Disclamer: I have put comments on the models and programs to explain them

I went in more detail in the notebook

Report 1

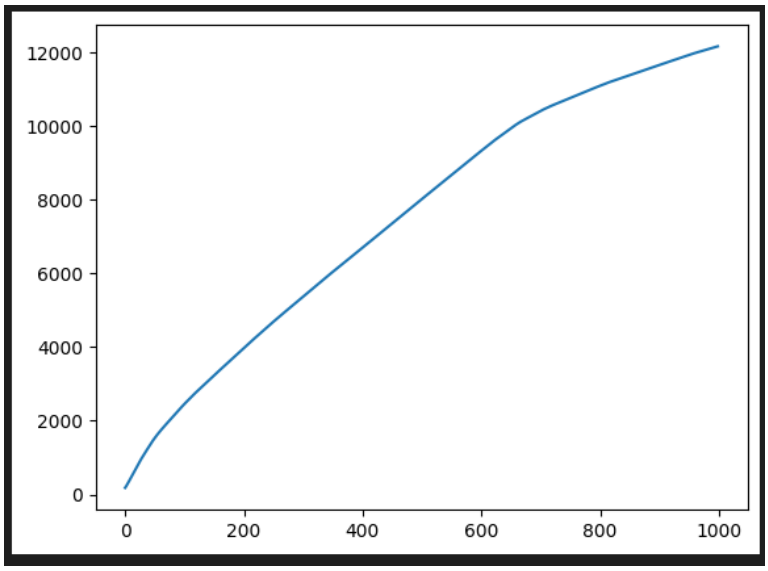
1. Generating data

2. Set par0

3. Train the model

4. plot the graph ℓ(D, λ(t)), versus t, t = 1, . . . , Tepochs, to show that the value of the objective function decreases during training

For question 1, the learning rate determines how quickly we update the parameters; if it is too large, we may overshoot. Whilst in contrast if the rate is too small many iterations will be need for coverage to the best values. In this case our learning rate is 0.01 with this initially the score is lower, however after 100 epochs, the score is near equal to the first one. An epoch can be defined as a single iteration through our training dataset, a epoch to small may be affected by the learning rate and also lead to underfitting of the curve in the graph.



Report 2

1. we are training using an dataset (only 3 variables)

2. normalise this dataset

3. split it into dtrain and dtest

4. tune T and the learning rate so that the graph reaches a minimum

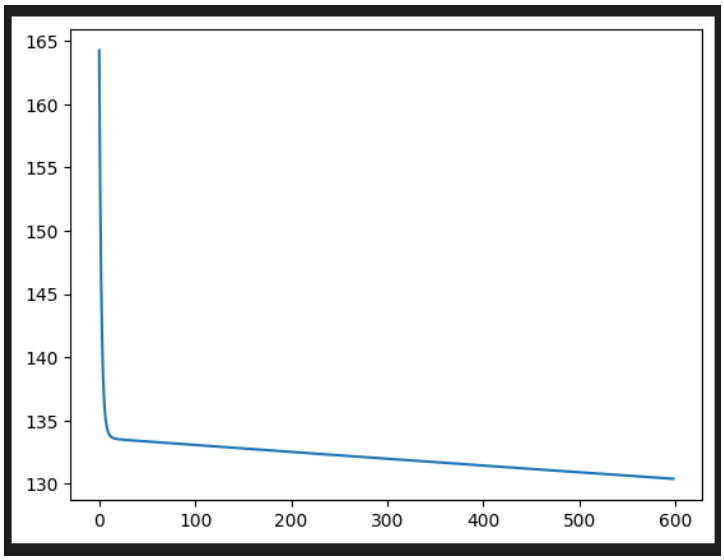
5. and then train it using D train

6. the par0 so d = 3

7. plot ℓ(λ(t), Dtrain) versus t, t = 0, 1, . . . , Tepochs

We expect ℓ(λ(t+1), Dtrain) ≤ ℓ(λ(t), Dtrain) as in ℓ(λ(t+1), Dtrain) we are estimating λ by minimizing through Tepochs gradient descent updates in contrast to what was done initially. This can be verified as it is shown in rule 3.

From my results I can infer that me graph may appear this way due to my sigma function



Report 3

1. we are training using an augmented dataset (includes origin this time)

2. normalise this dataset

3. split it into dtrain and dtest

5. tune T and the learning rate so that the graph reaches a minimum

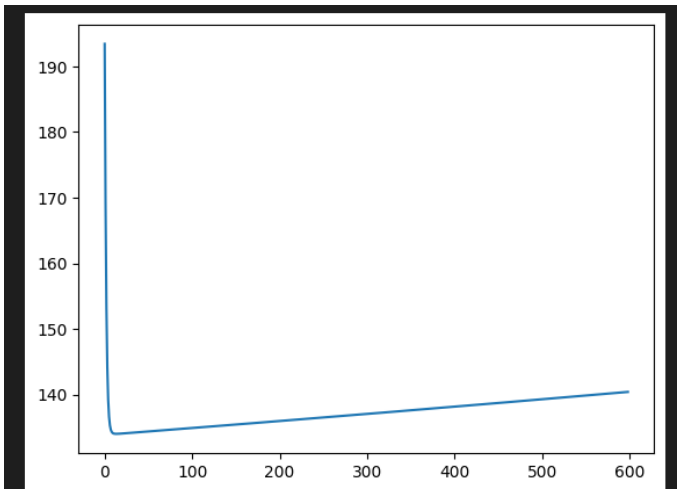
6. and then train it using augmented D train

7. the par0 differs a bit since you have a new variable so d = 4

8. comparing ER(original par, D test) and ER(augmented par, D test)

9. also comparing ER(original par, D train) and ER(augmented par, augmented D train)

You obtain the lowest error when dealing with ER(λ˜∗, Dtest). The inclusion of origin does improve the models performance as tested in Question 3 section.



Report 4

1. Using the original model (without nominal features) and the original (normalised)

2. normalise dataset

3. set par0

5. Create 10 different splits equal size

6. Set param

Could not finish so will make predictions -

The distinction between training and testing data sets is evident: training data trains the model, whereas testing data checks (tests) whether the model was built correctly. Although, you can still generate predictions based on their training data. However, I do not a better performance on the training set implies a better performance on the test set, the training set and test set are split separately and I feel the impact on on set does not impact the other. This can be seen in previous exercises where we see that the results of the training set and test set differ.