People who helped me on the homework: Madeline (never got last name), attended office hours and had some help from various students.

Honor Code:

“I certify that all solutions are entirely in my own words and that I have not looked at

another student’s solutions. I have given credit to all external sources I consulted.”

Signature:



Question 2:

A close-up of a paper with mathematical equations

Description automatically generated

A piece of paper with writing on it

Description automatically generated

Question 2e Graph:

A diagram of blue dots

Description automatically generated

Question 3:

A screenshot of a computer code

Description automatically generated

Question 4:

A graph with orange lines

Description automatically generated

A graph of a graph with numbers and lines

Description automatically generated with medium confidence

Please reference the code appendix at the bottom of this pdf for the code.

Question 5:

The values I used were: [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]

The accuracy of each value is: [0.8677, 0.8334, 0.8404, 0.8501, 0.8477, 0.8436, 0.8472, 0.8452]

Therefore, the best C value is 0.0001.

Please reference the code appendix at the bottom of this pdf for the code.

Question 6: (Code Appendix)

My C values are: [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000]

The results of those C Values are: [0.7172413793103448, 0.7559220389805097, 0.7991004497751125, 0.8125937031484257, 0.8299850074962519, 0.8410794602698651, 0.8461769115442278, 0.8515742128935532, 0.8233883058470765, 0.7904047976011993, 0.7940029985007495]

Therefore, the best C Value is 10.

Please reference the code appendix at the bottom of this pdf for the code.

Question 7:

Kaggle Score: Mnist: 0.915 Spam: 0.850

What I did to improve my Kaggle Score:

For Mnist, tinkering around with the C values, which I ended up with C=10, I also messed around with other hyperparameters I found in the documentation, but it seems like they only hurt or didn’t help the accuracy. The largest bump in accuracy was letting the model train on the full 50,000 points rather than the 10,000 points.

For the Spam dataset, it was stuck for a very long time at around 79%-80% and no hyperparameter was helping, eventually the only thing that brought it up so much was modifying featurize.py, by adding in a lot of words/phrases that I thought were used by spam emails a lot. I ended up looking at a bunch of spam emails.

Please reference the code appendix at the bottom of this pdf for the code.

Code Appendix:

Question 4:

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

Question 5:

A screenshot of a computer code

Description automatically generated

Question 6:

A screenshot of a computer code

Description automatically generated

Question 7:

A screenshot of a computer code

Description automatically generated

A computer screen shot of a program

Description automatically generated

Some websites I went to for help:

* <https://numpy.org/doc/stable/reference/index.html>
  + Went to this website a lot to look up numpy functions.
* <https://www.activecampaign.com/blog/spam-words>
  + This was looking for finding out notable key words to filter out of the spam email points.