***italicized text*Name**: [Elaina(Yaogui) Huang]

Net ID: [yh4310]

1a & 1b

1a

$$g(z_0) = \frac{e^{w_0^T x}}{e^{w_0^T x} + e^{w_1^T x}}$$

$$g(z_1) = \frac{e^{w_1^T x}}{e^{w_0^T x} + e^{w_1^T x}}$$

1b

$$(y=1 \mid z) = \frac{1}{1+e^{-(w^T x)}} \qquad z = w^T x$$

$$(y=0 \mid z) = \frac{e^{-(w^T x)}}{1+e^{-(w^T x)}}$$

2a.

LASSO(L1), will reduce numbers of parameters by zeroing-out some coefficients, making the w sparse.

2b.

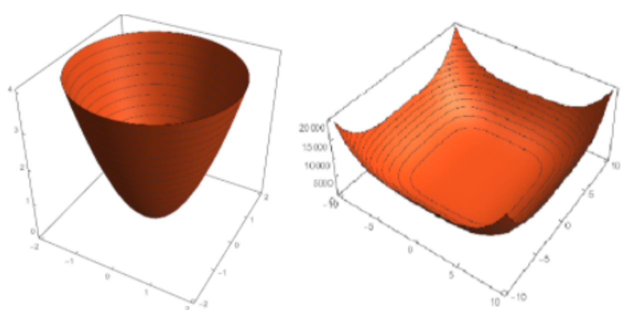
Ridge(L2), will keep all the numbers of paremeters, will penalize large coefficients, making many "small" coeffs.

2c.

Group LASSO, will shrink a group(in this case, the negative number group). Unless the whole group shrinks to 0, there might still have small negative coeffs exist, but ensure no large negative coeffs exist. Also by making positive number into a different group and assign lamba=0.

2d.

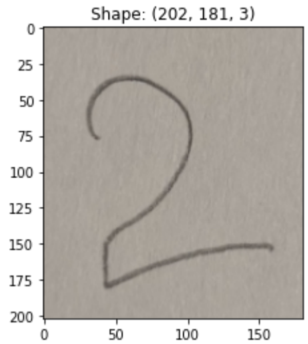
Bridge regression. For instance L4, based on the following visualization in a comparison between L2 and L4. It's clear to see that when the norm-level increases, the penalty shape is more flatten and smooth, eventually will adjust every coeffs to be similar like the previous ones.



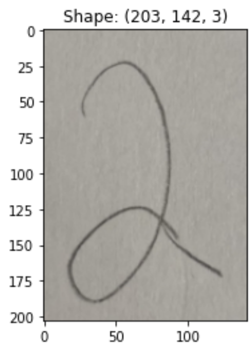
Left: L2, Right: L4

The visualization of your test image before pre-processing.

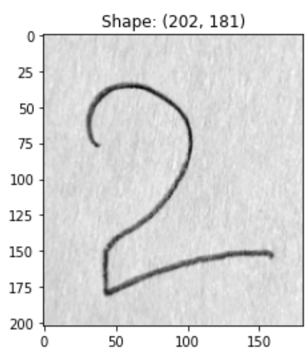
Good 2

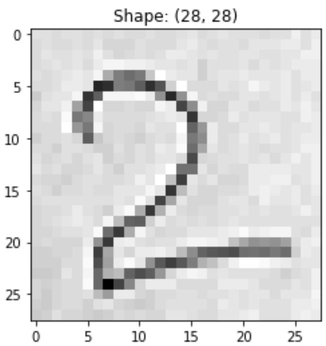


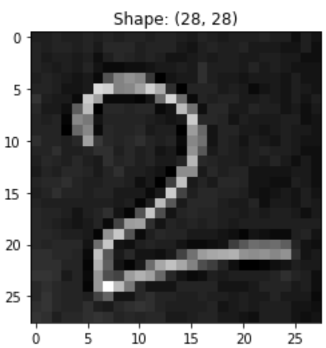
Bad 2

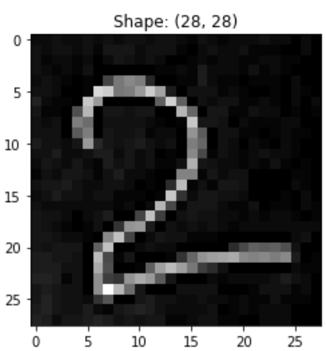


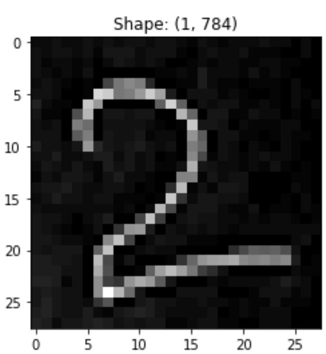
Good 2

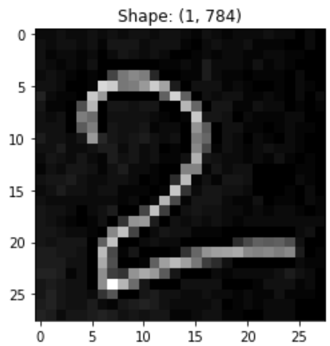



Shape: (28, 28)
0
5
10
15
20
25

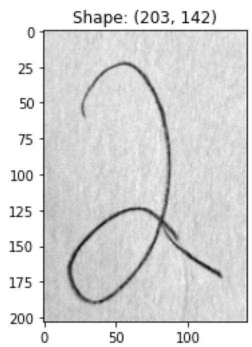

Shape: (28, 28)
0
5
10
15
20
25

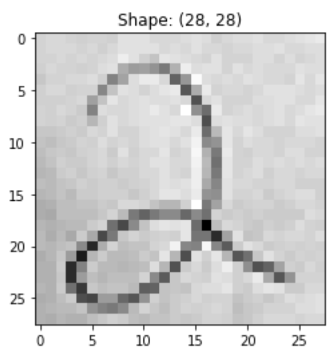

Shape: (28, 28)
0
5
10
15
20
25

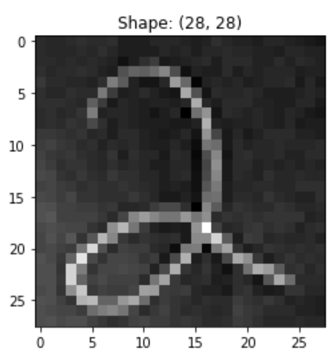

Shape: (1, 784)
0
5
10
15
20
25



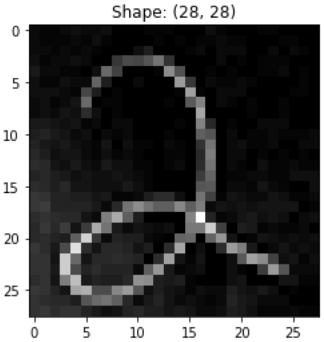
Bad 2



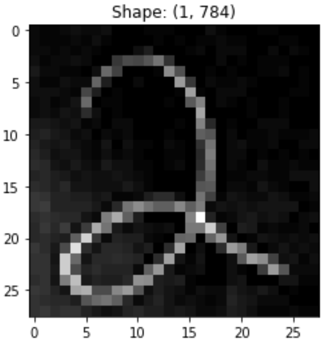




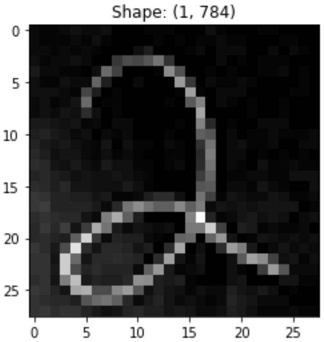
Shape: (28, 28)



Shape: (1, 784)

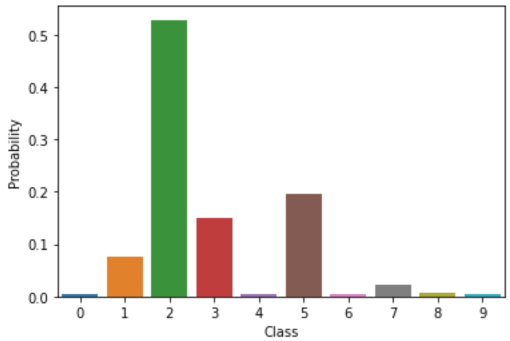


Shape: (1, 784)

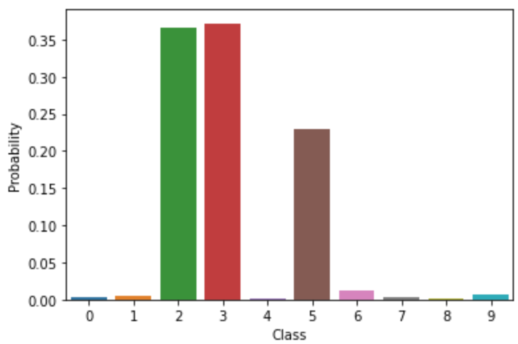


The bar plot showing the conditional probabilities per class for your test image.

Good 2



Bad 2



The predicted class label for your test image.

Good 2

```
test_pred = clf.predict(test_sample)
print("Predicted class is: ", test_pred)
```

```
Predicted class is:  ['2']
```

Bad 2

```
[ ] test_pred = clf.predict(test_sample)
    print("Predicted class is: ", test_pred)
```

```
Predicted class is:  ['3']
```

The figure from the “Explain the model prediction” section.

Good 2

Bad 2

- **In your own words**, list the classes for which the logistic regression predicted a high or moderately high probability. Using the figure from the “explain the model prediction” section, explain *why* the logistic regression estimates that these classes are very likely or moderately likely.
- Explain: how did you know what changes to make to your original drawing to create a modified version that would get a different predicted class label?

- I think for the "good 2", 1-pixel presents tremendous red color on the picture, the model is likely using the negativity connection to identify the possibility of my hand-drawing to be a 1. 2-pixel seems to have the most volume of blue color on the picture, despite those 3 and 5 pixels also seem correlated based on the model, the 2-pixel was chosen to be the finalized indicator. Also, on the 2 and 5 pixels, these are the only two pixels have dark blue color presented. 5-pixel seemingly to have more dark blue pixels, however, also more red pixels than the 2-pixel.
- The "bad 2" is drawn very differently from the "good 2." I specifically made a circle to confuse the model, but the circled-type-2 is still a very common drawing method. In this case, model have very difficult time to differentiate it from 2 and 3, despite that only 2 and 5 pixels have the dark blue color presented. Mostly because the circle, which adds up more degrees on curliness, the model finalized the picture to be a 3.
- Other number pixel has very similar pattern on the different 2s, with 1 and 4 have leading volume of red color. It's not possible to tell from human eye based on looking at the pixels from good and bad 2. The pixel's differentness is very subtle to human eye but very different to the model.
- Originally, I drew the good 2 with angular bottom, because I assume model would look into that area for a sharp turn, and I think my assumption was correct. Because when I drew the circled-2, though the majority of the pixels fitted into model's identifying process for a 2, the model eventually finalized a 3.