Introduction to Machine Learning
Assignment 2

Group 06
Lucas Pereira (s4507983) & Matteo Wohlrapp (s4974921)

September 28, 2021

Introduction

In this practical, we apply linear regression to a training data set composed of 500 different input vectors $x \in R^{25}$ and its corresponding target values $y \in R$. We use the data to calculate the optimal weight $w^* \in R^{25}$ which minimizes the MSE. We then apply w^* to a testing data set where we calculate the error. To find $w^* \in R$ we use different sizes P of training sets, ranging from $P = 30$ up to the complete training set $P = 500$ and later illustrate our findings in graphs.

Method

To implement linear regression we use MATLAB. First we load the given data which contains two sets of $500x25$ values, one used for training and the other one for testing. We also load two sets of 500 continuous targeting labels, again for training and testing. Each target value y corresponds to one input value $x \in R^{25}$.

For training, we then select P input vectors $x_1, .., x_P \in R^{25}$ from the $500x25$ training data set and its target values y to perform linear regression. The aim is to reduce the training error term for the Mean Squared Error MSE which is given by: $E_{train} = \frac{1}{P} \sum_{\mu=1}^{P} \frac{1}{2} * (w^* * x_{train}^{\mu} - y_{train}^{\mu})^2$. To achieve this, we calculate the gradient and set it equal to zero to find the minimum of the function. Rearranging the gradient leaves us with: $w^* = [X^T X]^{-1} X^T Y$ to calculate the optimal weight. Here, $w^* \in R^{25}$, $X \in R^{Px25}$ containing the first P members of the training vector and $Y \in R^{1xP}$ containing the target values. To calculate $[X^T X]^{-1}$ we first calculate $[X^T X]$ and then use the MATLAB function $pinv()$ on the result.

Once training is completed, we calculate the mean squared error for the testing set: $E_{test} = \frac{1}{1000} \sum_{\mu=1}^{500} *(w^* * x_{train}^{\mu} - y_{train}^{\mu})^2$ but this time without varying P.

As a bonus for more reliable curves, we calculate w^*, E_{train} and E_{test} for $P \in [10, 20, 30, .., 500]$, with $|[10, 20, 30, .., 500]| = 50$. Finally, we create a graph showing E_{train} and E_{test} for the varying P values. Furthermore we create bar graphs showing the values of w^* for different P. In the following section we discuss these results.

Results

In this section we present the results of our program executing the linear regression algorithm. This presentation will be done in two sections, one showing the graph for the MSE (Figure 1) for both E_{train} and E_{test} for each P value calculated, the other section consists of graphs showing the values of each weight in w^* for 6 different $P \in [30, 40, 50, 75, 100, 500]$ (Figure 2 - 7).

0.1. Mean squared error

0.1.1. Errors

In the following section we display the graphs for E_{train} and E_{test}. As a bonus, we increased the accuracy of the graph by testing with more values, with $P \in [10, 20, 30, .., 500]$, with $|[10, 20, 30, .., 500]| = 50$.

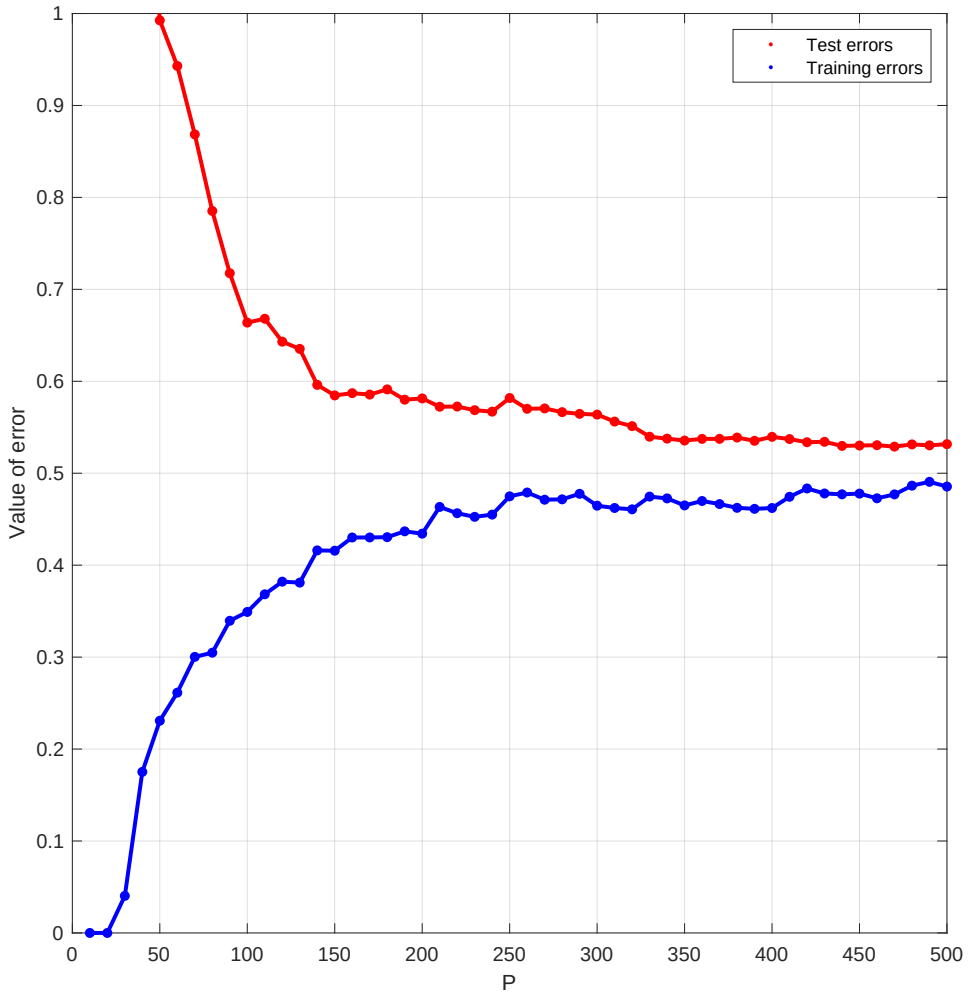

Figure 1: *Graph of the calculated E_{train} and E_{test} given a certain P*

0.2. Weights

In the following section we display the graphs for the weight w^* for $P \in [30, 40, 50, 75, 100, 500]$.

0.2.1. $P = 30$

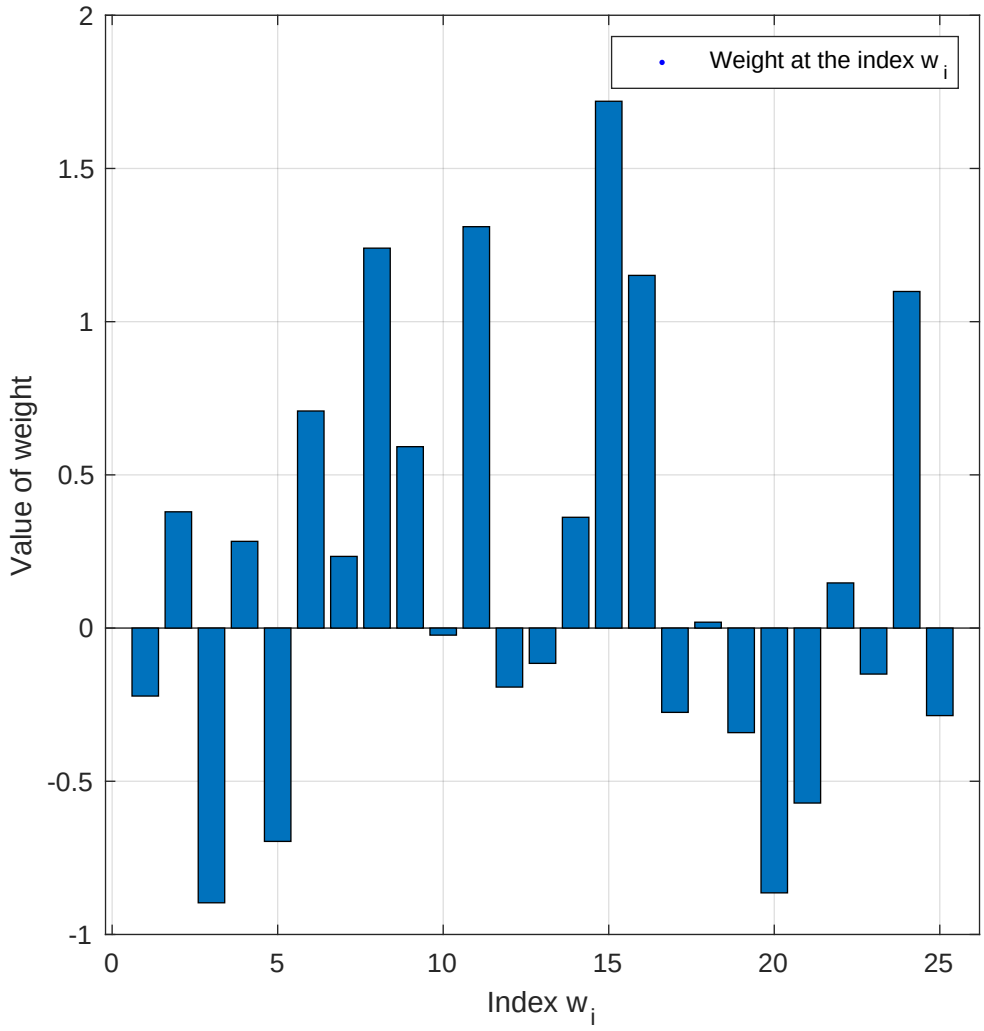

Figure 2: *Graph of the calculated weights, $P = 30$*

0.2.2. $P = 40$

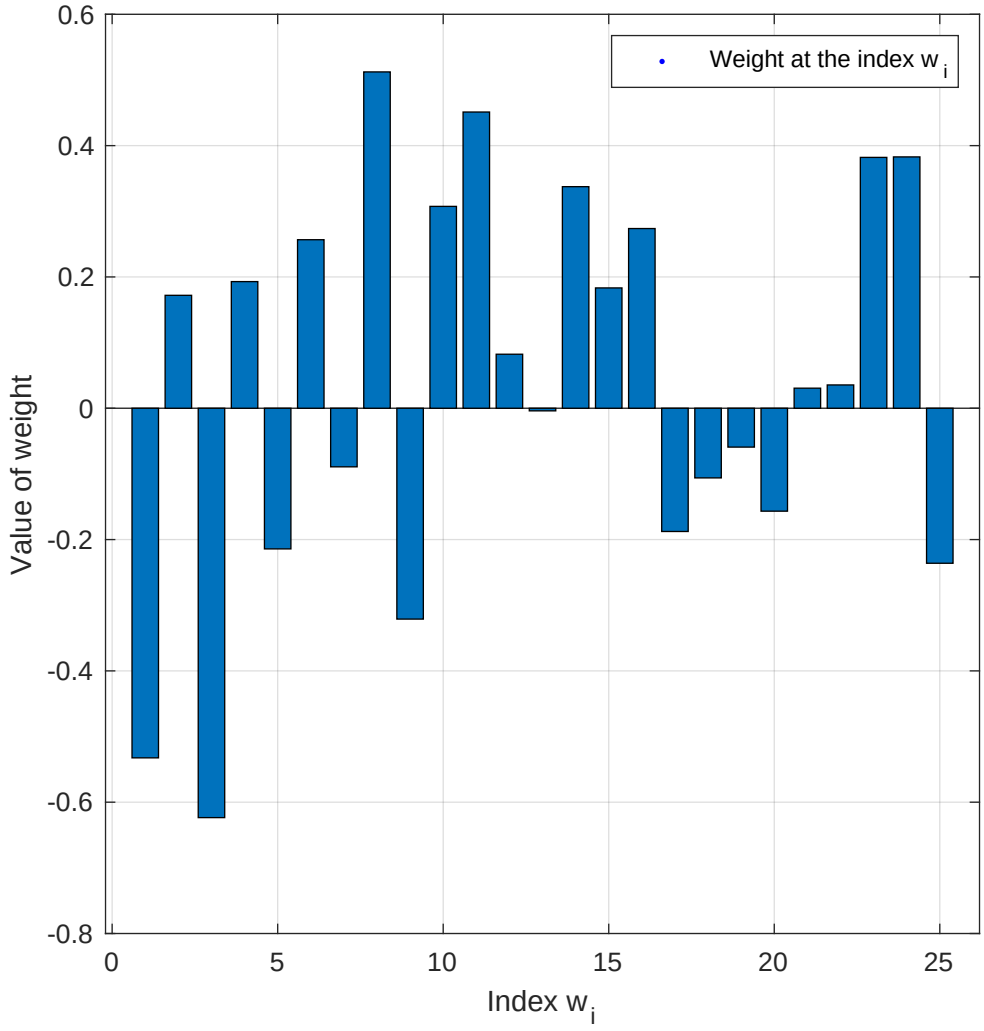

Figure 3: *Graph of the calculated weights given,* $P = 40$

0.2.3. $P = 50$

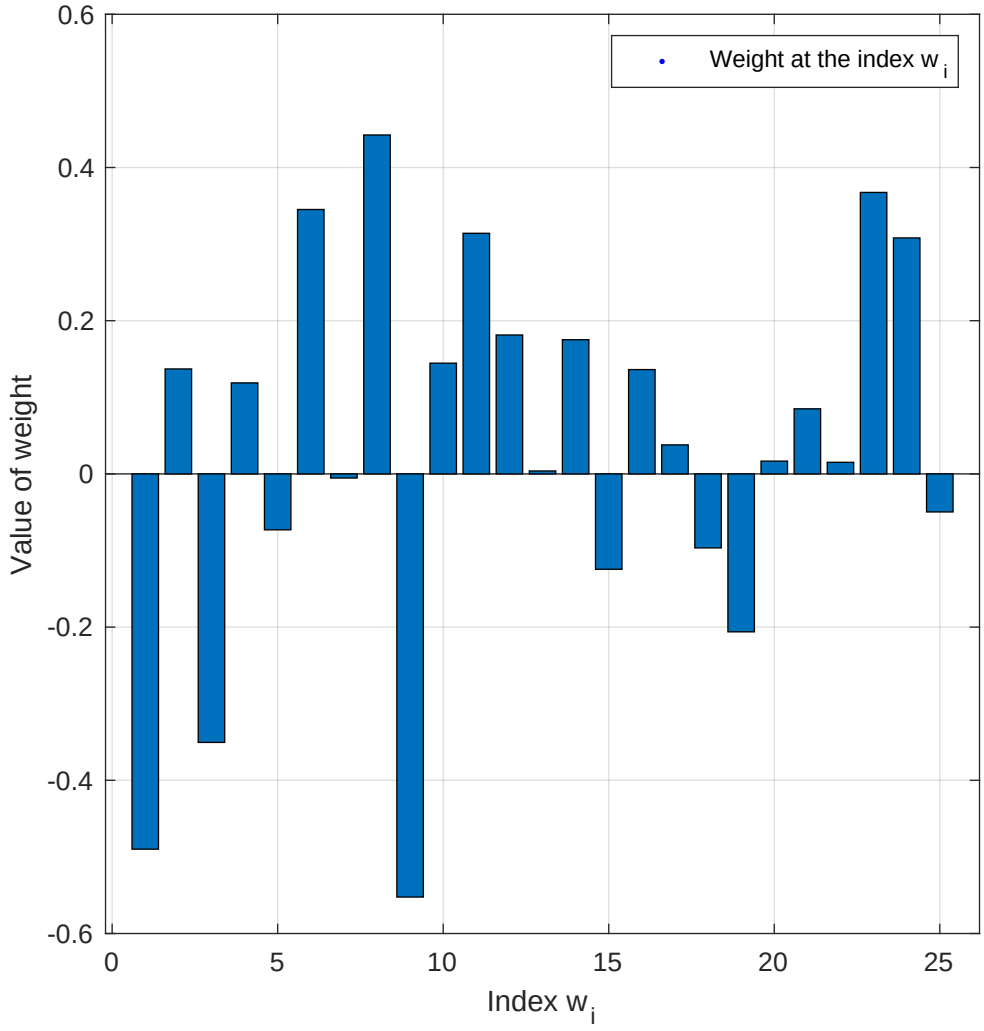

Figure 4: *Graph of the calculated weights,* $P = 50$

0.2.4. $P = 75$

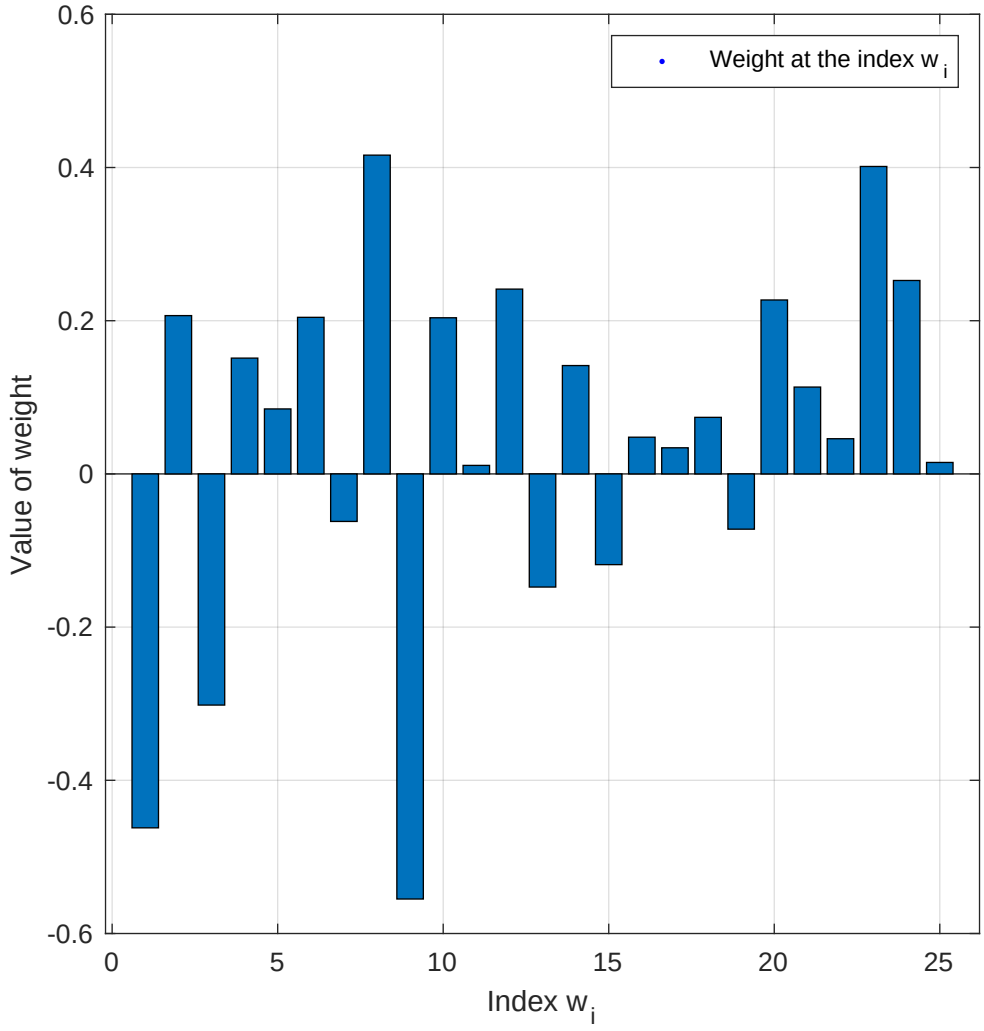

Figure 5: *Graph of the calculated weights,* $P = 75$

0.2.5. $P = 100$

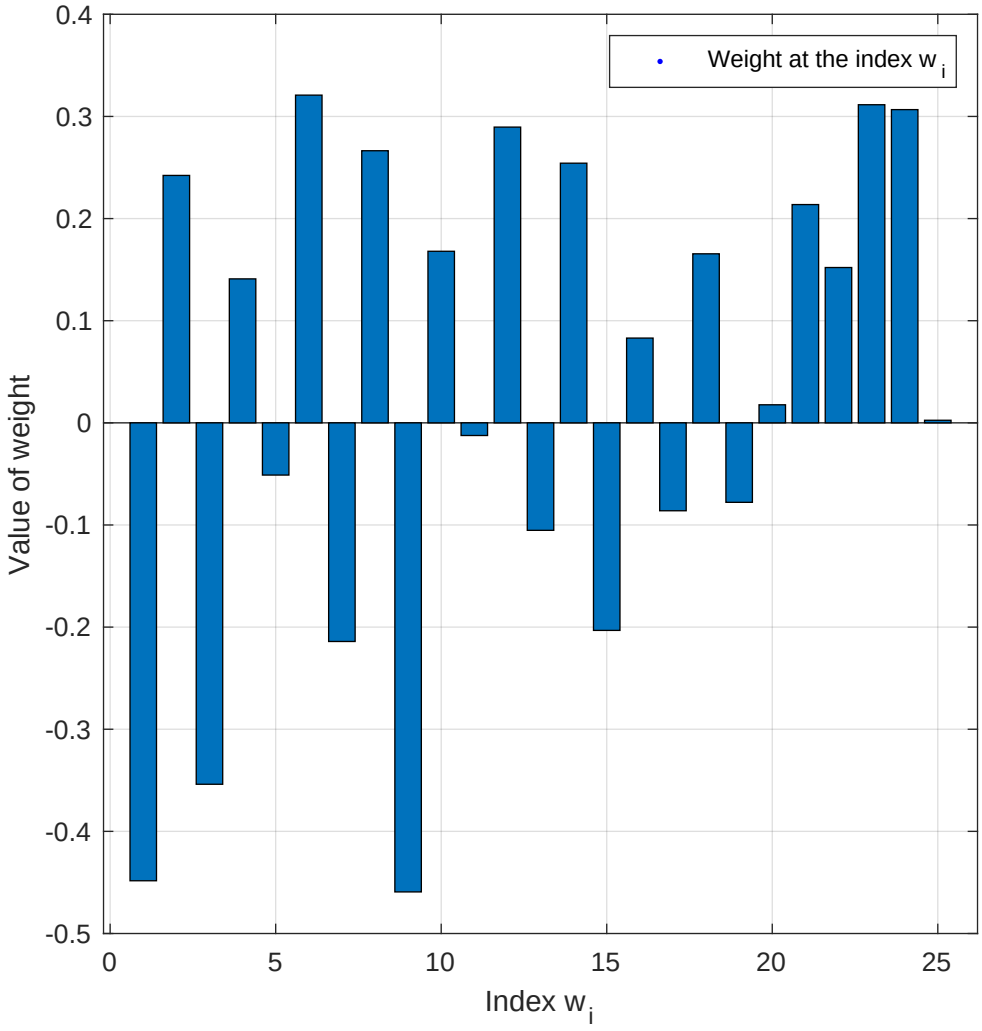

Figure 6: *Graph of the calculated weights,* $P = 100$

0.2.6. $P = 500$

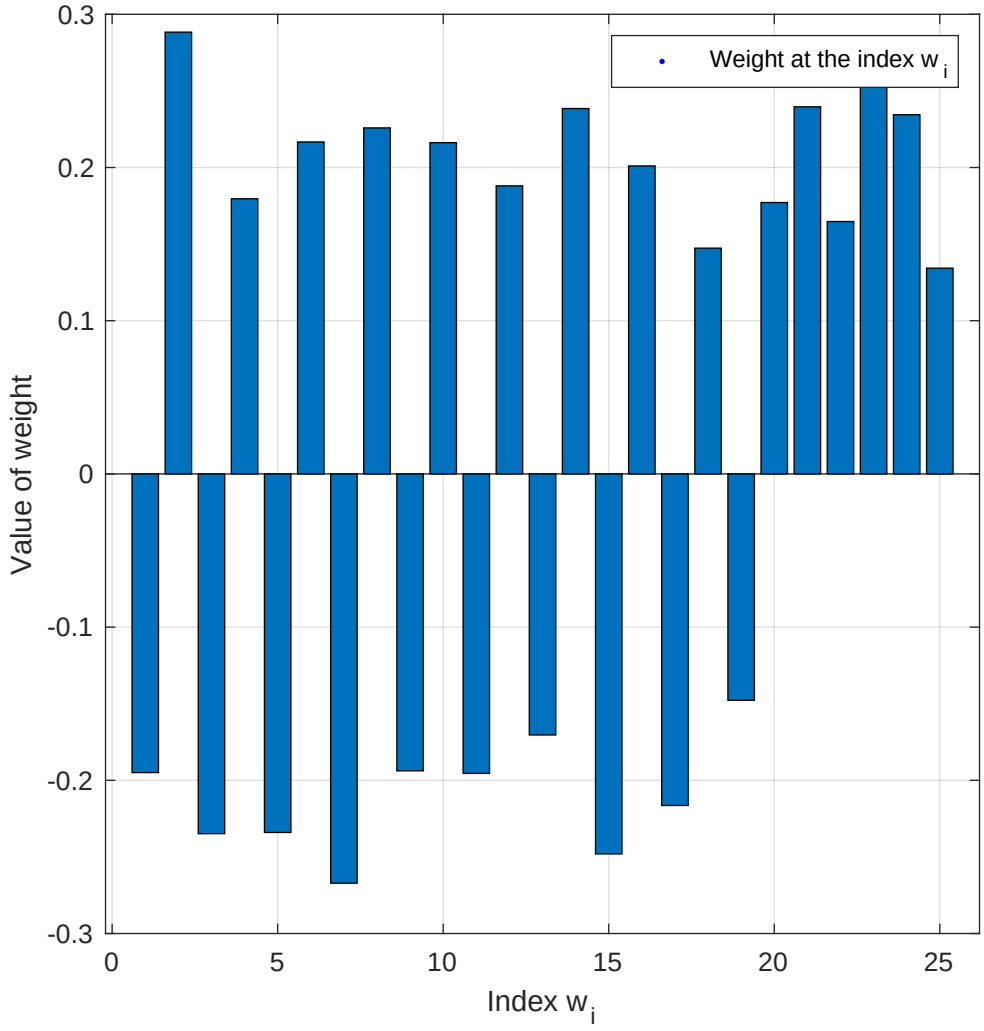

Figure 7: *Graph of the calculated weights,* $P = 500$

Discussion

The data of the graph depicting E_{train} and E_{test} behaves as expected. Due to the increased accuracy of the graph as part of the bonus, you can clearly see that while E_{train} follows a logarithmic function with increasing P, the curve of E_{test} can be described as $\frac{1}{x} + 0.5$, following an 'inverted' logarithmic function. The logarithmic increase in E_{train} is due to the fact, that a growing number of P leads to a situation where minimizing the distance between every input - target pair gets harder. However, since more data allows for more accurate weights, the increase is not linear but logarithmic. A similar argument can be used to explain the data of E_{test}. Due to an increasing accuracy for w^*, the target value can be better approximated with increasing P.

The data concerning w^* follows a clear trend as well. When observing Figure 2 to Figure 7, one can recognize a pattern evolving with an increasing number of P. It seems that with a larger training set, the values at neighboring indices in the weight graph seem to oscillate between positive and negative values, tending to have the same absolute value. In Figure 7, which shows the weight after training with the whole data set, the first 19 indices seem to switch between positive and negative sign, with the last six having the same sign.

Workload

The workload was split equally, with Matteo Wohlrapp focusing more on the the programming efforts, while Lucas Peireira complemented the work with documenting the results and refining the code. However, ideas and issues were discussed and solved as a team.