

به نام خدا



تمرین سری سوم درس BCI

محمد مهدی حبیبی

شماره دانشجویی: 401617033

استاد: دکتر شالچیان

پاییز- زمستان 1402

## فهرست مطالب

سوال 1	3
سوال 2	6
استخراج ویژگی با استفاده از واریانس	6
استفاده از channel selection	12
سوال سوم	13
جواب سوال سوم	17
استفاده از مدل یادگیری ماشین SVM	17
ارزیابی مدل یادگیری ماشین SVM	17
استفاده از مدل یادگیری ماشین KNN	18
ارزیابی مدل یادگیری ماشین KNN	19
سوال چهارم	20
سوال پنجم	21
سوال ششم	25



## تمرین ۳

- در این تمرین مراحل پیش پردازش، استخراج ویژگی و طبقه بندی سیگنال‌ها مورد ارزیابی قرار می‌گیرد.
- لطفاً با توجه به توضیحات زیر، فایل اسکریپت نوشته شده (m.) و گزارش (pdf) خود را در زمان تعیین شده در سامانه LMS ثبت بفرمایید.
- گزارش تمرین باید شامل تصاویر خروجی و توضیحات مربوط به آن و توضیحات مربوط به اسکریپت و فرآیند طی شده در برنامه باشد. (نیازی به بارگذاری دیتاست نیست در فایل زیپ گزارش نیست)
- تمام شبیه‌سازی‌های زیر را بر روی دیتاست تصور حرکت BCICIV\_calib\_ds1b\_100Hz انجام دهید.
- این مجموعه داده حاوی سیگنال‌های EEG برای تصور حرکت ۲ کلاسه است.

۱- در مرحله پیش پردازش، با تحلیل فرکانسی، در صورت لزوم از فیلترهای ناچ، میان‌گذر و ... استفاده کنید. آیا نیازی به استاندارد (نرمال) نمودن داده‌ها وجود دارد؟ در این صورت، داده‌ها را استاندارد کنید.

۲- در مرحله استخراج ویژگی، از ویژگی‌های مختلف آماری (واریانس، کورتوسیس)، توان باندهای مختلف (دلتا، تتا، آلفا و بتا) استفاده کنید. در این مرحله ویژگی‌های دو باند آلفا، دلتا و بتا، تتا را با استفاده از اسکتر پلات (برای دو جفت باند) رسم کرده و فضای ویژگی رسم شده را تحلیل کنید. آیا در این فضا اختلافی میان ویژگی‌ها قابل رویت است؟

۳- با استفاده از ویژگی‌های استخراج شده در مرحله قبل، مسئله طبقه بندی دو کلاسه را برای انواع ویژگی‌ها (ویژگی‌های آماری و ویژگی‌های فرکانسی) حل نمایید. در این مرحله به دلخواه می‌توانید ویژگی‌ها را با یکدیگر ترکیب کنید. راهنما: در این مرحله از دو کلاسیفایر SVM و KNN استفاده کنید و نتایج را باهم مقایسه کنید.

۴- با تحلیل نتایج فوق، مناسب‌ترین نوع ویژگی برای طبقه بندی این مسئله را تعیین نمایید.

۵- با تحلیل نتایج فوق، مناسب‌ترین نوع کلاسیفایر را برای طبقه بندی این مسئله را تعیین نمایید.

۶- از روش ICA استفاده کنید و آرتیفکت‌ها و نویزهای موردنظر را به انتخاب خود حذف کنید. سپس با سیگنال‌های جدید مرحله استخراج ویژگی (نوع ویژگی به دلخواه) و طبقه بندی (نوع کلاسیفایر به دلخواه) را تکرار کنید. چه تفاوتی در نتایج ایجاد شده است؟

**گام 1 :** ابتدا دیتا رو بارگذاری میکنیم و در ادامه power spectrum را برای یکی از کانال ها بطور دلخواه رسم میکنیم

که در اینجا برای کانال 21 ام اینکار را با دستور pwelch انجام دادیم

```
load('BCICIV_calib_ds1b_100Hz.mat');

cnt = double(cnt) *0.1;
fs =nfo.fs;
figure
plot(cnt(:,1))

hold on
pwelch(cnt(:,21));figure(gcf)

figure
hold on
pwelch(cnt(:,21),[],[],[],fs);figure(gcf);% plotting powerspectrum |
```

همانطور که از اطلاعاتی که در قسمت nfo.classes هست، سوژه قراره تصور دست راست و چپ را انجام بده در

nfo.classes			
	1	2	3
1	left	right	
2			
3			

نتیجه با یک مسئله ی دوکلاسه روبرو هستیم

## سوال 1

در مرحله پیش پردازش، با تحلیل فرکانسی در صورت لزوم از فیلترهای ناچ و میان گذر و... استفاده کنید. آیا نیازی به نرمال سازی دیپتا وجود دارد؟ در این صورت داده ها را استاندارد کنید.

از آنجا که فرکانس نمونه برداری دیتای مورد نظرمون برابر با 100 هرتز می باشد در نتیجه طبق قضیه نایکوئیست حداکثر اطلاعاتی که داریم تا 50 هرتز می باشد و همین طور در ادامه نیز چون می دونیم بیشتر اطلاعات تصور حرکتی بین بازه 8 تا 30 هرتز می باشد برای همین از یک فیلتر میان گذر با فرکانس 8 تا 30 هرتز استفاده میکنیم بنابراین اصلاً نیازی به طراحی فیلتر notch در این پروژه نمی باشد .

### نرمال سازی

```
bands=[1,4 ;4,8; 8,12; 12,30];
for band=1:4
    for ch=1:59
        disp(['Working on Bands:',num2str(band),'channel:',num2str(ch)]);

        %[b,a] =butter(order,wn,ftype)
        if band ==1
            [b,a] = butter(4,bands(band,2)/(fs/2));%make IIR filter (band pass filter )
        else
            [b,a] = butter(4,bands(band,:)/(fs/2));%make IIR filter (band pass filter )
        end
        temp_cnt = filtfilt(b,a,cnt(:,ch));%we use filtfilt for problem of filt that make phase shift
        temp_cnt = (temp_cnt-mean(temp_cnt))/std(temp_cnt);%Normalize
    end
end
```

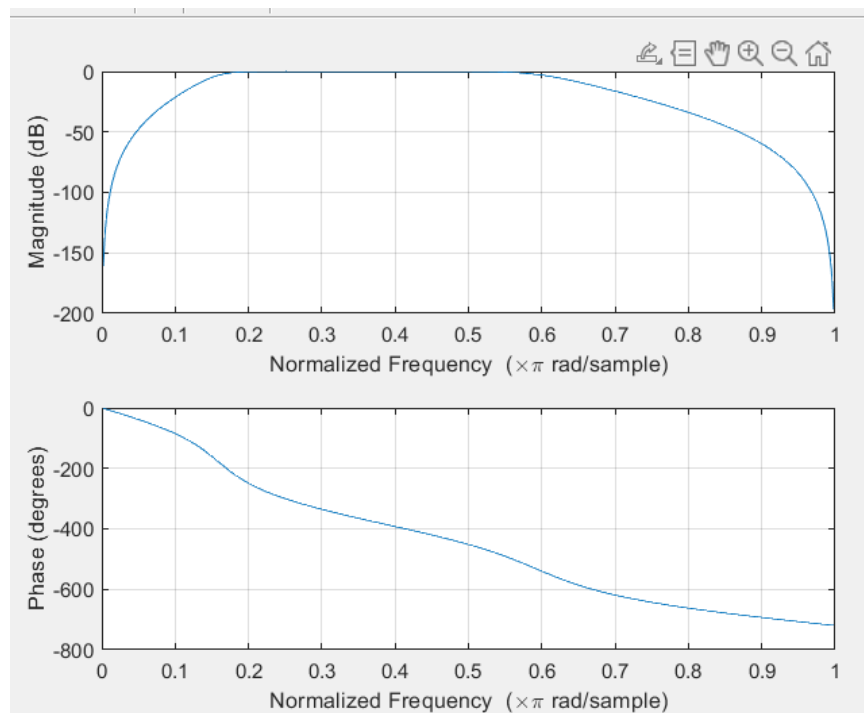
```

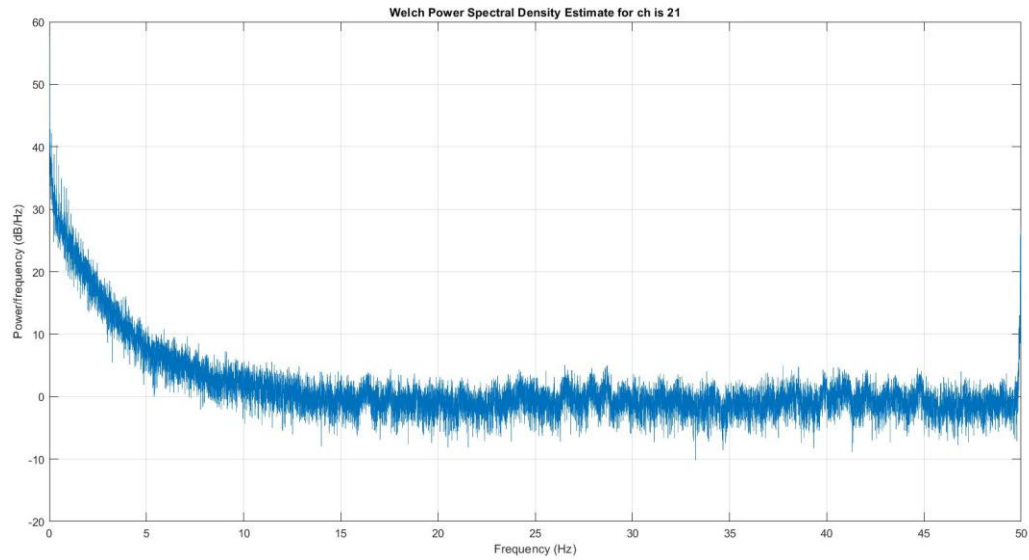
28 %%
29
30 % select data in beta and mu band
31 [b,a] = butter(4, [8 30]/((nfo.fs/2)));
32
33 new_cnt= filtfilt(b,a ,cnt);
34
35 pwelch(new_cnt(:,1));figure(gcf);
36 pwelch(new_cnt(:,1),[],[],[],fs);figure(gcf);

```

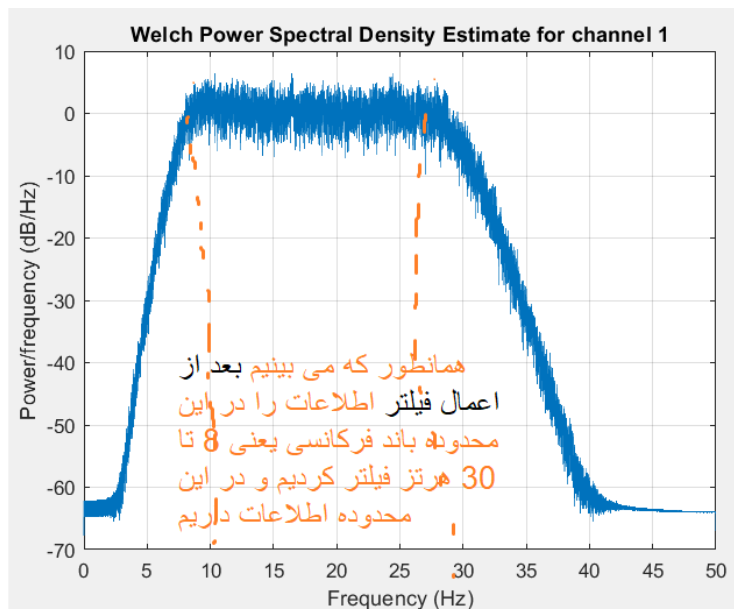
از اونجایی که می دانیم اطلاعات تصور حرکتی بیشتر در باند میو و بتا (8-30hz) می باشد، سیگنال مون را در این بازه فیلتر میکنیم تا صرفاً از اطلاعات محدوده ی 8 تا 30 هرتز استفاده کنیم .

\*از دستور filtfilt نیز استفاده میکنیم برای اعمال فیلتر تا دچار اختلاف فاز نشویم \*





در شکل بالا نمایش Power Spectral رو قبل از اعمال فیلتر میان گذر می بینیم و بعد از اعمال فیلتر میان گذر 8 تا 30 هرتز بصورت پایین میشود البته برای کانال های مختلف رسم کردیم ولی نتایج برای کانال های مشابه نیز همان می باشد





## سوال 2

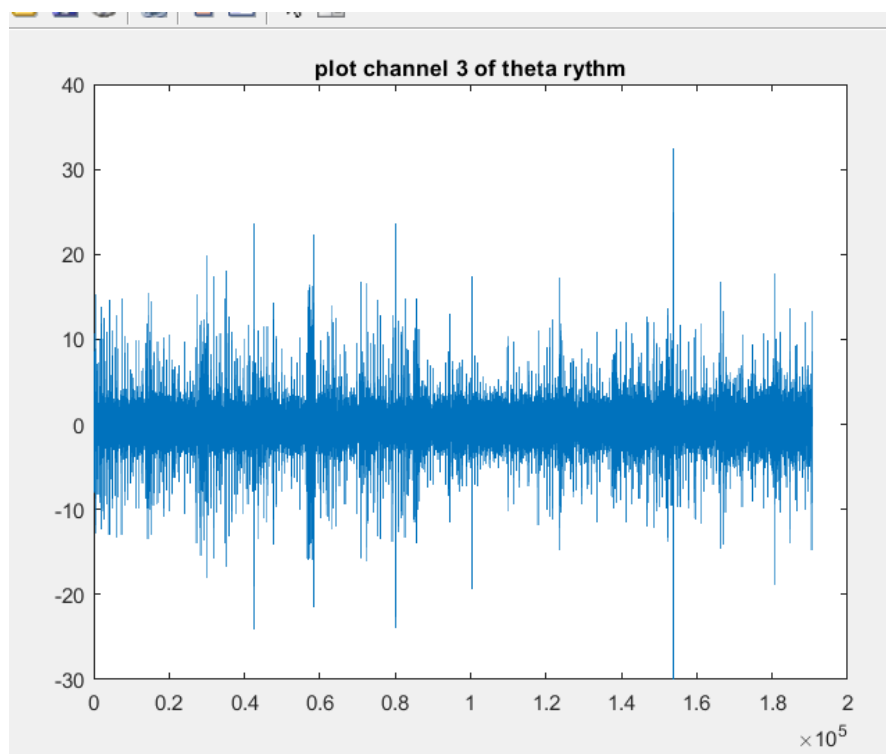
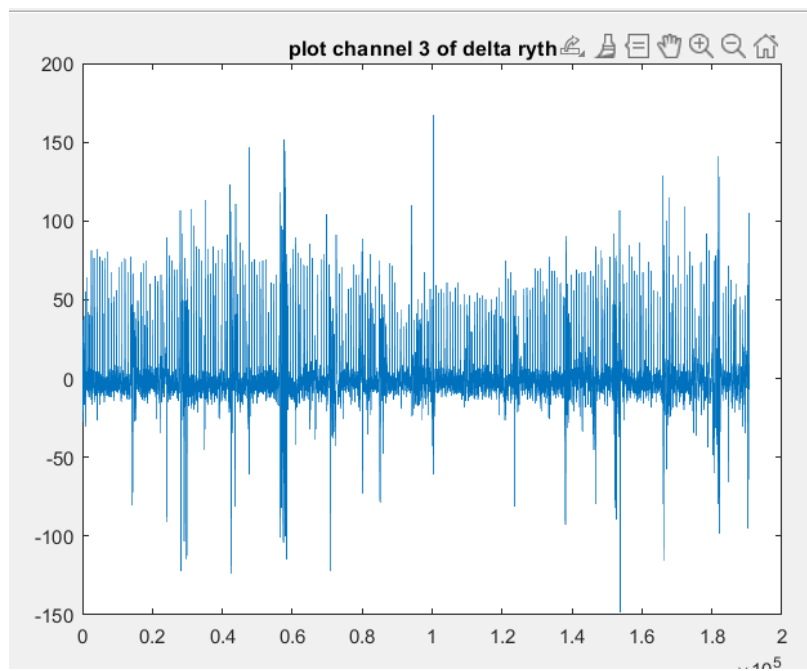
۲- در مرحله استخراج ویژگی، از ویژگی‌های مختلف آماری (واریانس، کورتوسیس)، توان باندهای مختلف (دلتا، تتا، آلفا و بتا) استفاده کنید. در این مرحله ویژگی‌های دو باند آلفا، دلتا و بتا، تتا را با استفاده از اسکتر پلات (برای دو جفت باند) رسم کرده و فضای ویژگی رسم شده را تحلیل کنید. آیا در این فضا اختلافی میان ویژگی‌ها قابل رویت است؟

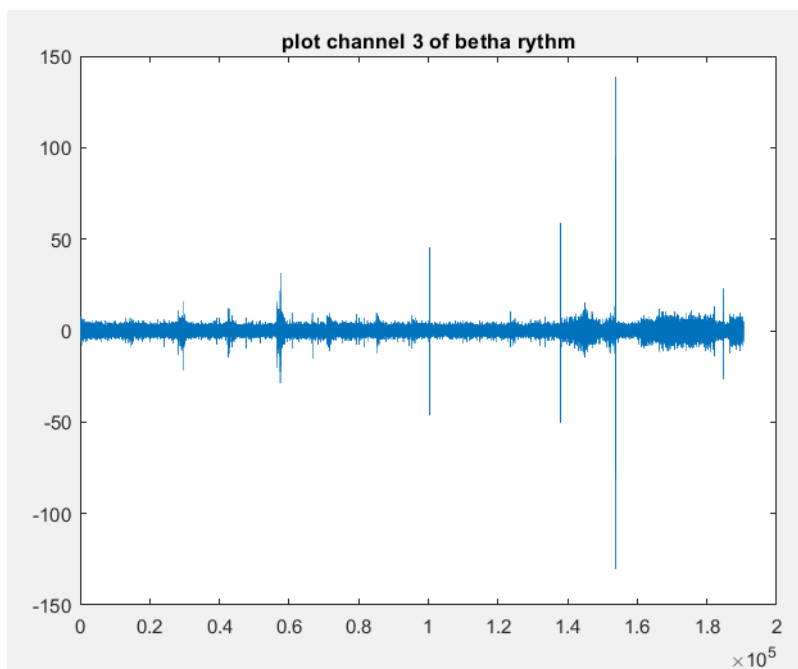
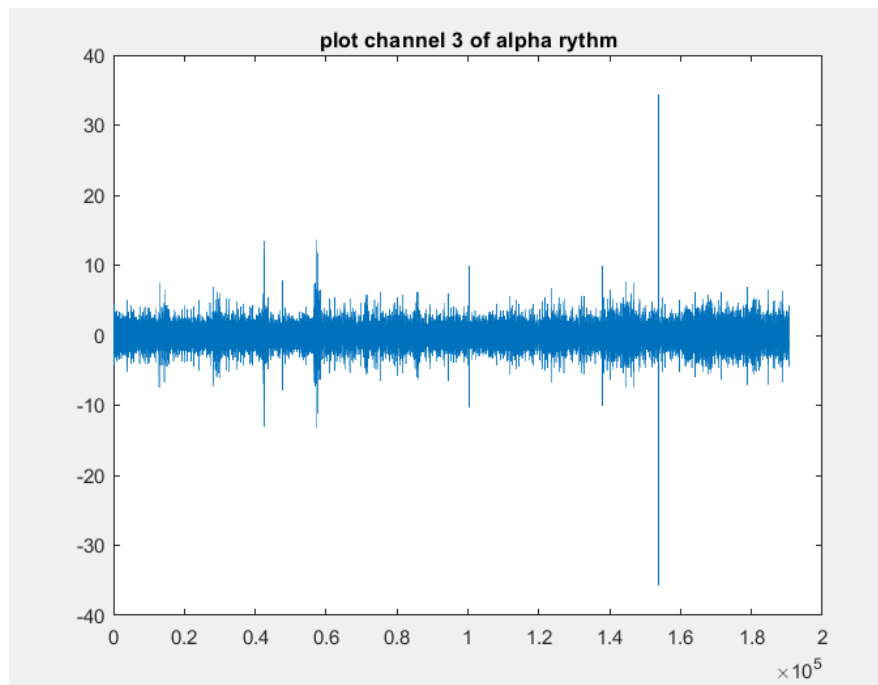
استخراج ویژگی با استفاده از واریانس

یکی از ویژگی‌های مهمی که می‌تونیم از سیگنال‌های EEG استخراج کنیم، واریانس سیگنال‌هاست. ما می‌توانیم با استفاده از دستور `var(signal)` می‌تونیم واریانس سیگنال مورد نظرمون رو بدست بیاریم.

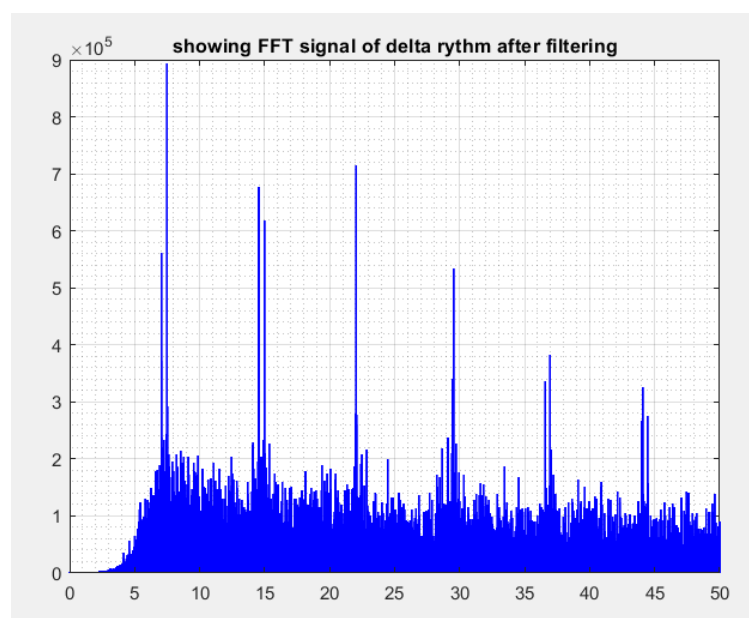
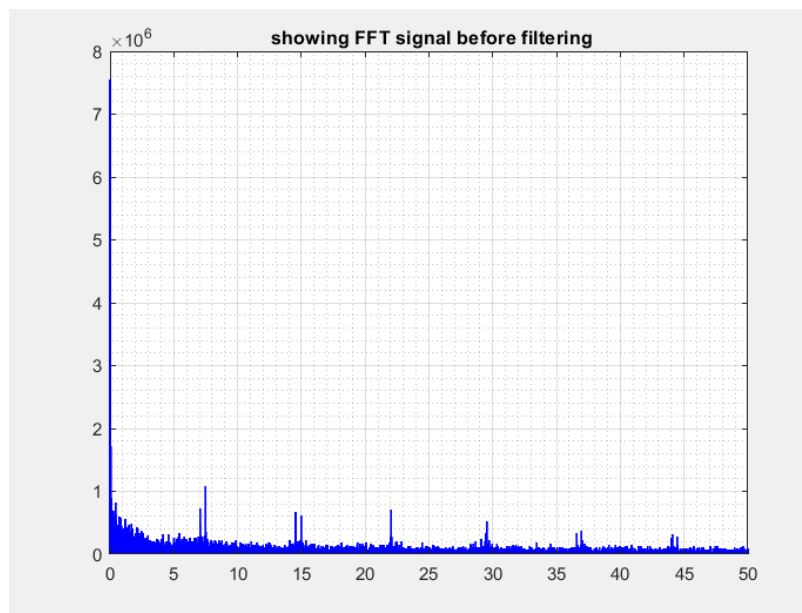
```
%% feature extraction
% create feature using variance
for i =1:length(class2)

    class1_Feature (i,:) = var(class1{i});
    class2_Feature (i,:) = var(class2{i});
end
```





استفاده از تبدیل فوريه



همین طور که در دو شکل صفحه قبل دیدیم وقتی سیگنال را در حوزه فوریه می بریم و محدوده فرکانسی آنرا محدود میکنیم، میتوانیم اطلاعات فرکانسی مهمی را از آن استخراج کنیم

استخراج ویژگی های آماری برای باند های مختلف فرکانسی دلتا، تتا، میو و بتا :

```
for i = 1:100
    var1_delta(i, :) = var(delta1{i});
    var1_theta(i, :) = var(theta1{i});
    var1_mu(i, :) = var(mu1{i});
    var1_betha(i, :) = var(beta1{i});

    kur1_delta(i, :) = kurtosis(delta1{i});
    kur1_theta(i, :) = kurtosis(theta1{i});
    kur1_mu(i, :) = kurtosis(mu1{i});
    kur1_betha(i, :) = kurtosis(beta1{i});

    power1_delta(i, :) = sum(abs(fft(delta1{i})).^ 2);
    power1_theta(i, :) = sum(abs(fft(theta1{i})).^ 2);
    power1_mu(i, :) = sum(abs(fft(mu1{i})).^ 2);
    power1_betha(i, :) = sum(abs(fft(beta1{i})).^ 2);
end

for i = 1:n2
    var2_delta(i, :) = var(delta2{i});
    var2_theta(i, :) = var(theta2{i});
    var2_mu(i, :) = var(mu2{i});
    var2_betha(i, :) = var(beta2{i});

    kur2_delta(i, :) = kurtosis(delta2{i});
    kur2_theta(i, :) = kurtosis(theta2{i});
    kur2_mu(i, :) = kurtosis(mu2{i});
    kur2_betha(i, :) = kurtosis(beta2{i});

    power2_delta(i, :) = sum(abs(fft(delta2{i})).^ 2);
    power2_theta(i, :) = sum(abs(fft(theta2{i})).^ 2);
    power2_mu(i, :) = sum(abs(fft(mu2{i})).^ 2);
    power2_betha(i, :) = sum(abs(fft(beta2{i})).^ 2);
end
```

از دستور روبرو نیز برای نمایش scatter دو جفت باند (الف و بتا) و (دلتا و تتا) استفاده میکنیم

```
figure
scatter(var_delta, var_alpha)
xlabel('Delta Variance')
ylabel('Alpha Variance')
title('Scatter Alpha Var vs Delta Var')

figure
scatter(kur_delta, kur_alpha)
xlabel('Delta Kur')
ylabel('Alpha Kur')
title('Scatter of Alpha Kur vs Delta Kur')

figure
scatter(power_delta, power_alpha)
xlabel('Delta Power')
ylabel('Alpha Power')
title('Scatter of Alpha Power vs Delta Power')
```

استفاده از channel selection

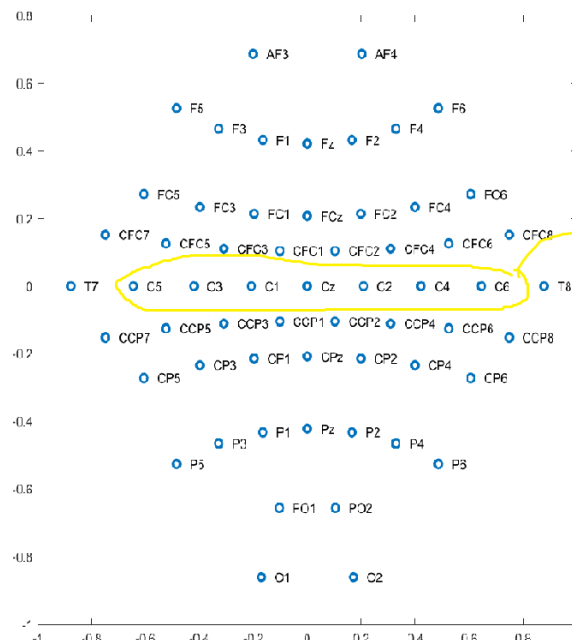
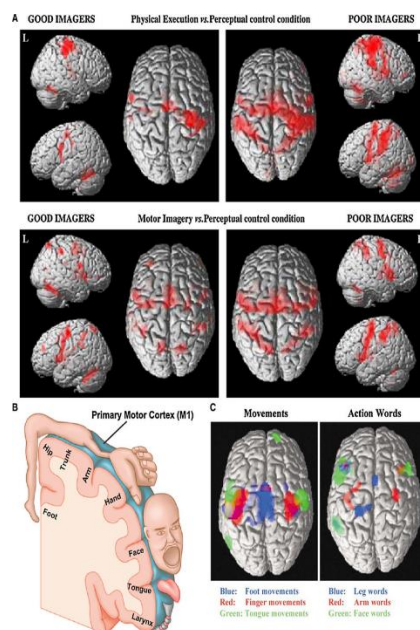
```
% channel selection
```

```
channel_index = 26:32; % select channels :c5 c3 c1 cz c2 c4 c6
```

```
channel_index(4) = []; % ignore information of Cz
```

```
cnt_ch = new_cnt(:,channel_index);
```

همانطور که انتظار داریم بیشتر اطلاعات تصور حرکتی مربوط به ناحیه primary motor area یا همان ناحیه M1 می باشد، بنابراین می توانیم بجای اینکه از اطلاعات تمامی 59 کانال ثبت شده استفاده کنیم، از اطلاعات کانال های مربوط به این task استفاده کنیم، در نتیجه یک عملیات channel selection نیاز داریم که بصورت زیر عمل میکنیم:



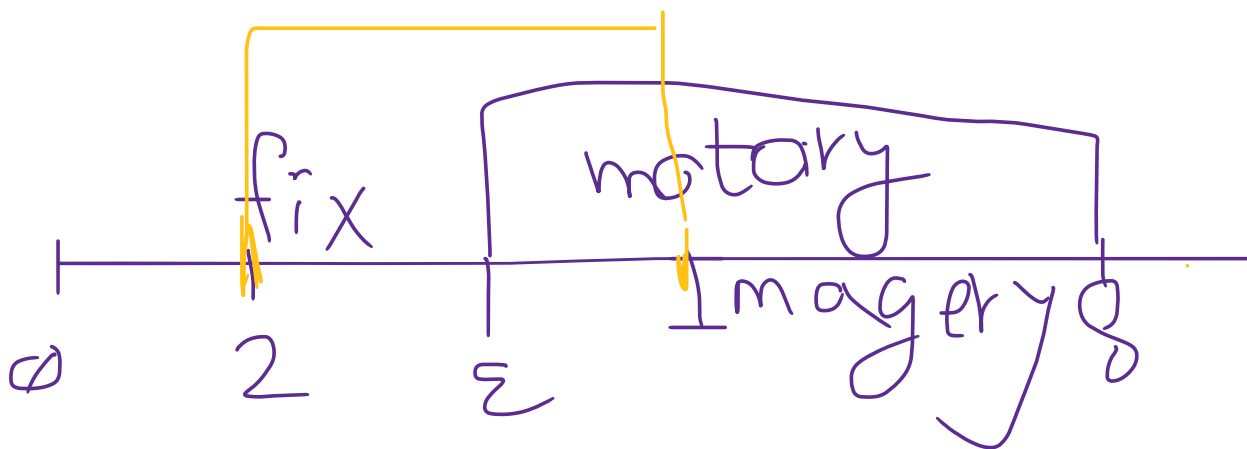
عادات تصور حرکتی  
این ناحیه می باشد  
یک چنل سلکشن  
نیاز داریم

۳- با استفاده از ویژگی‌های استخراج شده در مرحله قبل، مسئله طبقه بندی دو کلاسه را برای انواع ویژگی‌ها (ویژگی‌های آماری و ویژگی‌های فرکانسی) حل نمایید. در این مرحله به دلخواه می‌توانید ویژگی‌ها را با یکدیگر ترکیب کنید. راهنما: در این مرحله از دو کلاسیفایر SVM و KNN استفاده کنید و نتایج را باهم مقایسه کنید.

```
%%
number_class1 = sum(mrk.y == 1); %right hand
number_class2 = sum(mrk.y == -1); % left hand

class1 = cell(1,number_class1);
class2 = cell(1 , number_class2);
```

```
79 %% state 2 when using channel selection
80 for i =1:length(mrk.pos)
81     if mrk.y(i) ==1
82         class1{i}= cnt_ch(mrk.pos(i)+200 : mrk.pos(i)+600,:);
83     elseif mrk.y(i) ==-1
84         class2{i}= cnt_ch(mrk.pos(i)+200 : mrk.pos(i)+600,:);
85     end
86
87 end
88 |
```





```
%% creation and concat features & labels
features = [class1_Feature ; class2_Feature]; %concatition
label = [zeros(length(class1_Feature),1);
ones(length(class2_Feature),1)];
```

feature

Concat (چپ کردن)

label = 0  
کلاس 1

label = 1  
کلاس 2

طول بازه مربوط به تصور حرکتی را در واقع باید از 400 تا 800 بگیریم چراکه فرکانس نمونه برداری مون برابر با 100 هرتز می باشد و سوژه در بازه بین 4 تا 8 ثانیه تصور حرکت را انجام داده است، در این پروژه برای این که بررسی کنیم ببینیم آیا با تغییر طول بازه مورد نظر دقت بهتر میشود یا خیر برای همین کمی از قسمت ترایالی که به سوژه حتی تسک را نشان داده اند را نیز بررسی کردیم

طبق این کد نیز برای کلاس یک، label را برابر با 0 گذاشتیم و برای کلاس دوم یا همان تصور دست چپ، label را برابر 1 گذاشتیم

```
number_class1 = sum(mrk.y == 1);
```

با دستور **هایلایت** شده روبرو می توانیم تعداد ترایال هایی که سوژه تصور دست راست یا کلاس اول را کرده است را بدست بیاوریم که برابر با 100 می شود .  
برای تعداد ترایال های تصور دست چپ نیز برابر با 100 می شود .

دوتا سلول نیز برای کلاس های یک و دو با دستور روبرو درست میکنیم

```
|
class1 = cell(1,number_class1);
class2 = cell(1 , number_class2);
```

```
%%
class1 = class1(~cellfun ('isempty' , class1));
class2 = class2(~cellfun ('isempty' , class2));
|
```

%% state 2 when using channel selection

```
for i =1:length(mrk.pos)
    if mrk.y(i) ==1
        class1{i}= cnt_ch(mrk.pos(i)+200 : mrk.pos(i)+600,:);
    elseif mrk.y(i) ==-1
        class2{i}= cnt_ch(mrk.pos(i)+200 : mrk.pos(i)+600,:);
    end
end

end
```

بعد در اینجا نیز ترايال هایی که سوژه تصور دست راست کرده را در کلاس یک و ترايال هایی که سوژه تصور دست چپ کرده را در کلاس دوم می ریزیم

class1												
1x200 cell												
1	2	3	4	5	6	7	8	9	10	11	12	13
01x6 doub...		401x6 doub...		401x6 doub...	401x6 doub...	401x6 doub...	401x6 doub...	401x6 doub...		401x6 doub...		

حالا وقتی class1 را باز کنیم یک سلول 1\*200 می بینیم که ترايال هایی که در کلاس یک تصور دست راست کرده مقدار پیدا کردند و مابقی تهی شدند برای اینکه این مقادیر تهی را نیز از بین ببریم از دستور پایین استفاده میکنیم :

```
class1 = class1(~cellfun ('isempty' , class1));
class2 = class2(~cellfun ('isempty' , class2));
```

حالا دیگه ابعاد class1 و class2 بصورت 100\*1 می باشد که دیگه مقادیر تهی نداریم .

در گام بعد نوبت این هست از ویژگی های آماری که استخراج کردیم استفاده کنیم برای همین انها را در یک ماتریس features می ریزیم که بصورت زیر اینکار را میکنیم:

```
37 %% feature selection
38 mean_cnt = mean(new_cnt);
39 ZeroMeanCnt = new_cnt - mean_cnt ;
40 CovMatrix = cov(ZeroMeanCnt);
41
42 Number_EigVec = 35; % we choose only 35 principal components
43 [EigVec , EigVal] = eig(CovMatrix);
44 [Sort_EigVec , Sort_Index] = sort(diag(EigVal),'descend');
45 chooseEigVec = Sort_Index(1:Number_EigVec);%only 10 principal components
46 cnt_pca = cnt * EigVec(:,chooseEigVec);
```

همین طور label گذاری را نیز بصورت بالا انجام می دهیم

از اونجایی که label ها رو خودمون درست کردیم در نتیجه برای صدتای اول label ای که گذاشتیم برابر با 0 و برای صدتای دوم نیز label برابر با 1 هست ،برای اینکه کار یادگیری ماشین مون درست باشه بهتره ترتیب این label گذاری مون بصورت رندوم باشه ،برای اینکار نیز از دستور روبرو استفاده میکنیم :

```
%% creation and concat features & labels
features = [class1_Feature ;class2_Feature];%concatition
label = [zeros(length(class1_Feature),1);
         ones(length(class2_Feature),1)];

%% state when using kurtosis
```

استفاده از PCA

در قسمت بالا نیز همانطور که می بینیم از PCA استفاده کردیم تا ابعاد فیچر مون رو کاهش بدیم تا بتونیم عملکرد مدل را بهبود بدیم

%% %shuffling

%randperm(200)

```
idx = randperm(200);
features = features(idx , :);
label = label(idx);
```

توضیح مختصر PCA: ابتدا باید میانگین رو صفر کنیم و بعدش از ماتریس COV استفاده میکنیم و در اینجا ما از 35تا principal component اولیه استفاده کردیم در نتیجه ابعاد فیچر مون از 59\*19000 به 35\*19000 کاهش پیدا کرده است.

جواب سوال سوم  
استفاده از مدل یادگیری ماشین SVM

```
%% % Classification using SVM  
SVMModel = fitcsvm(features , label , 'KernelFunction','gaussian', 'ClassNames',[0;1]);
```

با دستور روبرو یک object می سازیم که مدل svm رو روی دیتامون fit کنیم، نوع کرنل ماشین بردار پشتیبان مون رو نیز برابر gaussian قرار دادیم

ارزیابی مدل یادگیری ماشین SVM  
از اونجایی که دیتامون یک دیتای نوروساینسی می باشد و حجم دیتا بسیار کم هست، بهتره از kfold cross validation یا random subsampling استفاده کنیم، تا ارزیابی که انجام می دهیم یک ارزیابی دقیق تر باشد

همین طور که از کد بالا نیز می بینیم، k را برابر 5 قرار دادیم و یک object بنام PartitionModel ایجاد کردیم

برای بدست آوردن صحت یا همون Accuracy نیز از کد پایین استفاده کردیم :

```
ACC=1-kfoldLoss(PartitionModel , 'LossFun','classiferror');
```

**نتیجه**

ACC for using SVM model is :0.53

```
% Classification using KNN
```

```
KnnModel = fitcknn(features , label , "NumNeighbors", 5);
```

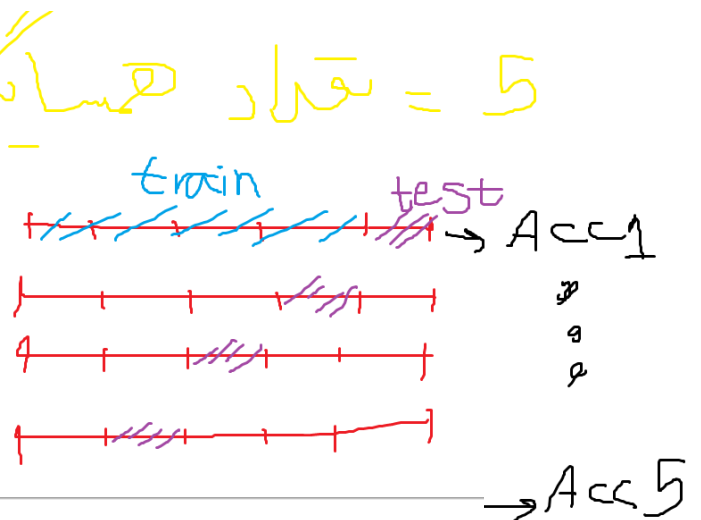
```
% validation
```

```
PartitionModel = crossval(KnnModel , 'kfold' , 5);
```

```
ACC=1-kfoldLoss(PartitionModel , 'LossFun','classiferror');
```

k-fold

Cross validation



```
% Classification using KNN
```

```
KnnModel = fitcknn(features , label , "NumNeighbors",5);
```

```
% validation
```

```
PartitionModel = crossval(KnnModel , 'kfold' , 5);
```

```
ACC=1-kfoldLoss(PartitionModel , 'LossFun','classiferror');
```

```
% validation using k-fold cross validation
```

```
PartitionModel = crossval(SVMModel , 'kfold' , 5);
```

در این قسمت نیز از مدل knn استفاده کردیم با  $k = 5$  nearest neighbors و از متد  $k$  fold cross validation استفاده کردیم که در ادامه نیز بطور خلاصه این متد ارزیابی را در صفا بعد توضیح دادیم

ACC for using KNN model is :0.585

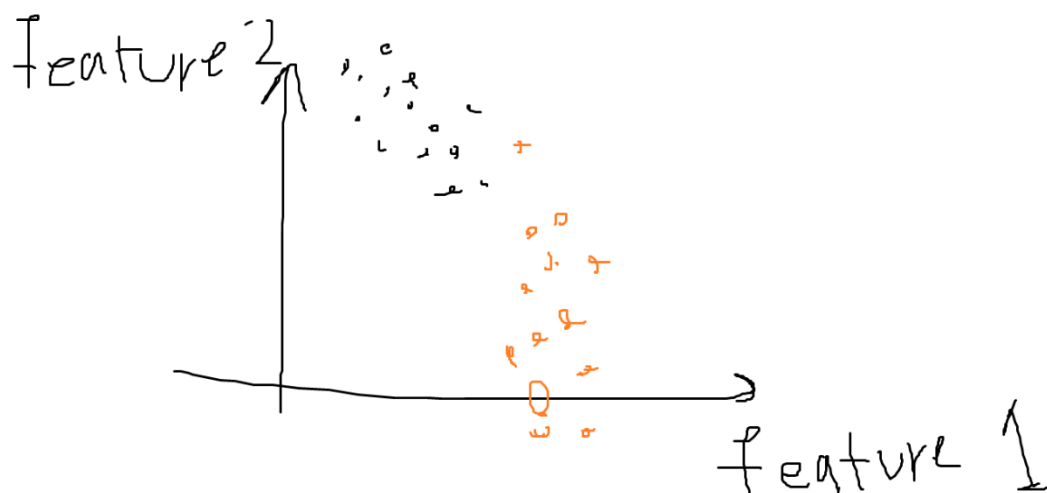
#### ارزیابی مدل یادگیری ماشین KNN

برای اینکه دقتی که بدست می آوریم قابل اعتماد باشد و به این صورت نباشد که بصورت شانسی مدل مون در قسمت خوب دیتا فیت شده باشد، دیتا را به  $k$  تا fold می شکنیم و یک از این قسمت ها را برای test و مابقی را برای train می زاریم و هر سری یک ACC بدست می آوریم و مطابق شکل بالا این کار را تکرار میکنیم و در آخر یک میانگین گیری از تمامی دقت های بدست آمده انجام می دهیم تا به صحت و عملکرد با اطمینان بیشتری دست پیدا کنیم .

## سوال چهارم

۴- با تحلیل نتایج فوق، مناسب ترین نوع ویژگی برای طبقه بندی این مسئله را تعیین نمایید.

مناسب ترین ویژگی با توجه به تحلیل هایی که انجام دادیم این می باشد که ابتدا سیگنال را از حوزه زمان به حوزه فوریه ببریم و سپس از باند های الف و بتا استفاده کنیم در واقع یک فیلتر میانگذر 8 تا 30 هرتز بزنیم و بعدش از power سیگنال و واریانس سیگنال استفاده کنیم و در اخر از یک فیلتر فضایی CSP استفاده کنیم و این گونه می شود که فضای ویژگی بصورت شکل پایین می شود :



## سوال پنجم

۵- با تحلیل نتایج فوق، مناسب ترین نوع کلاسیفایر را برای طبقه بندی این مسئله را تعیین نمایید.

برای اینکه به نتیجه های دقیق تری برسیم از محیط گرافیکی متلب نیز کمک گرفتیم تا بتوانیم مدل های یادگیری ماشین بیشتری را نیز با پارامترهای مختلف امتحان کنیم، که در ادامه دوتا از مدل ها یعنی SVM و KNN با پارامتر های مختلف را نتایج شون را آوردیم

در این قسمت از محیط گرافیکی متلب یعنی "classificationLearner"

استفاده کردیم و Dataframe مون را وارد این محیط میکنیم

از رستی Channel selection استفاده کردم

**Data set**

**Data Set Variable**

DataFrame 200x7 double

☒ Use columns as variables

☐ Use rows as variables

**Response**

☒ From data set variable

☐ From workspace

column\_7 double 0 .. 1

**Predictors**

	Name	Type	Range
<input checked="" type="checkbox"/>	column_1	double	8.06101 .. 167.22
<input checked="" type="checkbox"/>	column_2	double	6.56858 .. 168.538
<input checked="" type="checkbox"/>	column_3	double	6.65096 .. 157.359
<input checked="" type="checkbox"/>	column_4	double	7.03207 .. 160.761

Add All Remove All

[How to prepare data](#)

**Validation**

☒ Cross-Validation

Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold.

Cross-validation folds: 3

☐ Holdout Validation

Recommended for large data sets.

Percent held out: 25

☐ Resubstitution Validation

No protection against overfitting. The app uses all the data for both training and validation.

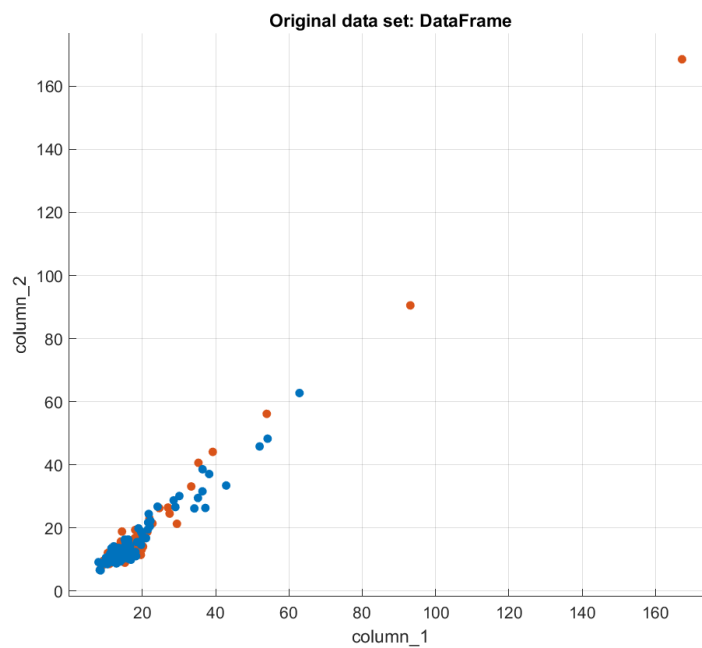
[Read about validation](#)

Response variable is numeric. Distinct values will be interpreted as class labels.

Start Session Cancel



توزیع دیتا وقتی که از channel selection استفاده کردیم بصورت زیر می باشد

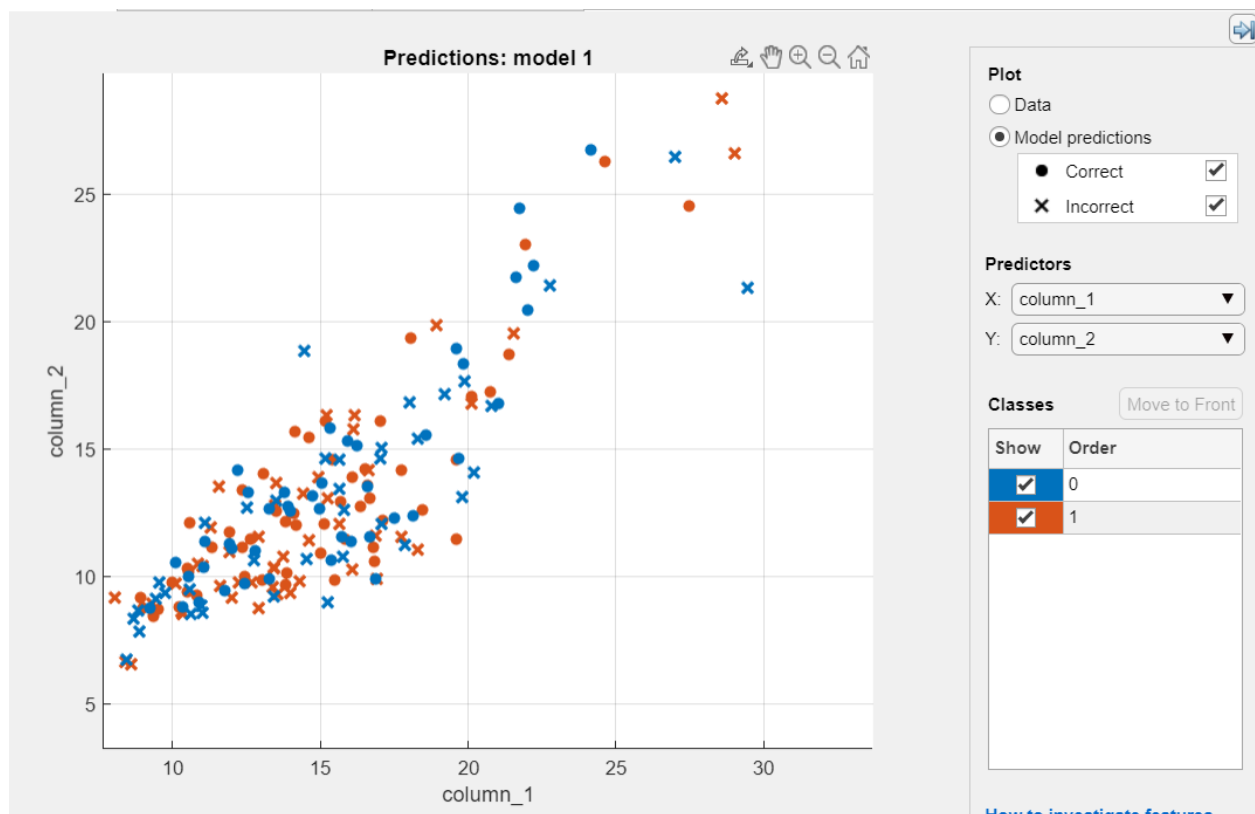
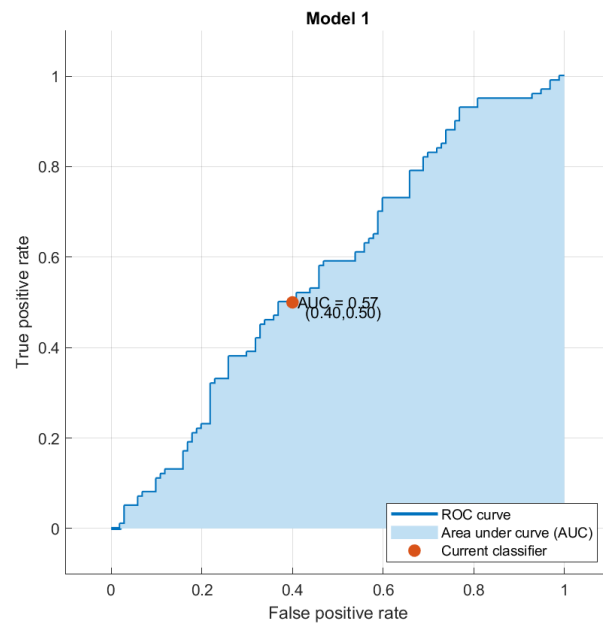
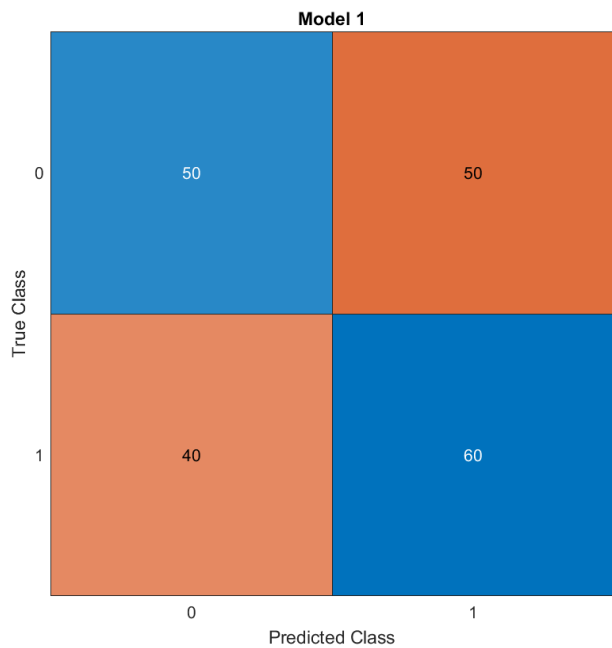


☆ 1 SVM

Accuracy (Validation): **55.0%**

Last change: Quadratic SVM
6/6 features

## نتایج استفاده از SVM



☆

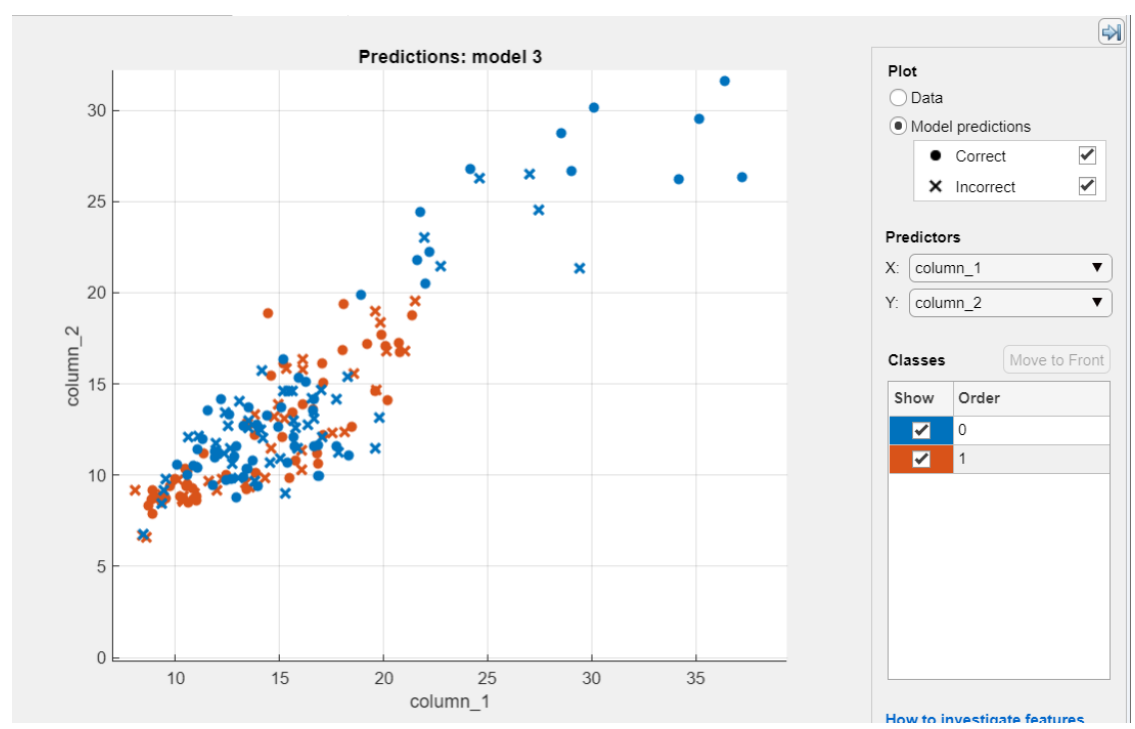
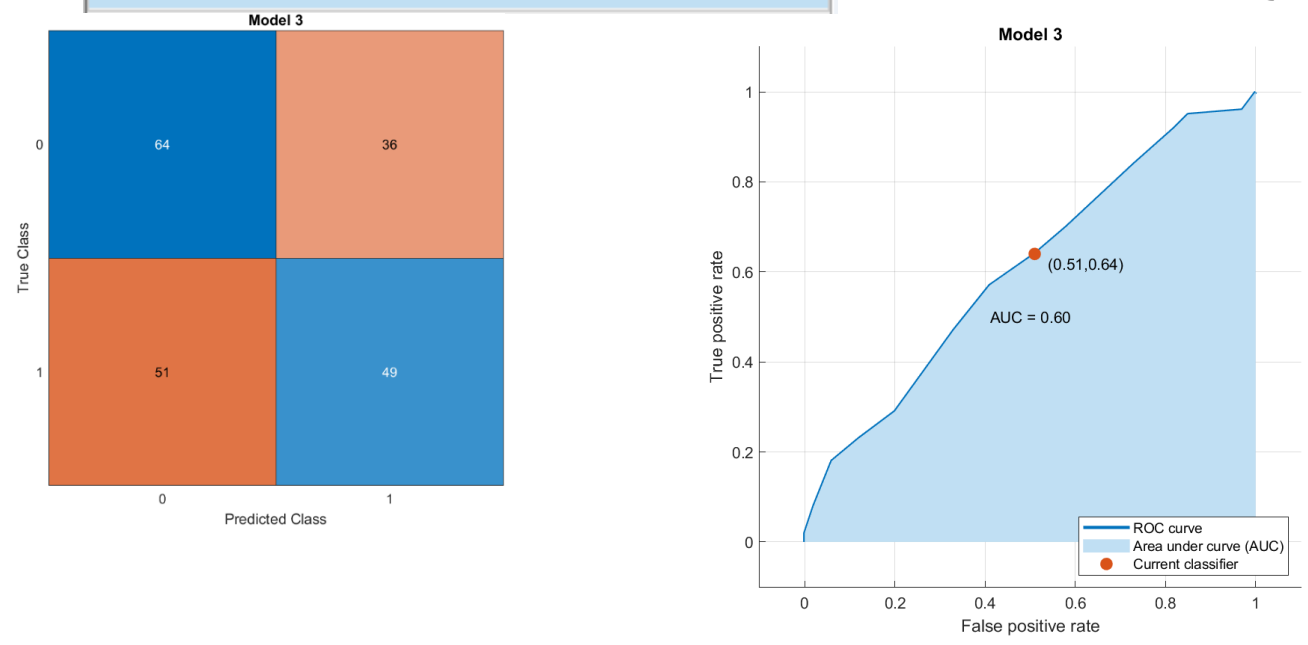
3 KNN

Accuracy (Validation): 56.5%

Last change: Medium KNN

6/6 features

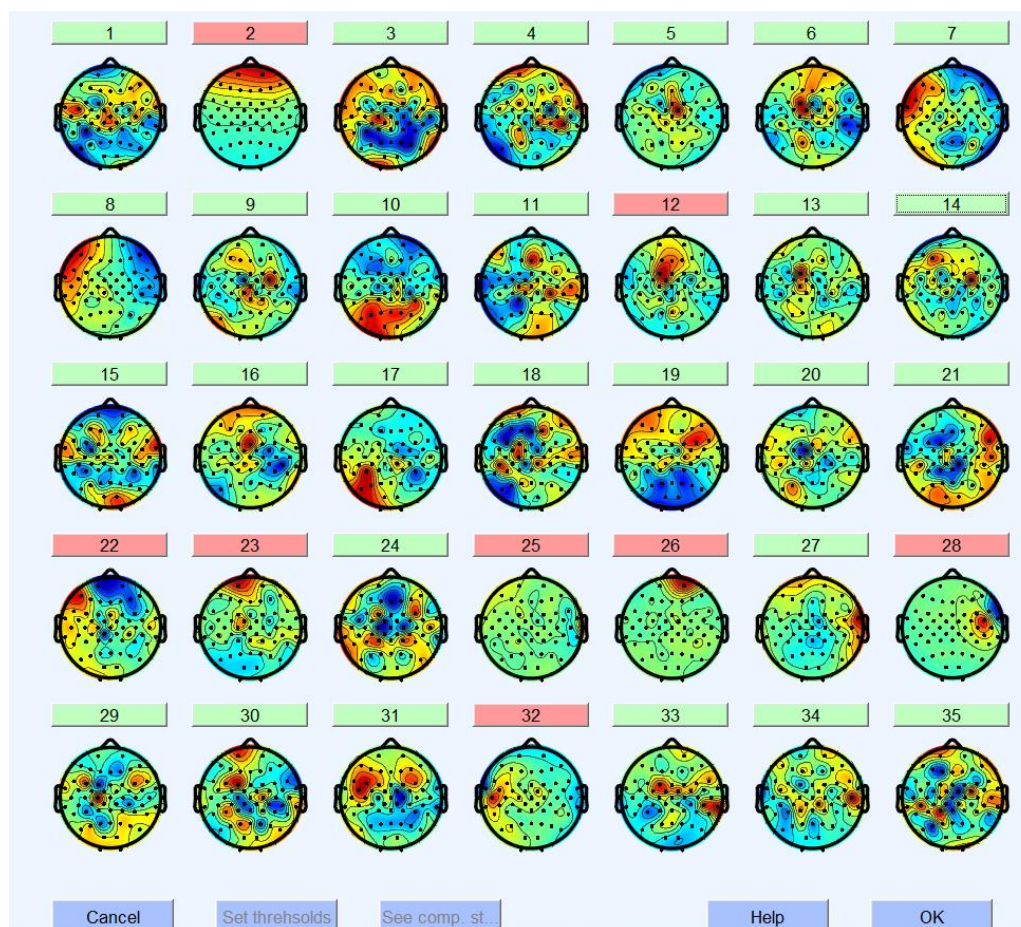
# نتایج استفاده از KNN

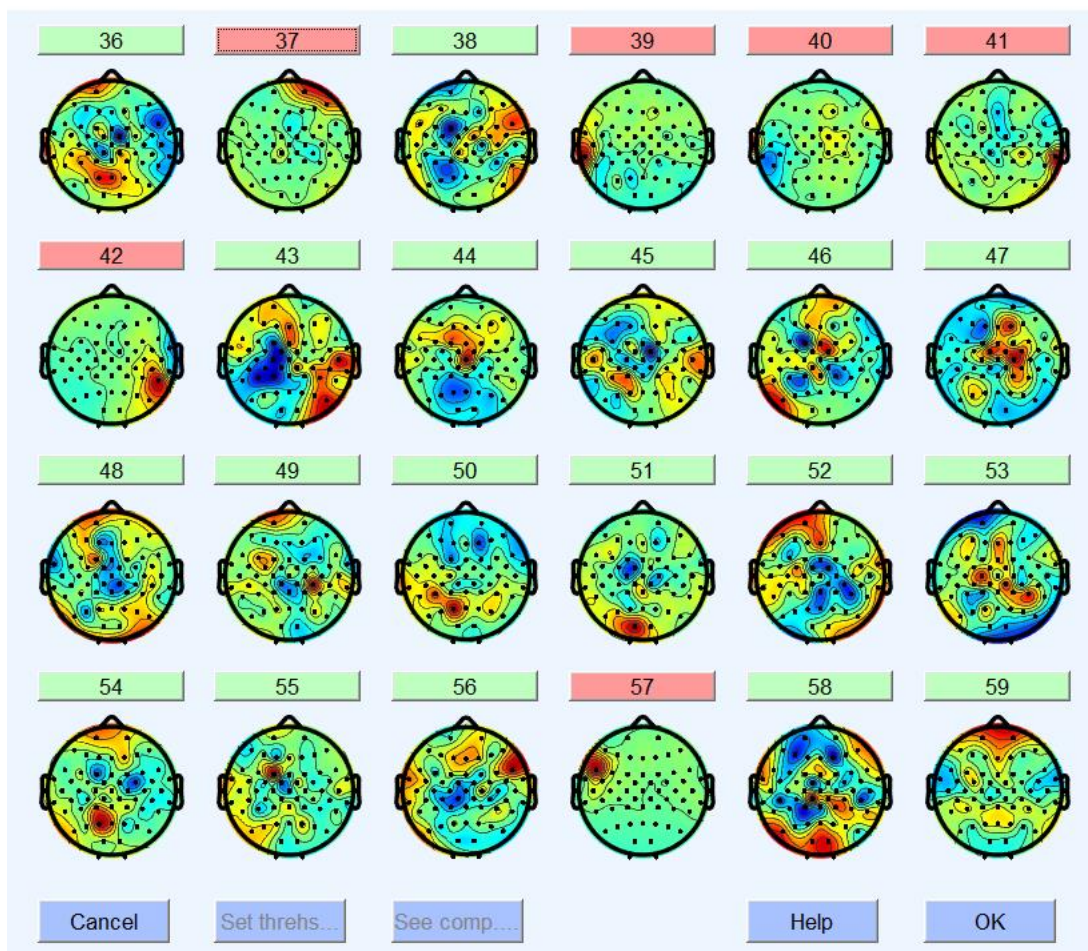


## سوال شیشم

۶- از روش ICA استفاده کنید و آرتیفکتهای و نویزهای موردنظر را به انتخاب خود حذف کنید. سپس با سیگنالهای جدید مرحله استخراج ویژگی (نوع ویژگی به دلخواه) و طبقه بندی (نوع کلاسیفایر به دلخواه) را تکرار کنید. چه تفاوتی در نتایج ایجاد شده است؟

اول که دیتا را مانند تمرین سری دوم وارد محیط eeglab میکنیم بعدش موقعیت کانالها را مشخص میکنیم و در ادامه رفرنس گذاری را بر مبنای Common Average Refrence می زاریم ،حالا از ICA استفاده میکنیم و بعدش در ادامه Componentهایی که نویزی هستند و حاوی اطلاعات مفیدی نمی باشند را حذف میکنیم که در ادامه Componentهایی که می خواهیم حذف بکنیم را علامت گذاری کردیم و بصورت زیر می باشد





حالا که component هایی که نویزی بودند را مشخص کردیم آنها را حذف میکنیم

Components [2,12,22,23,25,26,28,32,37,39,40,41,42,57] flagged for rejection.

Do you want to remove these components?

Note: we recommend removing components in STUDY instead

Cancel

Manual rej.

Yes

بعدش سیگنال تمیز شده را از قسمت export می گیریم و نام سیگنال مون را برابر با `cntRemoveBadComponents` می گذاریم