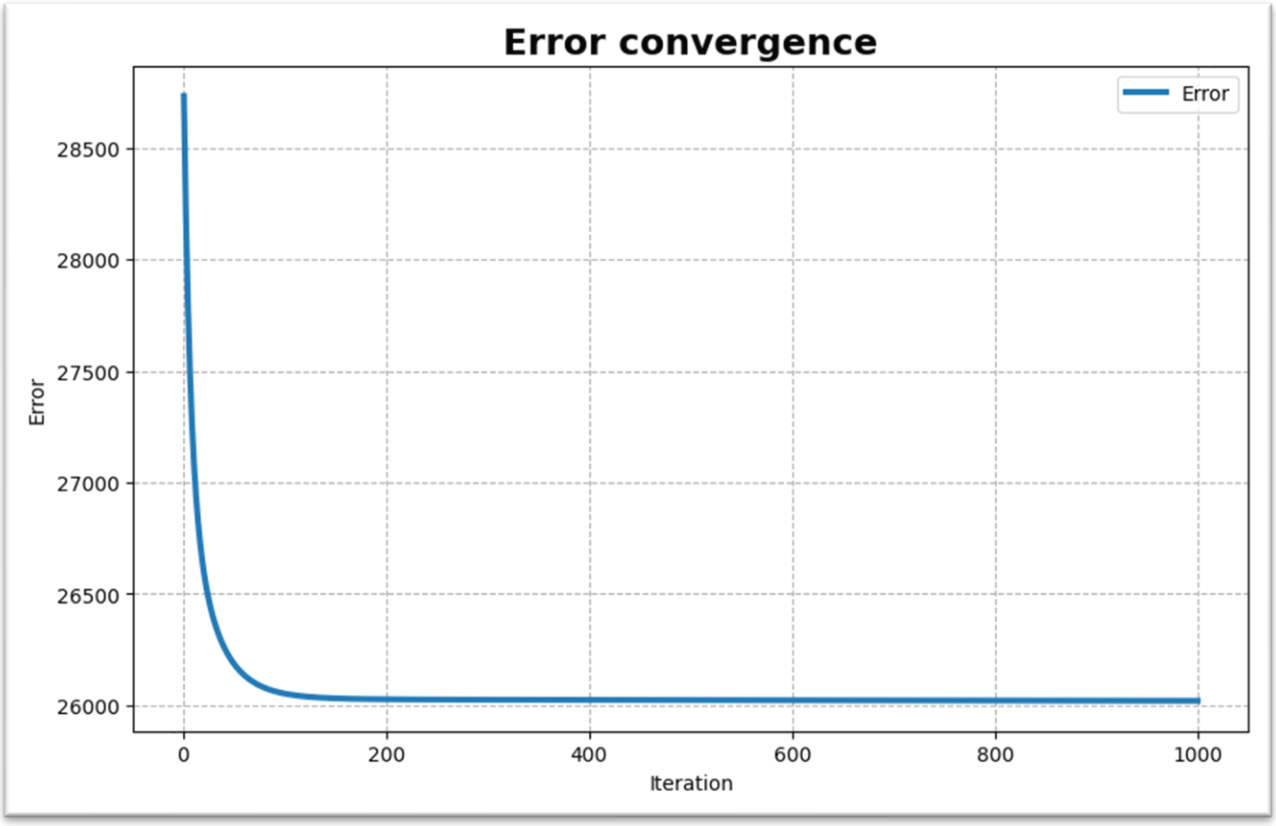
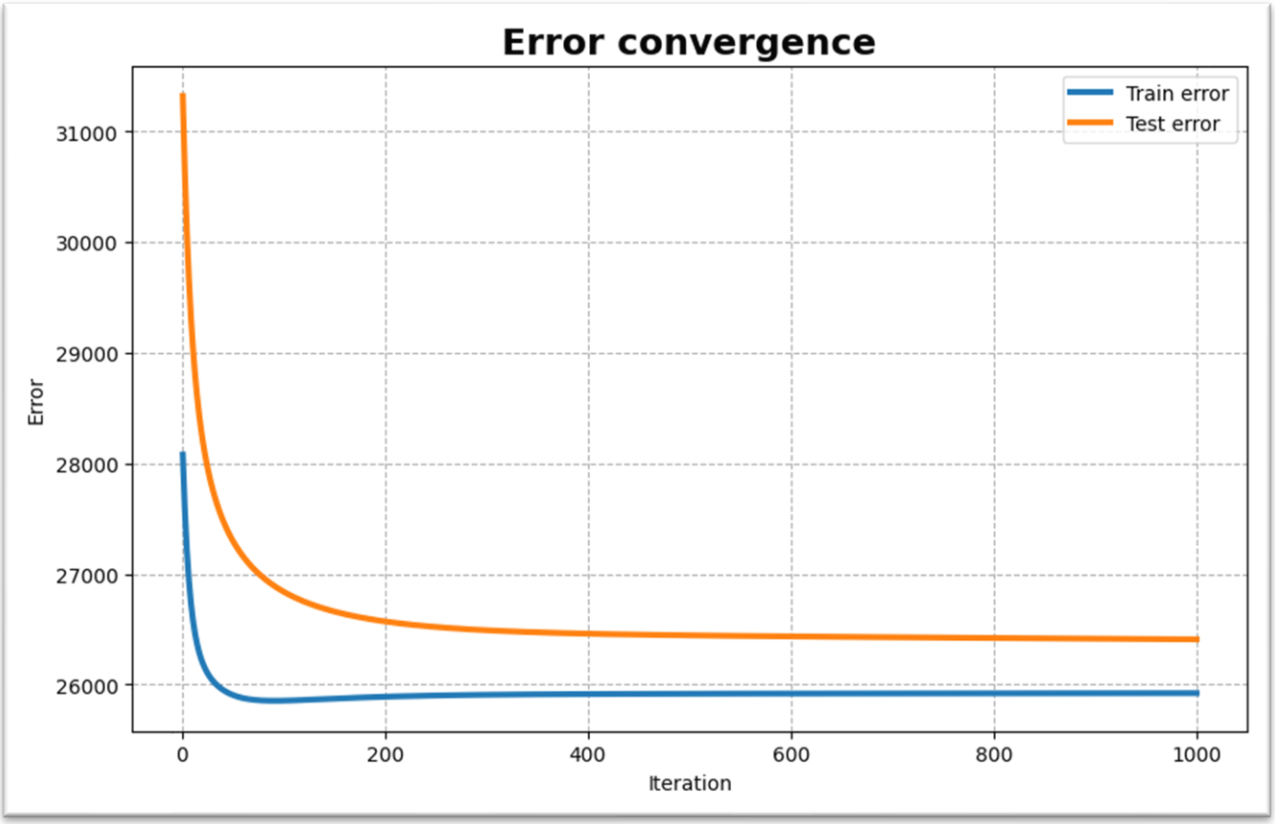
1. Error plot of linear least squares problem on the entire diabetes dataset using gradient descent:



2.



Using a random 80/20 split of the data into a train and a test set respectively, we can observe that the predictor that was trained on the train set gives lower error on the train set than on the test set.

```
iterations: 1000
last train error: 25922.176407732277
last test error: 26409.06367277156
```

Considering the values we ended up with, it seems that the relative difference is small enough to **not** be considered overfitting of the predictor to the training data: $\frac{difference}{training\ err} = \frac{26409-25922}{25922} \approx 1.8\%$.

Also, comparing the average squared error values to value of the **true** labels:

```
max label = 346.0
min label = 25.0
avg of labels = 152.13348416289594
median of labels = 140.5
```

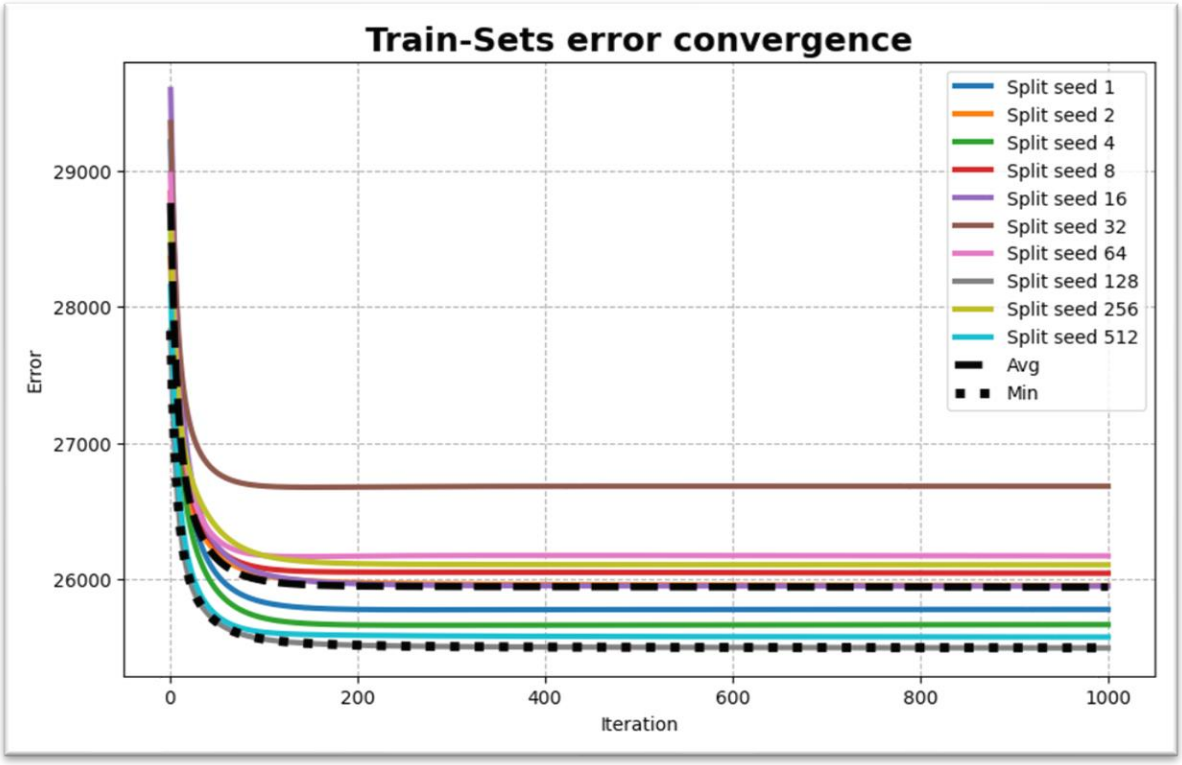
we can see that the average **difference**, between the predicted label from the corresponding true label is too high (relative to the average of the labels), to consider its predictions accurate **even on the training data**:

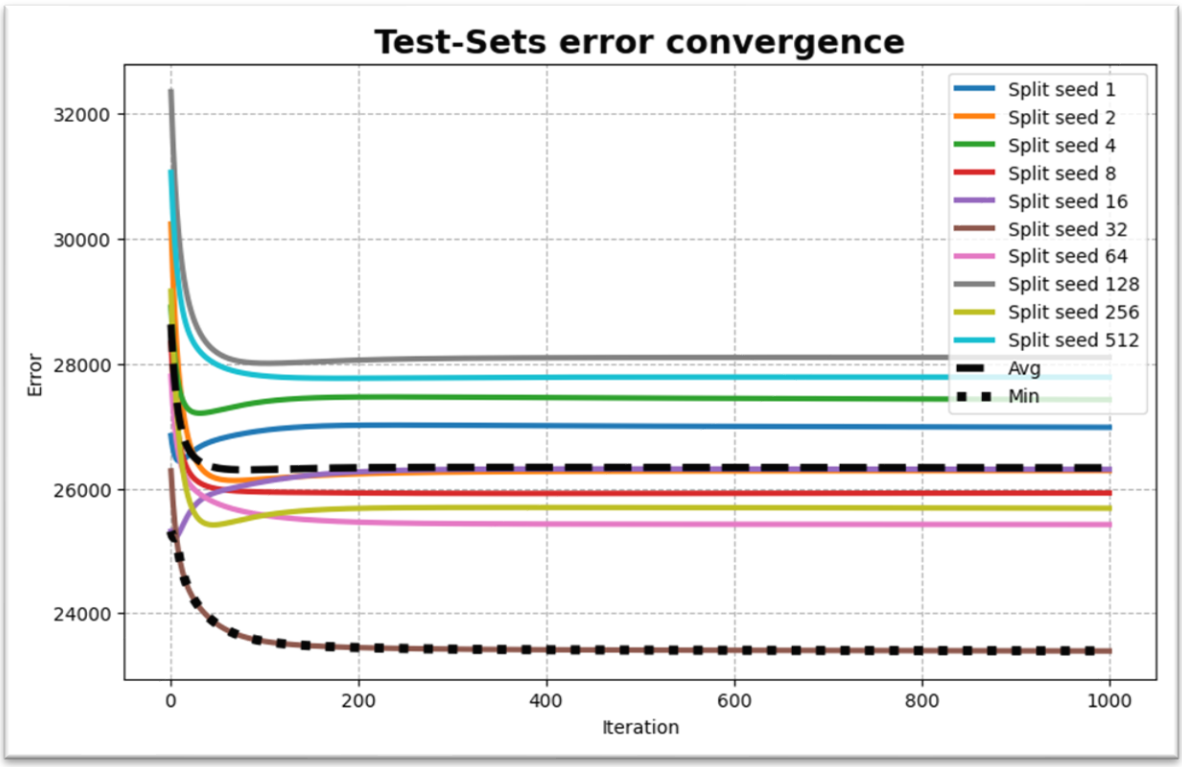
$$avg\ \boldsymbol{diff}\ of\ true\ label\ from\ prediction > \sqrt{25922} > 161$$

$$\frac{avg\ \boldsymbol{diff}}{avg\ true\ label} > \frac{161}{153} > 100\%.$$

Thus, we conclude that the model is **underfit** for the given data and is not suitable for learning on it.

3.


Train-Sets error convergence
Split seed 1
Split seed 2
Split seed 4
Split seed 8
Split seed 16
Split seed 32
Split seed 64
Split seed 128
Split seed 256
Split seed 512
Avg
Min
29000
28000
27000
26000
Error
0
200
400
600
800
1000
Iteration


Test-Sets error convergence
Split seed 1
Split seed 2
Split seed 4
Split seed 8
Split seed 16
Split seed 32
Split seed 64
Split seed 128
Split seed 256
Split seed 512
Avg
Min
32000
30000
28000
26000
24000
Error
0
200
400
600
800
1000
Iteration

Using 10 different random 80/20 splits of the data into a train and a test sets respectively, we can observe changes in the behavior of the train and test error graphs pairs, depending on the split.

```
last train error avg=25942.040515377685
last train error min=25495.950136536852
```

```
last test error avg=26330.276818852148
last test error min=23395.647831043338
```

Considering that the error values we end up with from the graphs of the **averages** aren't very different from the values we got in the previous question, we **can't say** that we observe any significant change in the fitting of the model.

The same could be said about the graphs of the **minimums**.

The minimum graph from the test sets plot is given by the same predictor that produces the maximum graph in the train sets plot, and vice versa. This behavior of the minimum graphs indicates, that the more the predictor that the algorithm found fits the train set, the less suitable this predictor will be for the test set. The balance point for finding the optimal Xk vector is represented by the average graphs, where the average error of the train is close to the average error of the test.

It can be seen, that the average error graph on both of the plots, models more closely the graph from the first question, where the algorithm tried to fit the predictor to the whole data set, than the minimum error graph, or most individual pairs of train-test graphs. Which would in turn indicate that using this kind of splitting of the dataset and averaging the graphs, and under the assumption of i.i.d. of seen and unseen data, we could closely predict the error that the predictor would have if it was tested on all possible data.