ML Assignment-1

Akshala Bhatnagar
2018012

1. Linear Regression

Dataset 1

The value of k chosen for this dataset is 3. The dataset is large so if a large value of k is taken then computation time would be high. Even at 3 folds there would be enough training samples because the dataset is large.

The file is read into a string. Then it is split on \n. This list of lines is further split on the basis of space and the values are added to a pandas dataframe. The column names are taken from the first line of the data file. In the sex column values present were M, F, I. Only numerical data can be used therefore they were converted to a one hot encoding. The sex column is deleted and three columns namely, M, F and I are added. These columns have the value 0 or 1 depending upon the sex. Rows which contain NaN value are dropped. The rows in the dataframe were shuffled. The rings column is taken as the y value and the rest of the dataframe is X.

Dataset 2

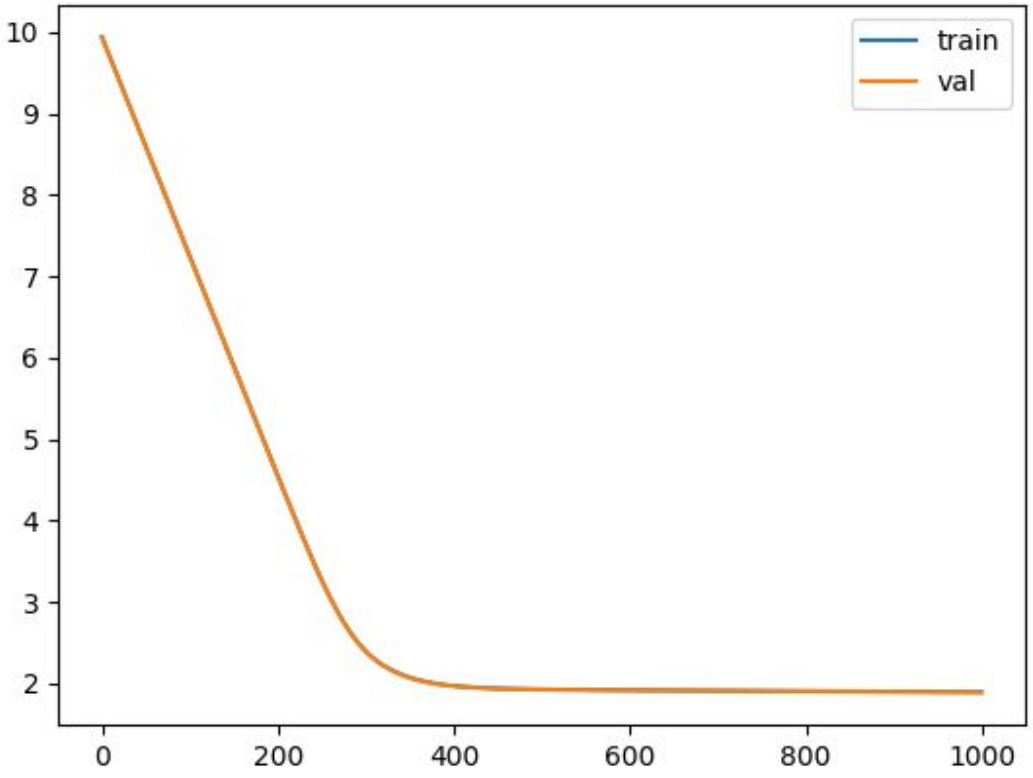
The value of k chosen for this dataset is 3. The dataset is large so if a large value of k is taken then computation time would be high. Even at 3 folds there would be enough training samples because the dataset is large.

The csv file is directly read into a pandas dataframe. Rows which contain NaN value are dropped. Rows in which user score is tbd are also dropped. The rows in the dataframe were shuffled. The global sales column is taken as the y value and the rest of the dataframe is X

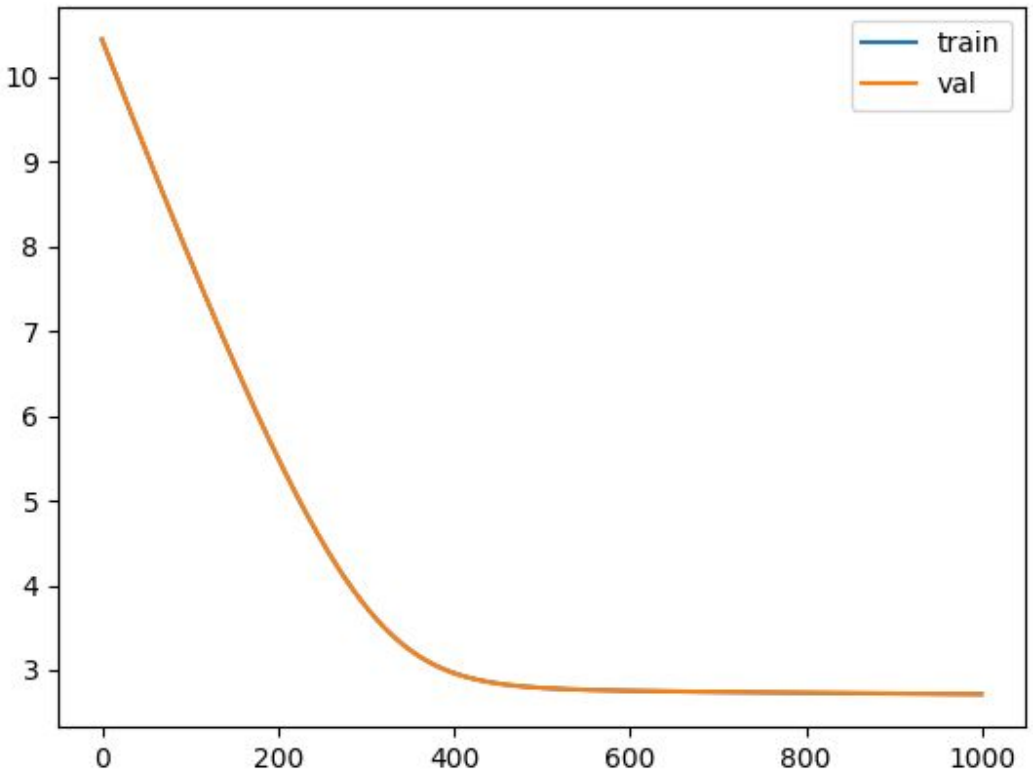
a) Train loss vs iterations, validation loss vs iterations

Dataset 1

MAE loss - learning rate: 0.01, no. of epochs: 1000

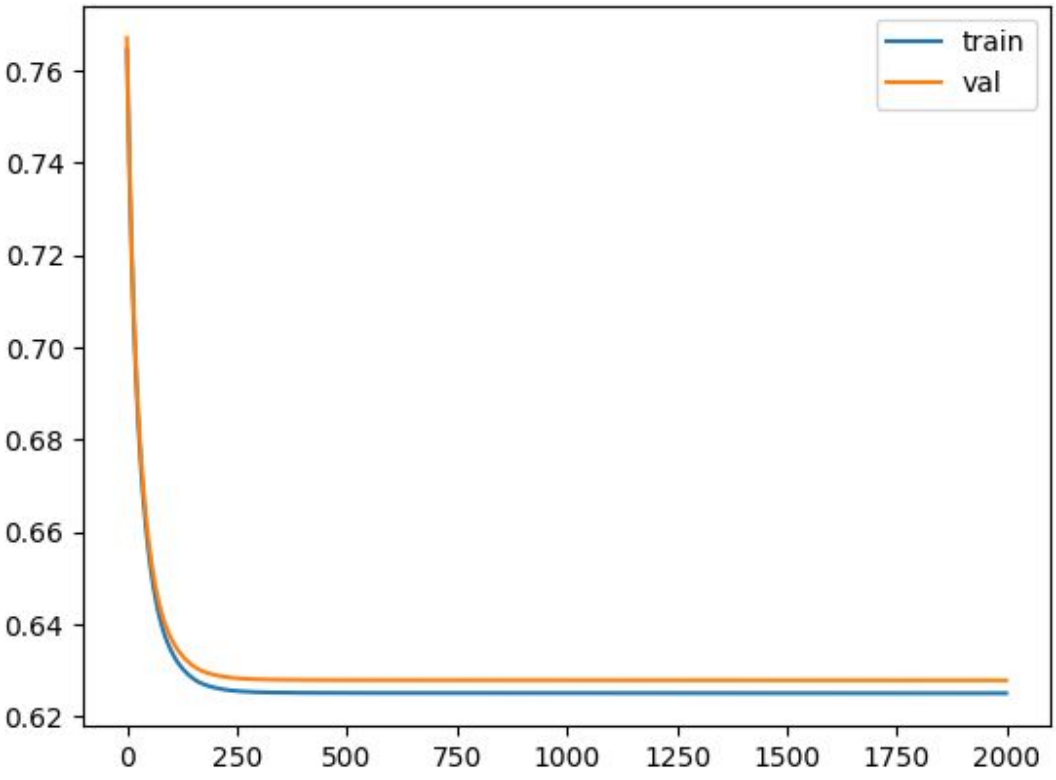


RMSE loss - learning rate: 0.01, no. of epochs: 1000

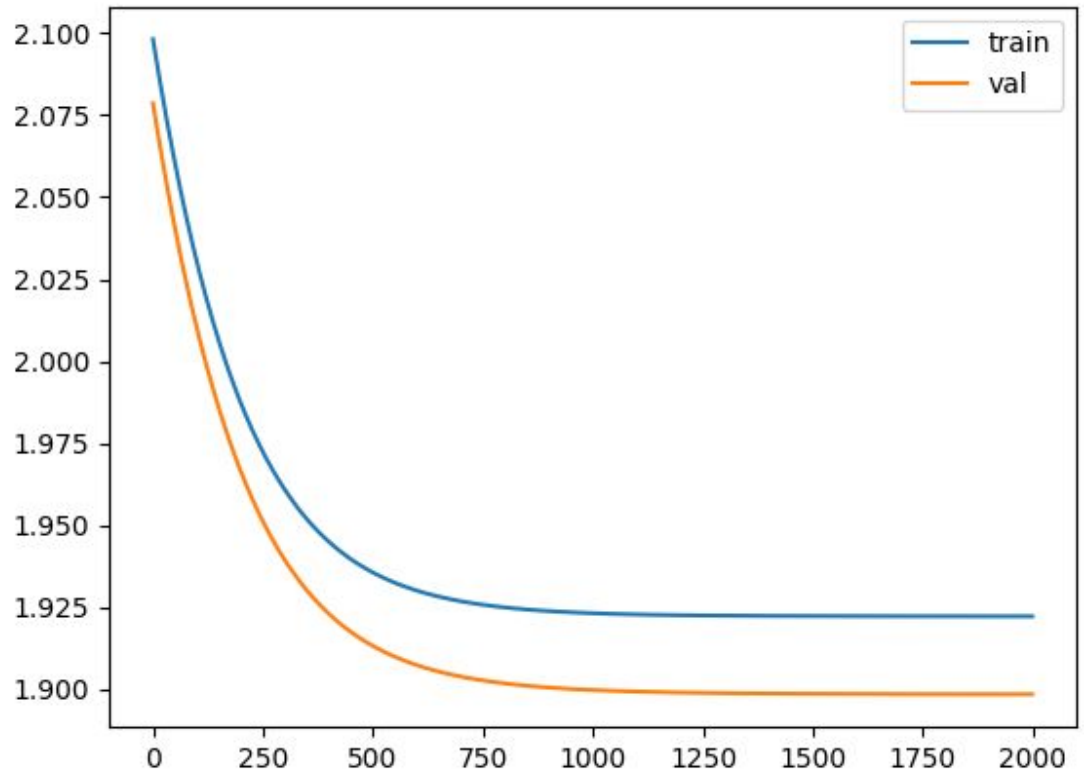


Dataset 2

MAE loss - learning rate: 0.000001, no. of epochs: 2000



RMSE loss - learning rate: 0.000001, no. of epochs: 2000



b) Dataset1 MAE

Fold	Train loss	Test loss
0	1.923762668553756	1.851253638429136
1	1.852453347746632	1.948478202594888
2	1.898331455902084	1.866288279414634

Considering both train and val loss fold 2 is the best.

Dataset1 RMSE

Fold	Train loss	Test loss
0	2.022441468011275	1.882179972548051
1	1.924322197889352	1.996254814010948
2	1.920610015486229	1.999874173678606

Considering both train and val loss fold 1 is the best.

Dataset2 MAE

Fold	Train loss	Test loss
0	0.621942718030754	0.623012355804239
1	0.639015460464578	0.605406055299133
2	0.614164129805743	0.655104048605861

Considering both train and val loss fold 0 is the best.

Dataset2 RMSE

Fold	Train loss	Test loss
0	0.761722915441703	0.772437364796754
1	0.798096024262932	0.764055763312567
2	0.763977252798790	0.794926249089246

Considering both train and val loss fold 0 is the best.

c) As it can be seen from the table above, MAE gives better performance in terms of loss in both the datasets. MAE gives lower values of loss therefore MAE should be preferred. It can be seen from the graphs shown above that the training and validation loss are much closer to each other in the case of MAE than RMSE for both the datasets.

d) RMSE loss is always greater than or equal to MAE. RMSE and MAE are equal when all errors have the same magnitude. In this case RMSE should be preferred because it penalises outliers more. Also RMSE does not have the modulus function. MAE takes the absolute value and is not differentiable. Generally also MSE is preferred over MAE so for the same reason RMSE should be preferred here.

e) Following is the theta obtained using normal equation

[[-0.45833542]

[11.07510254]

[10.7615367]

[8.97544462]

[-19.78686686]

[-10.58182703]

[8.7418058]

[1.22316618]

[1.1654505]

[0.34057424]

[2.72919092]]

Using MAE with the above value of theta -

Train loss: 1.5826879670386953

Val loss: 1.5690230535690535

2. Logistic Regression

There are 2 classes given in this dataset, namely 0 and 1. The dataset is such that all occurrences of one class are present together, therefore the data needs to be shuffled. The features present in the dataset are-

1. Variance

   Max value: 6.8248, Min value: -7.0421, mean: 0.436

2. Skewness

   Max value: 12.9516, Min value: -13.7731, mean: 1.92

3. Curtosis

   Max value: 17.9274, Min value: -5.2861, mean: 1.396

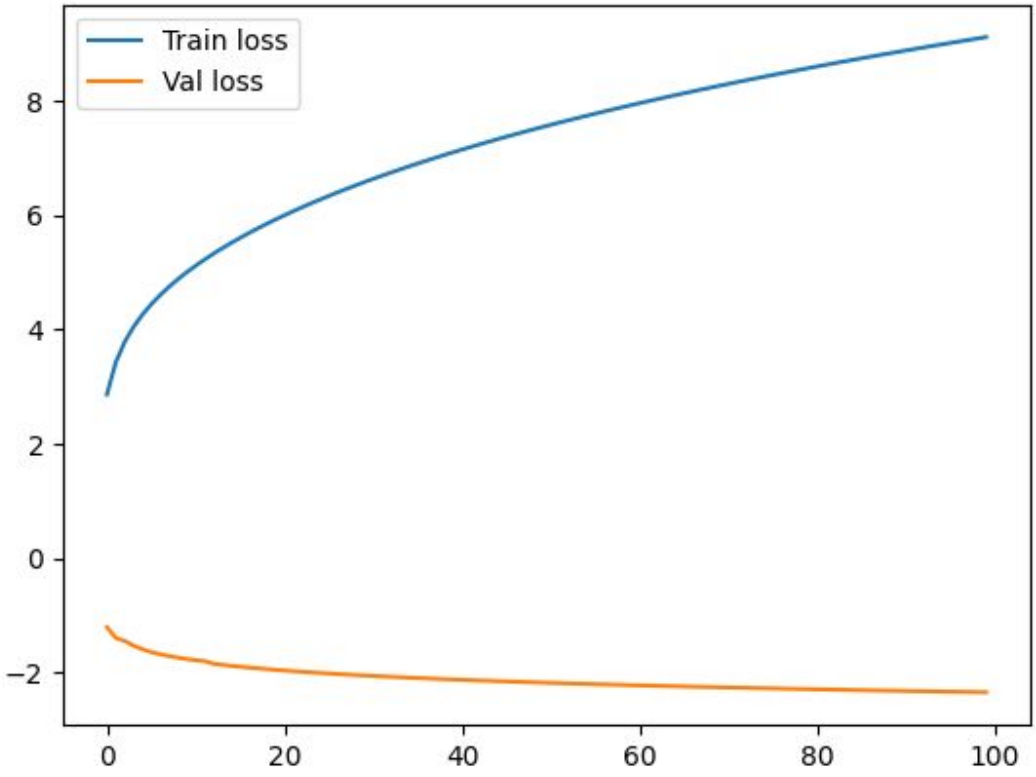
4. Entropy

   Max value: 2.4495, Min value: -8.5482, mean: -1.193

   SGD

a.

Learning Rate	No. of epochs	Train accuracy	Test accuracy
0.01	100	0.9885297184	0.9890510948



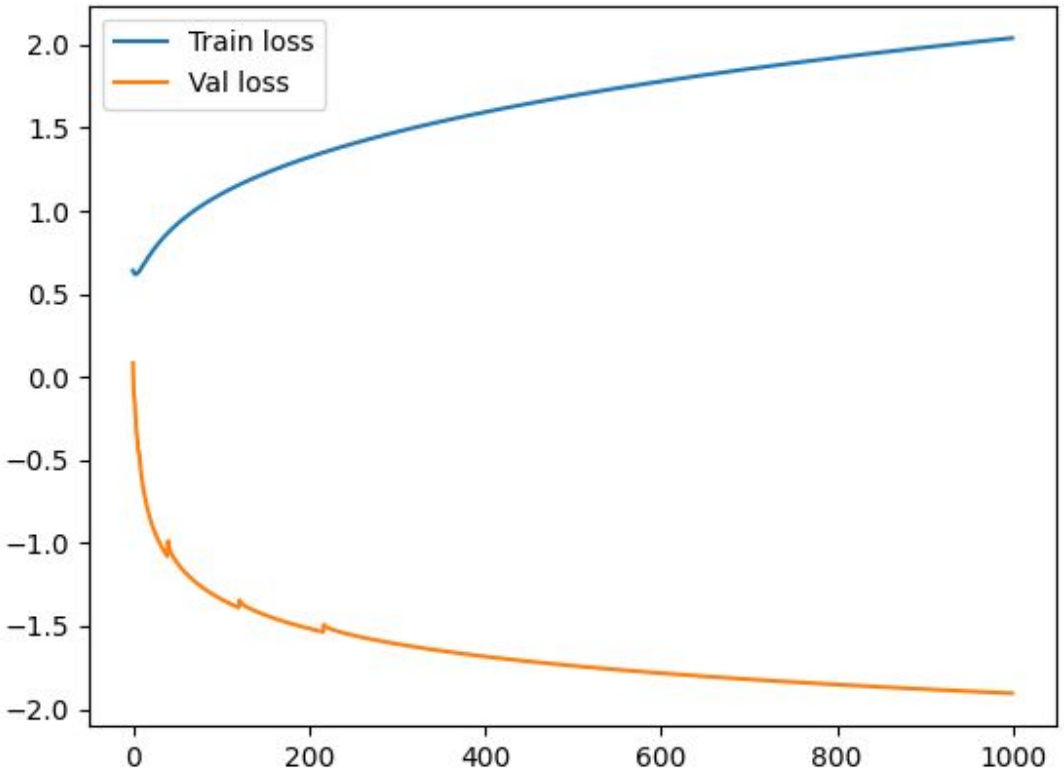
b.

c.

Learning Rate	No. of epochs	Train accuracy	Test accuracy
0.0001	1000	0.9906152241	0.9854014598
0.01	100	0.9885297184	0.9890510948
10	10	0.9833159541	0.9744525547

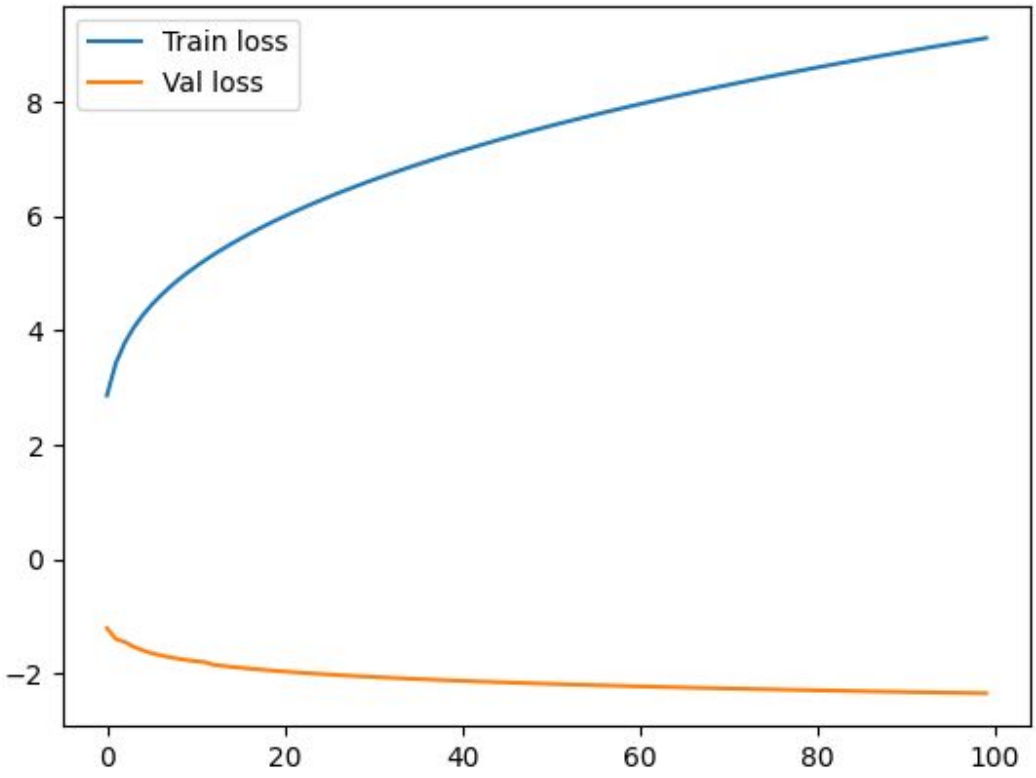
d. Learning Rate: 0.0001

No.of epochs: 1000



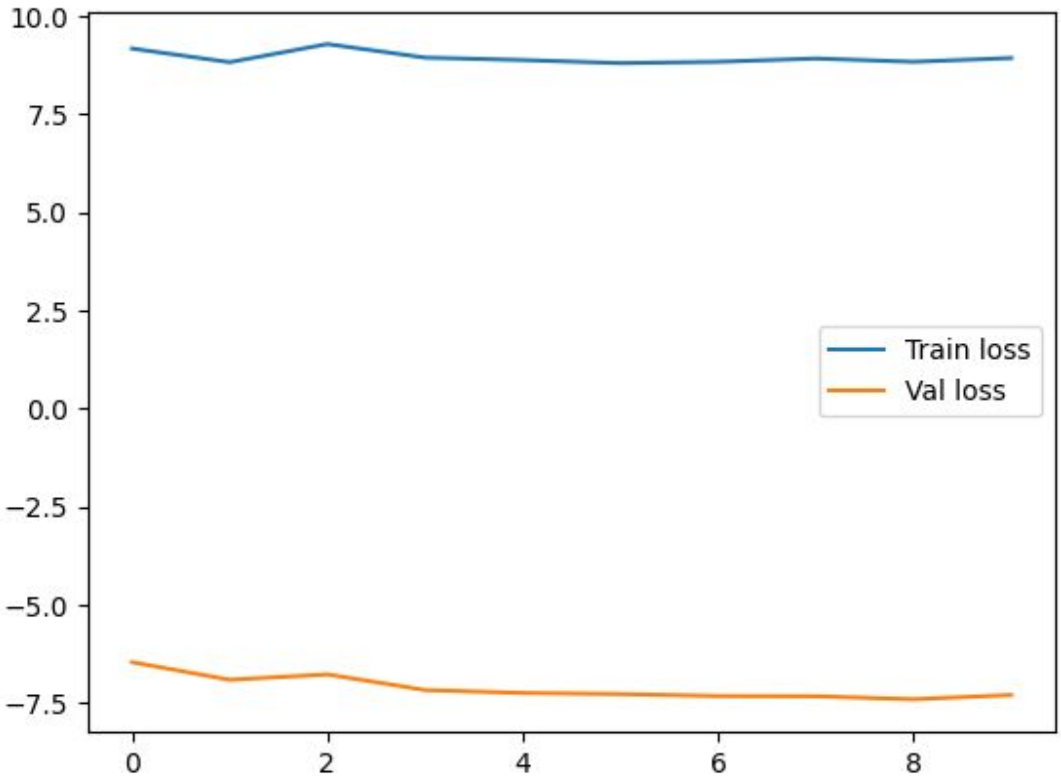
Learning Rate: 0.01

No.of epochs: 100



Learning Rate: 10

No. of epochs: 10

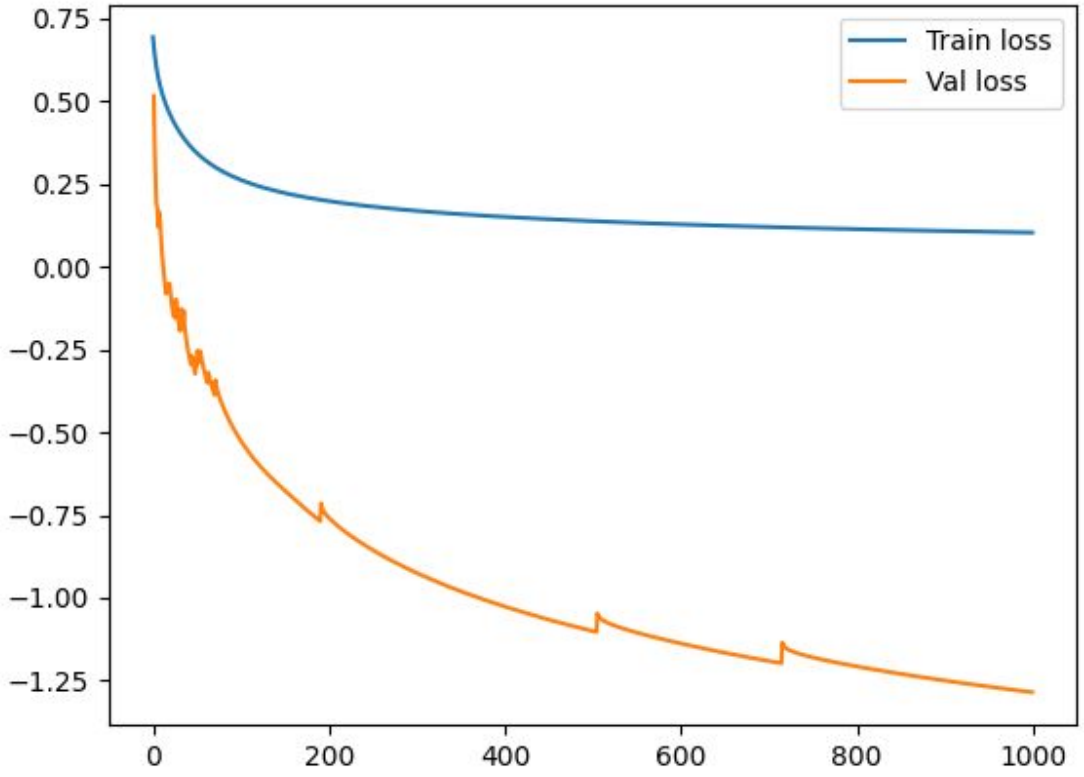


BGD

a.

Learning Rate	No. of epochs	Train accuracy	Test accuracy
0.01	1000	0.9749739311	0.9671532846

b.

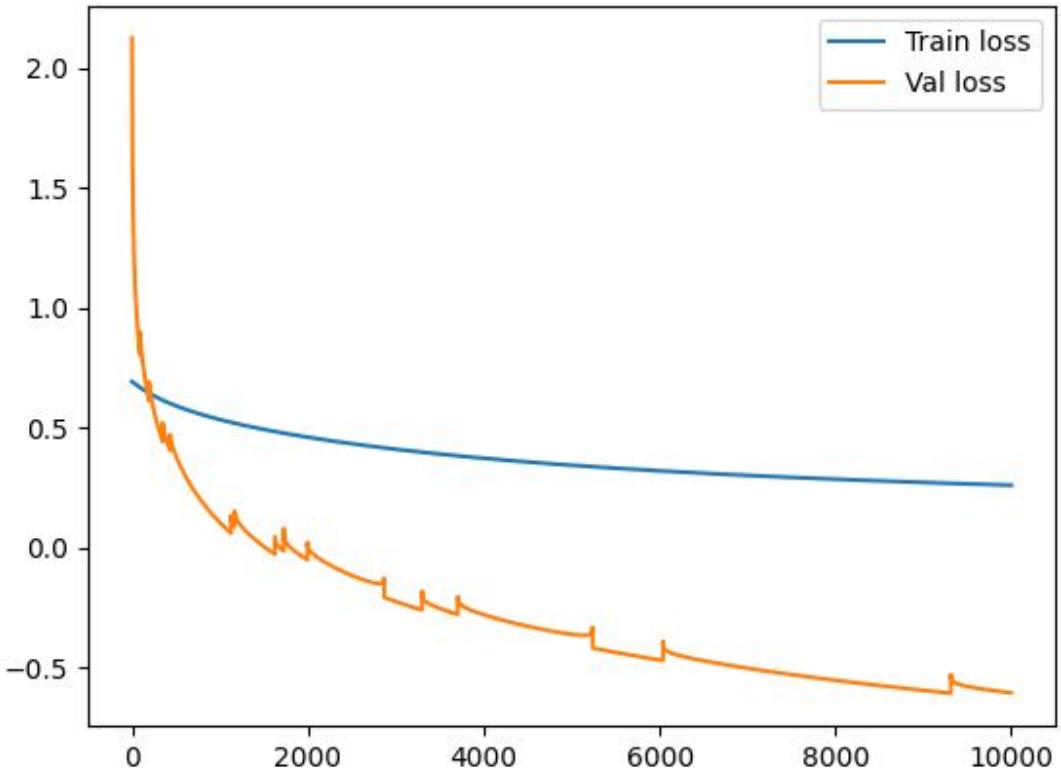


c.

Learning Rate	No. of epochs	Train accuracy	Test accuracy
0.0001	10000	0.9332638164	0.9416058394
0.01	1000	0.9708029197	0.9927007299
10	100	0.9864442127	0.9854014598

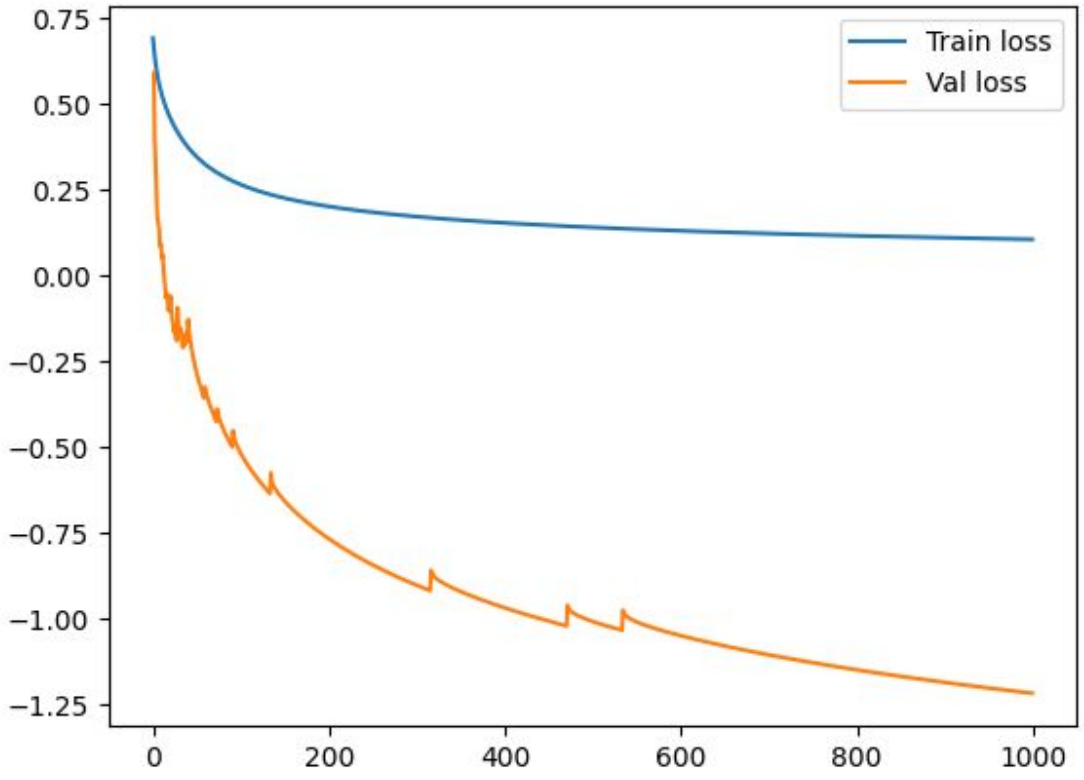
d. Learning rate: 0.0001

No. of epochs: 10000



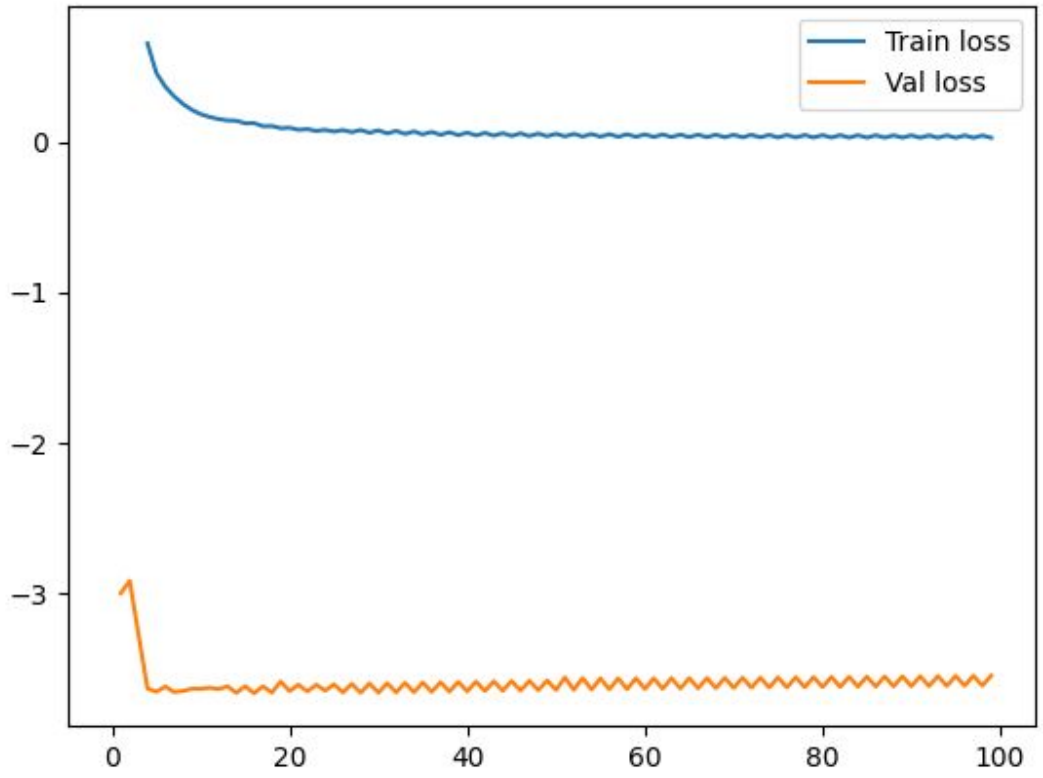
Learning Rate: 0.01

No. of epochs: 1000



Learning Rate: 10

No. of epochs: 100



Comparing BGD and SGD

a. Loss Plot: Convergence point is reached very soon for SGD as compared to BGD. The graphs for SGD are much more smooth as compared to BGD. The horizontal line for loss curve is reached much sooner.
b. As it can be seen from the tables above, for the same learning rate, less number of epochs are needed by SGD to

converge as compared to BGD. This is because on each epoch theta is being updated for every data point.

c. Accuracy using sklearn logistic regression: 0.9890909090909091
d. Test accuracy using sklearn SGD classifier: 0.9854545454545455

   Train accuracy using sklearn SGD classifier: 0.9890510948905109

   The above accuracies for sklearn SGD are obtained on 0.01 learning rate and 10000 no. of epochs

3. For logistic regression,

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$, (Hypothesis)

MSE loss, $J = \sum_{i=1}^{n} (h_\theta(x) - y)^2 = \sum_{i=1}^{n} \left[\frac{1}{1+e^{-\theta^T x}} - y\right]^2$

$h_\theta(x) = y\_pred$

$y = y\_true$

$$\frac{dJ}{d\theta} = \frac{d}{d\theta}\left[\sum_{i=1}^{n} \left(\frac{1}{1+e^{-\theta^T x}} - y\right)^2\right]$$

$$= \sum_{i=1}^{n} \frac{d}{d\theta}\left[\left(\frac{1}{1+e^{-\theta^T x}} - y\right)^2\right]$$

$$= 2\sum_{i=1}^{n} \left(\frac{1}{1+e^{-\theta^T x}} - y\right) \frac{d}{d\theta}\left[\left(\frac{1}{1+e^{-\theta^T x}}\right) - y\right]$$

$$= 2\sum_{i=1}^{n} \left(\frac{1}{1+e^{-\theta^T x}} - y\right) \frac{d}{d\theta}\left(1+e^{-\theta^T x}\right)^{-1}$$

$$= 2\sum_{i=1}^{n} \left(\frac{1}{1+e^{-\theta^T x}} - y\right) \left(\frac{-1}{(1+e^{-\theta^T x})^2}\right) \left(e^{-\theta^T x}\right)(-x)$$

$$\frac{dJ}{d\theta} = 2\sum_{i=1}^{n} \left(\frac{1}{1+e^{-\theta^T x}} - y\right) \left(\frac{x e^{-\theta^T x}}{(1+e^{-\theta^T x})^2}\right) \quad \text{— (1)}$$

It is given that the model produces really wrong results.

Case I: $y\_true = 1$, $y\_pred \to 0$ (approaches)

$\Rightarrow \frac{1}{1+e^{-\theta^T x}} \to 0$

$\Rightarrow 1 + e^{-\theta^T x} \to -\infty$

$\Rightarrow e^{-\theta^T x} \to -\infty$ (approaches)

By ①

$$\frac{dJ}{d\theta} = 2\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x}} - y\right)\underbrace{\left(\frac{x e^{-\theta^T x}}{(1+e^{-\theta^T x})^2}\right)}$$

Consider the underlined term,

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

$$= \frac{x e^{-\theta^T x}}{1 + 2e^{-\theta^T x} + e^{-2\theta^T x}}$$

$$= \frac{x}{\frac{1}{e^{-\theta^T x}} + 2 + e^{-\theta^T x}}$$

$e^{-\theta^T x}$ approaches $-\infty$, therefore $\frac{1}{e^{-\theta^T x}}$ approaches 0

Thus the term becomes $\frac{x}{2+e^{-\theta^T x}}$

$e^{-\theta^T x}$ approaches $-\infty$, so $2+e^{-\theta^T x}$ also approaches $-\infty$

$\therefore \frac{x}{2+e^{-\theta^T x}}$ approaches 0

This implies, $\frac{dJ}{d\theta} = 2\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x}} - y\right) \times 0$

$\Rightarrow \frac{dJ}{d\theta} = 0$

Thus gradient calculated approaches 0

Case II

y_true = 0, y_pred → 1 (approaches)

$\rightarrow \frac{1}{1+e^{-\theta^T x}} \rightarrow 1$

$\Rightarrow e^{-\theta^T x} \rightarrow \infty$ (approaches)

DATE
PAGE

By ①

$$\frac{dJ}{d\theta} = 2\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x}} - y\right)\underline{\left(\frac{x e^{-\theta^T x}}{(1+e^{-\theta^T x})^2}\right)}$$

consider the underlined term,

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

$$= \frac{x e^{-\theta^T x}}{1+2e^{-\theta^T x}+e^{-2\theta^T x}}$$

$$= \frac{x}{\frac{1}{e^{-\theta^T x}}+2+e^{-\theta^T x}}$$

$e^{-\theta^T x}$ approaches ∞, therefore $\frac{1}{e^{-\theta^T x}}$ approaches 0

Thus the term becomes $\frac{x}{2+e^{-\theta^T x}}$

$e^{-\theta^T x}$ approaches ∞, so $2+e^{-\theta^T x}$ also approaches ∞

$\therefore \frac{x}{2+e^{-\theta^T x}}$ approaches 0

This implies, $\frac{dJ}{d\theta} = 2\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x}} - y\right) \times 0$

$$\rightarrow \frac{dJ}{d\theta} = 0$$

Thus gradient calculated approaches 0,

It can be seen from both the cases that when mean squared loss is used with logistic regression and for a datapoint for which the model produces really wrong results, gradient calculated during gradient descent approaches 0.

DATE
PAGE

Cross entropy loss, $J = -\frac{1}{n} \sum_{i=1}^{n} \left[y \log(h_\theta(x)) + (1-y) \log(1-h_\theta(x)) \right]$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$$\frac{dJ}{d\theta} = \frac{d}{d\theta} \left[-\frac{1}{n} \sum_{i=1}^{n} \left[y \log(h_\theta(x)) + (1-y) \log(1-h_\theta(x)) \right] \right]$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \frac{d}{d\theta} \left[y \log(h_\theta(x)) + (1-y) \log(1-h_\theta(x)) \right]$$

$$\frac{dJ}{d\theta} = -\frac{1}{n} \sum_{i=1}^{n} \left[\frac{y}{h_\theta(x)} - \frac{1-y}{1-h_\theta(x)} \right] \frac{d}{d\theta} (h_\theta(x)) \quad -(1)$$

$$\frac{d}{d\theta}(h_\theta(x)) = \frac{d}{d\theta} \left(\frac{1}{1+e^{-\theta^T x}} \right)$$

$$= \frac{-e^{-\theta^T x} \cdot x}{(1+e^{-\theta^T x})^2}$$

$$= -x \left[\frac{1-1+e^{-\theta^T x}}{(1+e^{-\theta^T x})^2} \right]$$

$$= -x \left[\frac{1+e^{-\theta^T x}}{(1+e^{-\theta^T x})^2} - \frac{1}{(1+e^{-\theta^T x})^2} \right]$$

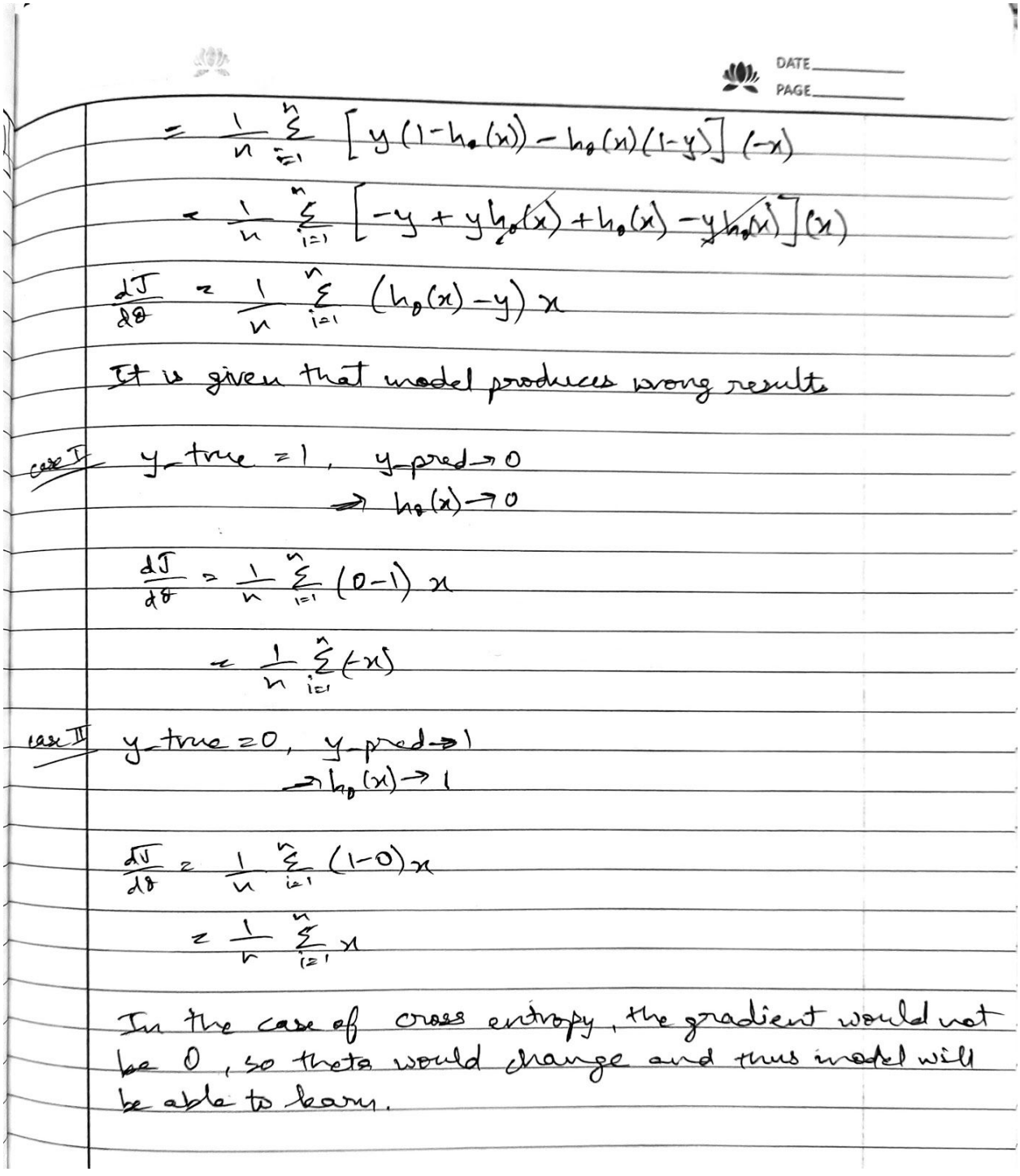
$$= -x \left[\frac{1}{1+e^{-\theta^T x}} - \frac{1}{(1+e^{-\theta^T x})^2} \right]$$

$$= -x \left[\frac{1}{(1+e^{-\theta^T x})} \left(1 - \frac{1}{1+e^{-\theta^T x}} \right) \right]$$

$$\frac{d}{d\theta}(h_\theta(x)) = -x \, h_\theta(x) \left(1-h_\theta(x)\right) \quad -(2)$$

using (2) in (1)

$$\frac{dJ}{d\theta} = -\frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{y}{h_\theta(x)} - \frac{1-y}{1-h_\theta(x)} \right\} (-x) \, h_\theta(x) \left(1-h_\theta(x)\right)$$



3. The model would not be able to learn effectively. The gradient calculated during gradient descent approaches zero, therefore

there would be no change in the model parameters during gradient descent.

Theta -= learning_rate * gradient

This problem would not arise if we use cross entropy loss which is log loss. The derivations for the same have been shown above.

4. Calculating the negative of log likelihood function for Bernoulli distribution is the same as calculating the cross entropy function for the Bernoulli distribution. Therefore logistic regression has been used in this question.
    a. B0 =-2.77121397 (coefficient of constant term)
    b. B1 = 0.04535902 (coefficient of X1: disease spread)
    c. B2 = 0.26146058 (coefficient of X2: age)
    d. Fitted response function:

        Y = 1 / (1 + e^(B0 + B1*X1 + B2*X2))

    e. exp(B1): 1.04640347
    f. exp(B2): 1.29882575

The value of exp(B2) is more than that of exp(B1), this means that the parameter X2 which is age is a more important feature in prediction.

    g. probability that a patient with 75% of disease spread and an age of 2 years will have a recurrence of disease in the next 5 years: 0.09847602

5. $Y = XB + \varepsilon$

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_{21} & \cdots & x_{k1} \\ \vdots & & & \vdots \\ 1 & x_{2n} & \cdots & x_{kn} \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ \vdots \\ B_k \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_k \end{pmatrix}$$

Let $e = XB + \varepsilon - Y$

$$MSE = \frac{1}{n} \varepsilon e^2$$

$$= \frac{1}{n} e^T e$$

$$= \frac{1}{n} (XB - Y + \varepsilon)^T (XB - Y + \varepsilon)$$

$$= \frac{1}{n} (B^T X^T - Y^T + \varepsilon^T)(XB - Y + \varepsilon)$$

$$= \frac{1}{n} (B^T X^T XB - B^T X^T Y + B^T X^T \varepsilon - Y^T XB + Y^T Y - Y^T \varepsilon + \varepsilon^T XB - \varepsilon^T Y + \varepsilon^T \varepsilon)$$

We need to find B^* that minimizes loss fn.

$$\frac{d(MSE)}{dB} = 2X^T XB - X^T Y + X^T \varepsilon - Y^T X + \varepsilon^T X$$

$\frac{d(MSE)}{dB} = 0$ (for minimizing)

$\therefore 2X^T XB^* - X^T Y + X^T \varepsilon - Y^T X + \varepsilon^T X = 0$

$\Rightarrow 2X^T XB^* = (X^T Y - X^T \varepsilon + Y^T \varepsilon - \varepsilon^T X)$

$\Rightarrow B^* = \frac{1}{2} (X^T X)^{-1} (X^T Y - X^T \varepsilon + Y^T \varepsilon - \varepsilon^T X)$

$X^T X$ needs to be invertible for the above solution to exist.

For this, X has to be an M*N matrix (M>=N) and the columns have to be linearly independent, ie- no two features depend on each other.