MP #2: Peer Feedback #2 (Pre-Feedback Work to Date) ⟹ Post-Feedback with annotation alongside.

Libo Zhang (lz200)

The structure of my work to date format will follow the recommended project milestones.

Note: Only the code written for testing my algorithm will be displayed to help peer review/feedback.

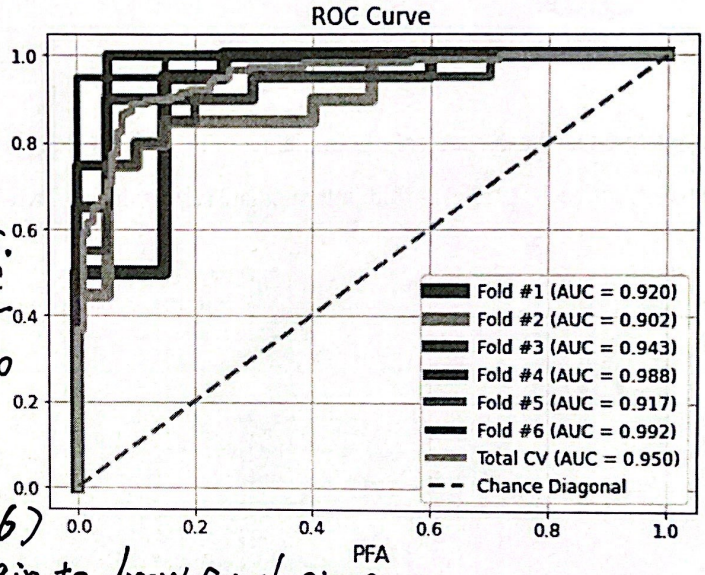
In Peer Feedback #1, I only completed Week 1 milestones. Therefore, in this session, to fully demonstrate that I have completed milestones for Week 2, 3, and 4, I decide to export the ROC plots for imaginary dataset and overt dataset first, as shown below (left blank space saved for Post-Feedback Annotation).

Overall peer feedback I received from other group members: Currently making demonstrable and good progress based on the Recommended Project Milestones.

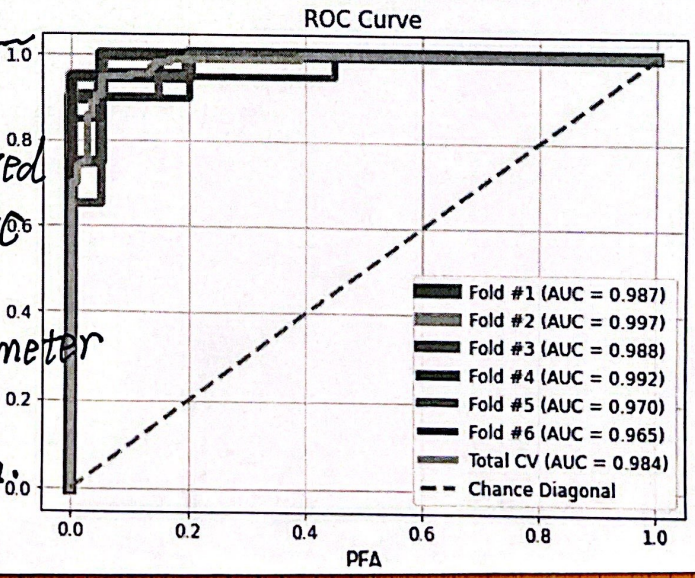
Notes to myself - What I will do next: Firstly, I will follow the Project Milestones (Week 3 and 4) to interpret my results and begin to draw conclusions. Secondly, I want to do more research and experiments on the ROC plots.

How the feedback I received influences my next steps: I realized that when using LinearSVC from sklearn, I should take $\frac{1}{\lambda}$ as the regularization parameter input due to the built in mathematical implementation.

[23]:
```
# Imaginary Dataset
# Provide the ROC for each 1st-level cross-validation fold (6 ROCs),
# and the total cross-validated ROC on a single graph.
plot_ROC(ds_img)
```

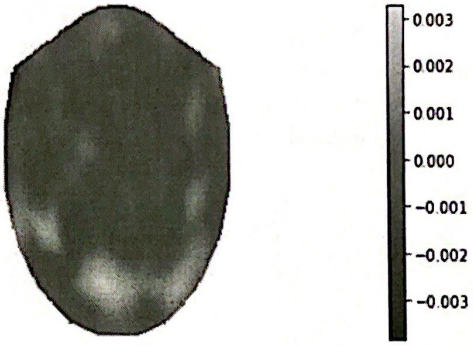


[24]:
```
# Overt Dataset
# Provide the ROC for each 1st-level cross-validation fold (6 ROCs),
# and the total cross-validated ROC on a single graph.
plot_ROC(ds_overt)
```

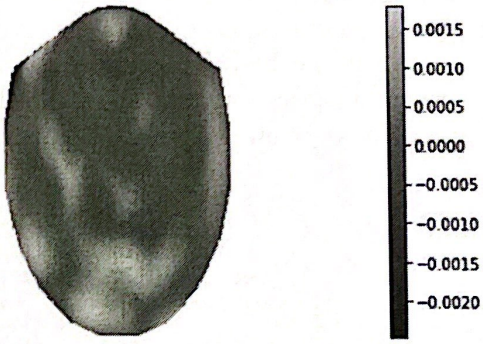


Then, let us visualize the 204×1 vector weights on the brain surface (Imaginary and Overt).

```
# Visualize the 204 * 1 weights for Imaginary Dataset
show_chanWeights(weights_img)
```




```
# Visualize the 204 * 1 weights for Overt Dataset
show_chanWeights(weights_overt)
```



To demonstrate that I can successfully implement the 2^{nd} level cross-validation to optimize the regularization parameter λ for each 1^{st} level fold, intermediate results during Two-Level Cross Validation are shown below.

How the feedback I provided to my peers influences my next steps: I helped one member identify one small error during discussion, he mistakenly thought we should also find the best λ for the first level cross-validation. Instead, what we should do for the 1st level is to optimize the classification accuracy.

[13]:
```
# Imaginary Dataset
# Extract the decision statistics as "ds_img"
# Extract the accuracy for each 1st level fold as "acc_img"
# The last accuracy value is the average accuracy
#                          (total cross-validated)
# Extract the 204 * 1 weights vector as "weights_img"
ds_img, acc_img, weights_img = BCI_Decode(img1, img2)
print(acc_img)
```

```
Current first level fold index is 1
Current optimal regularization parameter is 1.00
Current first level fold index is 2
Current optimal regularization parameter is 0.10
Current first level fold index is 3
Current optimal regularization parameter is 0.01
Current first level fold index is 4
Current optimal regularization parameter is 1.00
Current first level fold index is 5
Current optimal regularization parameter is 0.01
Current first level fold index is 6
Current optimal regularization parameter is 0.10
[0.825  0.85   0.9    0.95   0.825  0.975  0.8875]
```

How the exchange of information and ideas with my peers influences my next steps: I also learned a lot about how to correctly implement the [redacted] L2 regularization [redacted] (Ridge) given $\min_{W,C,\xi} \sum \xi_i + \lambda \cdot W^T W$, and how to implement the 2nd-level cross-validation in a convenient way, but I still prefer to extract indices from X and y, the most basic approach.