,2,2,1,2,2,2]
sinifSayisi	2
sonuc	[1,-1,1,0,-1,1,-1,1,0]
sonuc2	[0.0,0.0,0.0,0.1]
titles	1x4 string
toplam	2.1528e+04
toplam1	77.7000
toplamSinif	7
veri	4
veriler	4097x40 double
verisetiAEBANDF...	4097x10 double
verisetiAEFIRFilter	4097x10 double
verisetiAEIRFilter	4097x10 double
y	10
yRenk	2

IIRLP FİLTRE

The MATLAB R2020a interface shows the following code in the Editor window:

```
1 %clc;
2 OrnekSayisi = 10;
3 sinifSayisi = 2;
4 siniflar = randi(sinifSayisi,1,OrnekSayisi); %Siniflar baglangic olarak rastgele atiliyor.
5 dogruString = ["Saglikli","Hasta","Saglikli","Hasta","Hasta","Saglikli","Hasta","Saglikli","Saglikli","Saglikli"];
6 dogruSiniflar = [1,2,1,2,2,1,2,1,2,2,1]; %elde edilen sonucun dogruluğu hesaplanıyor.
7 oldClass = siniflar; %eskiSinif degerleri sinifta degisiklik olup olmadigini ogrenmek icin
8 center = randi(1,4097,sinifSayisi); %merkez degerleri tutacak dizi olusturuluyor.
9 Karisikveriseti = verisetiAEFIRFilter;
10 while true
11     for i=1:sinifSayisi
12         ortToplam = 0;
13         toplamSinif = 0;
14         for j=1:OrnekSayisi
15             if siniflar(j) == i
16                 ortToplam = ortToplam + Karisikveriseti(:,j);
17                 toplamSinif = toplamSinif + 1;
18             end
19         end
20     end
```

The Command Window shows the results of the KMeans clustering:

```
>> KMeans
Dogruluk oranı :0.6
Saglikli Hasta Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli >> KMeans
Dogruluk oranı :0.6
Saglikli Hasta Saglikli Hasta Hasta Saglikli Hasta Hasta Hasta >> KMeans
Dogruluk oranı :0.7
Saglikli Hasta Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli >> KMeans
Dogruluk oranı :0.6
Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli >> KMeans
Saglikli Hasta Saglikli Hasta Hasta Saglikli Hasta Hasta Hasta >>
```

The Workspace window shows the following variables:

Name	Value
i	4097
iIRLP	1x1 d2xos
j	10
k	4097
Karisikveriseti	4097x10 double
KarisikverisetiAE	4097x10 double
mean	[-8.6248,53.0768]
mean2	[56.0730,-8.2317,56.3018,-...
min	8.8625e+03
oldClass	[2,1,2,1,1,2,1,1,1]
OrnekSayisi	10
orta	7.6400
ortalama	4097x1 double
ortalamaSonuc	[0.7290,0.7620,0.7600,0.76...
ortToplam	4097x1 double
p	100
puan	0.4000
puan2	0.6000
renkler	ybgc'
siniflamaTekrarla...	100
siniflar	[2,1,2,1,1,2,1,1,1]
sinifSayisi	2
sonuc	[1,-1,-1,-1,1,0,0,0,-1]
sonuc2	[0.0,0.0,-1,1,-1,-1,0]
titles	1x4 string
toplam	1.0216e+04
toplam1	77.7000
toplamSinif	3
veri	4
veriler	4097x40 double
verisetiAEBANDF...	4097x10 double
verisetiAEFIRFilter	4097x10 double
verisetiAEIRFilter	4097x10 double
y	10
yRenk	2

FILTRESİZ

MATLAB R2020a

HOME PLOTS APPS EDITOR PUBLISH VIEW

File Edit Breakpoints Run Run and Advance Run and Time

Current Folder: C:\Users\dogus\Desktop\Bitirme

Editor: C:\Users\dogus\Desktop\Bitirme\KMeans.m

```
1 %clc;
2 OrnekSayisi = 10;
3 sinifSayisi = 2;
4 siniflar = randi(sinifSayisi,1,OrnekSayisi); %siniflar baglangic olarak rastgele ataniyor.
5 dogruString = ["Saglikli","Hasta","Saglikli","Hasta","Hasta","Saglikli","Hasta","Saglikli","Saglikli","Hasta"];
6 dogruSiniflar = [1,2,1,2,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1]; %elde edilen sonucun dogruluğu hesaplanıyor.
7 oldClass = siniflar; %eskiSinif degerleri sinifta degisiklik olup olmadigini ogrenmek icin
8 center = randi(1,4097,sinifSayisi); %merkez degerleri tutacak dizi olusturuluyor.
9 Karisikveriseti = KarisikverisetiAE;
10 while true
11     for i=1:sinifSayisi
12         ortToplam = 0;
13         toplamSinif = 0;
14         for j=1:OrnekSayisi
15             if siniflar(j) == i
16                 ortToplam = ortToplam + Karisikveriseti(:,j);
17                 toplamSinif = toplamSinif + 1;
18             end
19         end
20     end
21 end
```

Workspace:

Name	Value
BANDPASS	1x1 dfir
center	4097x2 double
dogruluk	0.8000
dogrulukTekrarla..	10
dogruSiniflar	2x10 double
dogruString	1x10 string
FIRLP	1x1 dfir
i	4097
IIRLP	1x1 df2sos
j	10
k	4097
Karisikveriseti	4097x10 double
KarisikverisetiAE	4097x10 double
mean	[11.0925,7.9899]
mean2	[56.3029,-8.1225,56.7186,...]
min	10000
oldClass	[1,2,1,2,1,2,1,1,1]
OrnekSayisi	10
orta	7.6400
ortalama	4097x1 double
ortalamaSonuc	[0.7290,0.7620,0.7600,0.76...
ortToplam	4097x1 double
p	100
puan	0.8000
puan2	0.2000
renkler	'ybgc'
siniflamaTekrarla..	100
siniflar	[1,2,1,2,1,2,1,1,1]
sinifSayisi	2
sonuc	[0,0,0,-1,0,0,0,0,-1]
sonuc2	[-1,1,-1,0,1,-1,1,-1,-1,0]
titles	1x4 string
toplam	3.5648e+04
toplam1	77.7000
toplamSinif	3

Command Window:

```
>> KMeans
Dogruluk oranı :0.5
Hasta Saglikli Hasta Saglikli Hasta Saglikli Hasta Saglikli Saglikli Saglikli >> KMeans
Dogruluk oranı :0.8
Saglikli Hasta Saglikli Hasta Saglikli Saglikli Saglikli Saglikli Saglikli Hasta >> KMeans
Dogruluk oranı :0.6
Saglikli Saglikli Saglikli Saglikli Saglikli Saglikli Hasta Saglikli Saglikli Saglikli
Dogruluk oranı :0.8
Saglikli Hasta Saglikli Saglikli Hasta Saglikli Hasta Saglikli Saglikli Saglikli >>
```

UTF-8 script Ln 12 Col 23

Burada filtresiz ve filtrelenmiş verilerin K-Means ile sınıflandırıldıktan sonra ortalama doğruluk değerleri aşağıda verilmiştir. Bu verilerimiz rastgele olarak A(sağlıklı ve gözleri açık) ve E(nöbet esnasında) veri setlerinden 5'er kişi alınarak oluşturulmuştur.

