All code can be seen on the git repository

Prep tasks:

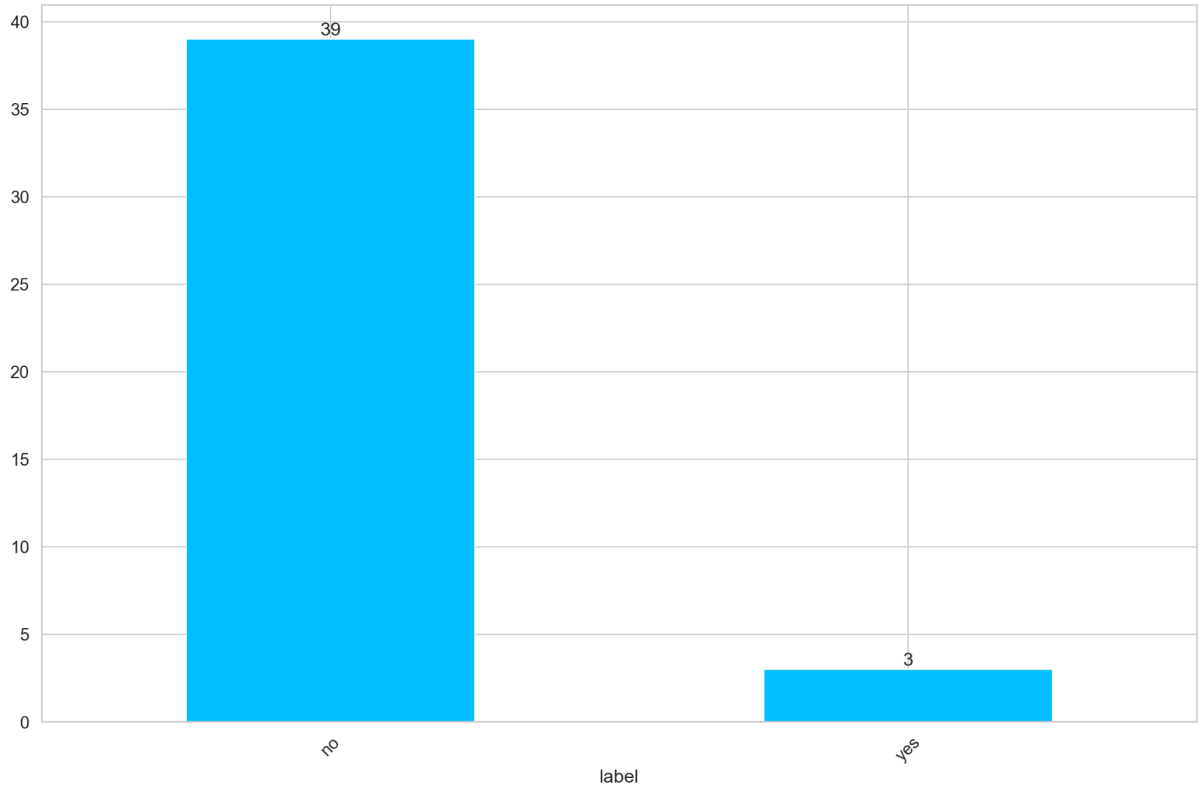
1) We have chosen to work with 1 second time collections in order to maximize the number of data points to increase the odds of producing a good classifier.
2) We've looked over the videos to single out where the interruptions occur in the data collected by watching when we tap the sensors on video and finding the marker in the data.
3) We've then marked the start and end of each interruption on the data.
4) We wrote a python script (seen here) to find the minimum number of segments per data type (ECG and EDA) so that each interruption will have its own segment and because of one interruption being 18 seconds we have concluded the minimum segmentation to be 20 (segments of 20 time units) for the EDA and 25 (segments of 25 time units) for the ECG (after having to crop a bit of the start of the long interruption).
5) We created the segments and labeled them accordingly.

TS tasks:

1) We took the data prepared and found each segment's: mean, standard deviation and median.
2) We displayed the number of yes's and the number of no's as seen below.
3) We've trained 3 types of SVM classifiers using different common kernels: linear, rbf and sigmoid.
4) For each of these we trained them on both Vlad's experiments and Lior's experiments and then tested them on each of the others' experiments respectively. (results displayed below). Using the features extracted on the previous task.
5) We built TSC using three different algorithms: Rocket Classifier using multirocket, WEASEL and shapelet transform classifier (all from sktime library).
6) We used each of above algorithms to train 3 models (9 models in total):
    a) A model that is trained only on Lior's 1st experiment data.
    b) A model that is trained on Lior's 1st and 2nd experiments data.
    c) A model that is trained on Lior's 1st and 2nd experiments data and Vlad's 1st experiment data.
7) All models were tested on Vlad's experiment number 2.
8) We chose these experiments for training because they had more interruptions compared to the others and were always tested on Vlad's 2nd experiment because this experiment had 2 interruptions compared to 1 interruption of the 1st experiment.
9) All results shown below.
10) We weren't able to run any classifiers on the data from the ECG because that data from the ECG was missing and had mostly empty chunks for the hr column which was the only relevant column in the data.
11) For the EDA classifiers Eda Arousal mean was used as the feature.

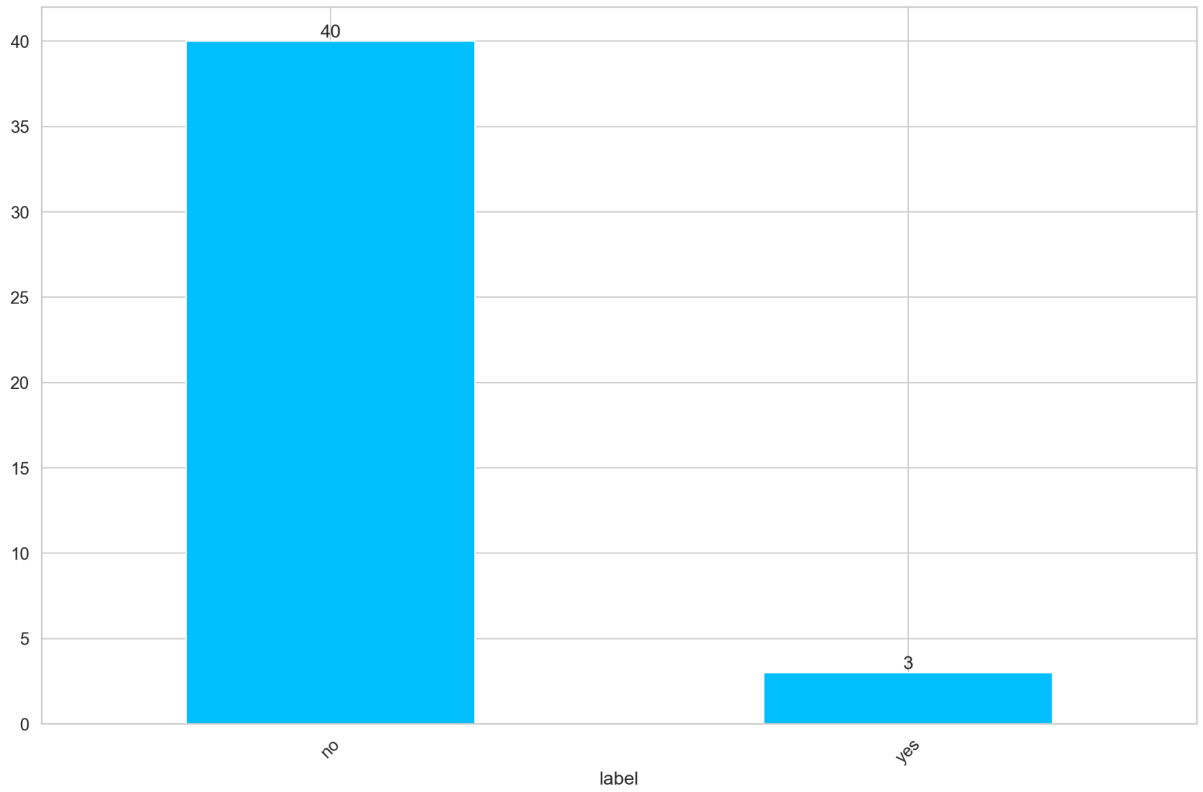
Lior’s experiment 1:

Start segment: 6
End segment: 47
Total: 42

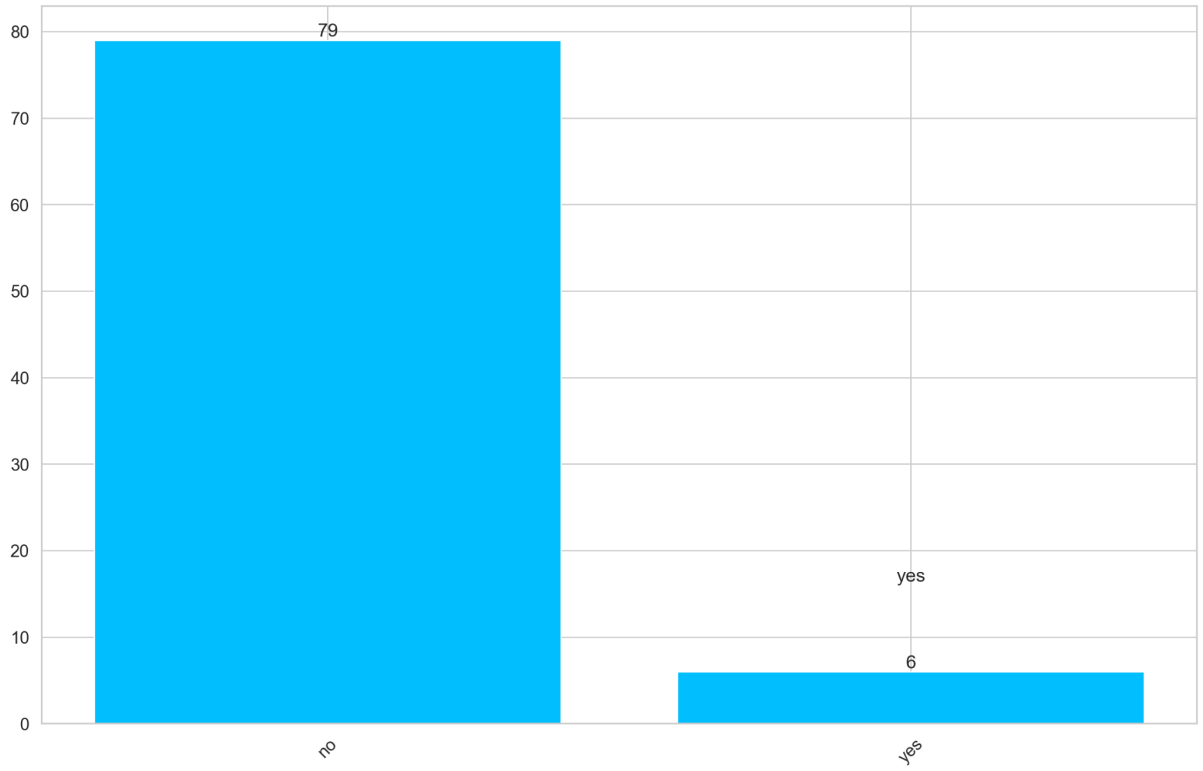


Lior’s experiment 1:

Start segment: 5
End segment: 47
Total: 43

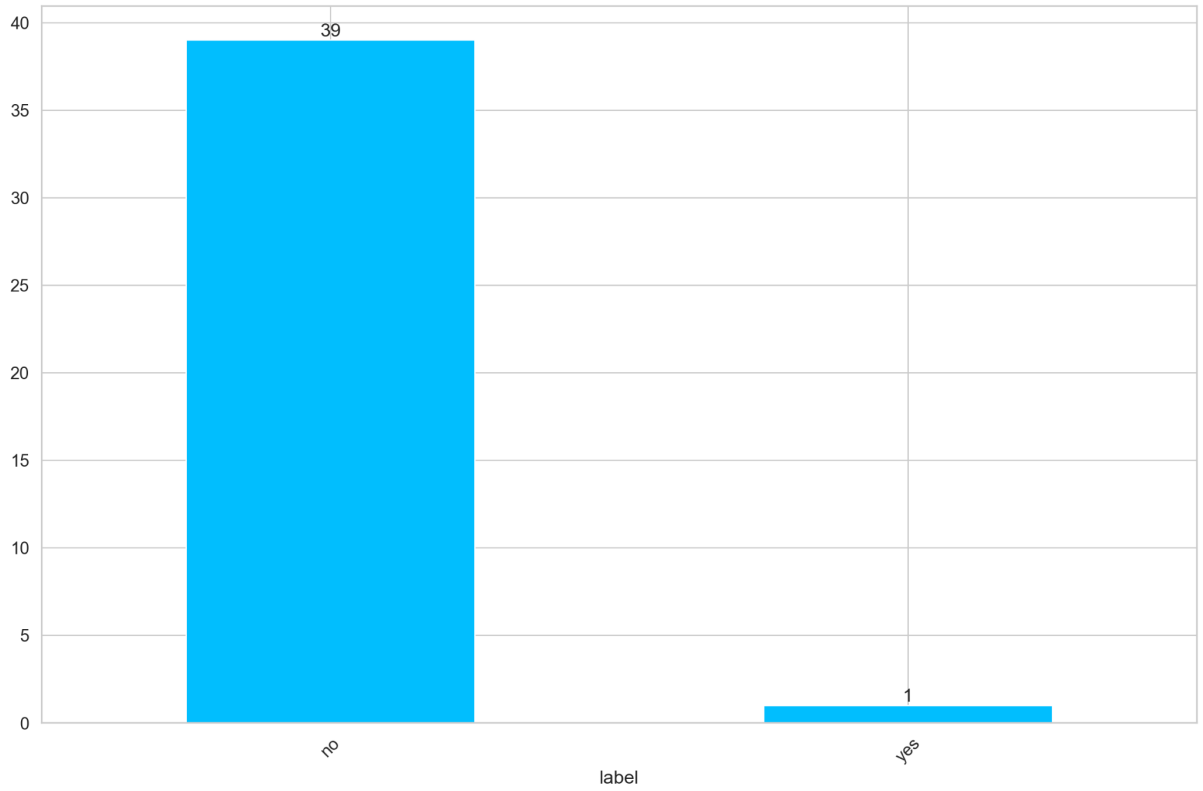


Lior’s experiments combined:



Vlad’s experiment 1:

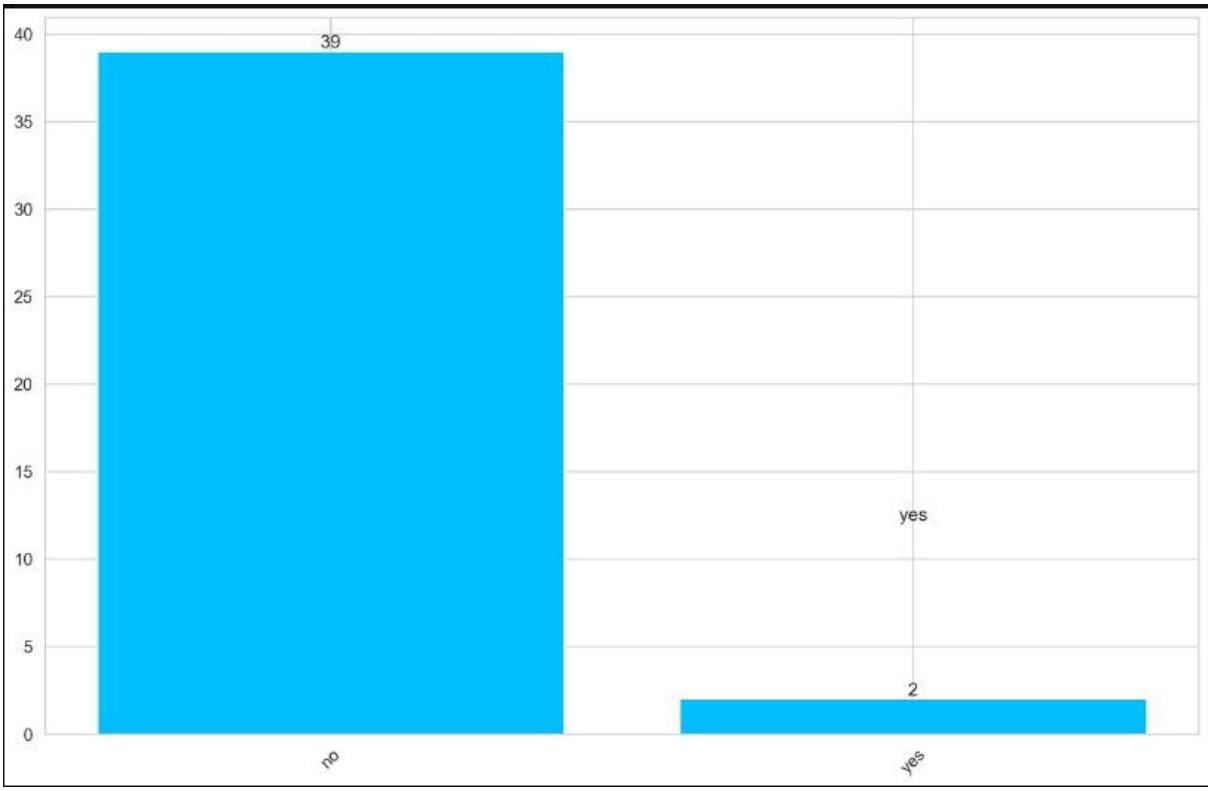
Start segment: 19
End segment: 58
Total: 40



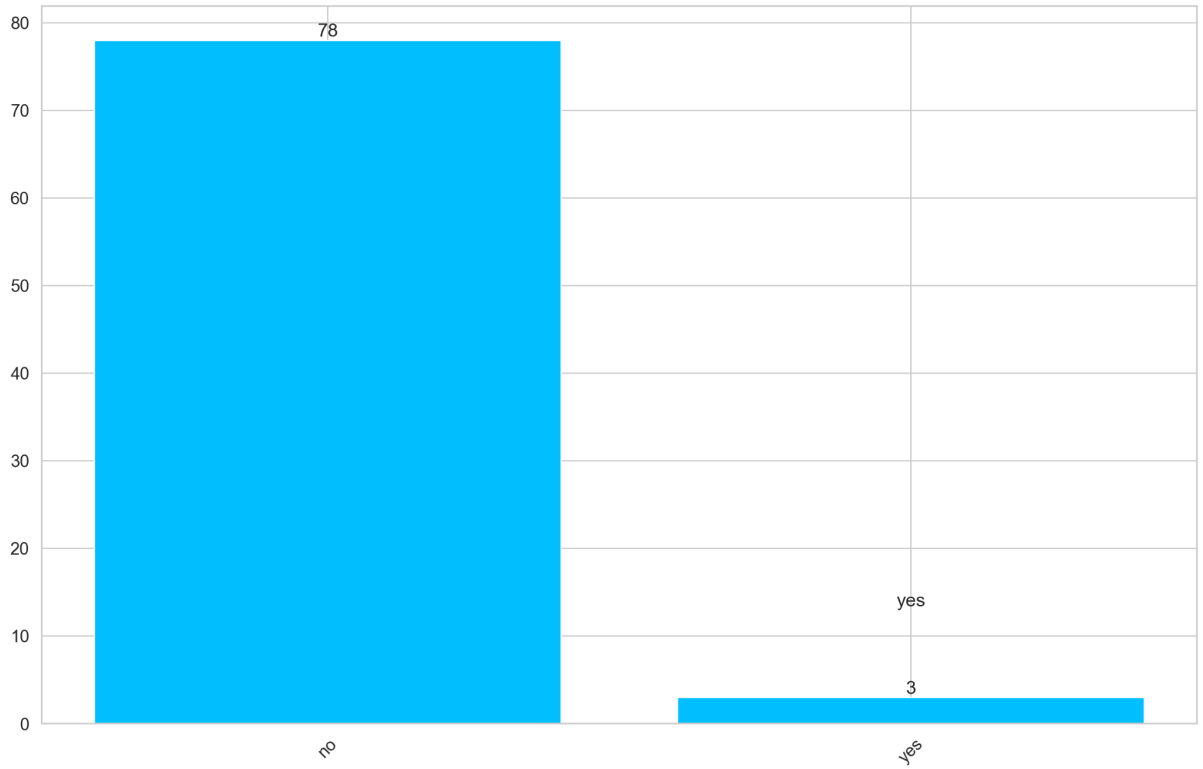
Vlad’s experiment 2:

Always used for testing

Start segment: 3
End segment: 43
Total: 41



Vlad’s experiments combined:



Actual labels (for Vlad experiment 2):

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])
```

Rocket:

Model trained on Lior’s experiment 1:

Predicted labels:

```
# Predict labels for the test data
y_pred = loaded_clf.predict(X_test_np)
# Print predictions
y_pred
```

```
array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0])
```

Report:

```
Accuracy: 87.80%
              precision    recall  f1-score   support

           0       1.00      0.87      0.93        39
           1       0.29      1.00      0.44         2

    accuracy                           0.88        41
   macro avg       0.64      0.94      0.69        41
weighted avg       0.97      0.88      0.91        41
```

Model trained on Lior’s 1st and 2nd experiment:

Predicted labels:

```
# Predict labels for the test data
y_pred1 = loaded_clf1.predict(X_test_np)
# Print predictions
y_pred1
```

```
array([1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0])
```

Report:

```
Accuracy: 80.49%
              precision    recall  f1-score   support

           0       1.00      0.79      0.89        39
           1       0.20      1.00      0.33         2

    accuracy                           0.80        41
   macro avg       0.60      0.90      0.61        41
weighted avg       0.96      0.80      0.86        41
```

Model trained on Lior’s 1st and 2nd experiment and Vlad’s 1st experiment:

Predicted labels:

```
# Predict labels for the test data
y_pred2 = loaded_clf2.predict(X_test_np)
# Print predictions
y_pred2
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Report:

```
Accuracy: 95.12%
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        39
           1       0.00      0.00      0.00         2

    accuracy                           0.95        41
   macro avg       0.48      0.50      0.49        41
weighted avg       0.90      0.95      0.93        41
```

WEASEL:

Model trained on Lior's experiment 1:

Predicted labels:

```
# Predict labels for the test data
y_pred = loaded_clf.predict(X_test_np)
# Print predictions
y_pred
```

```
array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0])
```

Report:

```
Accuracy: 85.37%
              precision    recall  f1-score   support

           0       0.95      0.90      0.92        39
           1       0.00      0.00      0.00         2

    accuracy                           0.85        41
   macro avg       0.47      0.45      0.46        41
weighted avg       0.90      0.85      0.88        41
```

Model trained on Lior’s 1st and 2nd experiment:

Predicted labels:

```
# Predict labels for the test data
y_pred1 = loaded_clf1.predict(X_test_np)
# Print predictions
y_pred1
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0])
```

Report:

```
Accuracy: 90.24%
              precision    recall  f1-score   support

           0       0.95      0.95      0.95        39
           1       0.00      0.00      0.00         2

    accuracy                           0.90        41
   macro avg       0.47      0.47      0.47        41
weighted avg       0.90      0.90      0.90        41
```

Model trained on Lior’s 1st and 2nd experiment and Vlad’s 1st experiment:

Predicted labels:

```
# Predict labels for the test data
y_pred2 = loaded_clf2.predict(X_test_np)
# Print predictions
y_pred2
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Report:

```
Accuracy: 95.12%
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        39
           1       0.00      0.00      0.00         2

    accuracy                           0.95        41
   macro avg       0.48      0.50      0.49        41
weighted avg       0.90      0.95      0.93        41
```

Shapelet:

Model trained on Lior's experiment 1:

Predicted labels:

```
# Predict labels for the test data
y_pred = loaded_clf.predict(X_test_np)
# Print predictions
y_pred
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0])
```

Report:

```
Accuracy: 87.80%
              precision    recall  f1-score   support

           0       0.95      0.92      0.94        39
           1       0.00      0.00      0.00         2

    accuracy                           0.88        41
   macro avg       0.47      0.46      0.47        41
weighted avg       0.90      0.88      0.89        41
```

Model trained on Lior’s 1st and 2nd experiment:

Predicted labels:

```
# Predict labels for the test data
y_pred1 = loaded_clf1.predict(X_test_np)
# Print predictions
y_pred1
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0])
```

Report:

```
Accuracy: 95.12%
              precision    recall  f1-score   support

           0       1.00      0.95      0.97        39
           1       0.50      1.00      0.67         2

    accuracy                           0.95        41
   macro avg       0.75      0.97      0.82        41
weighted avg       0.98      0.95      0.96        41
```

Model trained on Lior’s 1st and 2nd experiment and Vlad’s 1st experiment:

Predicted labels:

```
# Predict labels for the test data
y_pred2 = loaded_clf2.predict(X_test_np)
# Print predictions
y_pred2
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0])
```

Report:

```
Accuracy: 87.80%
              precision    recall  f1-score   support

           0       0.95      0.92      0.94        39
           1       0.00      0.00      0.00         2

    accuracy                           0.88        41
   macro avg       0.47      0.46      0.47        41
weighted avg       0.90      0.88      0.89        41
```

Classifier comparison SVM Vs. TSC:

The best results from the TSC were as follows (all TSC were tested on Vlad's 2nd experiment):

Rocket trained on Lior exp 1:

Accuracy: 87.80%

	precision	recall	f1-score	support
0	1.00	0.87	0.93	39
1	0.29	1.00	0.44	2
accuracy			0.88	41
macro avg	0.64	0.94	0.69	41
weighted avg	0.97	0.88	0.91	41

WEASEL derived from Lior exp 1 and 2:

Accuracy: 90.24%

	precision	recall	f1-score	support
0	0.95	0.95	0.95	39
1	0.00	0.00	0.00	2
accuracy			0.90	41
macro avg	0.47	0.47	0.47	41
weighted avg	0.90	0.90	0.90	41

And shapelet trained on Lior exp 1 and 2:

```
Accuracy: 95.12%
              precision    recall  f1-score   support

           0       1.00      0.95      0.97        39
           1       0.50      1.00      0.67         2

    accuracy                           0.95        41
   macro avg       0.75      0.97      0.82        41
weighted avg       0.98      0.95      0.96        41
```

While every option of SVM yielded only predictions of 0's:

Linear SVM trained on both of Vlad's experiments and tested on each of Lior's experiments respectively:

Tested on Lior's exp 1:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
Accuracy: 97.50%
              precision    recall  f1-score   support

           0       0.97      1.00      0.99        39
           1       0.00      0.00      0.00         1

    accuracy                           0.97        40
   macro avg       0.49      0.50      0.49        40
weighted avg       0.95      0.97      0.96        40
```

Tested on Lior's exp 2:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
Accuracy: 95.12%
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        39
           1       0.00      0.00      0.00         2

    accuracy                           0.95        41
   macro avg       0.48      0.50      0.49        41
weighted avg       0.90      0.95      0.93        41
```

We've also tried RBF and Sigmoid kernels to the same effect and testing Vlad's experiments on models trained by Lior's experiments yielded the same results (i.e., the models predicted every label to be 0).

Discussing results:

All the SVM models were the same and produced all predictions as 0, so while having a 95.12% accuracy they produced a 0% recall for the surprises. Therefore they are not very good, but this was to be expected seeing as we had very little segments labeled as 1 compared to segments labeled as 1.
Surprisingly the Time Series Classifiers (TSC) performed better as can be seen in these examples:
Rocked classifier trained on Lior's 1st experiment:

```
Accuracy: 87.80%
              precision    recall  f1-score   support

           0       1.00      0.87      0.93        39
           1       0.29      1.00      0.44         2

    accuracy                           0.88        41
   macro avg       0.64      0.94      0.69        41
weighted avg       0.97      0.88      0.91        41
```

Produced a good accuracy while predicting both the 1 labels but had a low f1 and precisions scores.

And the best performing classifier we could produce was the Shaplet transform classifier trained on both of Lior's experiments:

Accuracy: 95.12%

	precision	recall	f1-score	support
0	1.00	0.95	0.97	39
1	0.50	1.00	0.67	2
accuracy			0.95	41
macro avg	0.75	0.97	0.82	41
weighted avg	0.98	0.95	0.96	41

Had a good accuracy, managed to find both of the 1 labels and had the best f1 and precision scores we could produce.

Everytime Vlad's experiment 1 data was introduced to the training the classifier predicted everything to be 0's.

The classifiers are not very reliable but again this was to be expected because of the following reasons:

1. The way the experiment was conducted, we were both the programmers and the subjects of the experiments, therefore for the most part we knew about how all the interruptions will look like and how the robot program is going to run.
    a. We mitigated these factors by:
    b. Creating interruptions for each other without the other seeing them.
    c. Introducing randomness in every parameter possible. The number of interruptions, what interruptions were to occur, at what time and so on were all chosen randomly.
    d. Two interruptions were created by another person who is not one of us, so neither one of us knew what they would look like.
2. The number of 0's is overwhelmingly more than the number of 1's so the classifier's had a pretty easy time classifying something as 0 while a hard time classifying something as 1.
3. In general the amount of data was very low around 40 segments of data per experiment. (156 segments in total from all 4 experiments).
4. Hard to tell what surprise actually looks like and if when interruption happened were we actually surprised or if we were surprised during other times not very clear from the data.

Suggestions for improvement:

1) Increase the volume of available data by incorporating additional artificial data designed to simulate interruptions, or conduct further experimental trials with subjects to generate additional data.

2) With respect to the ECG sensors, it is our conjecture that the absence of data may be attributed to the absence of chest shaving prior to their utilization.
3) It is advisable to minimize the duration of interruptions, as interruptions exceeding the minimum time duration of a segment (n time units) result in excessively extended interruptions.