1.2.a) Refer to the documentation, what is the functionality of the tol parameter in the Perceptron class? (2 marks)

*tol* represents the improvement in the model's error. The model will return a solution when the improvement in error between two iterations is smaller than *tol*, i.e. the improvement is considered small and a good solution has been found.

1.2.b) If we set max_iter=5000 and tol=1e-3 (the rest as default), does this guarantee that the algorithm will pass over the training data 5000 times? If not, ensure that the algorithm will pass over the training data 5000 times? (2 marks)

No it does not. To ensure that the algorithm will pass over the training data 5000 times, set *tol* to 0.

1.2.c) How can we set the weights of the model to a certain value? (2 marks)

First fit the model. Then modify the *coef_* and *intercept_* attribute manually. The *coef_* attribute sets the weight for each feature, and the *intercept_* sets the bias term.

1.2.d) How close is the performance (through confusion matrix) of your NumPy implementation in comparison to the existing modules in the scikit-learn library? (2 marks)

My implementation's confusion matrix:

```
[[8. 0.]
 [3. 9.]]
```

Scikit-learn library's confusion matrix:

```
[[8. 0.]
 [2. 10.]]
```

My implementation misclassified one more point in class +1 than the library's module.

2.1.a) When we input a singular matrix, the function linalg.inv often returns an error message. In your fit_LinRegr(X_train, y_train) implementation, is your input to the function linalg.inv a singular matrix? Explain why. (2 marks)

The input is indeed a singular matrix because its determinant is 0, making it non-invertible.

2.1.b) As you are using linalg.inv for matrix inversion, report the output message when running the function subtestFn(). We note that inputting a singular matrix to linalg.inv sometimes does not yield an error due to numerical issue. (1 marks)

It outputs ERROR.

2.1.c) Replace the function linalg.inv with linalg.pinv, you should get the model’s weight and the “NO ERROR” message after running the function subtestFn(). Explain the difference between linalg.inv and linalg.pinv, and report the model’s weight. (2 marks)

Linalg.pinv performs pseudo inverse, which can handle both singular and non-singular matrices. Linalg.inv is used for non-singular matrices.

The weight is: [1.04360964e-14, 2.00000000e-01, 4.00000000e-01]