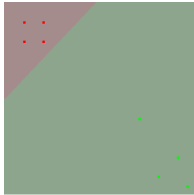


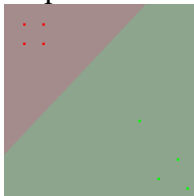
Question 1.

The visualization of the model learned on the UnitTest using 10 iterations of gradient descent, $\text{stepSize}=1.0$, $\text{convergence}=0.005$

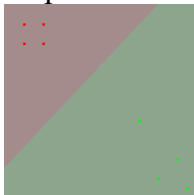
Step 10:



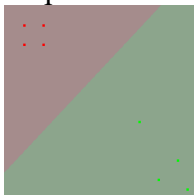
Step 20:



Step 30:



Step 36:



Question 2.

Hand in your completed LogisticRegression.py &

EvaluateBinaryProbabilityEstimate.py. Make sure they are clear.

If the TA has a hard time finding and spot-checking the key components (e.g. gradient calculation in `__gradientDescentStep`, the logic in `incrementalFit`, and loss calculation) they will have to deduct credit.

See Attached Files.

Question 3.

Tune the hyperparameter 'convergence' by trying [0.01, 0.001, 0.0001, 0.00001] (with stepSize of 1.0). Produce a table showing:

<convergence parameter>, <steps to convergence>, <validation set accuracy>

for each setting of the convergence hyperparameter.

Convergence Parameter	Steps to converge	Validation set accuracy
0.01	12	0.8409
0.001	60	0.8665
0.0001	191	0.9269
0.00001	530	0.9269

Question 4.

Describe in 2-3 sentences your interpretation of the output of this hyperparameter sweep. Include a justification for trying more parameters or stopping there. If you do think you should try more values of the 'convergence' hyperparameter, what is the next one you would try?

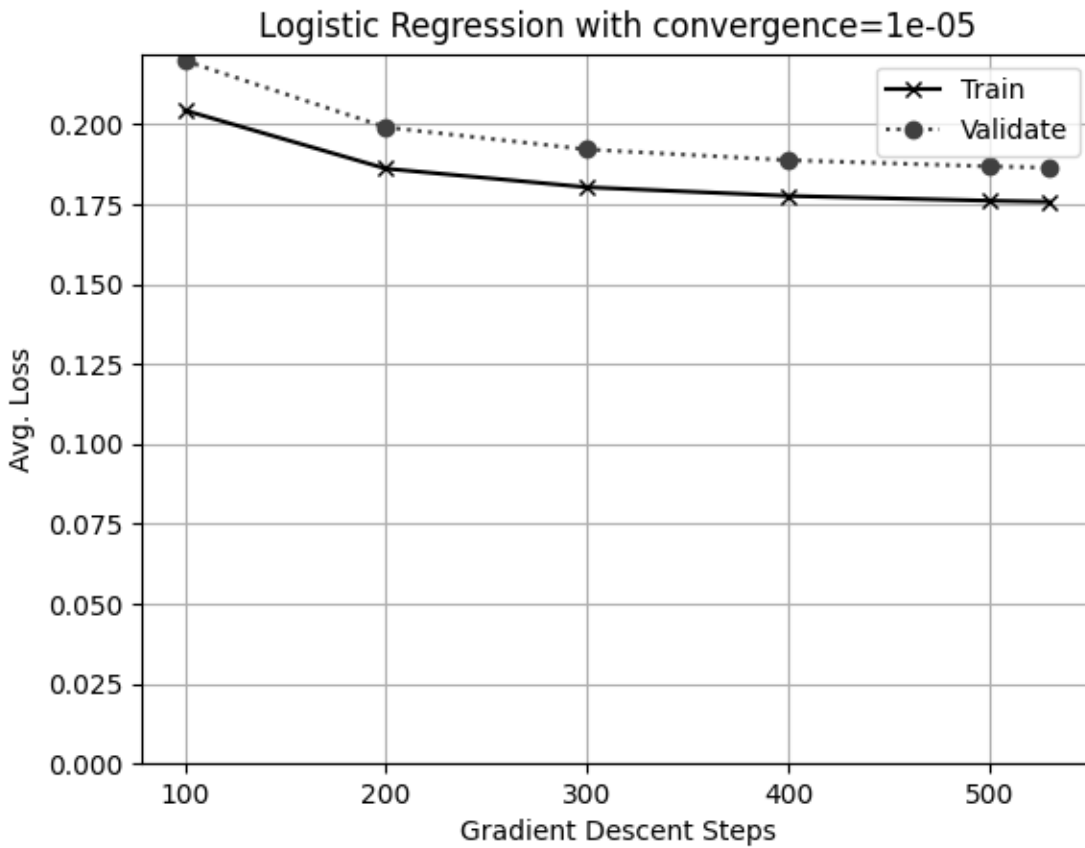
Using a smaller convergence parameter allows for the regression to achieve a higher accuracy at the expense of higher number of steps to converge. In this example, the 2 of the smallest parameters result in the same accuracy with different losses on validation sets. This model is likely to not improve in accuracy with smaller convergence parameters because the loss has almost flatlined by the end of the 530th step.

Question 5.

Produce a plot showing the training set loss and the validation set loss every 100 steps during the run

with the best convergence value you found in the parameter sweep and stepSize of 1.0. In no more than 100

words describe the difference between the training and validation loss. Which set does the model have better loss on? Why?



The best model is the one with convergence = 0.00001 because of lower log loss on the validation set. At the end of the training, the train set loss is at 0.1758 whereas the validation set loss is at 0.1865. The difference in loss between the training set and validation set results due to the model most likely being overfitted with manually crafted vocabulary.