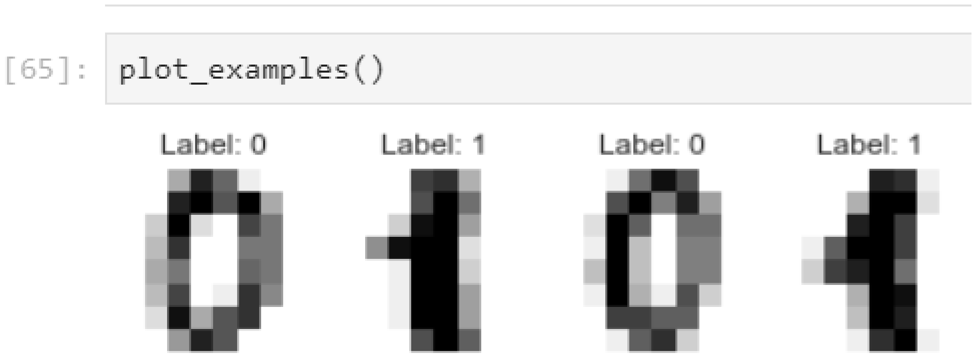
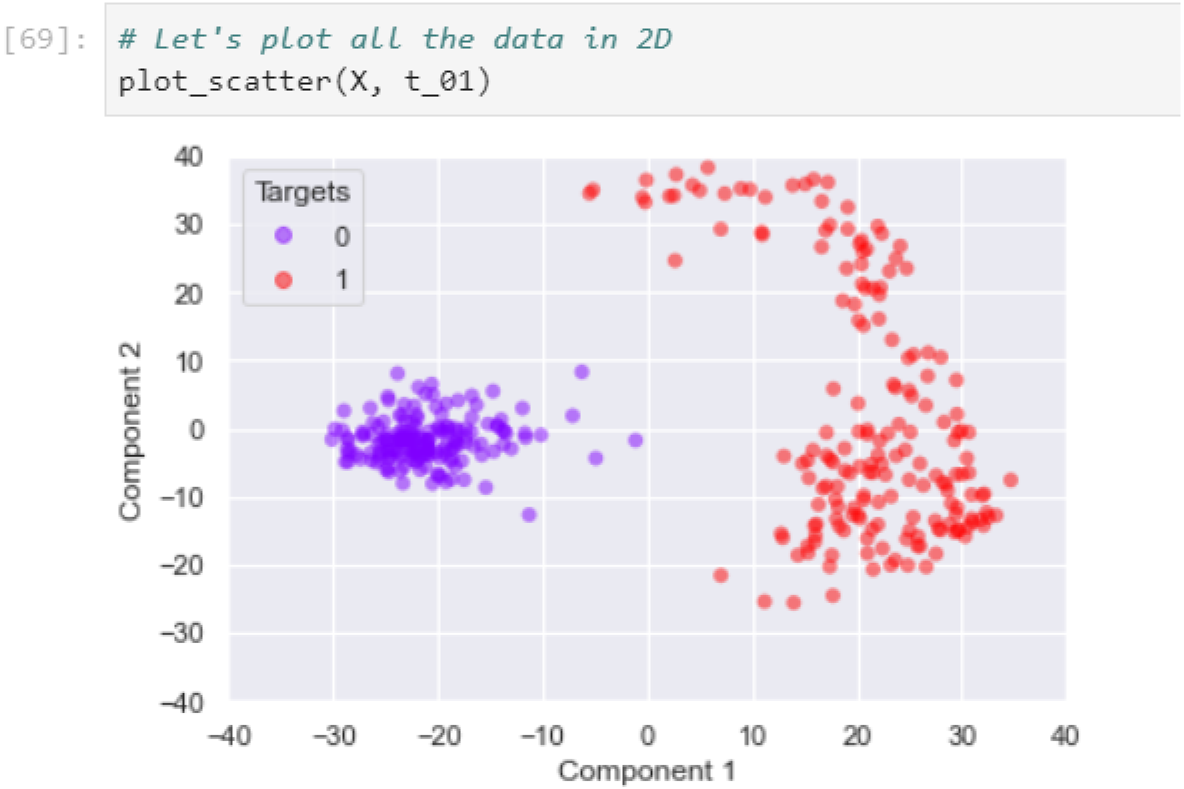
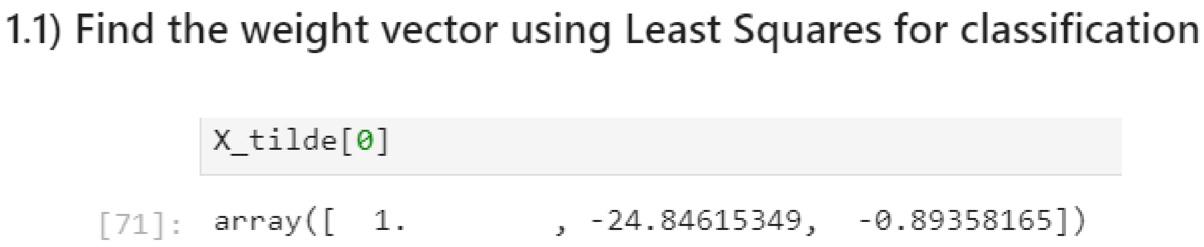
Week 7



```
[65]: plot_examples()
```



```
[69]: # Let's plot all the data in 2D
      plot_scatter(X, t_01)
```



1.1) Find the weight vector using Least Squares for classification

```
X_tilde[0]

[71]: array([  1.        , -24.84615349,  -0.89358165])
```

```
W_tilde
```

```
[72]: array([[0.50555556, 0.02182423, 0.00478603]])
```

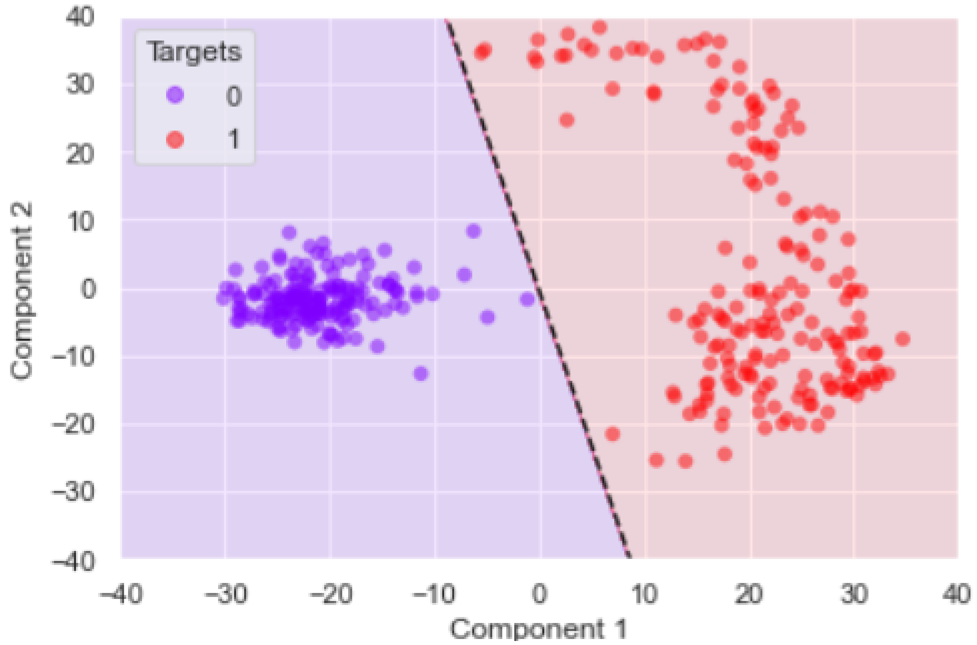
1.2) Create class predictions using the weight vector

```
[74]: preds = predict(W_tilde, X_tilde)
```

```
[75]: np.array_equal(t_01, preds)
```

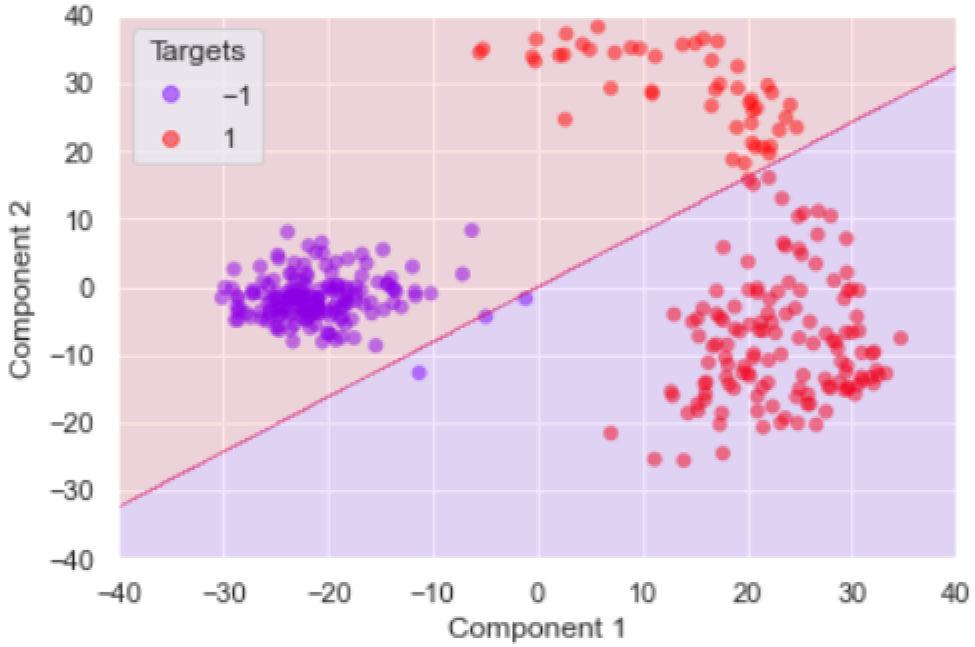
```
[75]: True
```

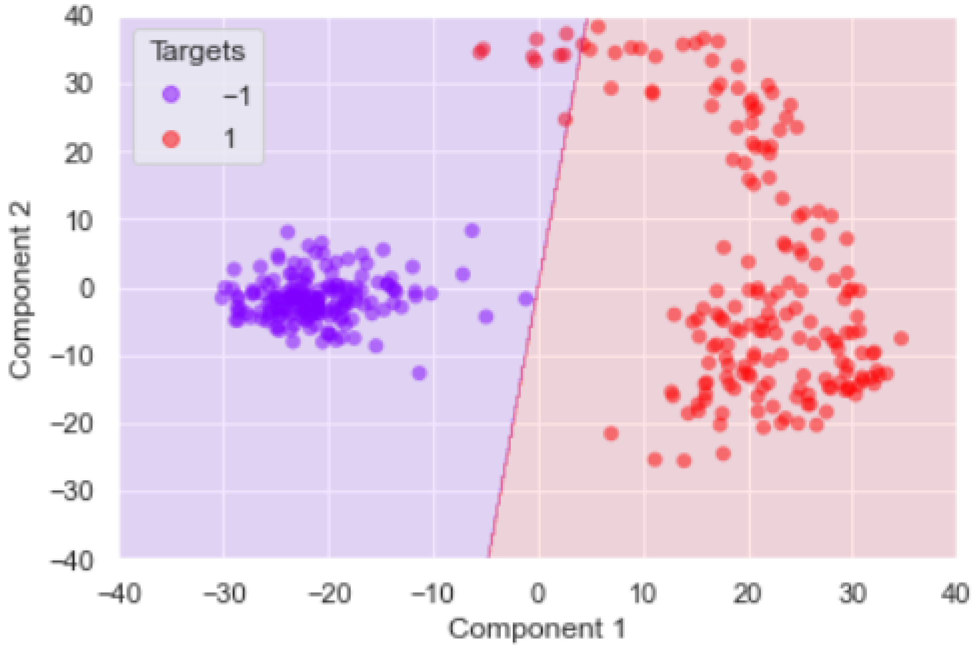
1.3) Plot the decision boundary



2) Perceptron

```
W_0 = [[ 0.55540482 -0.52491756  0.64855707]]
```



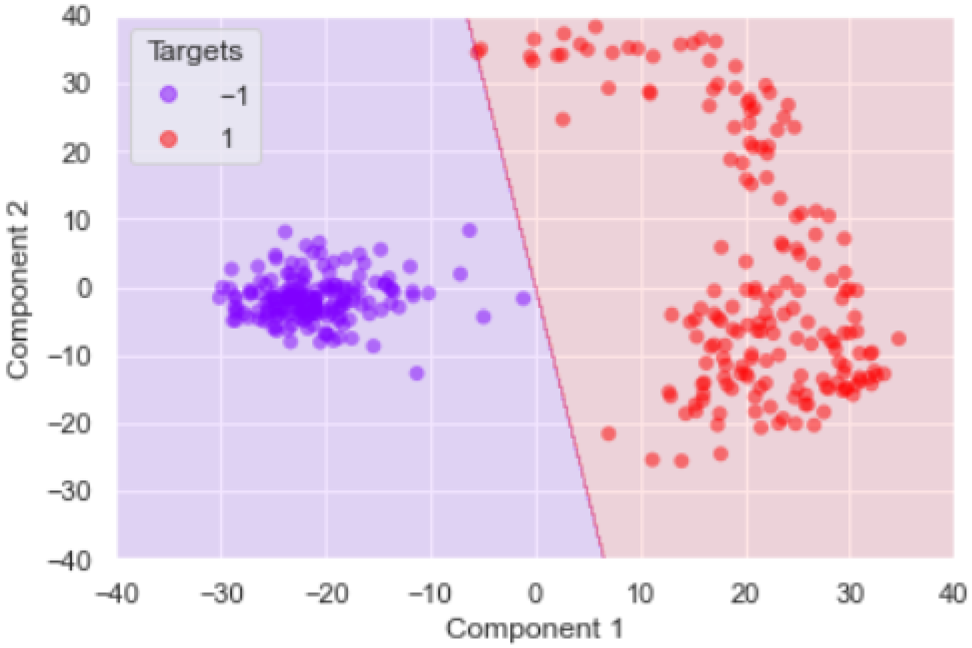


2.2) Perform class-predictions

```
np.array_equal(preds, t_11)
```

```
[83]: True
```

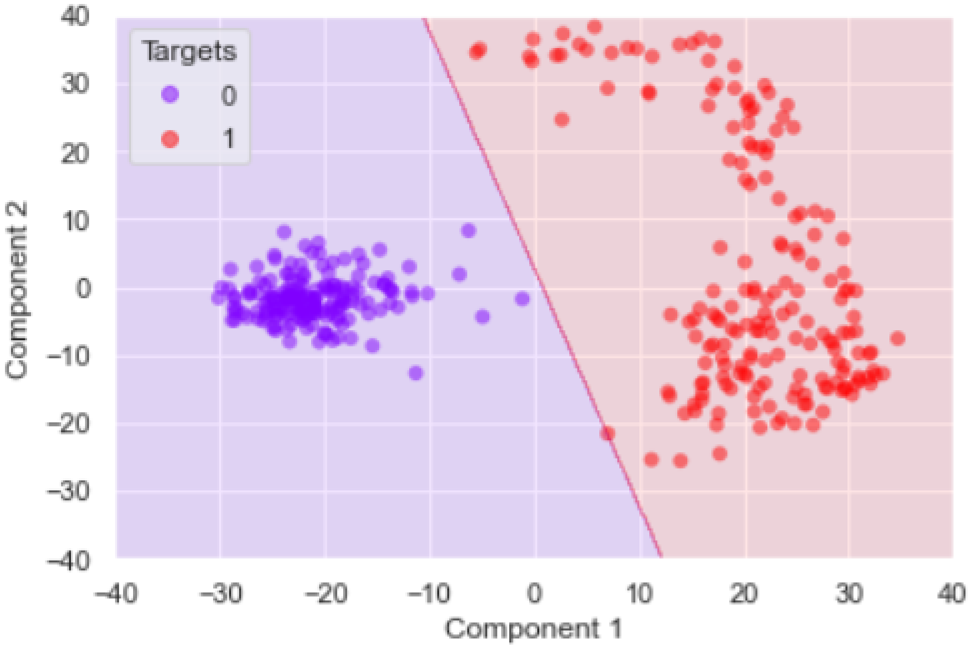
2.3) Plot the decision boundary



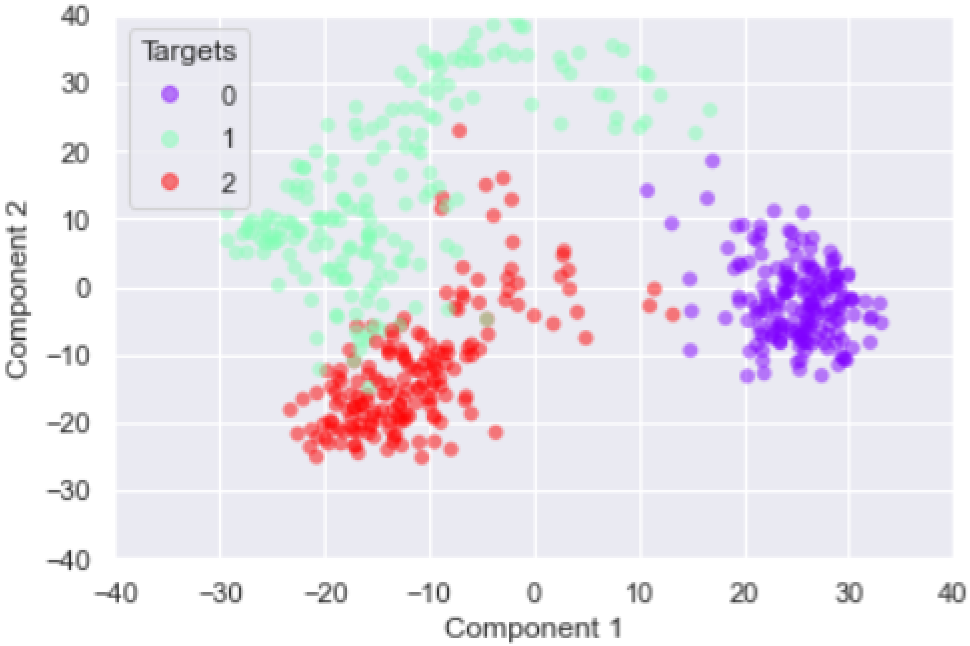
3.2) Perform class-predictions

```
np.array_equal(preds, t)
```

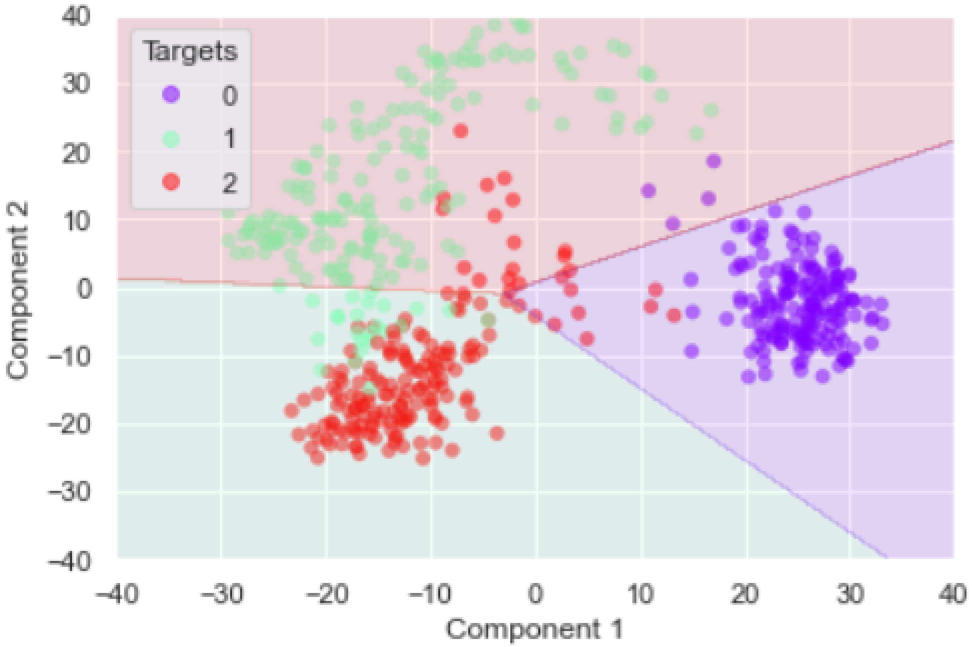
```
True
```



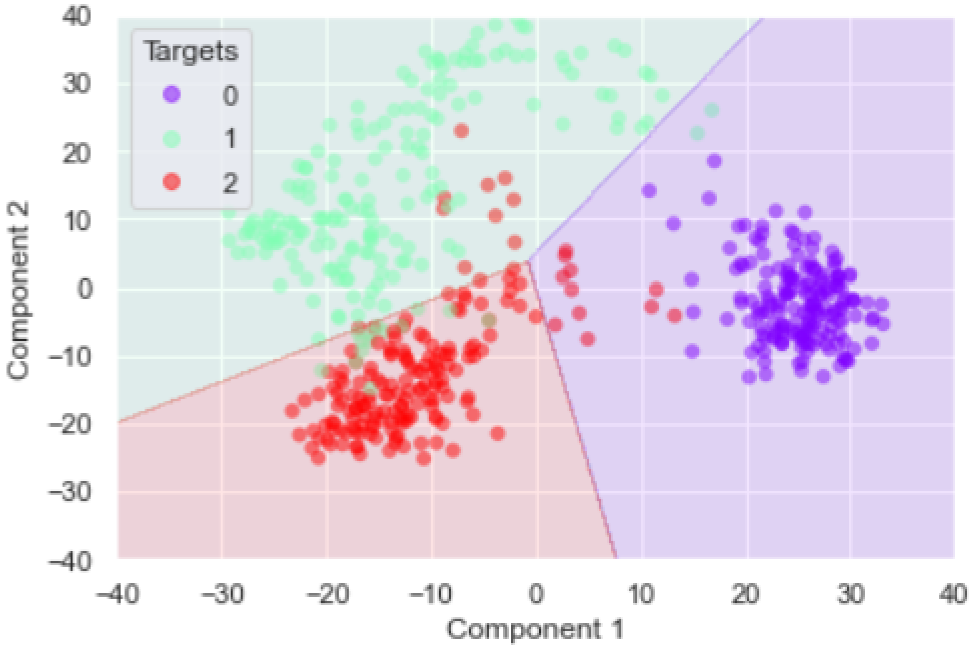
4) Multi-class logistic regression



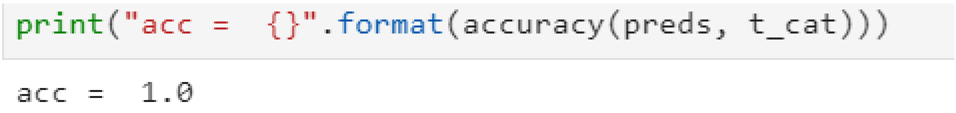
```
epoch = 0
acc =  0.40037243947858475
```



```
epoch = 49
acc =  0.931098696461825
```



5) Multi-class logistic regression on original data representation



```
print("acc =  {}".format(accuracy(preds, t_cat)))
```

```
acc =  1.0
```