Homework 4 - Written Answer Key

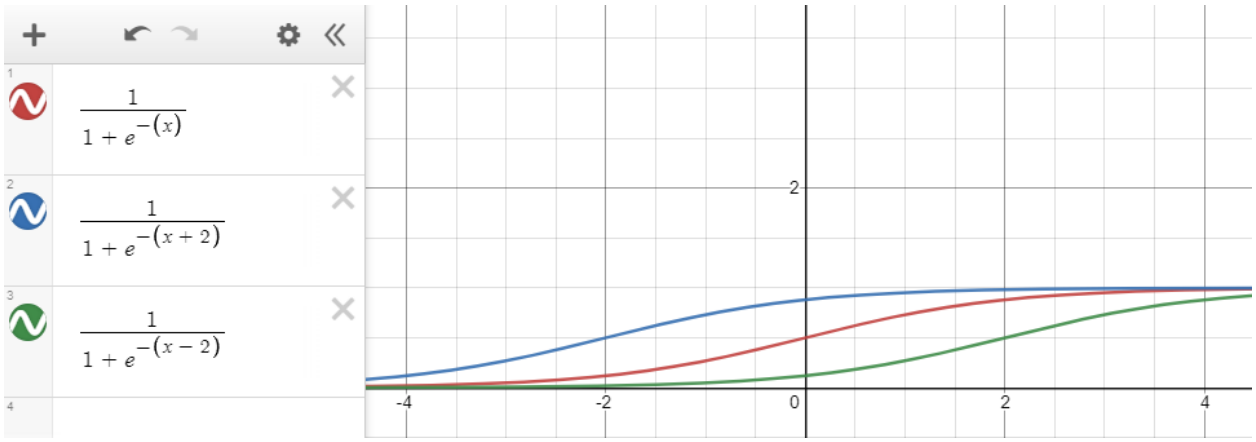
Question 1:

(Question)

How does the logistic function change when w_0 changes? e.g shifted left/right. You can just run some simulations and describe what you notice. (Or state mathematically what happens)

(Answer(s))

When w_0 changes, the midpoint of the line representing the logistic function will shift. If w_0 increases, the midpoint of the line will shift to the left. If w_0 decreases, the midpoint of the line will shift to the right.



Red Line:

$$\frac{1}{1+e^{-(x)}}$$

Blue Line:

$$\frac{1}{1+e^{-(x+2)}}$$

Green Line:

$$\frac{1}{1+e^{-(x-2)}}$$

Question 2:

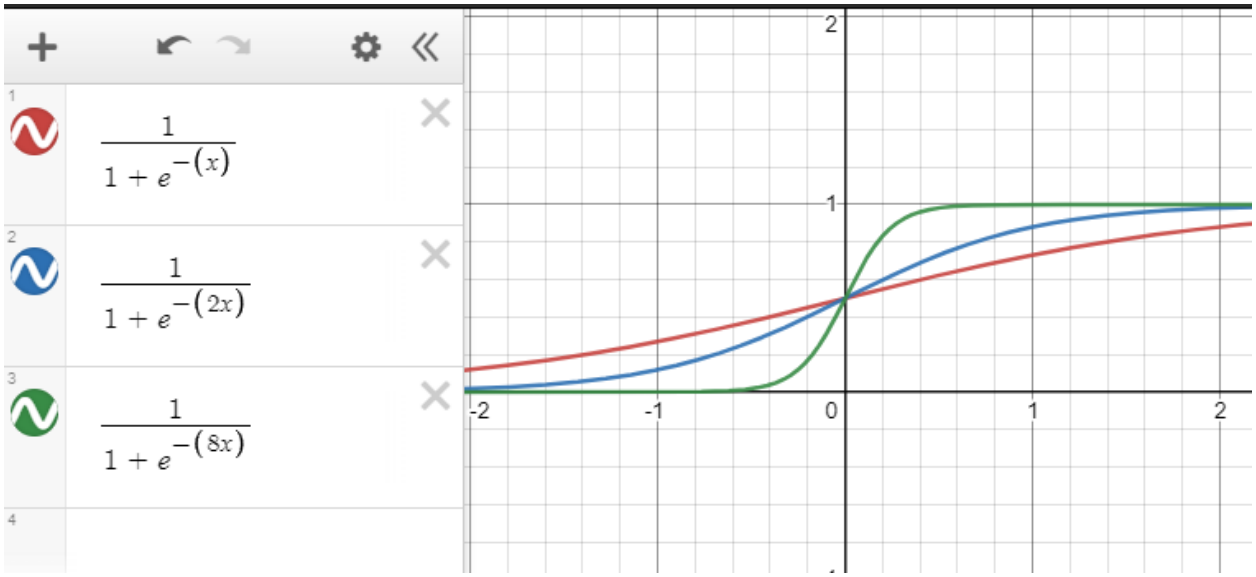
(Question)

How does the logistic function change if you use $\mathbf{w}' = 2\mathbf{w}$ instead of $\mathbf{w}$? You can just run some simulations and describe what you notice. (Or state mathematically what happens.) Try increasing by larger factors than the number 2. Use your observations to argue why a solution with large weights can cause logistic regression to overfit.

(Answer(s))

Using $w' = 2w$ instead of w will not change the predicted class of any sample. This is because the predicted class of a sample depends on the sign of $w^T x$, and multiplying w by 2 does not change the sign of the entire term. However, the magnitude of the value will increase. This means that the magnitude of the number being sent to the sigmoid function will be larger, resulting in a likelihood farther away from 0.5 and closer to either 0 or 1.

Thus, a solution with large weights has the potential to cause logistic regression to overfit because the model will magnify its predictions, becoming more "sure" of its classifications. Samples that would originally be very close to the decision boundary, which would traditionally represent a degree of uncertainty in their classifications, would be pushed farther away from the logistic function, making the function more of a "binary classifier." This will lead to overfitting.



Red Line:

$$\frac{1}{1+e^{-(x)}}$$

Blue Line:

$$\frac{1}{1+e^{-(2x)}}$$

Green Line:

$$\frac{1}{1+e^{-(8x)}}$$

Question 3:

(Question)

Suppose you are in the middle of training a logistic classifier on a data set (below) where the current coefficients are: $\mathbf{w}^T = [0.66, -2.24, -0.18]$.

In the table below $h_{\mathbf{w}}(x) = \frac{1}{1+e^{-(w_0+w_{1:k}^T \mathbf{x})}}$

	x_1	x_2	$h_{\mathbf{w}}(x)$	y
1	0.49	0.09	0.389	0
2	1.69	0.04	0.042	0
3	0.04	0.64	0.613	0
4	1.	0.16	0.167	0
5	0.16	0.09	0.572	1
6	0.25	0.	0.526	1
7	0.49	0.	0.393	1
8	0.04	0.01	0.638	1

(a) What is the equation for the decision boundary?

(b) For a decision boundary of 0.5 create the confusion matrix.

(c) Plot the points on a graph and draw the decision boundary (I would suggest using some sort of plotting library and a image editor)

(d) For the data set above what is the FPR?

(e) For the data set above what is the TPR?

(f) What is the accuracy?

(g) What is the recall?

(h) What is the precision

(i) In logistic regression, we are trying to maximize the log likelihood

$$\ell(\mathbf{w}) = \sum_{i=1}^{N} (y^{(i)} \ln(h(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)})))$$

which is the same as minimizing the error function

$$- \left(\sum_{i=1}^{N} (y^{(i)} \ln(h(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)}))) \right)$$

This quantity is sometimes called the *cross-entropy* of the classifier on the dataset. Using the initial weights, what is the cross-entropy of the classifier on the given training set?

(j) Given $\mathbf{w}$ as described above and $\mathbf{w}' = (1.33, -2.96, -2.77)^T$, which is more likely to be the correct decision boundary given access only to the data above.

(k) Perform one step of gradient ascent using the $\mathbf{w}$ above and learning rate 0.1

(l) How did the data points near the decision boundary contribute to the new value of $\mathbf{w}$?

(m) How did the data points which were correctly classified and far away from the decision boundary contribute to the new value of $\mathbf{w}$?

(n) How did incorrectly classified points contribute to the new value of $\mathbf{w}$?

(o) Did the cross-entropy (error) go up or down after one iteration of the gradient ascent? Is this what you expected? Why or why not?

(Answer(s))

1.

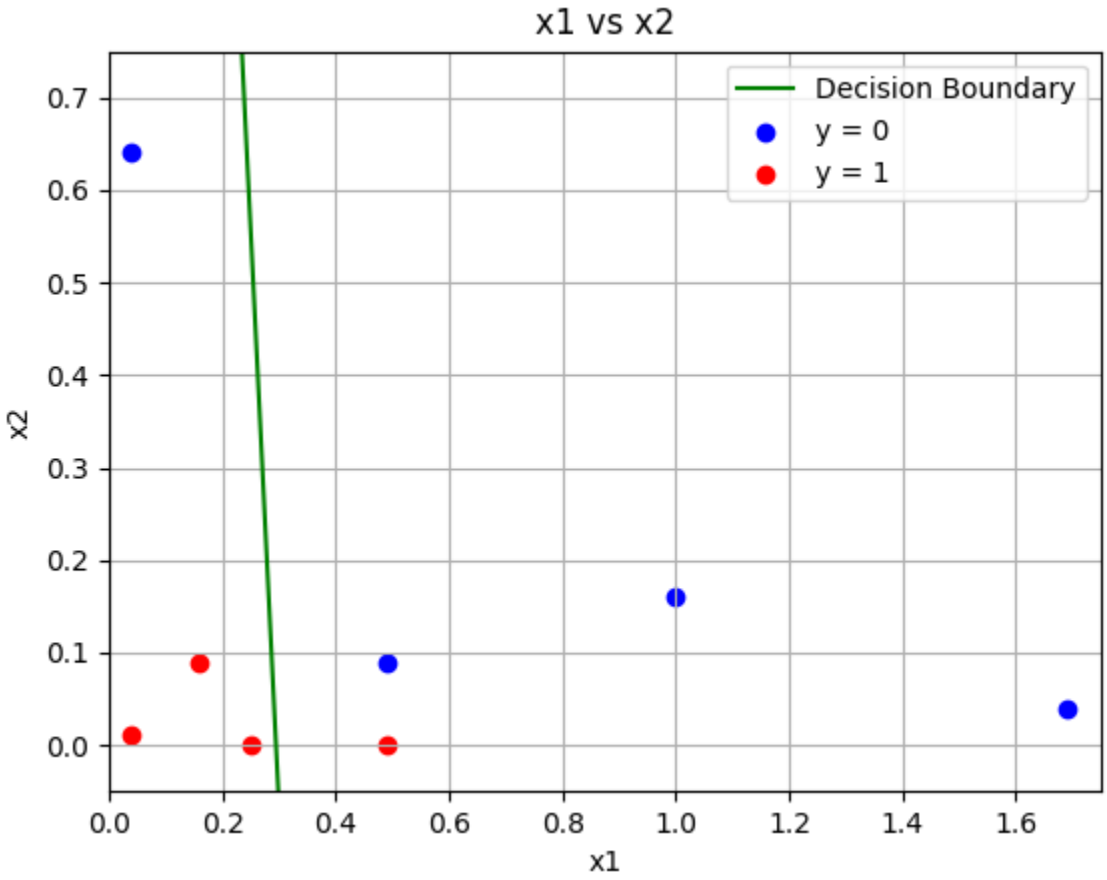
$$w^T x = 0$$

$$0.66 - 2.24x_1 - 0.18x_2 = 0$$

2.

	Actually Positive	Actually Negative
Predicted Positive	3	1
Predicted Negative	1	3

3.



4.

$$\text{FPR} = \frac{\text{False Positive}}{\text{True Negative} + \text{False Positive}} = \frac{1}{4}$$

5.

$$\text{TPR} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{3}{4}$$

6.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Number of Examples}} = \frac{6}{8} = \frac{3}{4}$$

7.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{3}{4}$$

8.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{3}{4}$$

9.

$$\text{cross-entropy} = -(\sum_{i=1}^{N}(y^{(i)}ln(h(x)) + (1-y^{(i)})ln(1-h(x))))$$

$$\begin{aligned}\text{cross-entropy} = &-((0)(ln(0.389)) + (1-0)(ln(1-0.389)) + ((0)(ln(0.042)) + (1-0)(ln(1-0.042))\\ &+((0)(ln(0.613)) + (1-0)(ln(1-0.613)) + ((0)(ln(0.167)) + (1-0)(ln(1-0.167))\\ &+((1)(ln(0.572)) + (1-1)(ln(1-0.572)) + ((1)(ln(0.526)) + (1-1)(ln(1-0.526))\\ &+((1)(ln(0.393)) + (1-1)(ln(1-0.393)) + ((1)(ln(0.638)) + (1-1)(ln(1-0.638)))\end{aligned}$$

$$\text{cross-entropy} = -((-0.493) + (-0.43) + (-0.949) + (-0.183) + (-0.559) + (-0.642) + (-0.934) + (-0.449))$$

$$\text{cross-entropy} = -(-4.252)$$

$$\text{cross-entropy} = 4.252$$

10.

$$\text{cross-entropy} = -((-0.525) + (-0.022) + (-0.451) + (-0.119) + (-0.435) + (-0.441) + (-0.755) + (-0.267))$$

$$\text{cross-entropy} = 3.015$$

The cross-entropy for w' is lower than the cross-entropy for w, so w' is more likely to be the correct decision boundary when given access only to the data above.

11.

$$\nabla l(w) = X^T(y - \sigma(Xw))$$

$$\sigma(Xw) = \begin{bmatrix} 0.389 & 0.042 & 0.613 & 0.167 & 0.572 & 0.526 & 0.393 & 0.638 \end{bmatrix}^T$$

$$\nabla l(w) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0.49 & 1.69 & 0.04 & 1.0 & 0.16 & 0.25 & 0.49 & 0.04 \\ 0.09 & 0.04 & 0.64 & 0.16 & 0.09 & 0.0 & 0.0 & 0.01 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.389 \\ 0.042 \\ 0.613 \\ 0.167 \\ 0.572 \\ 0.526 \\ 0.393 \\ 0.638 \end{bmatrix} \right)$$

$$\nabla l(w) = \begin{bmatrix} 0.660 \\ 0.046 \\ -0.414 \end{bmatrix}$$

$$w = w + \frac{\alpha}{N} \nabla l(w)$$

$$w = \begin{bmatrix} 0.66 \\ -2.24 \\ -0.18 \end{bmatrix} + \frac{0.1}{8} \left(\begin{bmatrix} 0.660 \\ 0.046 \\ -0.414 \end{bmatrix} \right)$$

$$w = \begin{bmatrix} 0.668 \\ -2.239 \\ -0.185 \end{bmatrix}$$

12.

The data points that were close to the decision boundary contributed heavily to the new value of w.

13.

The data points that were correctly classified and far from the decision boundary barely contributed to the new value of w.

14.

The incorrectly classified point contributed most to the new value of w.

15.

$$\text{cross-entropy} = -((-0.494) + (-0.043) + (-0.949) + (-0.184) + (-0.557) + (-0.641) + (-0.931) + (-0.446))$$

$$\text{cross-entropy} = 4.245$$

Cross-entropy went down after one iteration of gradient ascent. This is what I expected because gradient ascent move w in the direction of the w-value that minimizes cross-entropy.

Question 4:

(Question)

Suggest possible target variables (response variables) and features (predictors) for the following classification problems. For each problem, indicate how many classes there are. There is no single correct answer.

(a) Given an audio sample, to detect the gender of the voice.

(b) A electronic writing pad records motion of a stylus and it is desired to determine which letter or number was written. Assume a segmentation algorithm is already run which indicates very reliably the beginning and end time of the writing of each character.

(Answer(s))

1. Possible target variables for the gender of a voice in an audio sample are the possible genders (male, female, nonbinary). In this case, there would be 3 classes. Predictors include pitch, loudness, and shrillness.

2. Possible target variables for the classification of a written letter or number are the possible letters and digits (26 lowercase letters, 26 uppercase letters, 10 digits). In this case, there would be 62 classes. Predictors include time spent writing the character, pressure applied to the stylus, and number of darkened pixels in the character.

Question 5:

(Question)

Regularization:

- Add lasso regularization to the log likelihood function for logistic regression
- Add ridge regularization to the log likelihood function for logistic regression
- Determine the derivative of log likelihood function for logistic regression with ridge regularization.
- Implement logistic regression with ridge regularization:
  - Add the ridge regularization to your programming assignment for logistic regression
  - Using 5-fold cross validation, find the optimal λ. Did the regularization help? How did the regularization affect the error on the training data. How did it affect the error on the validation set?
    You will turn in your answer by:
    describing your results on Gradescope
    submitting a separate .ipynb file for your code

(Answer(s))

1.

$$E_{lasso} = \sum_{i=1}^{N}(y^{(i)}ln(h(x)) + (1 - y^{(i)})ln(1 - h(x))) - \lambda(|w_1| + |w_2| + ... + |w_d|)$$

2.

$$E_{ridge} = \sum_{i=1}^{N}(y^{(i)}ln(h(x)) + (1 - y^{(i)})ln(1 - h(x))) - \lambda(w_1^2 + w_2^2 + ... + w_d^2)$$

3.

$$\nabla E_{ridge} = \nabla(\sum_{i=1}^{N}(y^{(i)}ln(h(x)) + (1 - y^{(i)})ln(1 - h(x)))) - \nabla(\lambda(w_1^2 + w_2^2 + ... + w_d^2))$$

$$\nabla E_{ridge} = X^T(y - \sigma(Xw)) - 2\lambda I'w \text{ (as shown in class)}$$

4.

```
def sigmoid(z):
    result = 1 / (1 + np.exp(-z))
    return result
```

```
def hypothesis(X_train_1, w):
    z_score = X_train_1.dot(w)
    y_hat = sigmoid(z_score)
    return y_hat
```

```
def likelihood(X_tr, y_tr, w, n):
    y_hat = hypothesis(X_tr, w) # is a vector of all probability of each example, basically h(x^(i)) for all i = 0 .. n
    likelihood = np.sum(y_tr * np.log(y_hat) + (1 - y_tr) * np.log(1 - y_hat))
    return likelihood
```

```
# Write the gradient ascent function
def Gradient_Ascent(X_train_1, y_2d_train, learning_rate, num_iters, alpha):
    # Number of training examples.
    N = X_train_1.shape[0]
    # Initialize w(<np.ndarray>). Zeros vector of shape X_train.shape[1],1
    w = np.zeros((X_train_1.shape[1], 1))
    # Initiating list to store values of likelihood(<list>) after few iterations.
    likelihood_values = []
    for i in range(num_iters):
        y_hat = hypothesis(X_train_1, w)
        error = y_2d_train - y_hat
        gradient = X_train_1.T.dot(error)
        # Updating Parameters
        ridgeDeriv = 2 * w
        w = w + (learning_rate / N) * gradient - alpha *  ridgeDeriv
        if (i % 100) == 0:
            likelihood_values.append(likelihood(X_train_1,y_2d_train,w,N))

    return w, likelihood_values
```

```python
from sklearn.model_selection import KFold

def kfold_cross_validation(X_train_1, y):
    lambdas = [1,0.00001,0.001, 0.1, 10, 0.01, 0.0001, 0.00000001]
    #print("All possible lambdas:", lambdas)
    best_lambda = 0
    kfold_model = KFold(n_splits=5, random_state=None, shuffle=False)
    min_error = float('inf')
    w_best = []
    for l in lambdas:
        #print(l)
        error = []
        wval = []
        for train_index, test_index in kfold_model.split(X_train_1):
            X_tr = X_train_1[train_index]
            Y_tr = y[train_index].reshape((-1, 1))
            X_ts = X_train_1[test_index]
            Y_ts = y[test_index].reshape((-1, 1))
            w, likelihood = Gradient_Ascent(X_tr, Y_tr, 0.001, X_tr.shape[0], l)
            yhat = hypothesis(X_ts, w)
            error.append(np.sum((y-yhat)**2))
            wval.append(w)

        if (np.mean(error) < min_error):
            min_error = np.mean(error)
            w_best = wval[np.argmin(error)]
            best_lambda = l

        #print("Min error",min_error)

    return w_best, best_lambda
```

Question 6:

(Question)

A data scientist is hired by a political candidate to predict who will donate money. The data scientist decides to use two predictors for each possible donor:

- x_1 = the income of the person(in thousands of dollars), and
- x_2 = the number of websites with similar political views as the candidate the person follows on Facebook.

To train the model, the scientist tries to solicit donations from a randomly selected subset of people and records who donates or not. She obtains the following data:

Income (thousands \$), $x_1^{(i)}$	30	50	70	80	100
Num websites, $x_2^{(i)}$	0	1	1	2	1
Donate (1=yes or 0=no), $y^{(i)}$	0	1	0	1	1

(a) Draw a scatter plot of the data labeling the two classes with different markers.

(b) Find a linear classifier that makes at most one error on the training data. The classifier should be of the form,

$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } z^{(i)} > 0 \\ 0 & \text{if } z^{(i)} < 0, \end{cases} \qquad z^{(i)} = \mathbf{w}^{\mathsf{T}}\mathbf{x}^{(i)}$$

What is the weight vector $\mathbf{w}$ of your classifier?

(c) Now consider a logistic model of the form,

$$P(y^{(i)} = 1|\mathbf{x}^{(i)}) = \frac{1}{1+e^{-z^{(i)}}}, \qquad z^{(i)} = \mathbf{w}^{\mathsf{T}}\mathbf{x}^{(i)}$$

Using $\mathbf{w}$ from the previous part, which sample i is the *least* likely (i.e. $P(y^{(i)}|\mathbf{x}^{(i)})$ is the smallest). If you do the calculations correctly, you should not need a calculator.

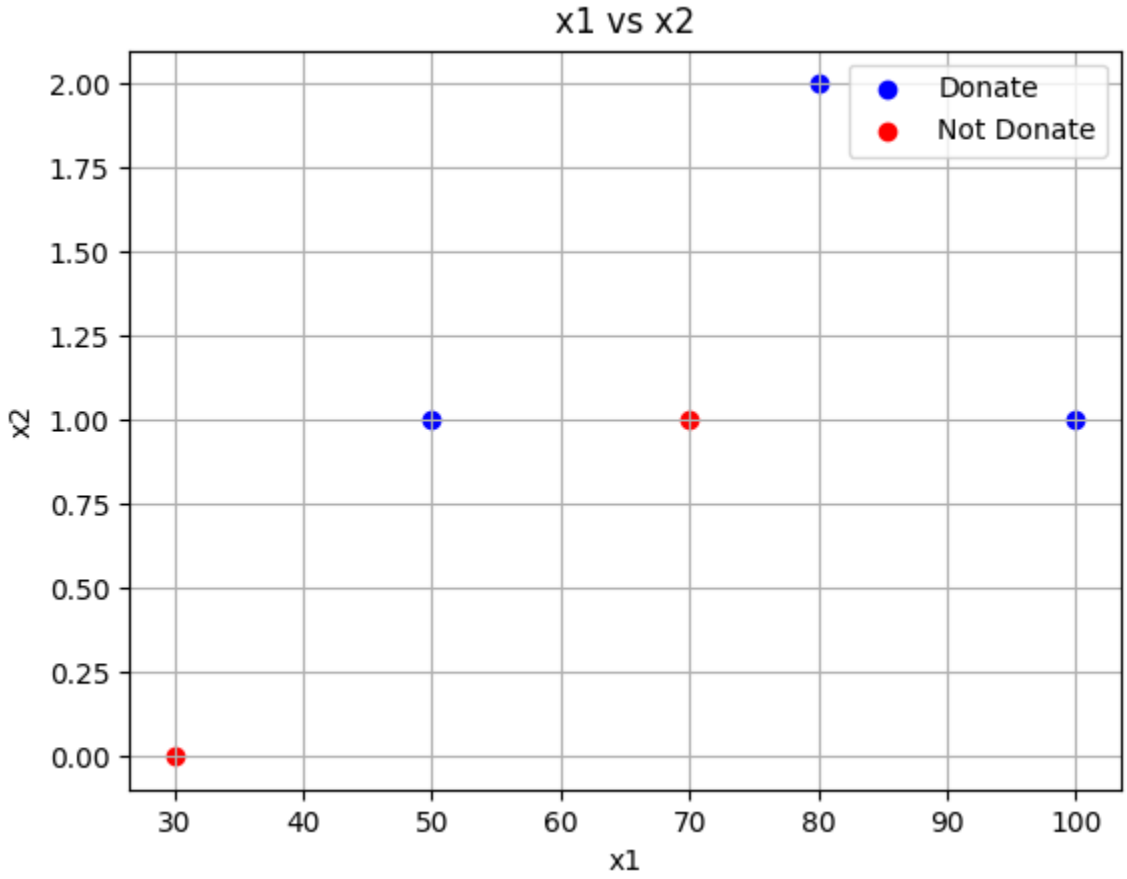
(d) Now consider a new set of parameters

$$\mathbf{w}' = \alpha\mathbf{w},$$

where $\alpha > 0$ is a positive scalar. Would using the new parameters change the values $\hat{y}$ in part (b)? Would they change the likelihoods $P(y_i|\mathbf{x}_i)$ in part (c)? If they do not change, state why. If they do change, qualitatively describe the change as a function of α.

(Answer(s))

1.

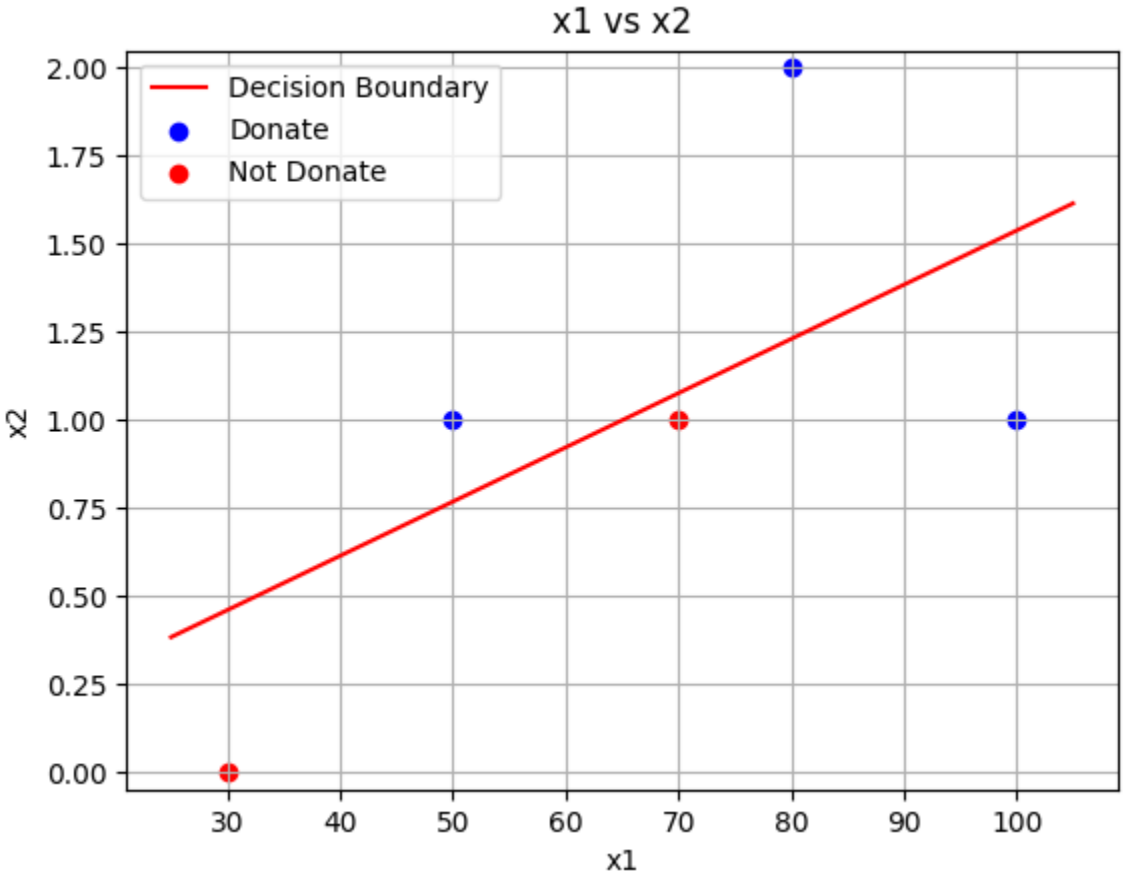

x1 vs x2
Donate
Not Donate
x2
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
30
40
50
60
70
80
90
100
x1

2.

$$z(x^{(i)}) = (0) + (-1)x_1 + (65)x_2$$

$$\hat{y}_i = \begin{cases} 1 & \text{if } z^{(i)} > 0 \\ 0 & \text{if } z^{(i)} < 0 \end{cases} z^{(i)} = w^T x^{(i)}$$

$$w = \begin{bmatrix} 0 \\ -1 \\ 65 \end{bmatrix}$$



3.

Of all these values, $P(y_5 = 1|(100, 1))$ must be the smallest because it's "farthest" below the chosen decision boundary (signalling the low likelihood that it will be classified as Class 1). Proof:

$$z^1 = \begin{bmatrix} 0 \\ -1 \\ 65 \end{bmatrix} \begin{bmatrix} 1 & 30 & 0 \end{bmatrix} = -30$$

$$P(y^{(1)} = 1|x^{(1)}) = \frac{1}{1 + e^{-(-30)}} = \frac{1}{1 + e^{30}}$$

$$z^2 = \begin{bmatrix} 0 \\ -1 \\ 65 \end{bmatrix} \begin{bmatrix} 1 & 50 & 1 \end{bmatrix} = 15$$

$$P(y^{(2)} = 1|x^{(2)}) = \frac{1}{1 + e^{-(15)}} = \frac{1}{1 + e^{-15}}$$

$$z^3 = \begin{bmatrix} 0 \\ -1 \\ 65 \end{bmatrix} \begin{bmatrix} 1 & 70 & 1 \end{bmatrix} = -5$$

$$P(y^{(3)} = 1|x^{(3)}) = \frac{1}{1 + e^{-(-5)}} = \frac{1}{1 + e^{5}}$$

$$z^4 = \begin{bmatrix} 0 \\ -1 \\ 65 \end{bmatrix} \begin{bmatrix} 1 & 80 & 2 \end{bmatrix} = 50$$

$$P(y^{(4)} = 1|x^{(4)}) = \frac{1}{1 + e^{-(50)}} = \frac{1}{1 + e^{-50}}$$

$$z^5 = \begin{bmatrix} 0 \\ -1 \\ 65 \end{bmatrix} \begin{bmatrix} 1 & 100 & 1 \end{bmatrix} = -35$$

$$P(y^5 = 1|x^{(5)}) = \frac{1}{1 + e^{-(-35)}} = \frac{1}{1 + e^{35}}$$

4.

Using these new parameters will not change the values $\hat{y}$ in part (b) because only the signs matter for predicting which class a sample belongs to. Multiplying $z(x^{(i)})$ by a positive scalar will not change its sign and therefore will not change $\hat{y}$.

However, using these new parameters will change the likelihoods $P(y_i|x_i)$ in part (c). Although multiplying $z(x^{(i)})$ by a positive scalar will not change its sign, it will change its magnitude. Then, when this new value is plugged into $\frac{1}{1+e^{-z(i)}}$, the resulting likelihood will change. If $\alpha < 1$, then likelihoods will come closer to 0.5. However, if $\alpha > 1$, then the likelihoods will move farther away from 0.5 and be more polarized towards 0 or 1.

Question 7:

(Question)

A multivariate logistic regression model has been built to predict the propensity of shoppers to perform a repeat purchase of a free gift that they are given. The descriptive features used by the model are the age of the customer, the socioeconomic band to which the customer belongs (a, b, or c), the average amount of money the customer spends on each visit to the shop, and the average number of visits the customer makes to the shop per week. This model is being used by the marketing department to deter-mine who should be given the free gift.

(a) The weights in the trained model are shown in the following table:

Feature	Weight
Intercept (w_0)	-3.82398
AGE	-0.02990
SOCIOECONOMIC BAND B	-0.09089
SOCIOECONOMIC BAND C	-0.19558
SHOP VALUE	0.02999
SHOP FREQUENCY	0.74572

Create the coefficient vector **w**.

(b) Rewrite the following data matrix using the dummy encoding so it works with the coefficients in the previous question.

ID	AG	SOCIOECONOMIC BAND	SHOP FREQUENCY	SHOP VALUE
1	56	b	1.60	109.32
2	21	c	4.92	11.28
3	48	b	1.21	161.19
4	37	c	0.72	170.65
5	32	a	1.08	165.39

(c) Use the model to make predictions for the data matrix you created in the previous question.

(d) In building logistic regression models, it is recommended that all continuous descriptive features be scaled. In this question the continuous features were normalized to the range $[-1, 1]$ using min/max normalization (and called range normalization). The following table shows a data quality report for the dataset used to train the model described above.

Feature	N	% Missing	min value	mean	max value	std. dev
Age	5,2000	6	18	32.7	63	12.2
SHOP FREQUENCY	5,2000	0	0.2	2.2	5.4	1.6
SHOP VALUE	5,2000	0	5	101.9	230.7	72.1

Feature	N	% Missing	# categories	mode
SOCIOECONOMIC BAND	5,2000	8	3	a
REPEAT PURCHASE	5,2000	0	2	no

On the basis of the information in this report, all continuous features were normalized using min/max normalization (aka range normalization).

After applying these data preparation operations, a logistic regression model was trained to give the weights shown in the following table.

Feature	Weight
Intercept (w_0)	0.6679
AGE	-0.5795
SOCIOECONOMIC BAND B	-0.1981
SOCIOECONOMIC BAND C	-0.2318
SHOP VALUE	3.4091
SHOP FREQUENCY	2.0499

For this question if we have any missing values, we will replaced them using mean imputation for continuous features, and mode imputation for categorical features.

Use this model to make predictions for each of the query instances shown in the following table (question marks refer to missing values).

ID	AG	SOCIOECONOMIC BAND	SHOP FREQUENCY	SHOP VALUE
1	38	a	1.90	165.39
2	56	b	1.60	109.32
3	18	c	6.00	10.09
4	?	b	1.33	204.62
5	62	?	0.85	110.50

(Answer(s))

1.

$$w = \begin{bmatrix} -3.82398 \\ -0.02990 \\ -0.09089 \\ -0.19558 \\ 0.02999 \\ 0.74572 \end{bmatrix}$$

2.

$$X = \begin{bmatrix} 1 & 56 & 1 & 0 & 1.60 & 109.32 \\ 1 & 21 & 0 & 1 & 4.92 & 11.28 \\ 1 & 48 & 1 & 0 & 1.21 & 161.19 \\ 1 & 37 & 0 & 1 & 0.72 & 170.65 \\ 1 & 32 & 0 & 0 & 1.08 & 165.39 \end{bmatrix}$$

3.

$$\hat{y} = Xw = \begin{bmatrix} 75.9808244 \\ 3.9113696 \\ 114.8888247 \\ 122.1495208 \\ 118.58624 \end{bmatrix}$$

4.

$$AG_4 = \text{mean\_age} = 32.7$$

$$SOCIOECONOMIC\_BAND_5 = \text{mode\_band} = a$$