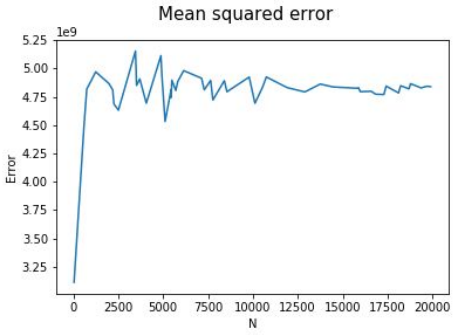
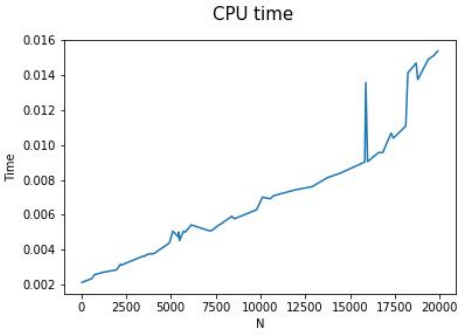
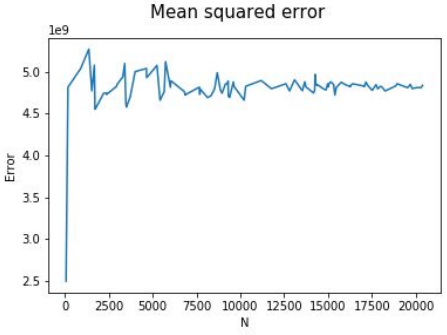
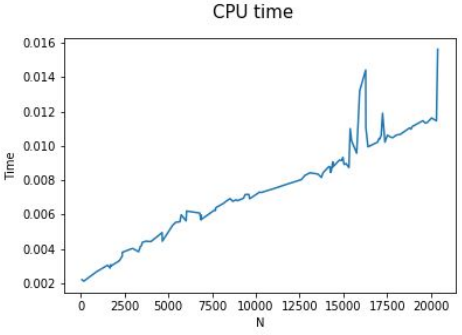
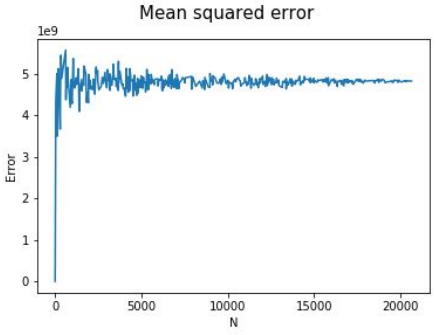
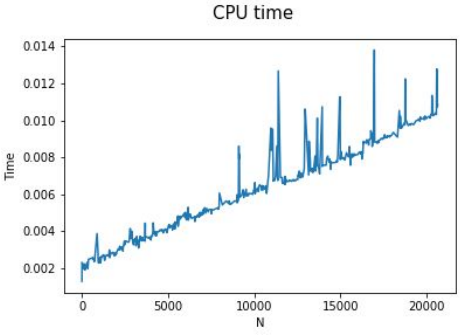
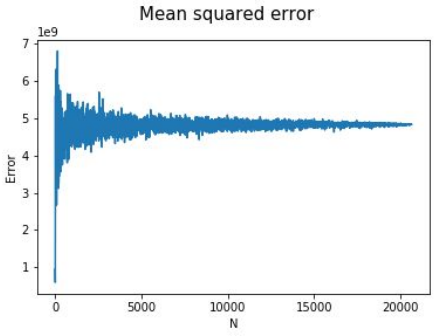
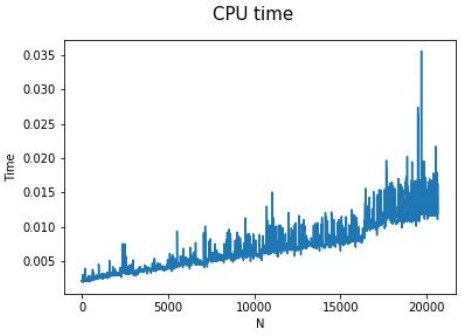
MIIS - Ana Mestre

LAB 1: Programming task

Regression task

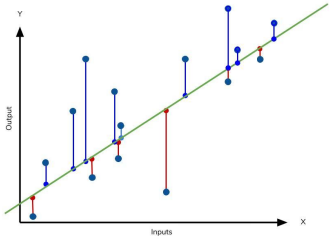
Regression dataset chosen: **cadata** with a data size of 20,640 and 8 features.

Iterations	Mean squared error	CPU time
50		
100		
500		
5000		

Observations

Taking a look at these results, we can see that regardless of how many samples we take, we achieve quite similar results.

Regarding the **mean squared error**, let's first remember that it measures the average squared difference between the estimated values and the actual value. Thus, the smaller the mean squared error, the closer we are to finding the line that best fits the model.



$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)}^{2}$$

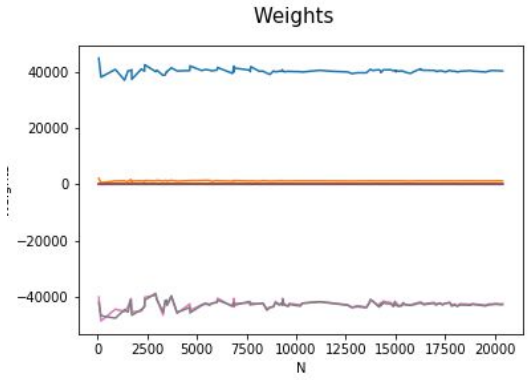
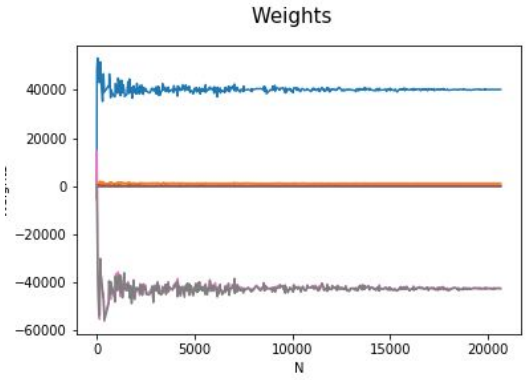
The square of the difference between actual and predicted

If we observe the results shown in the table above, we can see that the smaller the sample, the less stable the mse is. At the beginning this mse is usually lower. For large sized samples, the error tends to be quite similar. With this in mind, we can deduce that on the one hand, when there are fewer points (N small), the distance between them can either be close or distant. For this reason, the MSE can either be low or high. On the other hand, with a large value for N, the MSE reaches a point where it's very similar to the MSE obtained for N + 1 within the same dataset. This happens because when we are working with points inside the same dataset, these points are usually in the same value range. In addition, the larger the size of the dataset, the less important every point is individually. Therefore, outliers have less impact on our final MSE.

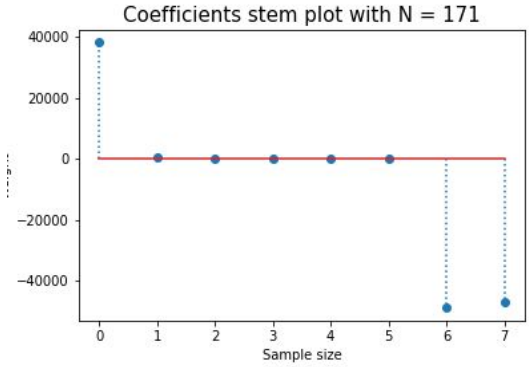
Despite the difference for smaller or larger datasets, we can observe that the overall mean squared error for this case is pretty low. Consequently, our prediction was quite accurate.

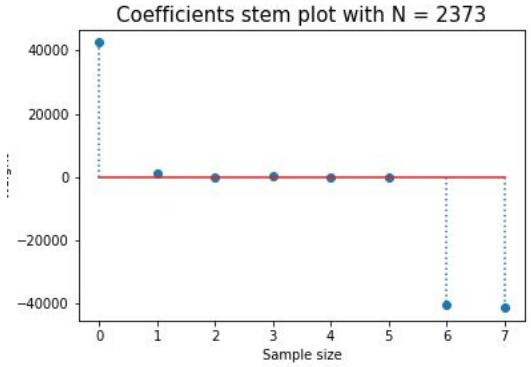
Concerning the **CPU time**, as it was expected, the larger the sample, the more time it takes to compute our results. Although we can see some unusual spikes, the increasing tendency is revealed on the plots.

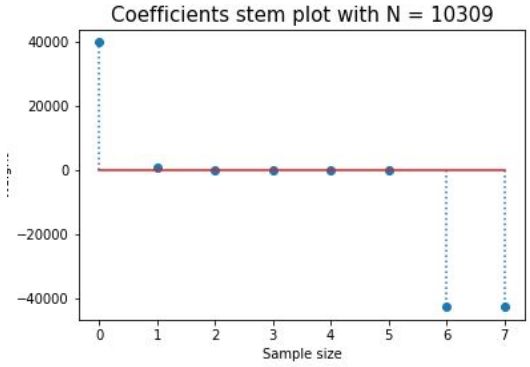
Looking into the **weights** results shown in the following table, we can observe in both plots (for 100 iterations and for 500 iterations) that the same behaviour as the MSE is happening here, the weights stabilize when the number of samples is larger.

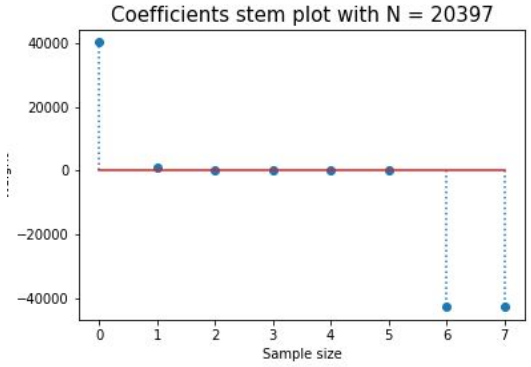
Weights plot for 100 iterations	Weights plot for 500 iterations
	

Looking at these previous general plots and at the stem plots below, we can see that for this case, our weights stabilize more or less between N = 10.000 and N = 15.000 depending on the feature.





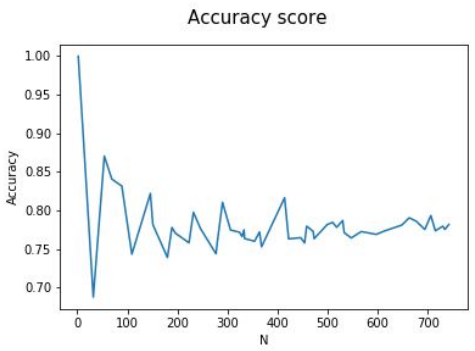
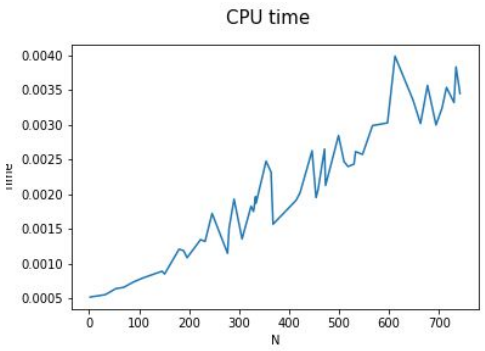
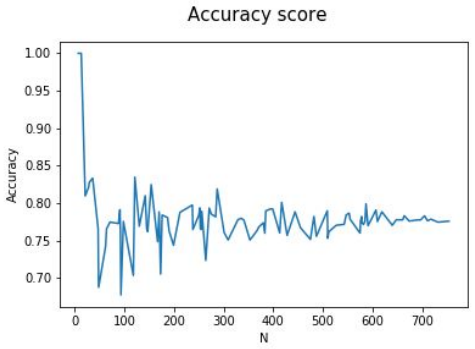
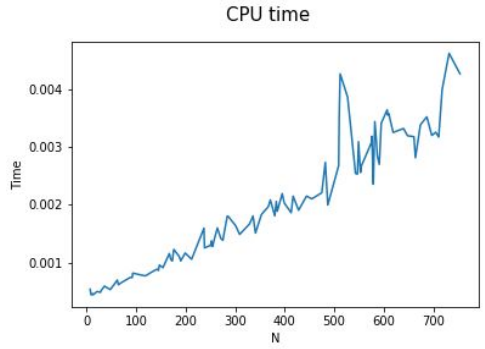
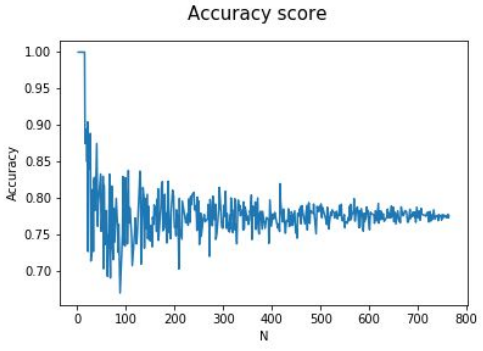
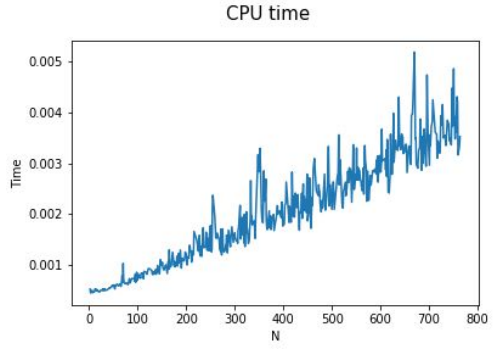




Classification task

Choose a classification dataset and apply logistic regression. Repeat the previous four steps using as error the mean accuracy.

Classification dataset chosen: **diabetes** with a data size of 768 and 8 features and 2 classes.

Iterat.	Accuracy	CPU time
50		
100		
500		

Observations

With regards to the **accuracy score** for our predictions, we can see that the values obtained are quite high since they are between 0.75 and 0.8 approximately. Also, the same conclusion retrieved on the first part of this assignment can be applied for this case. When the number of N gets larger, the accuracy score stabilizes. Since this dataset is smaller, this statement is not as obvious as before but it can still be appreciated.

Concerning the **CPU time**, as it was expected here as well, the larger the amount of data we have to predict, the more time it takes to do so. This tendency is clear on the CPU time plots above.

Taking a look at the weights shown in the general plots and the stem plots below, we can see the stabilization for weights as well. Although in this case, we can observe that there is a feature that doesn't stabilize as much as the other ones (the one shown in the pink line). We could deduce that this feature (feature 6) stabilizes a little at N = 700 approximately but it is not clear. The blue feature (number 0) stabilizes more than the pink one but its variety of values also highlights, for this case the weights maintain the same at approximately N = 600. Moreover, for the other six features, the stabilization is at approximately N = 500.

Weights plot for 100 iterations	**Weights plot for 500 iterations**