CSE4088 Introduction to Machine Learning
Homework 2

Berk Kırtay - 150118043

Generalization Error

The formula changes when we apply €=0.005 and M=1: $P[|E_{in}-E_{out}| > 0.05] <= 2 \cdot e^{\frac{-1}{200} \cdot N} <= 0.003$

1.

$$\text{For } N = 500 -> 2 \cdot e^{\frac{-5}{2}} = 0.164 \text{ which is greater than } 0.03$$

So we can calculate by trying different values of N.

$$\text{For } N = 1000 -> 2 \cdot e^{\frac{-1}{200} \cdot 1000} = 0.0134 \text{ which is less than } 0.03$$

Answer is N=1000, B.

2.

$$\text{For } N = 1500 -> 2 \cdot e^{\frac{-1}{200} \cdot 1500} = 0.011 \text{ which is less than } 0.03$$

Answer is N=1500, C.

3.

$$\text{For } N = 2000 -> 2 \cdot e^{\frac{-1}{200} \cdot 2000} = 0.009 \text{ which is less than } 0.03$$

Answer is N=2000, D.

The Programming Questions

For the rest of the questions, I wrote functions in python and I specified which one I run for each question. More details can be observed in the source code. To run the scripts, please run "hw2.py".
To plot classification line, please call the functions with plotting=True like in the following:

```python
run_pla(sample_size=10, plotting=False, question="Q4, Q5")
run_pla(sample_size=100, plotting=False, question="Q6, Q7")
run_linear_regression(sample_size=100, plotting=False, question="Q8")
run_linear_regression2(sample_size=100, plotting=False, question="Q9")
run_lr_pla(sample_size=10, plotting=False, question="Q10")
run_nonlinear_transformation(sample_size=1000, plotting=False, question="Q11")
run_nonlinear_transformation2(
    sample_size=1000, plotting=False, question="Q12, Q13")
```

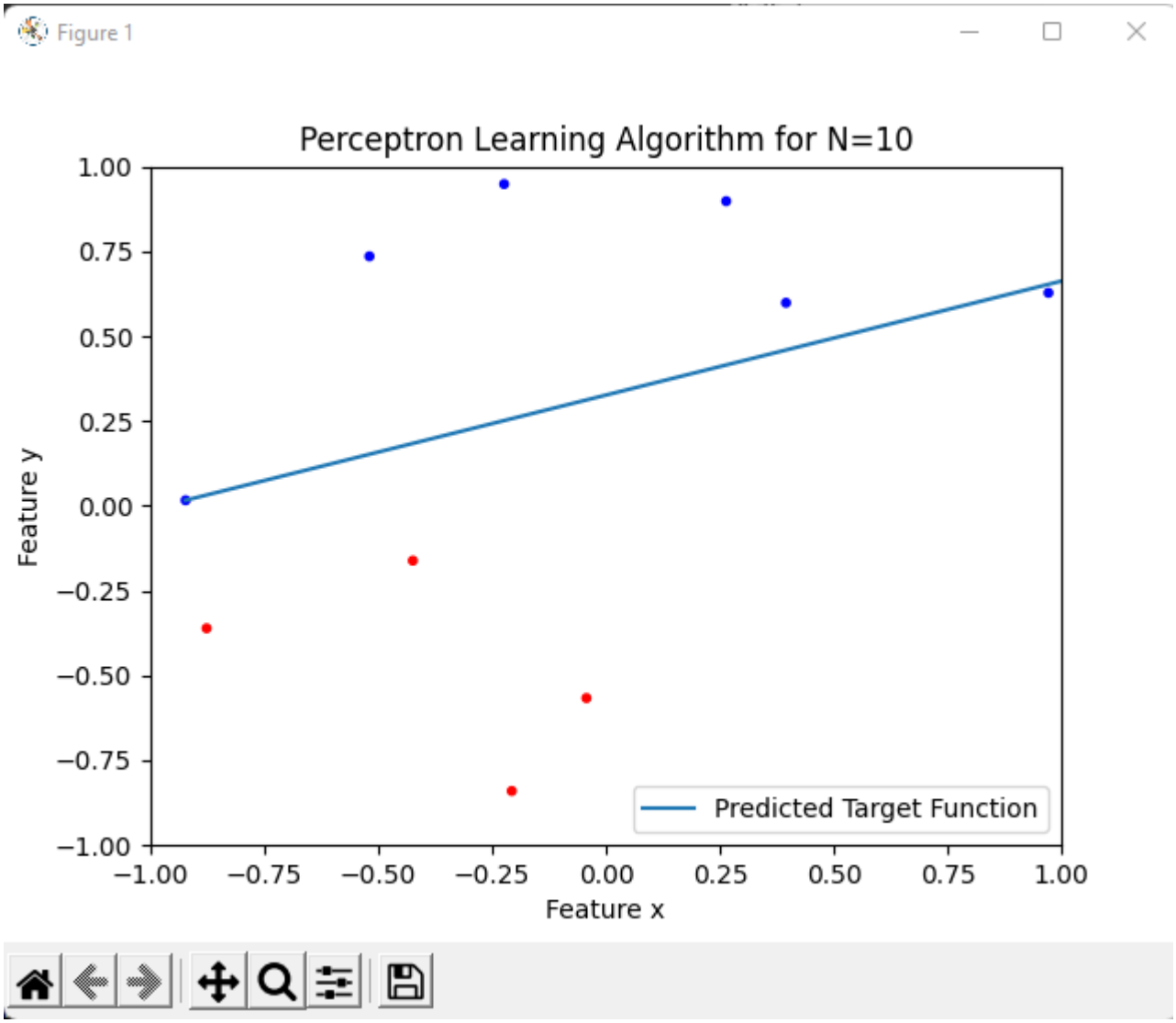
An example program run:

```
$ py hw2.py
Q4, Q5: For N=10, Disagreements: 0.1130999999999997, iterations until convergence: 10.977.
Q6, Q7: For N=100, Disagreements: 0.01334999999999987, iterations until convergence: 115.251.
Q8: For N=100, Ein: 0.039180000000000076.
Q9: For N=100, Eout: 0.04825800000000003.
Q10: For N=10, Disagreements: 0.003800000000000002, iterations: 1.733.
Q11: For N=1000, Ein: 0.5049030000000001.
Q12, Q13: For N=1000, Eout: 0.10063899999999996
```

The Perceptron Learning Algorithm

4. and 5.

I run run_pla() function with N = 10.
Sample plotting of predicted classification line:

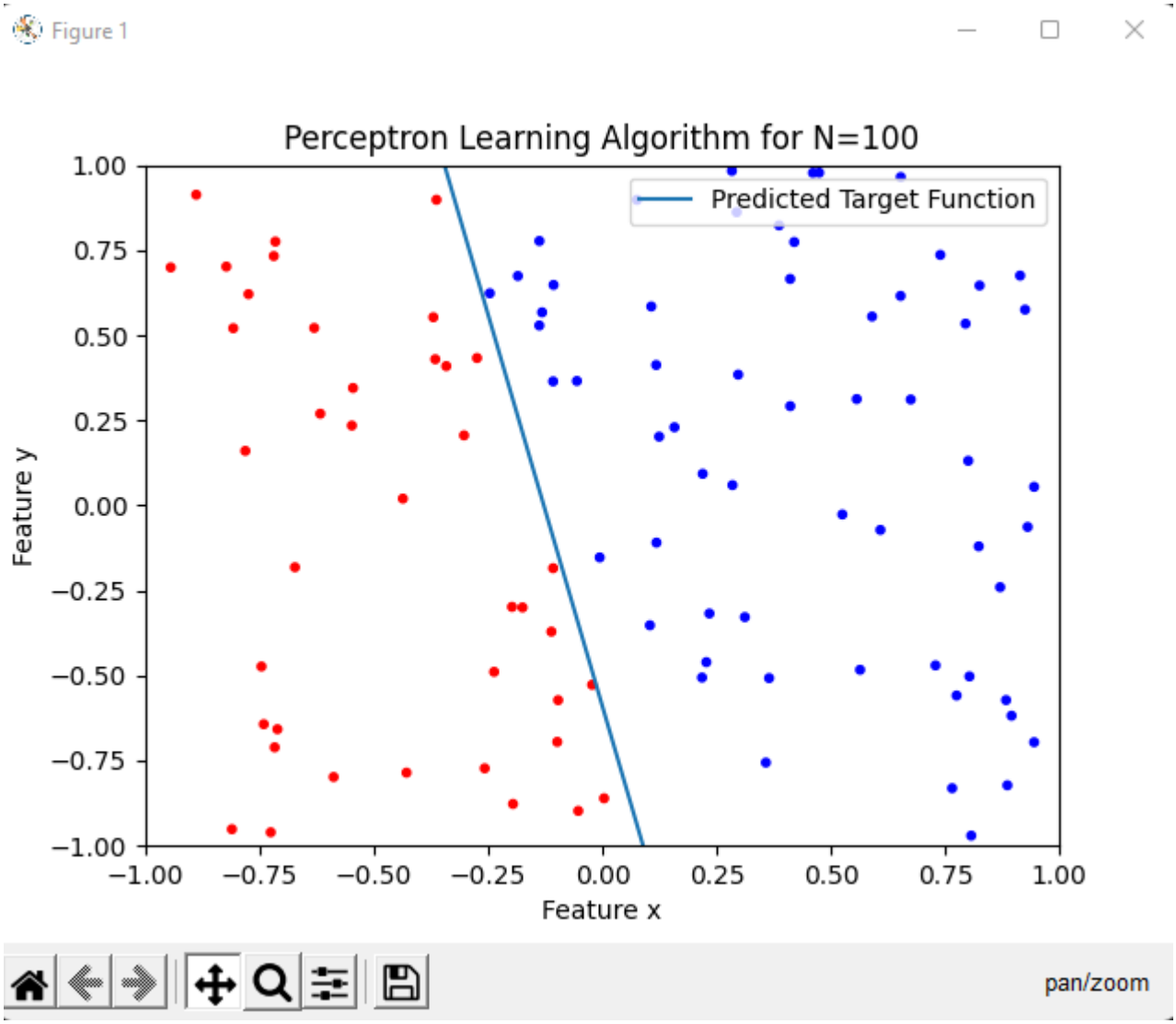


1000 runs average iterations until converge: 10.97, P[f(x) != g(x)]= 0.111

Answers: 4: B and 5: C

6. and 7.

I run run_pla() function with N = 100.
Sample plotting of predicted classification line:



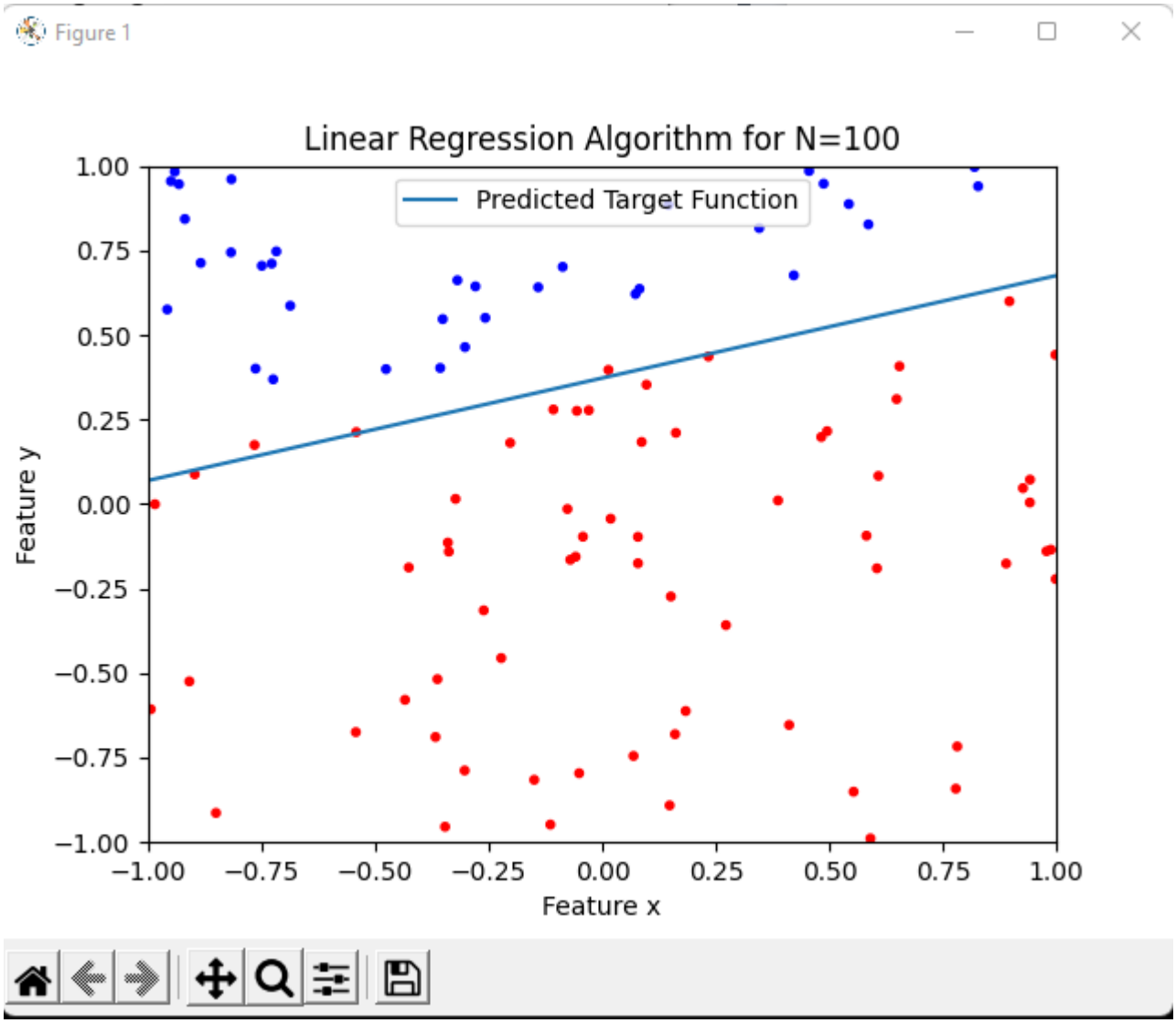
1000 runs average iterations until converge: 115.251, P[f(x) != g(x)]= 0.013

Answers: 6: B and 7: B

Linear Regression

8.

I run run_linear_regression() function with N = 100.
Sample plotting of predicted classification line:



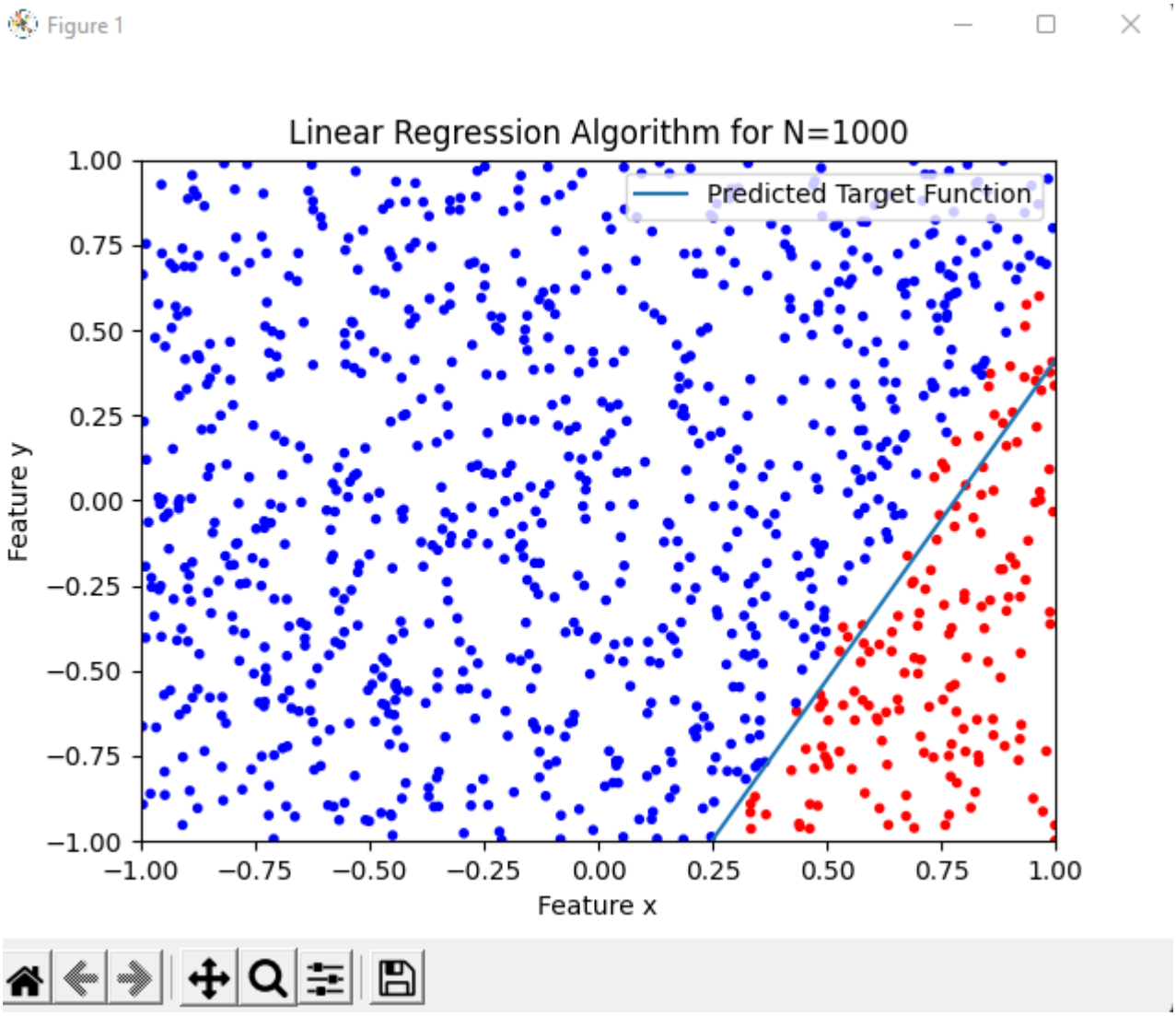
1000 runs average:

$$P[f(x)! = g(x)] = E_{in} = 0.0391$$

Answer: C

9.

I run run_linear_regression2() function with N = 100.
Sample plotting of predicted classification line with 1000 out of sample points:



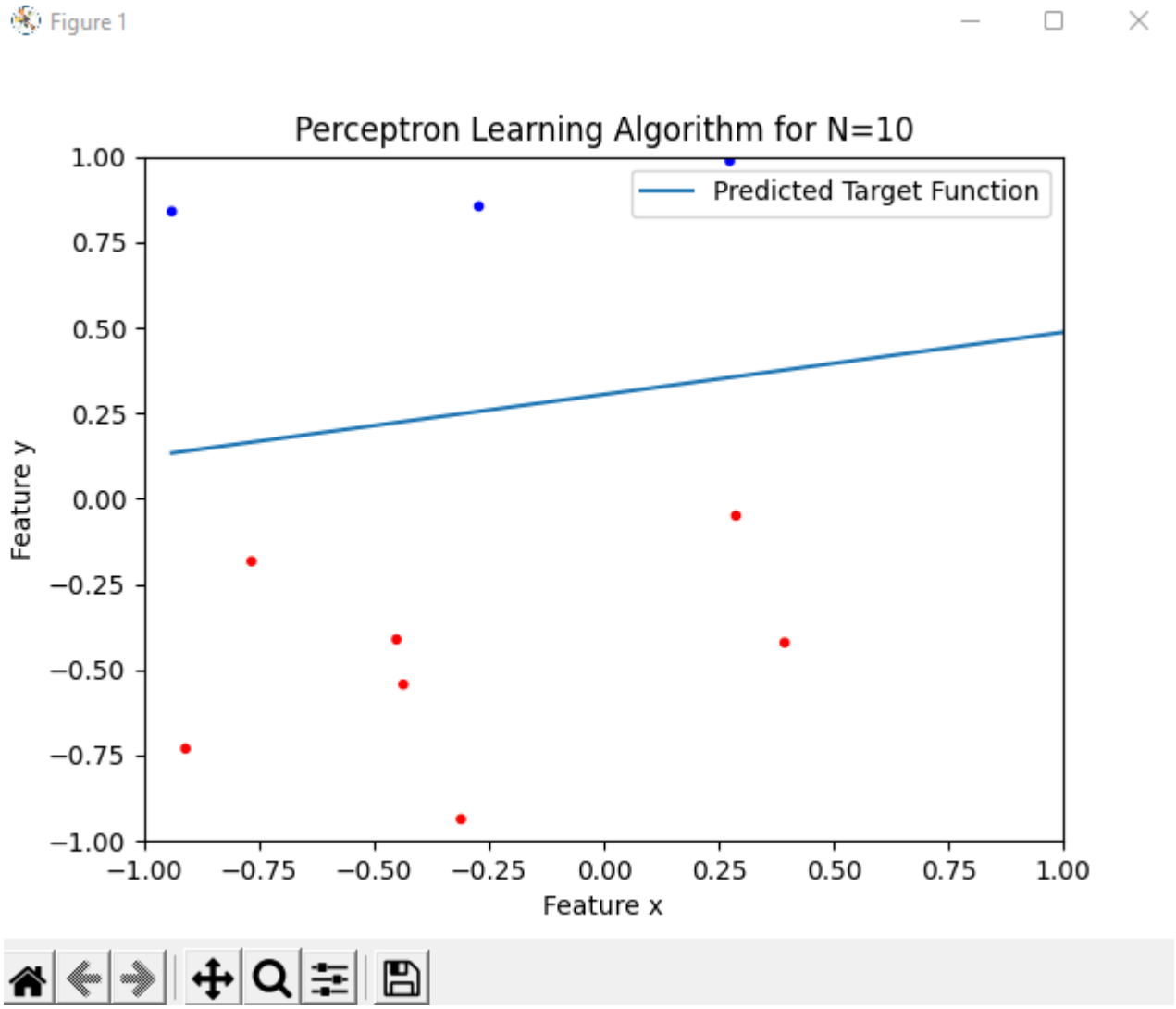
1000 runs average:

$$P[f(x)! = g(x)] = E_{out} = 0.048$$

Answer: C

10.

I run run_lr_pla() function with N = 10.
Sample plotting of predicted classification line:



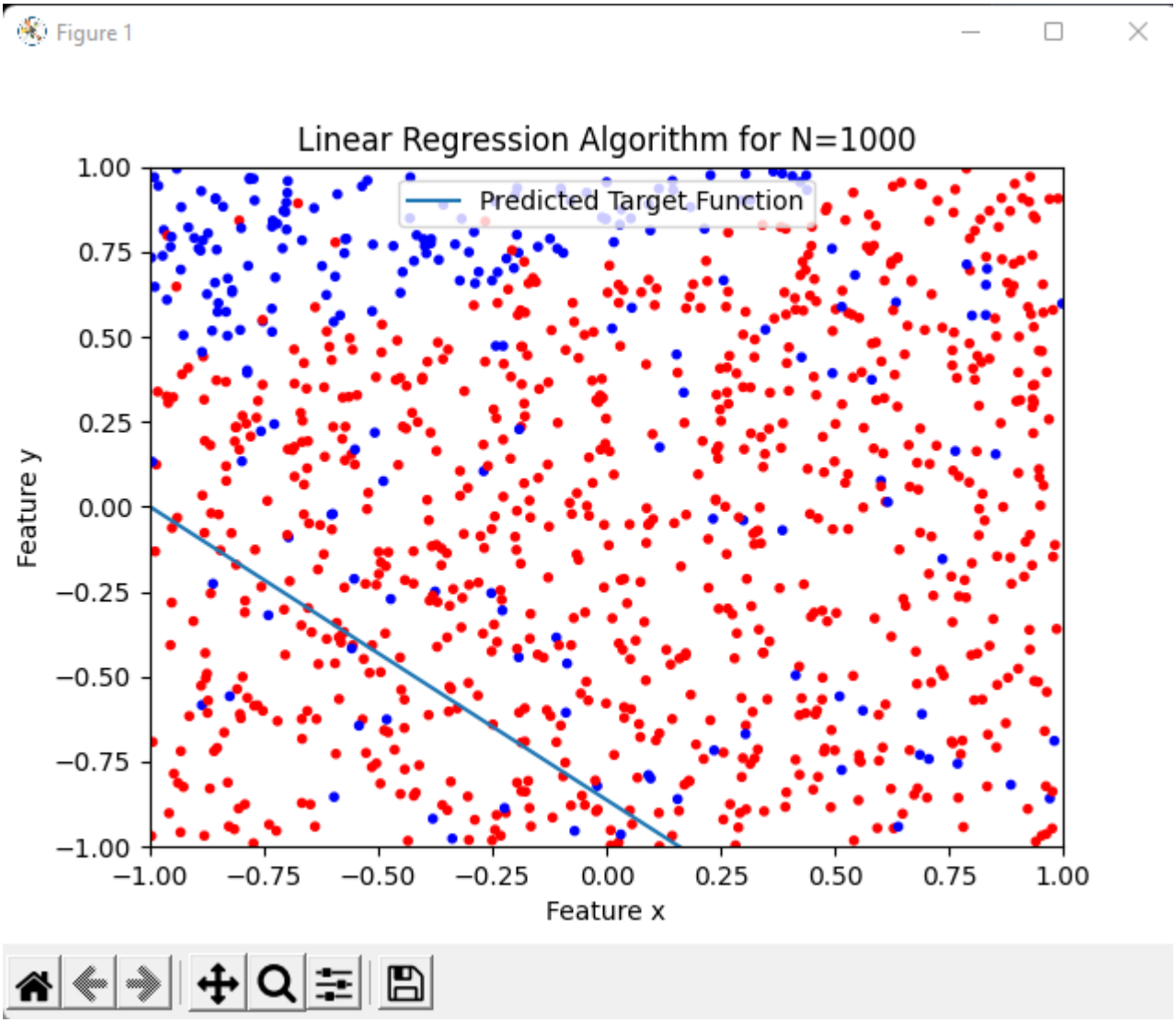
1000 runs average:
Iterations until converge: 1.733

Answer: A

Nonlinear Transformation

11.

I run run_nonlinear_transformation() function with N = 1000.
Sample plotting of predicted classification line:



1000 runs average:

$$P[f(x)! = g(x)] = E_{in} = 0.504$$

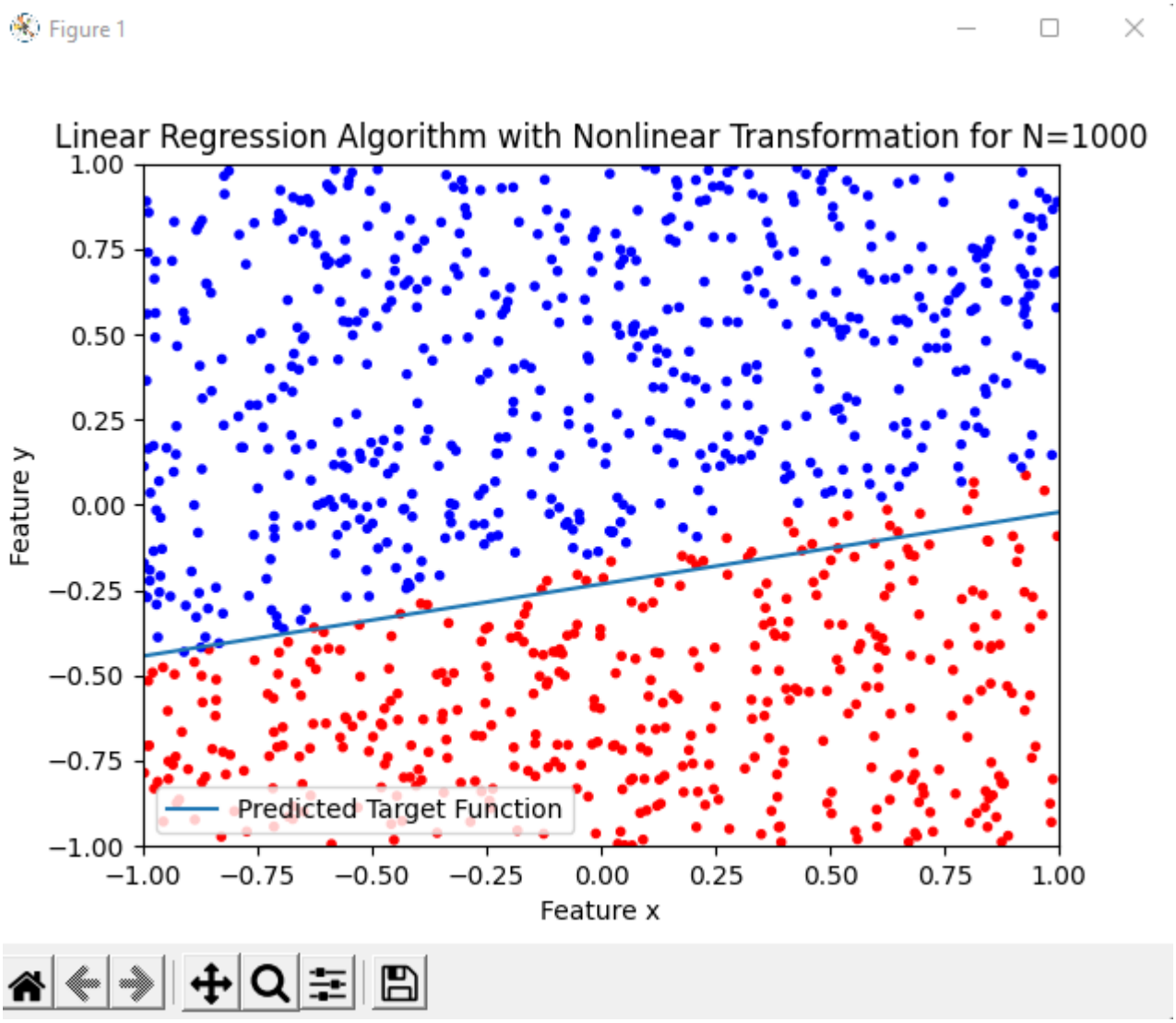
Answer: D

12.

When we tested each function in the choices, the hypothesis in the A choice gives the least disagreements with our predicted target function.
Answer: A

13.

I run run_nonlinear_transformation2() function with N = 1000.
Sample plotting of predicted classification line with 1000 out of sample points:



$$P[f(x)! = g(x)] = E_{out} = 0.101$$

1000 runs average:
Answer: B