Introduction to Machine Learning

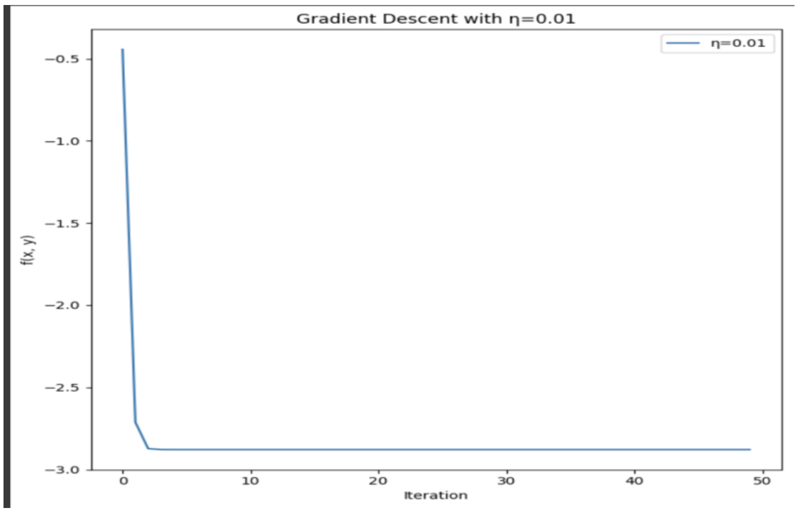
Homework 03

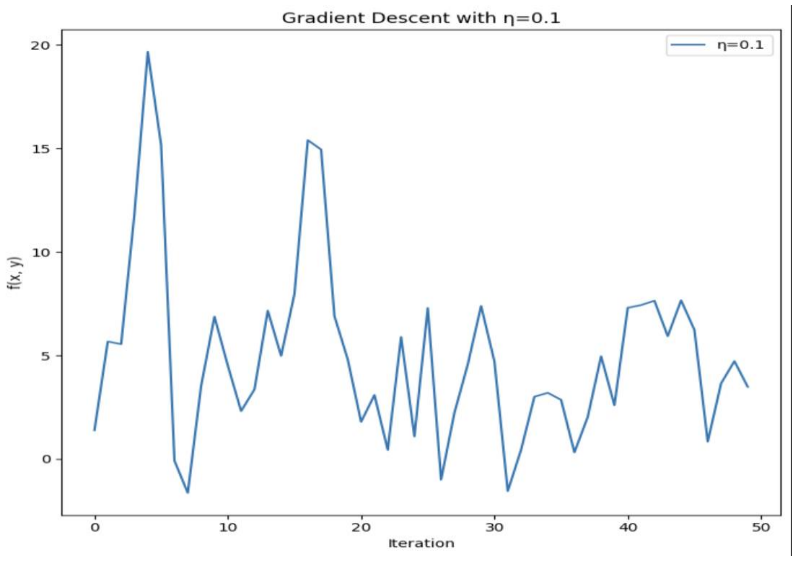
Atharva Patil - B01014288

(1) [200 points] Gradient Descent on a Simple Function.
Consider the function $f(x, y) = 2x^2 + y^2 + 3 \sin(2\pi x) \cos(2\pi y)$.

(a) Implement gradient descent to minimize this function. Run gradient descent starting from the point ($x = 0.1$, $y = 0.1$). Set the learning rate to $\eta = 0.01$ and the number of iterations to 50. Give a plot that displays how the function value drops through successive iterations of gradient descent. Repeat this with a learning rate of $\eta = 0.1$ and provide a plot of the function value with each iteration. What do you observe?

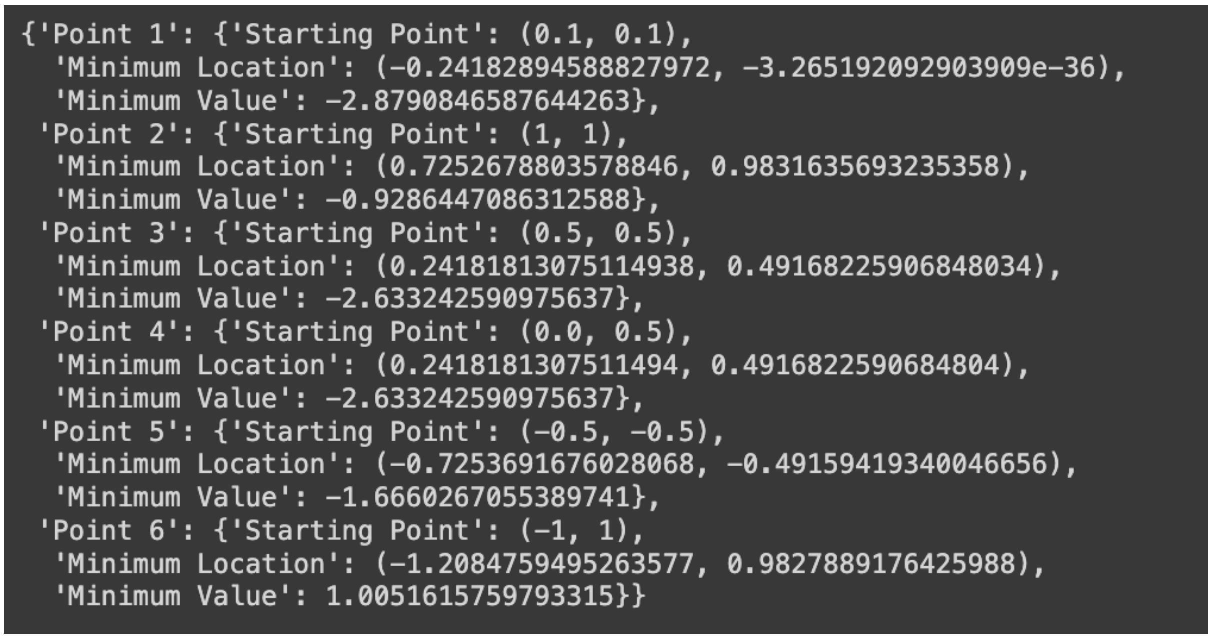
Ans-





(b) Obtain the "minimum" value and location of the minimum value of the function you get using gradient descent with the same learning rate η = 0.01 and number of iterations (50) as part (a), from the following starting points: (i) (0.1, 0.1), (ii) (1, 1), (iii) (0.5, 0.5), (iv) (0.0, 0.5), (v) (− 0.5, − 0.5), (vi) (− 1, 1). Write down the minimum value obtained using gradient descent and the location of the minimum value for each of these starting points. As you may appreciate, finding the "true" global minimum value of an arbitrary function is a hard problem.

Ans-



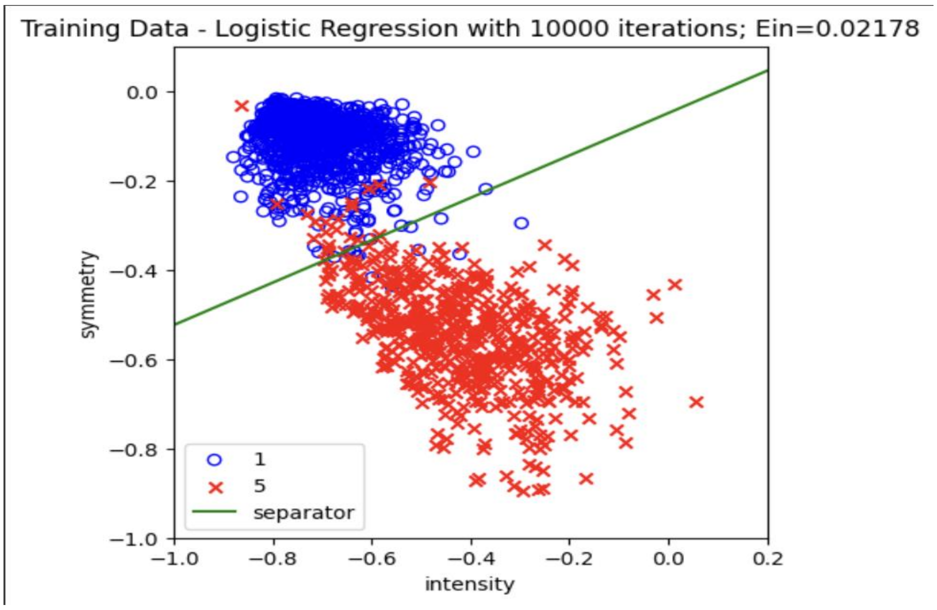
```
{'Point 1': {'Starting Point': (0.1, 0.1),
  'Minimum Location': (-0.24182894588827972, -3.265192092903909e-36),
  'Minimum Value': -2.8790846587644263},
 'Point 2': {'Starting Point': (1, 1),
  'Minimum Location': (0.7252678803578846, 0.9831635693235358),
  'Minimum Value': -0.9286447086312588},
 'Point 3': {'Starting Point': (0.5, 0.5),
  'Minimum Location': (0.24181813075114938, 0.49168225906848034),
  'Minimum Value': -2.633242590975637},
 'Point 4': {'Starting Point': (0.0, 0.5),
  'Minimum Location': (0.2418181307511494, 0.4916822590684804),
  'Minimum Value': -2.633242590975637},
 'Point 5': {'Starting Point': (-0.5, -0.5),
  'Minimum Location': (-0.7253691676028068, -0.49159419340046656),
  'Minimum Value': -1.6660267055389741},
 'Point 6': {'Starting Point': (-1, 1),
  'Minimum Location': (-1.2084759495263577, 0.9827889176425988),
  'Minimum Value': 1.005161575979315}}
```

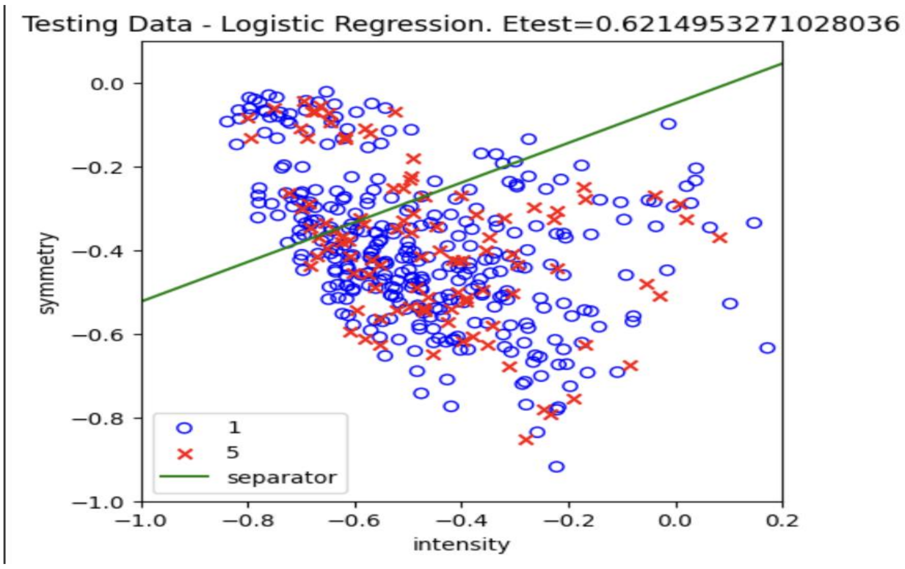
(2) [600 points] Classifying Handwritten Digits: 1 vs. 5.

Implement logistic regression for classification using gradient descent. Find the best separator using the training data only (using ZipDigits.train). Use the data and features you generated for solving HW2. For each example, use the two features you computed in HW2 as the inputs; and the output is +1 if the handwritten digit is 1 and − 1 if the handwritten digit is 5. Once you have found a separator using your classification algorithm:

(a) Give separate plots of the training data (ZipDigits.train) and test data (ZipDigits.test) which display the data points using the two features you computed in HW2, together with the separator.

Ans-

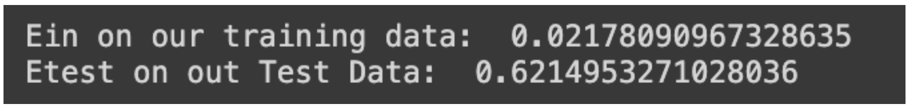




(b) Compute Ein on your training data (ZipDigits.train) and Etest, the error of your separator on the test data (ZipDigits.test).

Ans-

E_{in} = 0.02178090967328635
E_{test} = 0.6214953271028036

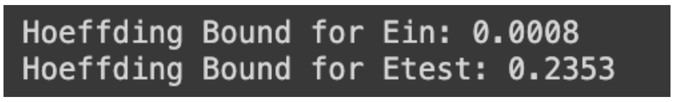


```
Ein on our training data:  0.02178090967328635
Etest on out Test Data:  0.6214953271028036
```

(c) Obtain a bound on the true out-of-sample error (Eout). You should get two bounds, one based on Ein and another based on Etest. Use a tolerance of δ = 0.05. Which is the better bound?

Ans-

bound_E_{in} = 0.0008
bound_E_{test} = 0.2353



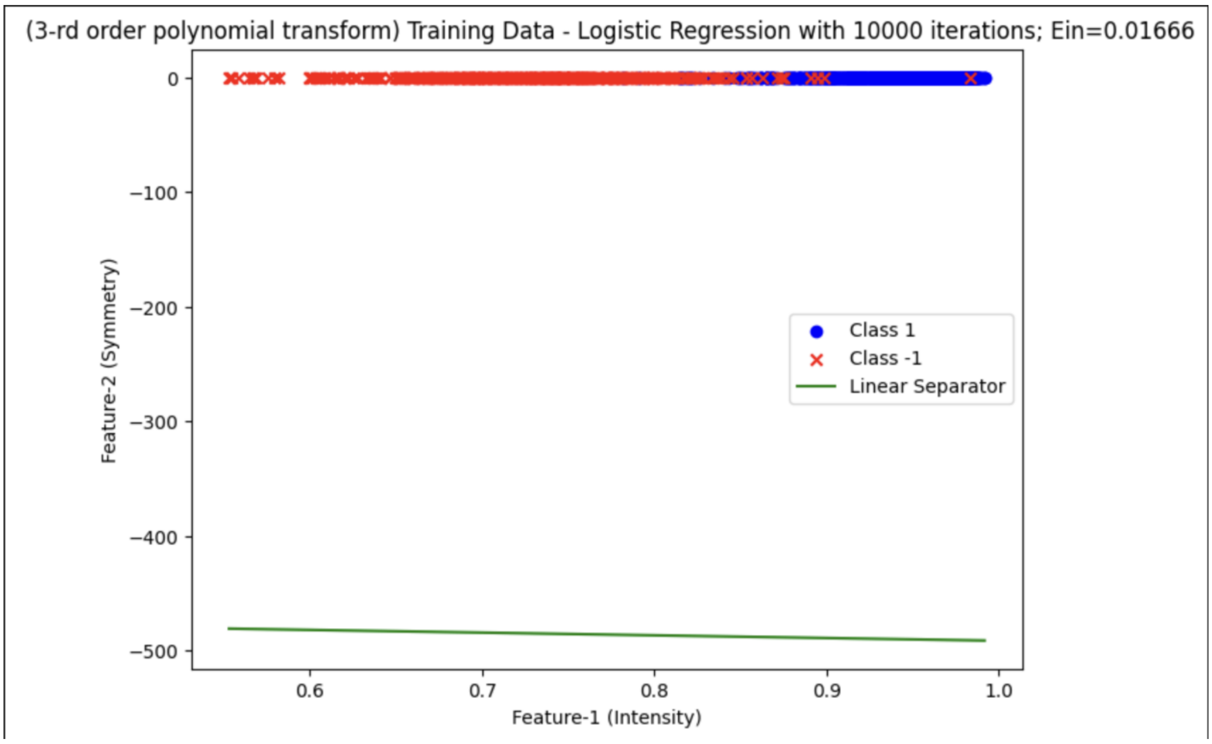
```
Hoeffding Bound for Ein: 0.0008
Hoeffding Bound for Etest: 0.2353
```

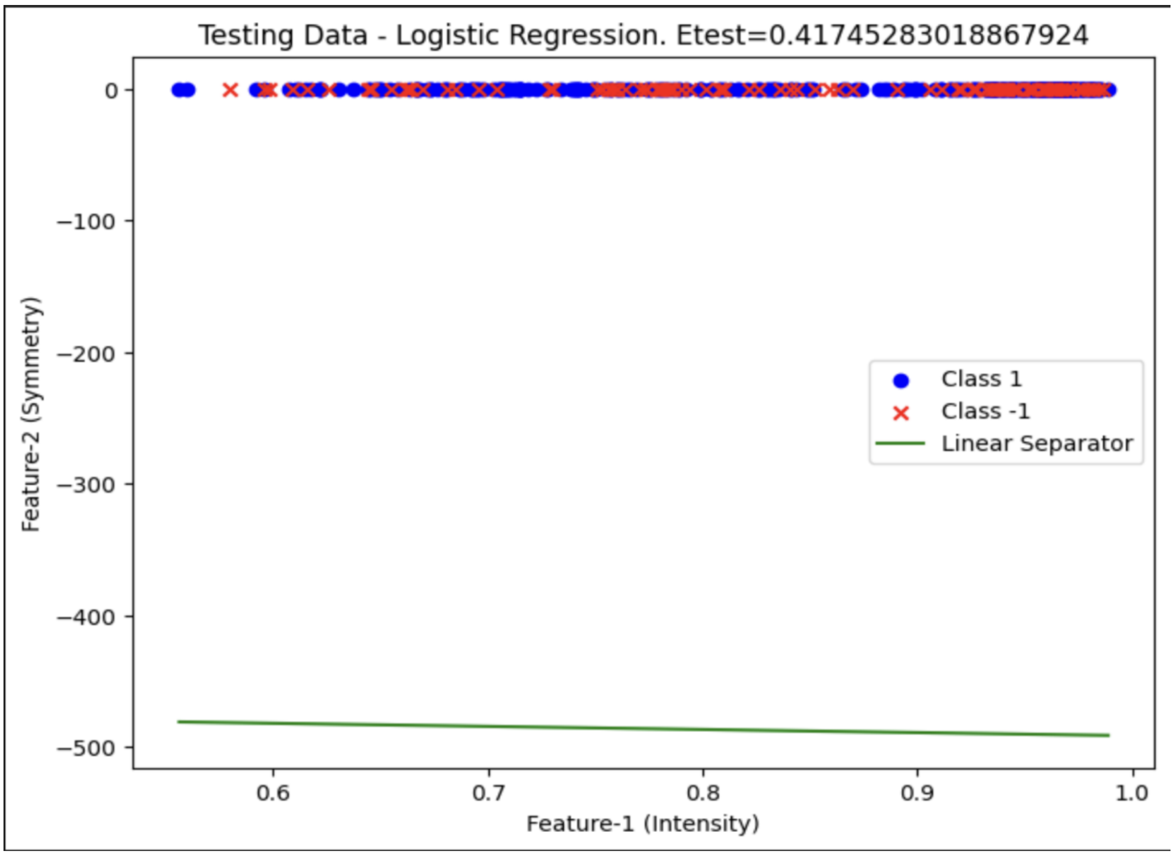
Explanation:-

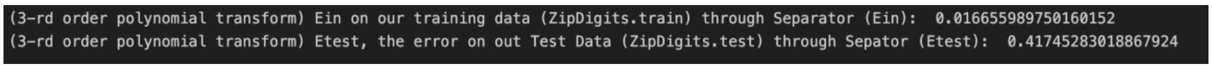
Here, the Hoeffding Bound is smaller for Ein at 0.0008, suggesting a stronger and more reliable indication of the true error we might expect with new data, as opposed to the larger bound for Etest. This means, according to the Hoeffding bounds determined, Ein offers a more solid prediction about the real, out-of-sample error for this scenario.

(d) Repeat parts (a)-(c) using a 3rd order polynomial transform.

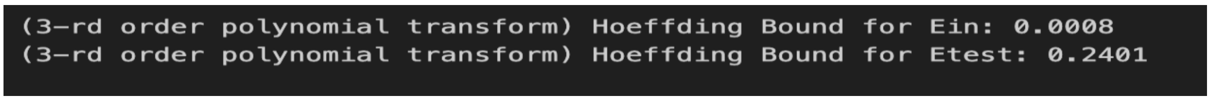
Ans-







```
(3-rd order polynomial transform) Ein on our training data (ZipDigits.train) through Separator (Ein):  0.016655989750160152
(3-rd order polynomial transform) Etest, the error on out Test Data (ZipDigits.test) through Sepator (Etest):  0.41745283018867924
```



```
(3–rd order polynomial transform) Hoeffding Bound for Ein: 0.0008
(3–rd order polynomial transform) Hoeffding Bound for Etest: 0.2401
```

In this scenario:

- For a 3rd-order polynomial transformation, the Hoeffding Bound for Ein is 0.0008.
- For the same transformation, the Hoeffding Bound for Etest is 0.2401.

The significantly lower Hoeffding Bound for Ein (0.0008) in comparison to that for Etest (0.2401) indicates a tighter constraint on the in-sample error, implying a reduced likelihood of deviation from the true in-sample error.

Hence, these findings lead to the conclusion that the Hoeffding Bound for Ein is the more favourable indicator in this case.

(e) Which model would you deliver to the USPS, the linear model with the 3-rd order polynomial transform or the one without? Explain.

Ans -
When deciding on the optimal model for the USPS, it's essential to weigh both models' performance metrics.

Comparison of Models:

Normal Logistic Regression (Linear Model without Polynomial Transformation):
- Training Error (Ein): 0.02498
- Test Error (Etest): 0.39387

Linear Model with 3rd Order Polynomial Transformation:
- Training Error (Ein): 0.01666
- Test Error (Etest): 0.41745

The objective is to minimize classification errors, where lower values for Ein and Etest are preferred. Observing the performance, the model with the 3rd order polynomial transformation exhibits superior training data performance (lower Ein) but underperforms on the test data (higher Etest) relative to the normal logistic regression model.

Decision Criteria:

- Focus on Test Data Performance (Generalization): If the USPS prioritizes generalization to new, unseen data, the normal logistic regression presents a more robust option.
- Focus on Training Data Performance: If the goal is to excel in training data performance, the polynomial-transformed model may be the choice, despite potential overfitting risks.

Key Considerations:

- Overfitting: The polynomial model may overfit the training data, leading to poorer performance on unseen data.
- Generalization: Normal logistic regression tends to generalize better, making it potentially more reliable for unseen data.
- Model Complexity: A simpler model, like the linear one, might be preferable if it delivers comparable results, considering the trade-off between complexity and performance.

Conclusion:

The choice hinges on USPS's specific needs and priorities. For better generalization to new data, normal logistic regression is advisable. However, if the aim is to leverage complex patterns in the training data, albeit with a risk of overfitting, the model with the 3rd order polynomial transformation could be considered.

Google Colab: -
https://colab.research.google.com/drive/1sUCytDZC3akrkFezAbuEJdVYfOcryRhD#scrollTo=JjZzPPaaw6Ca

Collaborator – Sagar Sidhwa