CS 434
Implementation Assignment Report #1

Adrian Henle - Onur Sasmaz - Tristan Jong
A# - 932514094 - T#

Linear Regression

1. Learned weight vector:

[[3.67103960e+01]
[-1.10220511e-01]
[4.25270181e-02]
[9.94268803e-03]
[4.03688262e+00]
[-1.81193844e+01]
[3.91213593e+00]
[-3.24572263e-03]
[-1.61764599e+00]
[3.51469633e-01]
[-1.35490385e-02]
[-8.88849879e-01]
[9.33221332e-03]
[-5.87431614e-01]]

2.

Training data ASE:
21.6359643629017

Testing data ASE
19.442083367047132

The training data ASE is larger. We somewhat expected the testing data ASE to be larger since the training data is where the learned linear regression came from, but apparently the testing data fits the learned curve better than the training data does.

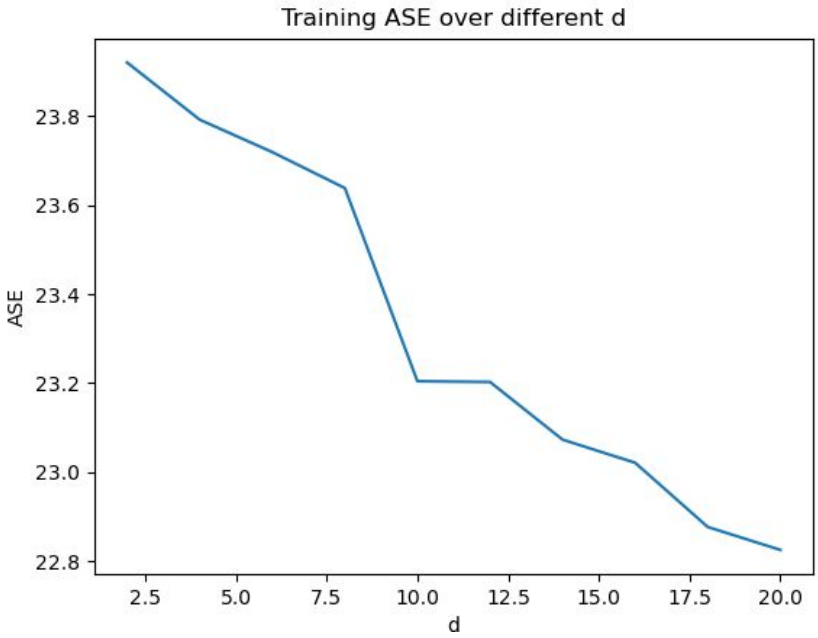
3.

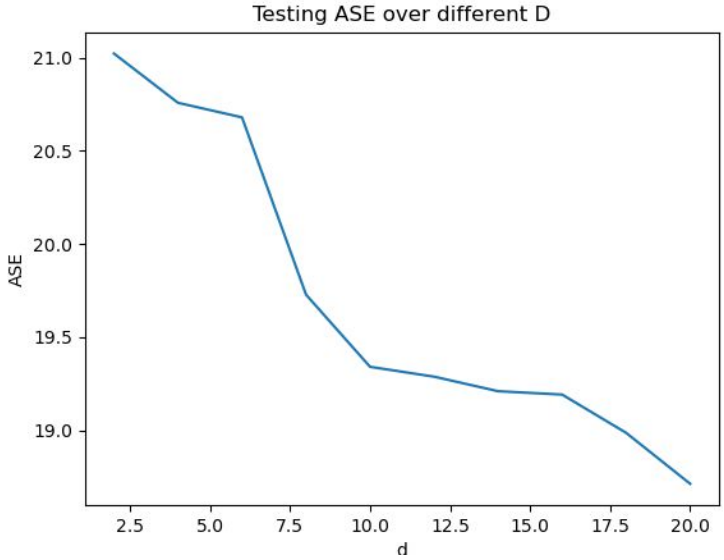
ASE over the training data:
23.96275049259437

ASE over the testing data:
21.108458913842224

Removing the dummy variables makes the ASE larger. Since we have removed one of the constants of the data set, it makes sense that the linear regression model would become slightly more inaccurate because the dummy data was an extra constant feature added, and since it was always 1, it reduced the error of the model.

4.



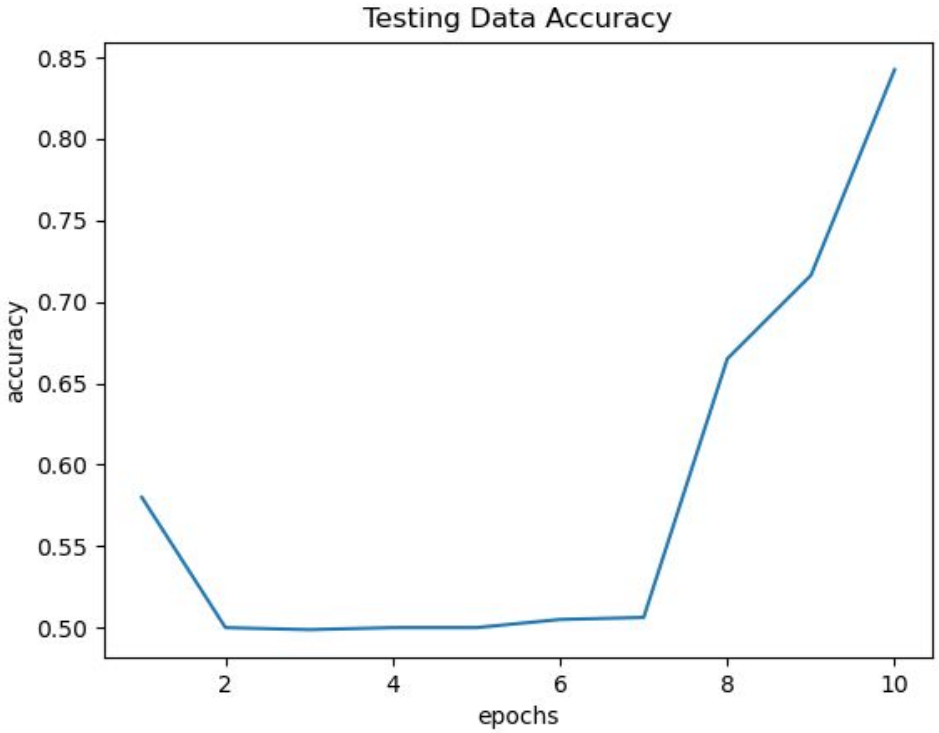


As we introduce more features that are sampled from the standard normal distribution, the ASE starts to decrease somewhat linearly. This indicates that as we add more features, assuming they do follow standard normal distribution, the accuracy of our linear regression model will increase linearly in regards to how many features we have introduced. This is most likely because as we introduce more features, we are introducing more ways for our linear regression model to be correctly developed to follow all of the data points for each sample in the data set.

Logistic regression with regularization

1. The accuracy for both the training and testing data tends to dip for the first few gradient descent iterations, but begins to increase rapidly after the initial dip.
The learning rate used was less than 0.1. Higher learning rates would cause overflow errors at runtime with the python script, even after normalization of the data sets. Execution will take some seconds depending on how many epochs are being iterated through.

Training and testing plots, respectively, for learning rate = 0.1 over 10 iterations:


Testing Data Accuracy
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
accuracy
2
4
6
8
10
epochs