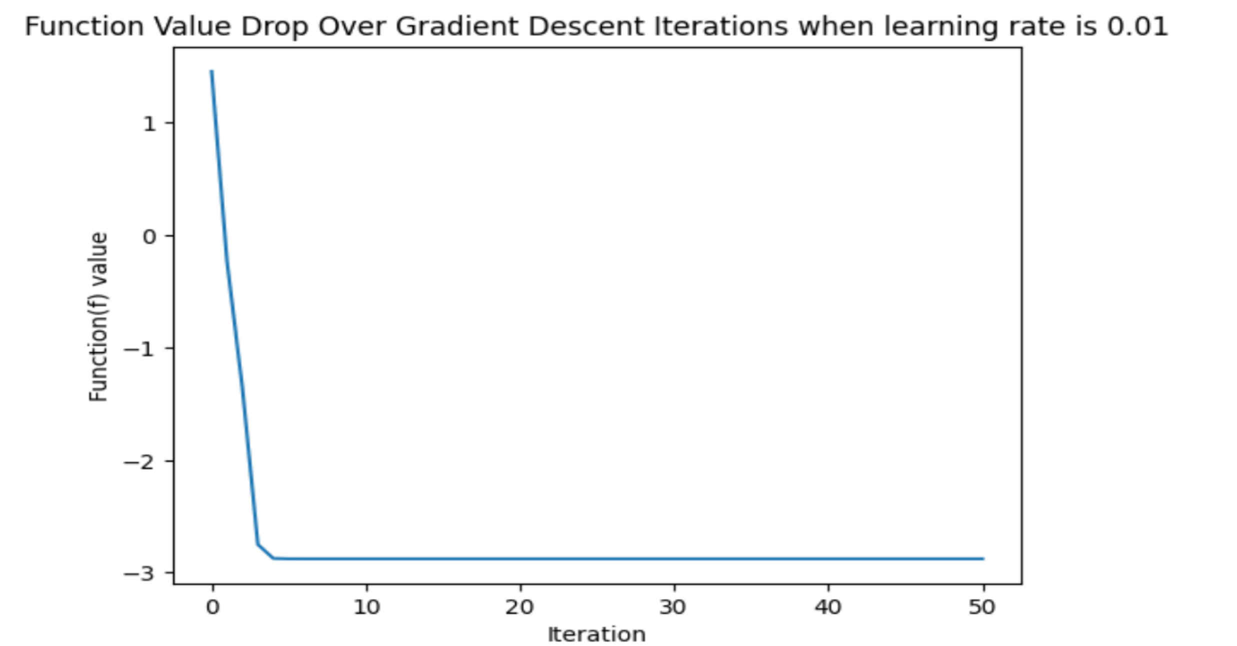
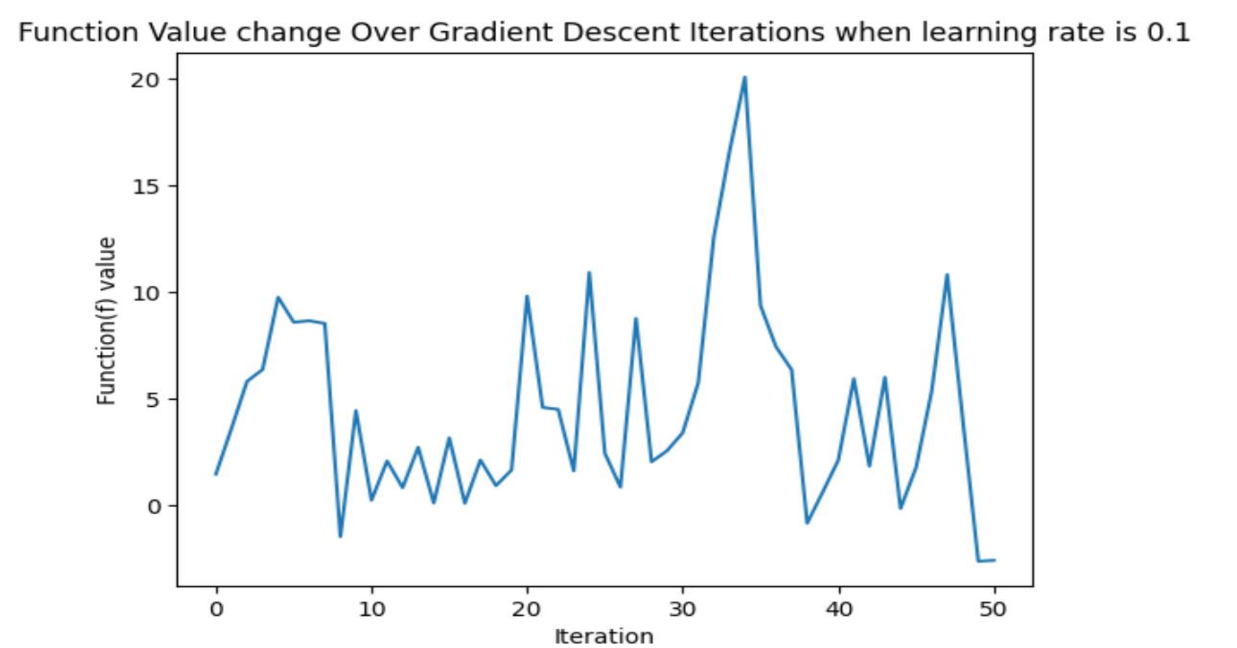
# Home Work – 3

Saiprakash Nalubolu
B01037579

Q1.

a.



For Learning rate ($\eta$) = 0.01, the descent is smooth and the function value decreases steadily as the iterations increases. There are no significant spikes in the function value, indicating that the learning rate is conservative enough to allow the algorithm to converge towards a local minimum without overshooting. The plot suggests a good rate of convergence without significant oscillations.

For Learning rate ($\eta$) = 0.1, There are substantial oscillations in the function value with increase in number of iterations. This indicates that the learning rate might be too high for this problem. The large learning rate causes the algorithm to overshoot the minimum and then attempt to correct itself, leading to a zig-zag pattern in the descent.

The learning rate $\eta$ plays a crucial role in the behavior of the gradient descent algorithm. A small learning rate can lead to a slow but steady convergence, but when the learning rate is higher it can cause overshooting and oscillations. If the learning rate is too small, convergence may be very slow, requiring many iterations to reach the minimum. On the other hand, if the learning rate is too large, the algorithm may never converge and instead keep bouncing around the minimum.

b. When starting point is at

(i) (0.1, 0.1)

```
Initial point: [0.1 0.1]
The minimum value of the function: -2.8790846587644263
The location of minimum value of the function at (x, y) = (-0.241828945494765, 1.2269903410060357e-10)
```

(ii) (1, 1)

```
Initial point: [1 1]
The minimum value of the function: -0.9286447086312597
The location of minimum value of the function at (x, y) = (0.7252678803578847, 0.9831635693235358)
```

(iii) (0.5, 0.5)

```
Initial point: [0.5 0.5]
The minimum value of the function: -2.6332425909756374
The location of minimum value of the function at (x, y) = (0.24181813075115313, 0.4916822590684799)
```

(iv) (0.0, 0.5)

```
Initial point: [0.  0.5]
The minimum value of the function: -2.6332425909756374
The location of minimum value of the function at (x, y) = (0.2418181308911715, 0.49168225905153434)
```

(v) (−0.5,−0.5)

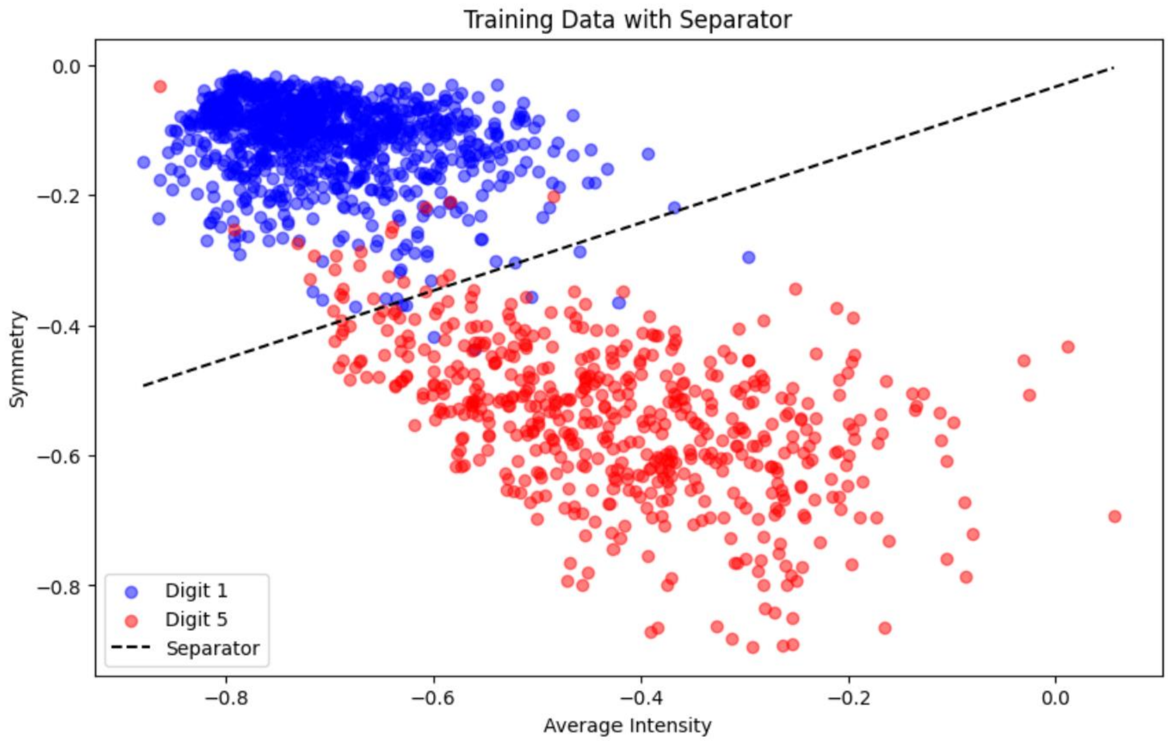
```
Initial point: [-0.5 -0.5]
The minimum value of the function: -1.6660267055389744
The location of minimum value of the function at (x, y) = (-0.7253691676027778, -0.49159419340047744)
```

(vi) (−1,1)

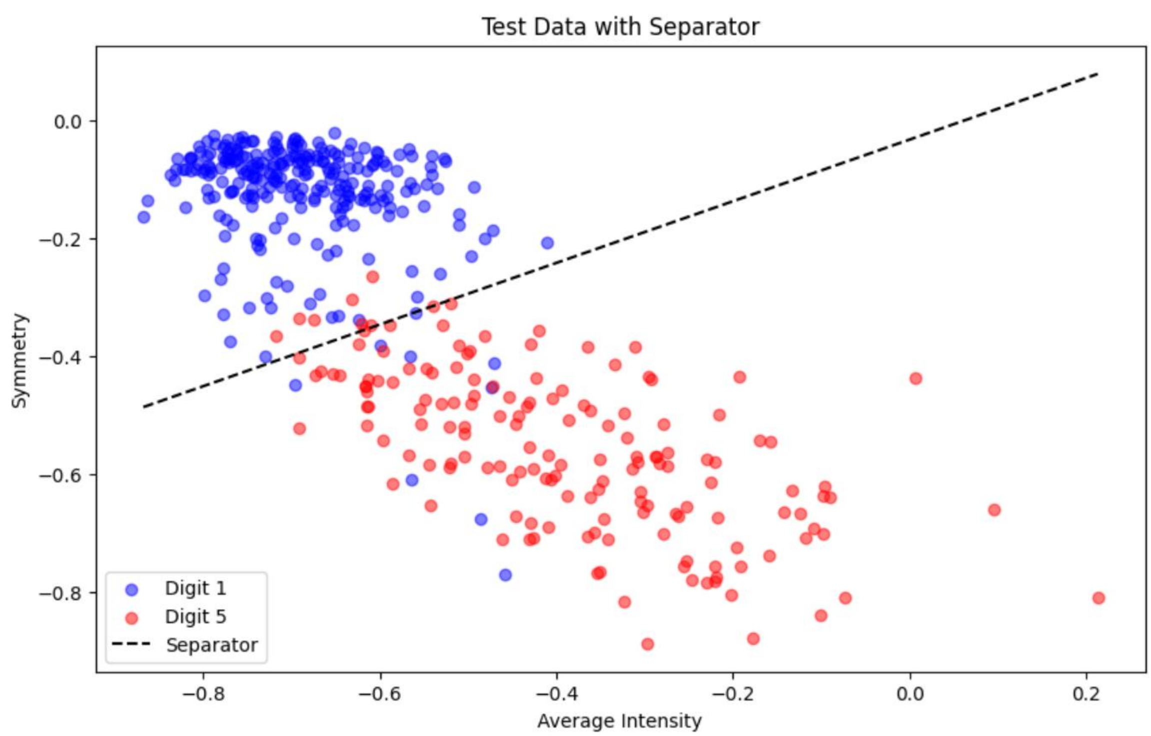
```
Initial point: [-1  1]
The minimum value of the function: 1.0051615759793306
The location of minimum value of the function at (x, y) = (-1.2084759495263497, 0.9827889176426049)
```

Q2,

a)



Training Data with Separator
Symmetry
Average Intensity
Digit 1
Digit 5
Separator
0.0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0.0

Test Data with Separator
Symmetry
Average Intensity
Digit 1
Digit 5
Separator
0.0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0.0
0.2

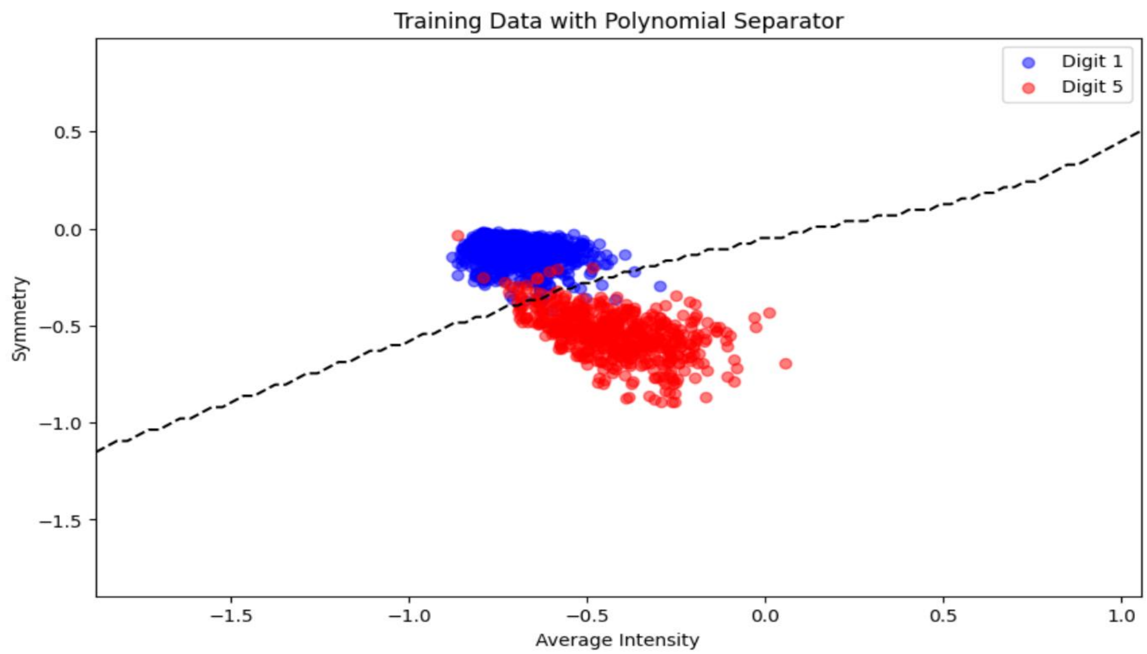
```
E_in (Training Error): 0.02114029468289558
E_test (Test Error): 0.03773584905660377
```

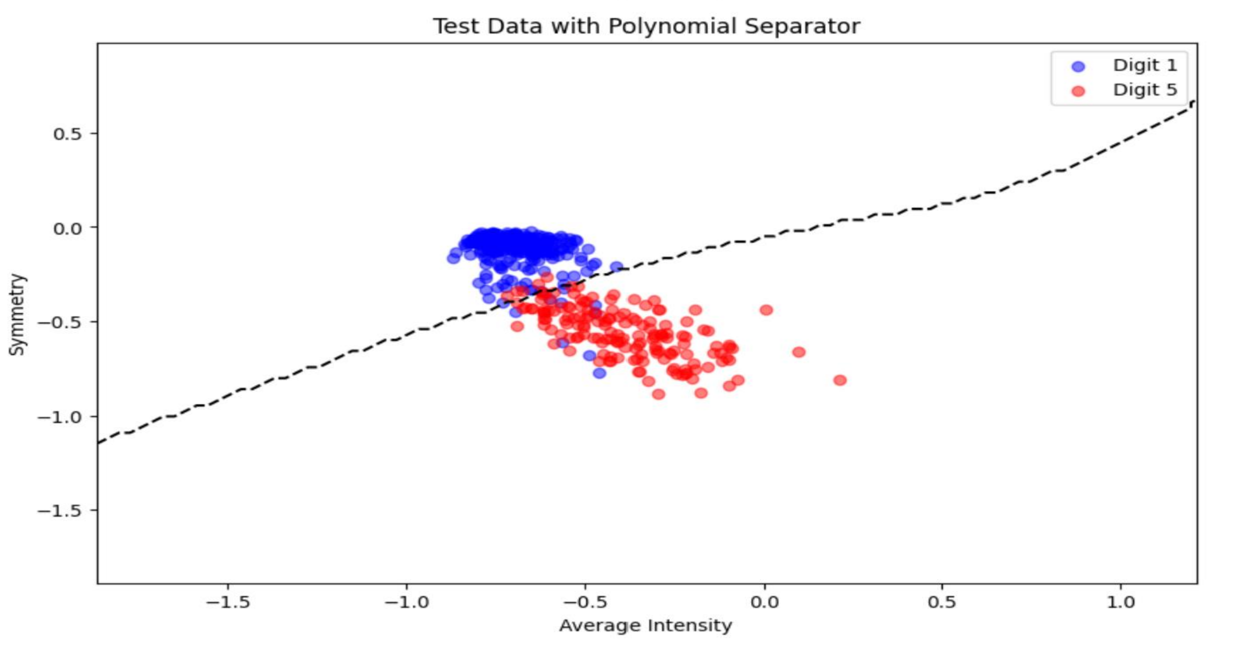
b)

```
Bound based on E_in: 0.055514350859392464
Bound based on E_test: 0.10369108915084554
The better bound is based on E_in.
```

c)

d)



```
E_in after polynomial transformation (Training Error): 0.021780909673286355
E_test after polynomial transformation (Test Error): 0.03537735849056604
Bound based on E_in after polynomial transforms: 0.05615496584978323
Bound based on E_test after polynomial transforms: 0.10133259858480781
```

The better bound is based on E_in.

e) The increase in E_in indicate that the polynomial transformation is leading to overfitting, where the model is too complex for the training data.
   However, the E_test(Test Error) is lower after the polynomial transformation (0.0353) compared to before the transformation (0.0377), which indicates that the polynomial model may generalize better on the test data.
   Hence, If the primary goal is generalization to unseen data (which it typically is), the polynomial model could be preferred because of its lower E_test. If the goal is to have a simpler model due to constraints like computation cost, interpretability, or if there is concern about potential overfitting with more complex models, one might choose the linear model.
   Considering the USPS context, where the model's ability to generalize is likely more important than a small increase in E_in, the model with the 3rd-order polynomial transform could still be the better choice due to its superior performance on the test set. However, this recommendation comes with a limitation that one should monitor for overfitting.

Colab link:

https://colab.research.google.com/drive/1_R2a7lejHmfaJdnB5tjKgEdBRHDRxp4g#scrollTo=gndvzGjd_gz9