Programming Assignment 1 - Report

Wesley Jameson Watkins (wjw16)

1. SVM

a)

This part of the question is answered in the file "./SVM/svm.py." The algorithm is implemented in the "train" function, where the convex optimization problem is solved using the CVXPY library. The optimization problem is then solved for many different values of C (dependent on the "trials" parameter given by the caller). For each trial, the error is calculated, and the value of C that produces the smallest error is chosen.

b)

The data for this part is generated in the file "./data.py". Then, the file "./SVM/b.py" trains the SVM model on the generated data and prints out the answers to each of the questions asked, along with the necessary plots.

Support Vectors: [(0.8140338, -2.3116527), (0.3581546, 1.1395161), (-0.1110767, -1.3506412)]

Margin: 0.8556

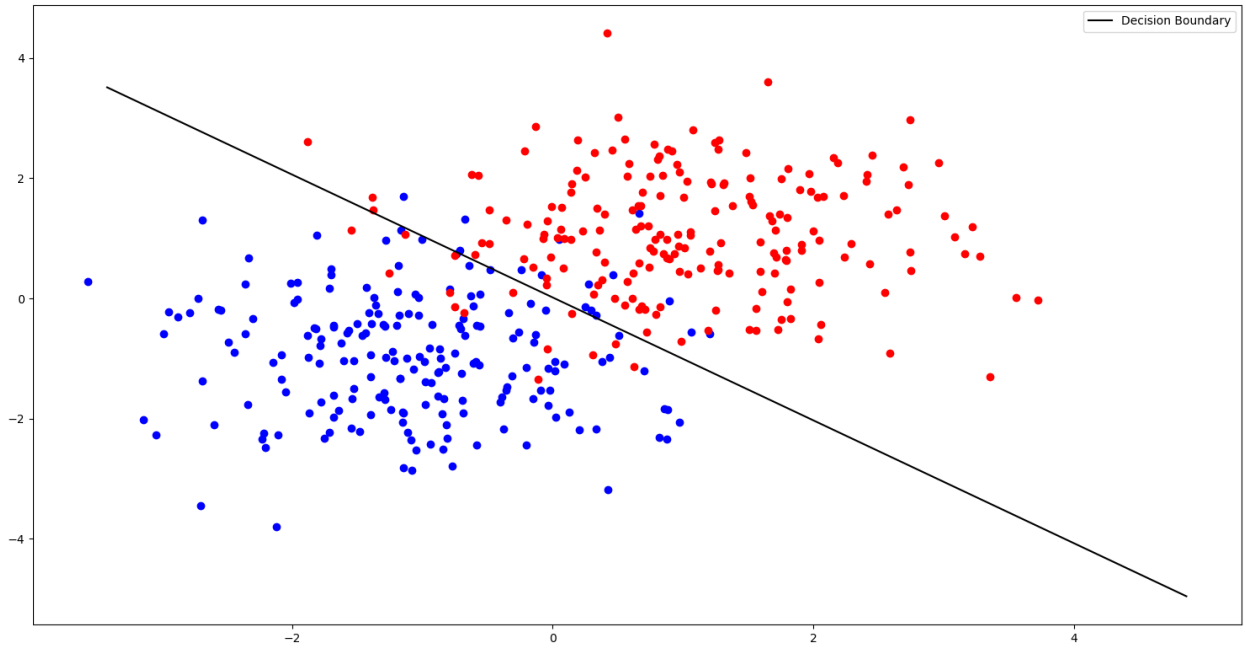
Leave One Out Cross-Validation Error: 0.0747

Here's tradeoff of C between the margin and the misclassification error:

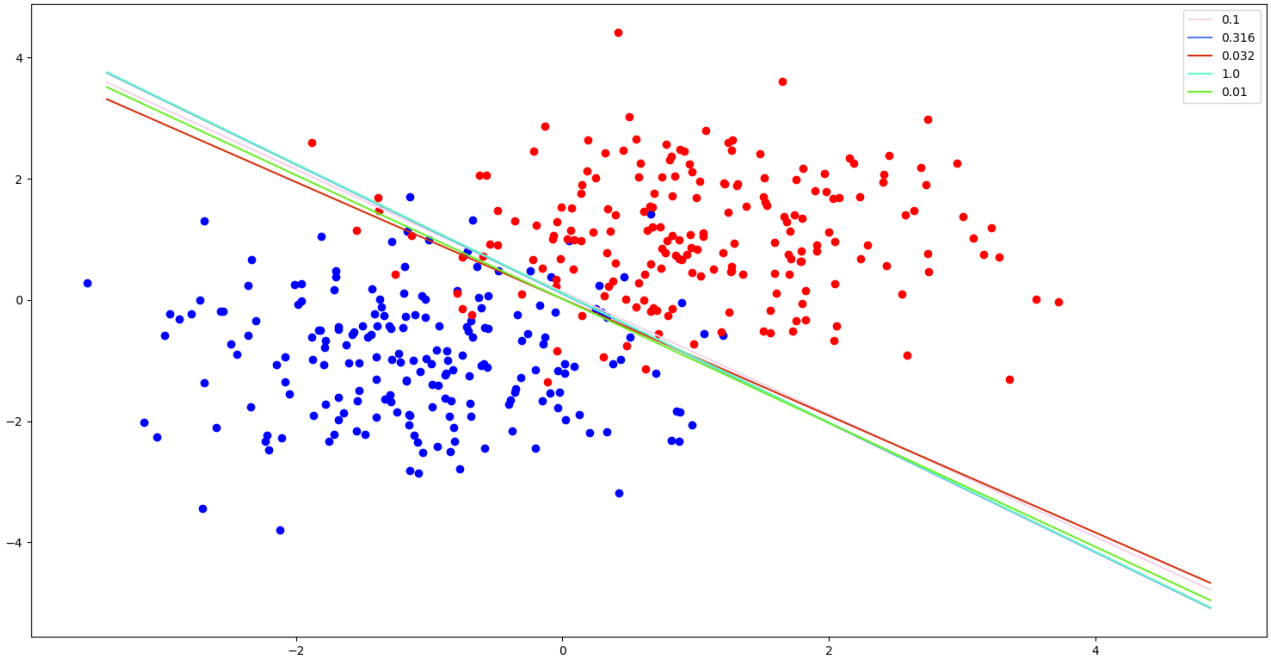
C	Margin
0.01	0.855607
0.031623	0.588880
0.1	0.491088
0.31623	0.453481
1	0.448914

C	Error (%)
0.01	8.25
0.031623	8.00
0.1	7.25
0.31623	7.50
1	7.50

Decision Boundary Plot:



Decision Boundary w/ Different C-Values:



c)

This part of the question is answered in "./SVM/c.py". The program starts by reading in the MNIST training and testing datasets and removing any non-0 and non-1 labeled points. Then, for training purposes, all the points labeled 0 are changed to -1. The SVM is then on trained on the training data. Once trained, the SVM runs on the test data and achieves the following results:

Points Misclassified: 234 out of 2115

Generalization Error: 0.111

Accuracy: ~89%

2. Regression

a)

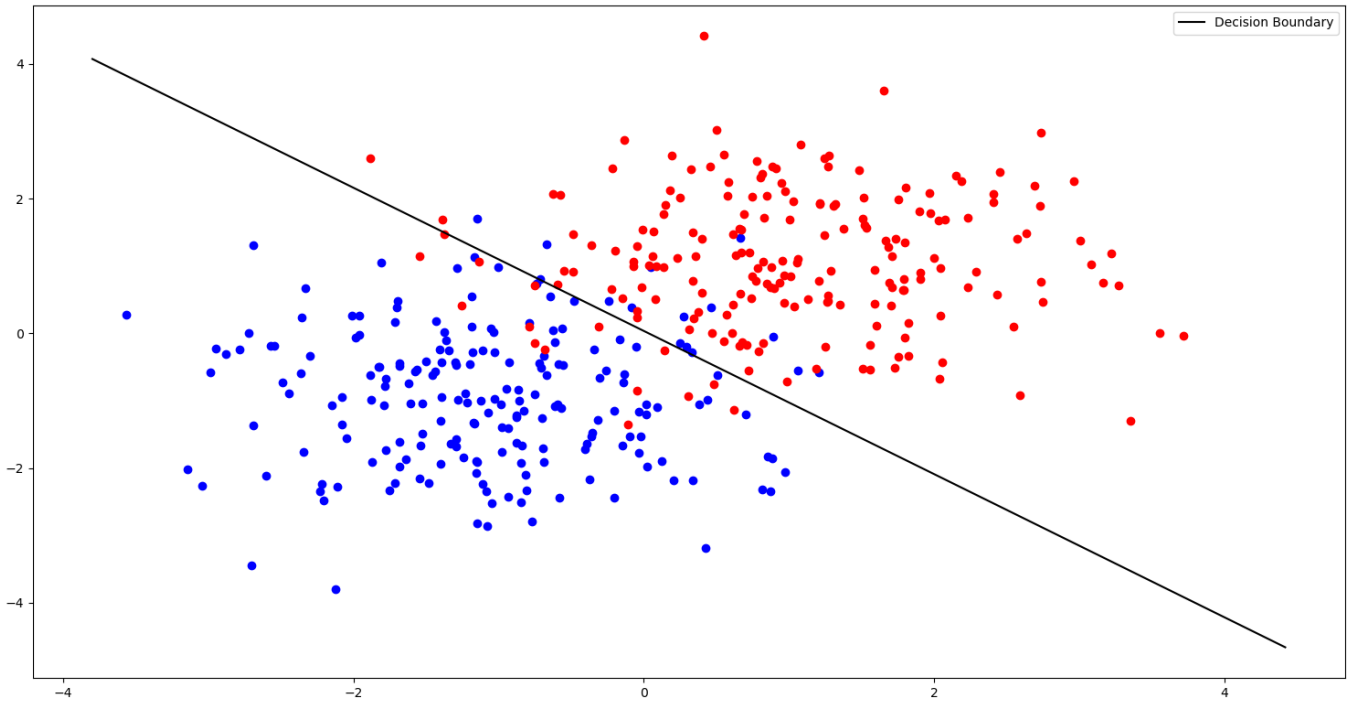
This part of the question is answered by two separated files. The implementation of linear regression is given in the file "./Regression/linear.py". Then, the file "./Regression/a.py" generates data (in the same way as question 1, part b), the model is trained on the training data, and the results from after being run on the test data are as follows:

Points Misclassified: 33 out of 400

Generalization Error: 0.0825

Accuracy: ~92%

Leave One Out Cross-Validation Error: 0.0747



b)

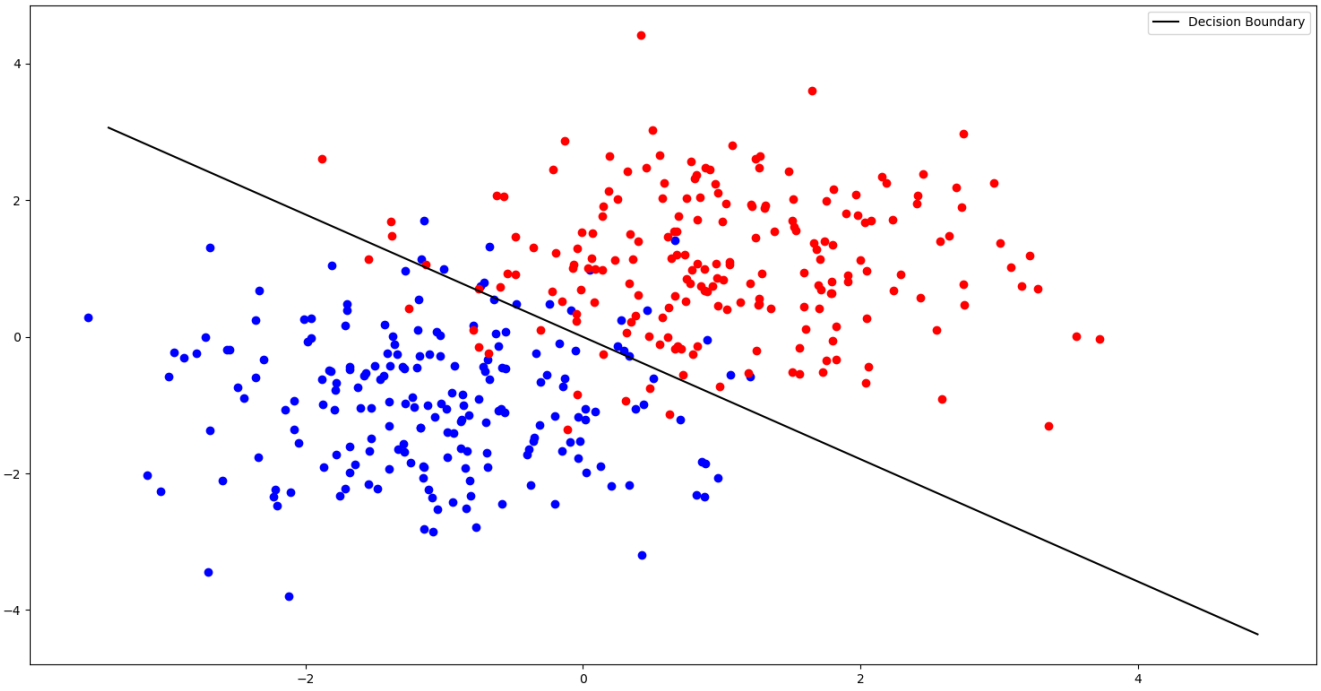
This part of the question is answered by two separated files. The implementation of logistic regression is given in the file "./Regression/logistic.py." Then, the file "./Regression/b.py" generates data (in the same way as question 1, part b), the model is trained on the training data, and the results from after being run on the test data are as follows:

Points Misclassified: 32 out of 400

Generalization Error: 0.08

Accuracy: ~92%

Leave One Out Cross-Validation Error: 0.0774



c)

This part of the question is answered in "./Regression/c.py". The program starts by reading in the MNIST training and testing datasets and removing any non-0 and non-1 labeled points. Then, for training purposes, all the points labeled 0 are changed to -1. The SVM, linear regression, and logistic regression models are then each trained on the MNIST training data. After training is complete, the test data is run on each model with the following results:

Support Vector Machine:

Points Misclassified: 234 out of 2115

Generalization Error: 0.111

Accuracy: ~89%

Linear Regression:

Points Misclassified: 13 out of 2115

Generalization Error: 0.006

Accuracy: ~99%

Logistic Regression:

Points Misclassified: 29 out of 2115

Generalization Error: 0.014

Accuracy: ~99%

Therefore, the SVM performed the worst while the linear regression model performed the best.