Matlab Assignment
Neuroscience & Neural Engineering

Ioannis Charalampous

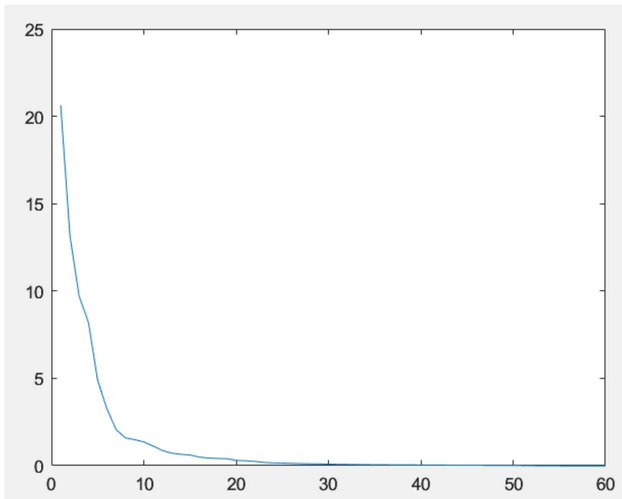
03500053

Dimitrios Papanagnou

03500045

A) 1. Apply PCA to the data to identify EEG channels that covary. How many PCs are needed to describe this dataset? *10 points*

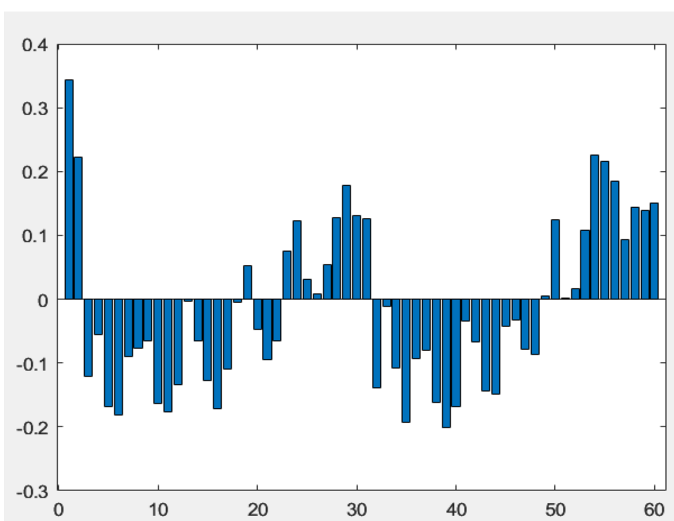
In AskisiA.m we applied PCA on EEGdata with 60 PCs equal to the number of channels in the EEE data so that we identify the channels that covary. We then plot the latent value of PCA which are the eigenvalues of PCs.



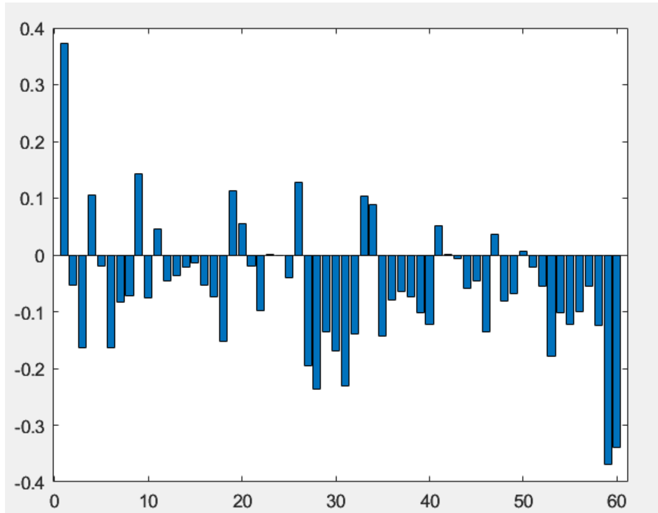
By using the "elbow" method we see that additional PC components after 8 PCs contribute diminishing returns in terms of explaining variance. So, we can decide that 8 PCs are needed to describe this dataset.

2. Plot a) the first 2 PCs (coeffs) and b) their trial-averaged activations (scores) for each one of the 2 stimuli. *10 points*

PC1:

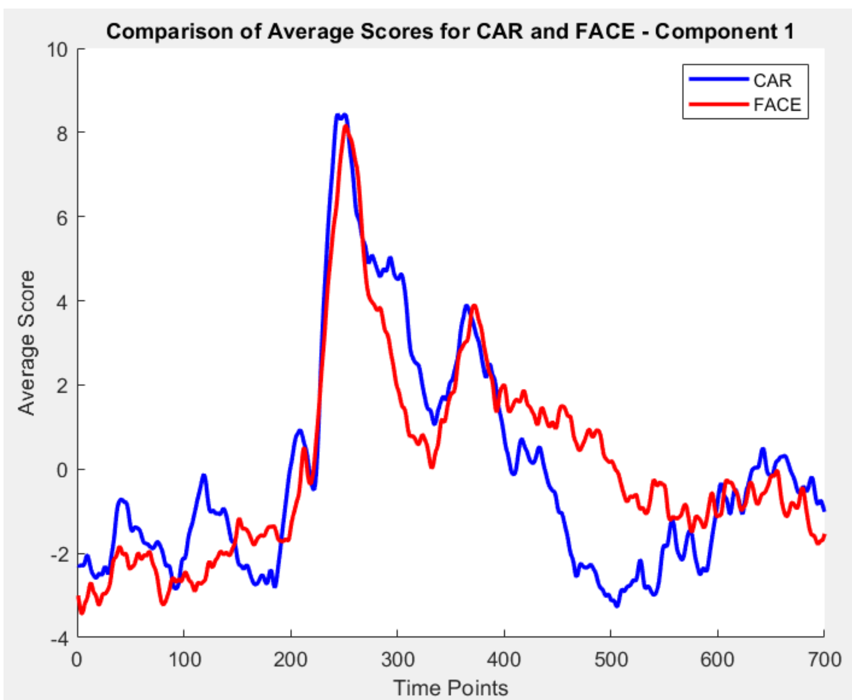


PC2:

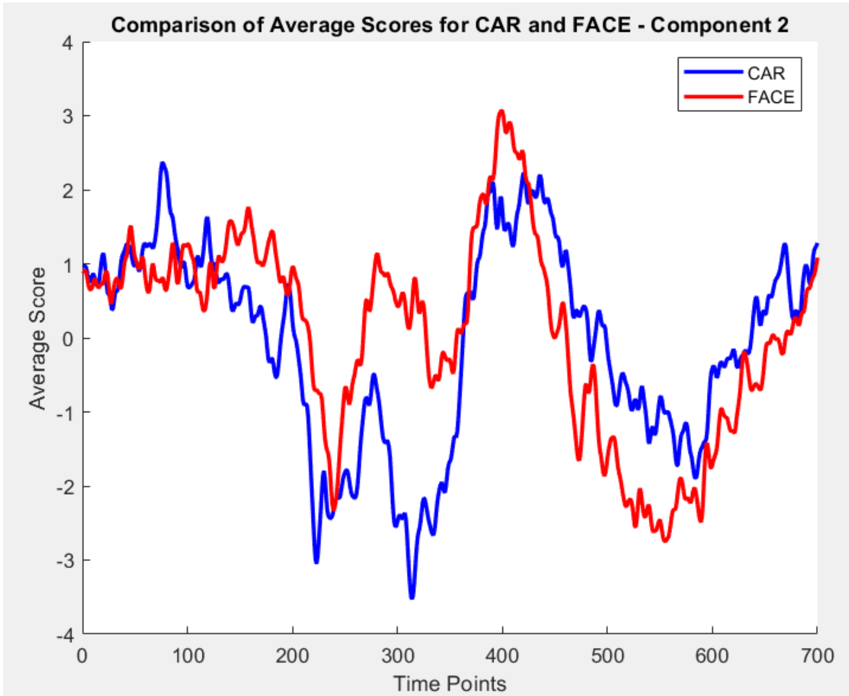


In the first two plots we can see the first two PCs and each bar indicates the contribution and relationship of each EEG channel to this principal component.

SCORE1:



SCORE2:



In these two plots we can see the average score of PC1 and PC2 in each different class (CAR, FACE) on 700 time points.

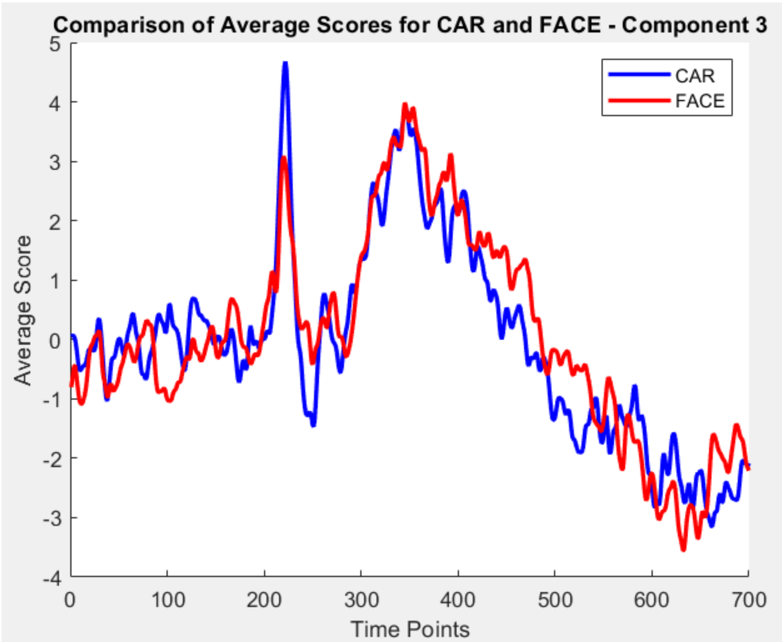
3. Do these PCs allow discrimination of the 2 stimuli and at which time windows? Would adding more PCs improve discrimination? Quantify this using a classification algorithm. *10 points*

These PCs allow discrimination of the two stimuli, and we can see this when the two lines (blue, red are different from each other, which means that the average scores for each stimuli have a lot of different values, thus allowing discrimination.

Especially on PC1 the time points [100:200]ms and [400,550]ms show a lot of potential for discriminatins the two stimuli.

On PC2 the time points [150:200]ms and [250,350]ms show some difference in the score values but we start to observe that it is harder to find differences in the score values of the two stimuli.

This is explained when we plot the average scores for the third PC:



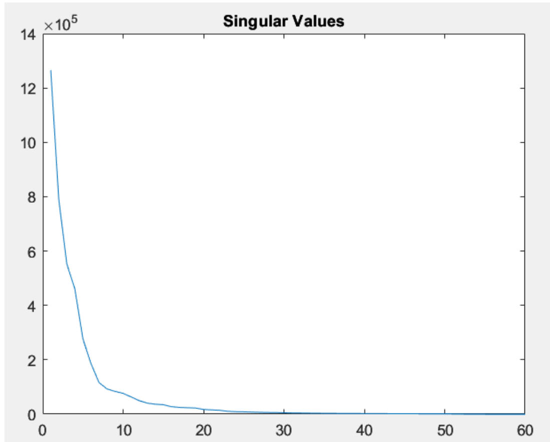
We see that the average scores for each stimuli start to become the same. The first few PCs explain the majority of the variance, and each subsequent PC explains progressively less. At some point, the additional variance explained by new PCs is trivial, adding little to discrimination of the two stimuli.

Next we use the ldadecoding.m file for classifying the average score of one PC with fisher linear discriminant classifier. After experimenting and leveraging from the observations from the plots we use the average score of PC1 with time window [400:550]ms which are classified with an accuracy of 76.25%.

B) Repeat A using a matrix or tensor decomposition algorithm of your choice. Explain why you chose this algorithm and compare its performance with PCA. *20 points*

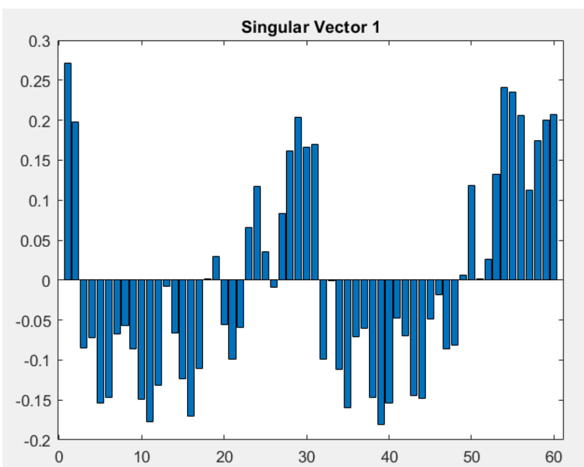
Code for Question B is in askisiB.m file. The matrix decomposition matrix that we used is the SVD and we follow the same steps as with Question A.

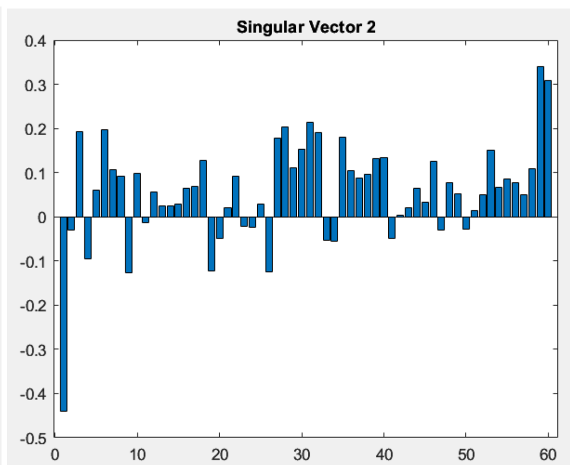
First, we plot the squared diagonal values (singular values) of matrix S, which can explain how many singular vectors (V matrix) we need to describe the dataset.



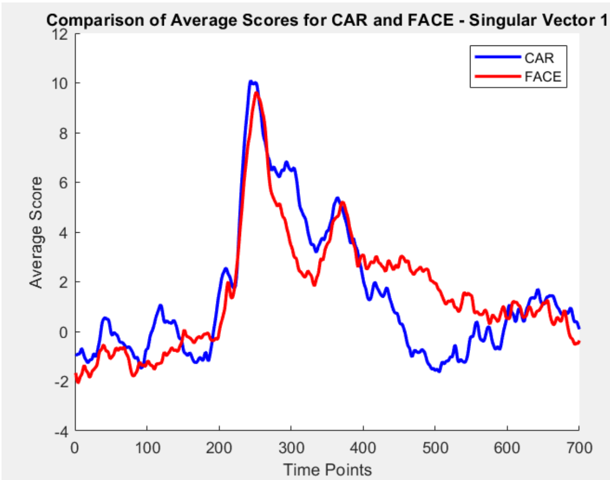
We observe that 8 singular vectors are enough to describe the dataset, which are the same results as with PCA.

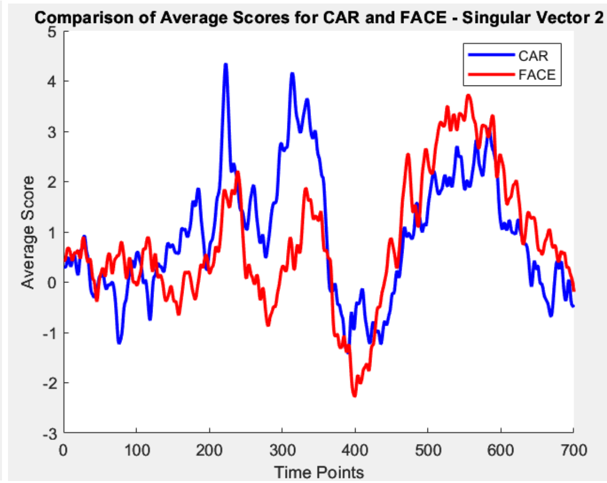
Next, we plot the first and second singular vectors:





Then we plot the average scores for each stimuli:





From the singular vectors we can observe the contribution and relationship of each EEG channel to this singular vector.

The score is computed from multiplying the U and S matrices then we compute the average score for each stimuli for singular vector 1 and 2.

We observe now more clearly that the first singular vector has better discrimination ability in time window [400:550]. This time window of the first singular vector is fed into lda decoder with performance 72.5%.

C) 1. Use the single_trial_analysis.m Matlab function to compute EEG classification performance (measured by the area under the ROC curve) Az in the time windows [201,250]ms and [451,500]ms. Compare performance of the two classification algorithms available (LDA vs Logistic Regression). *5 points*

In this question that is solved in AskisiC.m we select from the EEGdata the time windows [301,350]ms and [551,600]ms, because the stimulus starts at 100ms we need to select 100ms later than the time windows in the question. Then we perform four experiments with two classification algorithms (LDA, Logistic Regression) in each time window.

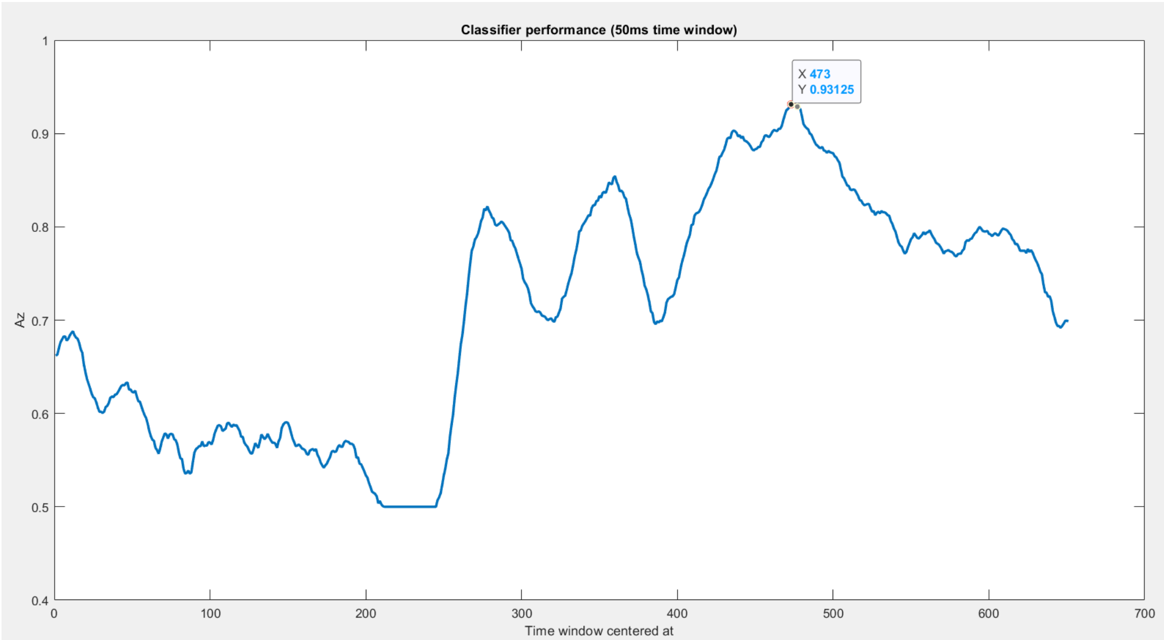
- First Time Window [301,350]ms:
  - Logistic Regression: Az = 0.7213
  - LDA: Az = 0.7444
- Second Time Window [551,600]ms:
  - Logistic Regression: Az = 0.7625
  - LDA: Az = 0.7888

Both classifiers perform better in the second time window [551,600]ms compared to the first [301,350]ms This shows that the EEG signals in the later time window are more informative or clearer for distinguishing between the two stimuli.

LDA often performs well in cases where the relationships between variables are approximately linear, and the classes exhibit multivariate normal distribution. If these conditions are met, LDA can outperform more complex models like Logistic Regression

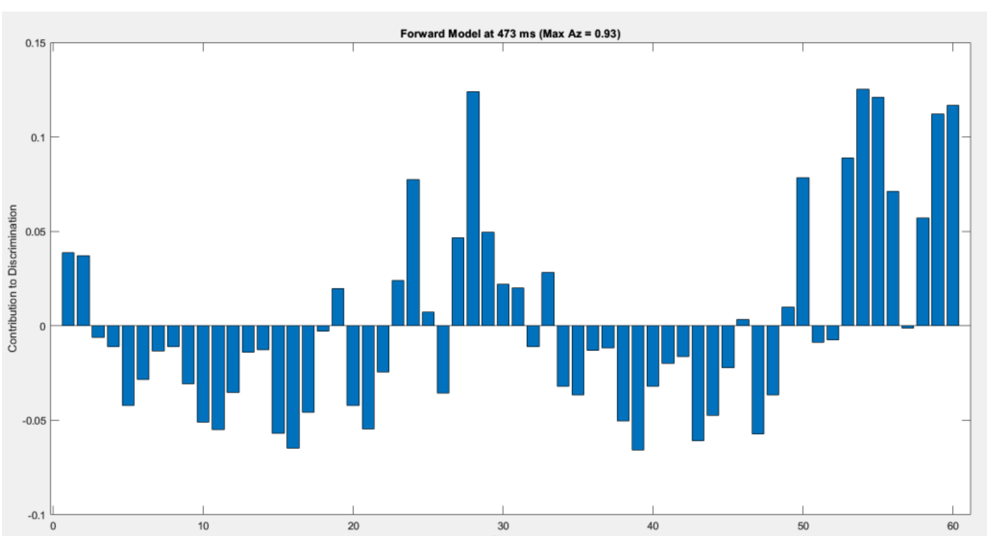
2. Plot an Az curve across time to illustrate how stimulus classification evolves over the duration of a 700ms trial. How long after stimulus presentation do we get the best stimulus classification from the EEG signals? Plot the forward model (indicating the channel contribution to discrimination) at the time of maximal discrimination. What happens if you change the duration of the time window? What happens if you change the overlap between consecutive time windows? *15 points*

For this question we use the single_trial_analysis.m to train the better performing LDA classifier on the whole EEGdata dataset with a sliding window of 50ms and an overlap of 49ms to thoroughly see how stimulus classification evolves over the duration of 700ms. Each window is classified and validated with Leave One Out cross-validation and the mean Az value is saved. After the whole training session, the following performance results are displayed.



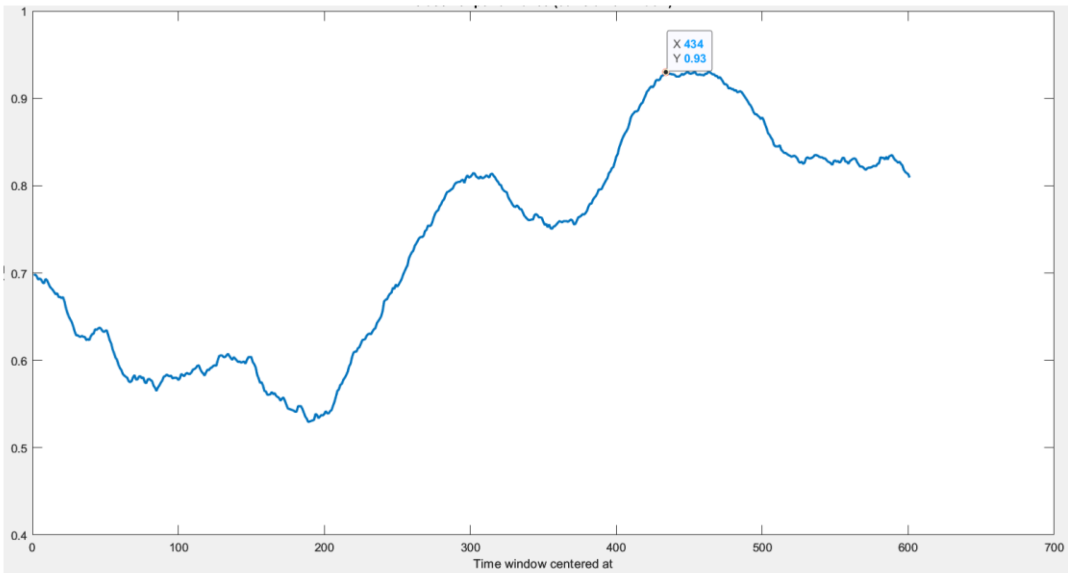
In this plot we observe that the classifier has a low Az performance before 100ms in the plot and then it slowly starts to increase reaching the Az value of 0.93 at 473 ms. This happens because the first 100 ms are the pre-stimulus recorded data and have no value in the discrimination of the stimuli and the classifier shows a good ability to recognize that. The highest accuracy occurs in the time window [473,522]ms meaning that this is the time window that has the highest discrimination ability between the stimuli.

Next, we plot the forward model (indicating the channel contribution to discrimination) at the time of maximal discrimination [473,522]ms. This is done by plotting the 'a' output of single_trial_analysis at the time where the area under the curve (Az) reached a maximum of 0.93.



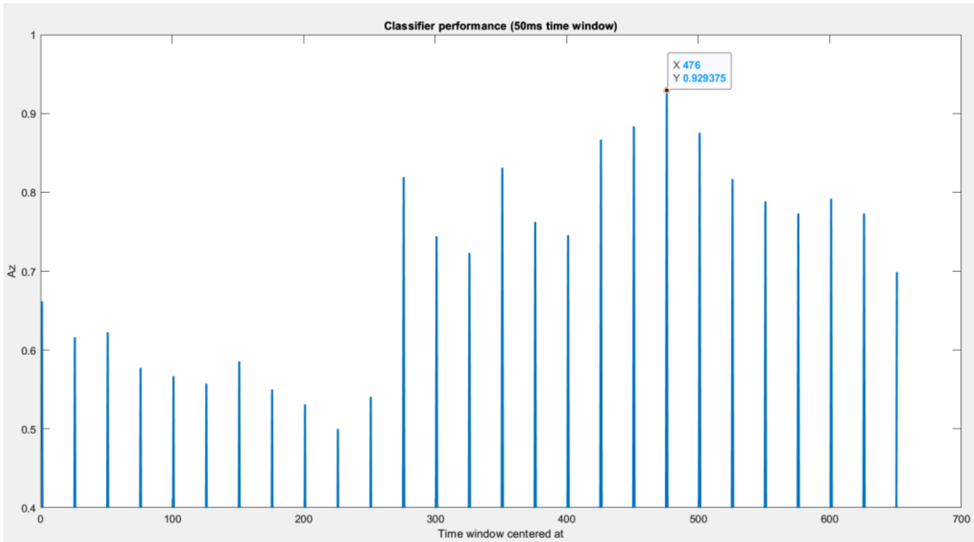
The graph shows a considerable variance in contributions across channels. Some channels have strong positive contributions, some have strong negative contributions, and others have contributions close to zero. Channels with taller bars (both positive and negative) are more influential in the discrimination task. These channels are likely located in brain regions actively involved in processing the stimulus or task at hand at this specific time point.

Following we experiment on the dataset with a sliding window of 100ms with the same overlap as before of 99 ms that allows us to check the performance on each possible window of the dataset.



It is observed that we have a similar recognition ability with low performances before 100ms (pre-stimulus) then it increases to the same Az value of 0.93 at 434ms.This means that the EEG data at the time window [434,533]ms has the highest discrimination ability. This time window contains the previous window [473,522]ms which further proves our results are correct and we have found the most important temporal part of the data.

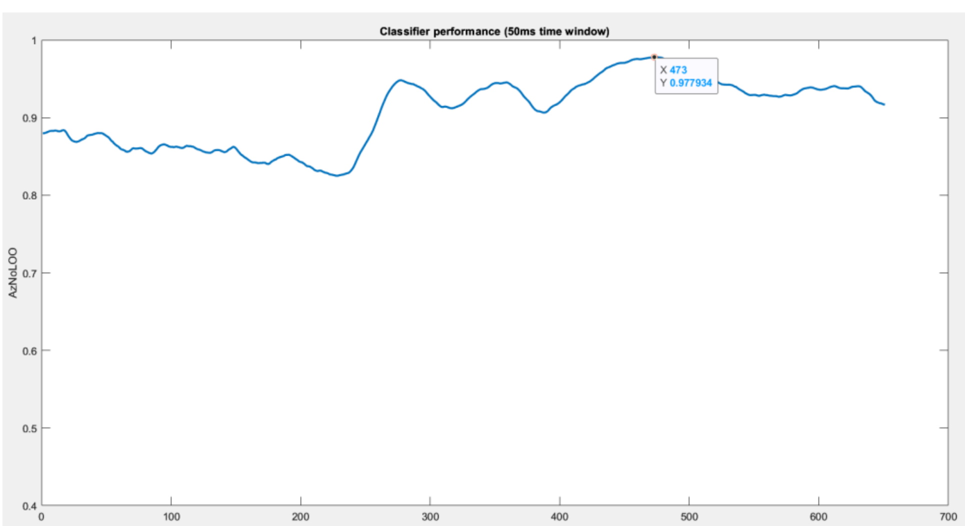
We also perform the classification of the dataset with 50ms windows and 50% overlap (25ms).



It is obvious that the Az values are less because we have less time windows and less available data for observation. We still observe though that the highest discrimination is observed in a similar time window [476,525] as before. Also, the plot has a similar shape as the previous one with low performance before the 100ms landmark then it steadily increases to the highest 0.93 Az performance.

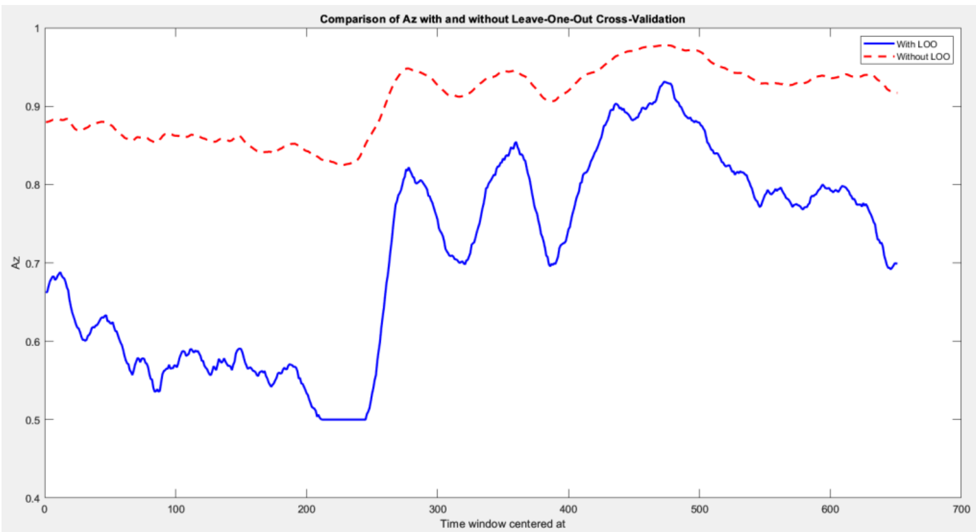
3. Compute the same curve without leave-one-out cross-validation (skipLOO=1). Observe the differences between the 2 curves focusing on the pre-stimulus Az values and comment on the role of cross-validation in classification analysis. *5 points*

To compute the Az curve without LOO cross-validation we change the appropriate flag on single_trial_analysis and we have the following Az plot:



The single_trial_analysis.m in this case uses the LDA classifier without a validation method as a result the classifier overfits on the dataset and we have no new data to test the performance. It is also important to note that as a result of the overfitting the classifier can't recognize the pre-stimulus period and showcases high performance in every time window with the highest at the same time window as before [473,522]ms with Az value of 0.977. This is the overfitted training accuracy that isn't representative of the actual discrimination that we want to achieve.

Both the Az curves with and without validation are shown in the following plot:



We can observe the differences in performances especially until the 100ms windows where the performances have the most deviations. It is important to note that the classifier has usually better performances on the training sets than the validation sets but also overfitting occurs when no validation occurs as shown here and the classifier loses its discrimination ability.

D) Now use a classification approach of your choice to discriminate faces from cars. Compare the performance of your approach with C and explain the differences. *20 points*

1. Model

Our classification approach for discriminating faces from cars from EEG signals was a deep learning approach. We employed a Bidirectional Long Short-Term Memory (Bi-LSTM) neural network for the EEG classification of our dataset. This model specializes in sequence classification tasks involving time series like our EEG signals. Each component of the model contributes to handling and processing the sequence data effectively, enabling the model to learn from both past and future context within the sequence.

```
% Bi-LSTM model
layers = [
    sequenceInputLayer(60)
    bilstmLayer(100, 'OutputMode', 'sequence')
    dropoutLayer(0.2)
    bilstmLayer(50, 'OutputMode', 'last')
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer
];
```

In the previous figure we can observe the Bi-LSTM model that we used.

- First we have a SequenceInput layer, which acts as the input layer for the data. It specifies that the input sequences consist of 60 features per time step. This corresponds to 60 channels of EEG data where each sequence is a series of observations from all the channels over time.
- Follows the Bi-LSTM layer, which processes the input sequence in both forward and backward directions, allowing it to capture dependencies and patterns that occur over time. With 100 units (cells), it learns to represent the input data into a higher dimensional space, capturing complex temporal relationships.
- Then a Dropout layer set at 20% dropout possibility of the cells helps to reduce overfitting.
- Another Bi-LSTM layer produces temporal features with 50 cells. The 'OutputMode' set to 'last' means that this layer only outputs the final LSTM output of the sequence processing, condensing the entire sequence's information into a single output vector.
- The feature vector produced before is used by the Fully Connected layer to produce the class probabilities, acting as the classification layer. This FC layer has two outputs for each stimuli (Face,Car) and uses the ‘softmax’ activation function. The ‘softmax normalizes the class probabilities so that they sum up to one, representing the probability distribution over the classes.
- Finally, the Classification layer specifies the objective of the network during training, which is to minimize the classification error. It computes the cross-entropy loss for

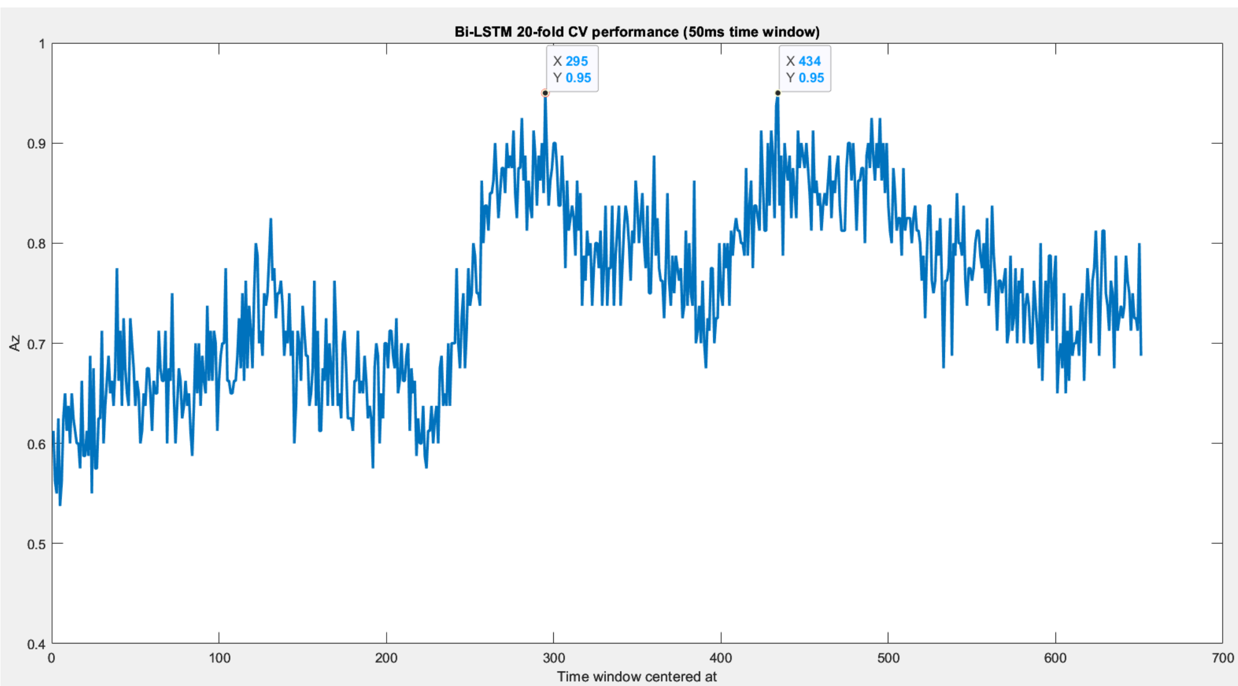
classification tasks, using the probabilities from the ‘softmax’ layer and the true label of each training example to adjust the model weights optimally.

2. Validation

Next, we tried to replicate the same experiments with the previous Question C in our AskisiD_20fold.m file. We used the same segmentation as before with a window of 50 ms and a 49 ms overlap. Then we tried to replicate the validation scheme that was used previously. We calculated the area under the ROC curve with the rocarea.m script that was provided for computing our classification performance. Also, we tried to replicate the Az calculation by studying the single_trial_analysis.m script. The Az scores under 0.5 value where set to 0.5 because values of 0.5 and lower indicates no discriminative ability (random performance).

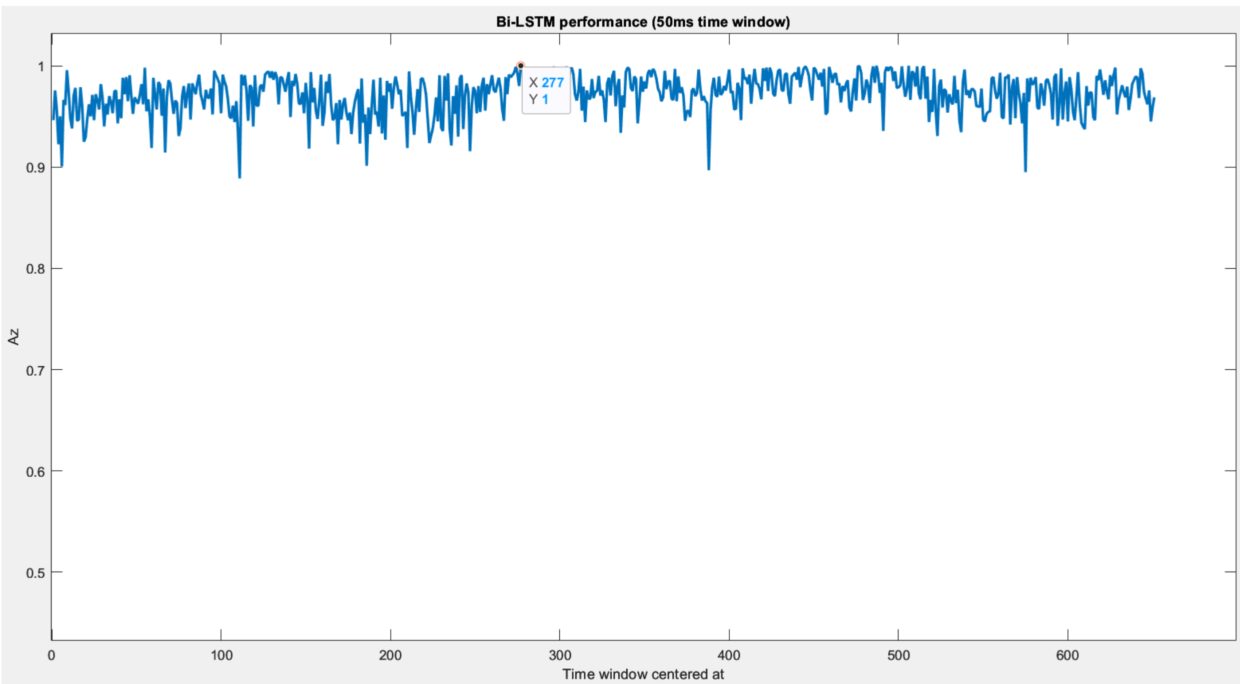
Regarding the validation of the results, we tried to use Leave One Out cross-validation but it was too computationally demanding for us and needed to much runtime. So instead of LOOCV which in our case, with 80 trials, means we need to use an 80-fold cross validation for every segment that we test our model on we employed 20-fold cross-validation which was not so demanding.

Below we present our results:



We observe that the Az values of the Bi-LSTM model shown also in bilstm_20fold.fig follow the same pattern as the LDA. It has low values pre-stimuli and then it starts to perform better. But in this case the Az scores seem to show more consistency and have higher values with higher performance of 0.95 at two time windows [295, 344] ms and [434,483] ms. The second time window seems to be close to the window with the best discrimination of the LDA classifier of 0.93 at [473,522] ms. With our model though now we observe a different time period that can also offer the impressive stimulus classification on the period [295, 344] ms. This improvement in performance occurs because the Bi-LSTM model allows capturing dependencies in the time series data from both past and future contexts and the model learns better the temporal dynamics of the EEG signals.

We also performed the classification as it was performed on Question 3 without validation in AskisiD.m. The model is trained on the whole segment each time and we can see only the Az performance of the whole segment without using new data to validate on so we have the following results which are impressive but are not representative on the models ability to discriminate on new data.



The performance of the model shown also in bilstm.fig is consistently high because of the impressive ability of the Bi-LSTM to learn on temporal data. It even reaches the Az value of 1 at [277, 326] ms. Most of the time periods have high Az values close to 1. Now we also observe that without validating on unseen by the classifier data we cannot have an actual representation of the EEG data's ability to discriminate between Faces and Cars because the model has good performance even on the time period before the stimulus was presented.