Assignment 1: Report
Pankil Kalra, 2018061

QUESTION 1

Preprocessing Strategy:

- Videogame sataset had some rows with null values. These rows were dropped from the dataset.
- In Abalone dataset, “Sex” was a categorical variable. This column was dropped and instead of this column three new columns were added to the dataset – “sex_m”, “sex_f” and “sex_i” which contained binary values indication the sex species. This was done since categorical variables cant be used directly as features for linear regression.
- Both the daatsets were shuffled, using a fixed seed. (df.sample(frac = 1, random_state = 0).
- Both the datasets was separated into X and y numpy arrays.
- For both the datsets ,the input features array: X was normalised to bring consistency in the feature values.

Value of k:
The value of k depends on the size of dataset. The Abalone dataset consisted of around 4000 samples after preprocessing whereas the Videogame consisted of around 7000 samples after the same. Judging by the size of the dataset we can say that the number of folds should be less than(or equal) 10. This is is because as value of k becomes larger the bias decreases but the variance and the computation time increases.

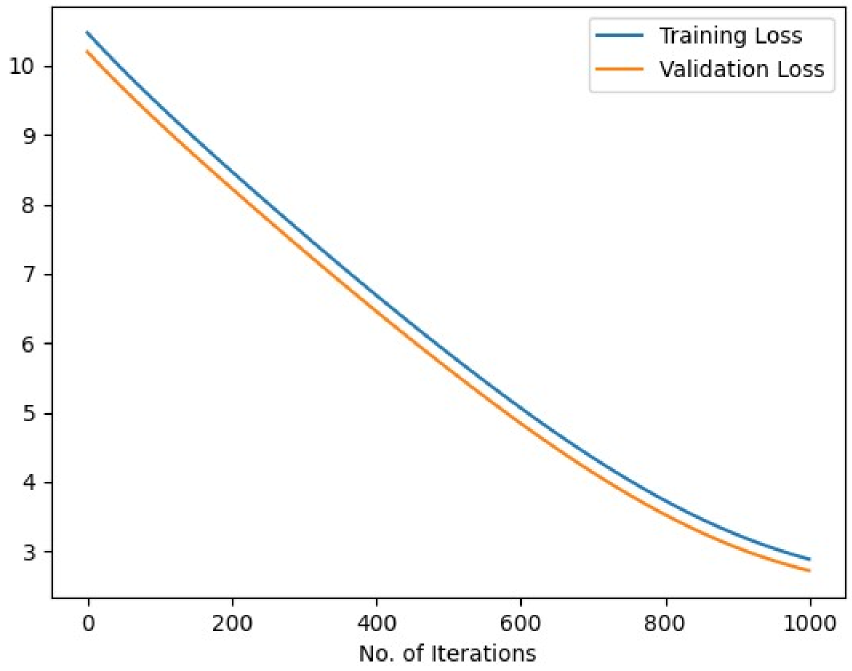
I tried all the k values from 3 to 10 and took the mean error for every split to get which k would be the most appropriate (for both RMSE loss and MAE loss).

It was found out that:
1) For Abalone dataset, k = 7 would be best for RMSE loss and k = 6 would be best for MAE loss.
2) For vidoegame dataset, k= 10 would be best for both RMSE and MAE losses.

Gradient descent has been implemented in scratch.py file.

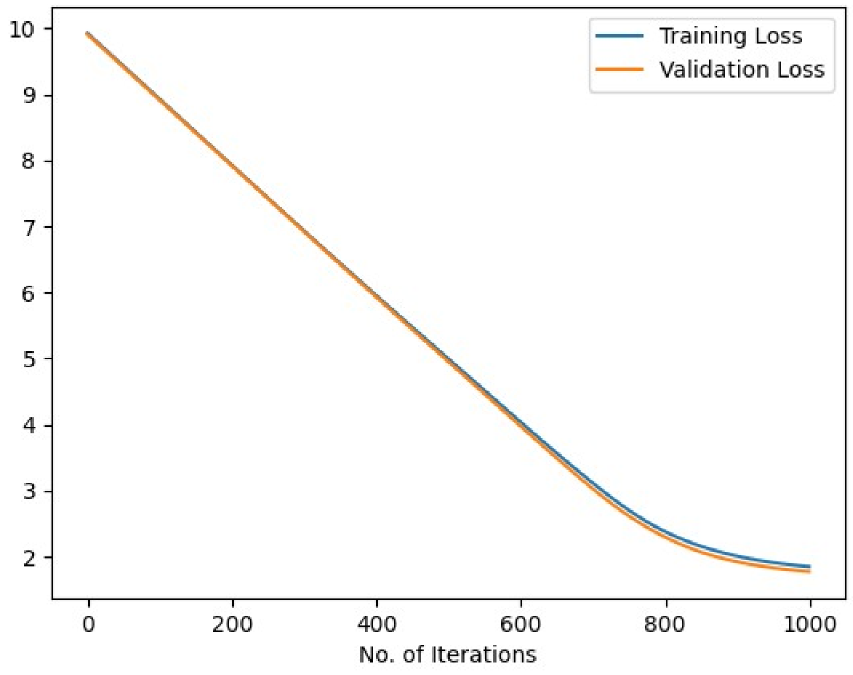
a) Abalone Dataset, Loss Used: RMSE

Graph for the best split when k was taken as 7. (The best split was when 4th fold was used as testing).



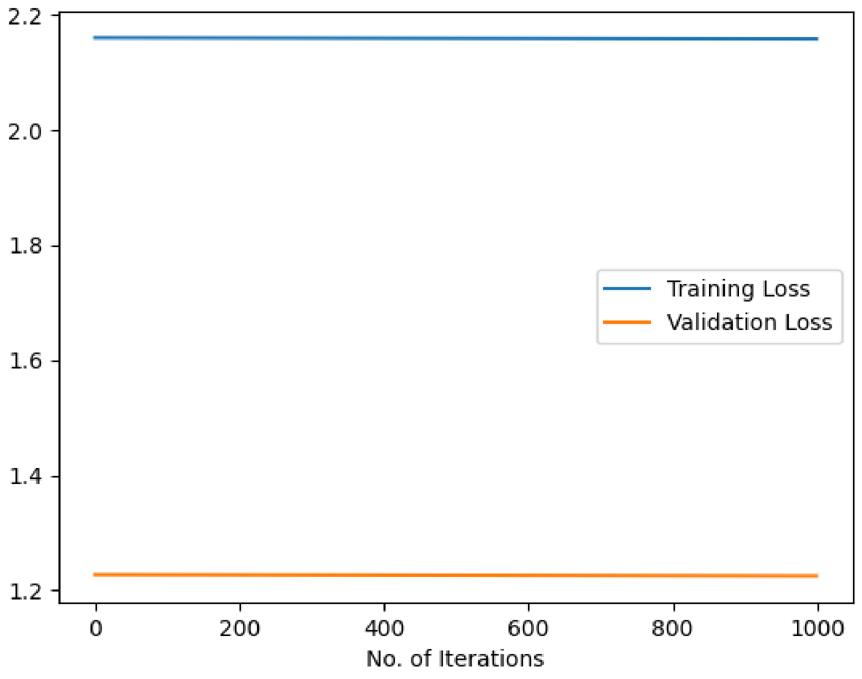
Abalone Datset, Loss Used: MAE

Graph for the best split when k was taken as 6. (The best split was when 4th fold was used as testing).



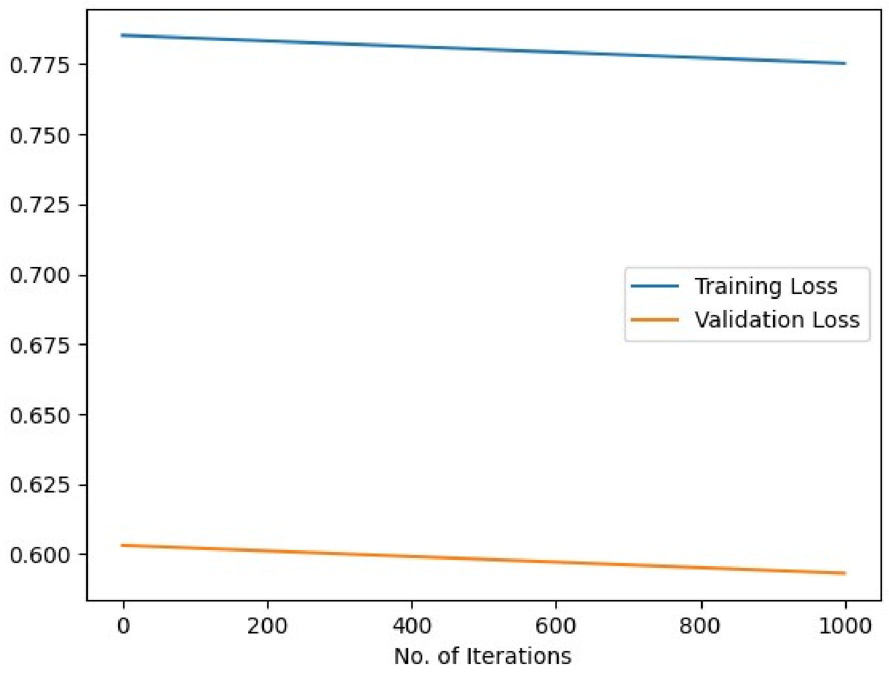
Videogame Dataset, Loss Used: RMSE

Graph for the best split when k was taken as 10. (The best split was when 5th fold was used as testing).



Videogame Dataset, Loss Used: MAE

Graph for the best split when k was taken as 10. (The best split was when 5th fold was used as testing).



b) For Abalone dataset, best RMSE value = 2.7210663479424273. K was taken to be 7, and this value was obtained when 4th fold was used for testing.

For Abalone dataset, best MAE value = 1.7737218783162927. K was taken to be 6, and this value was obtained when 4th fold was used for testing.

For Videogame dataset, best RMSE value = 1.2249934336816115. K was taken to be 10, and this value was obtained when 5th fold was used for testing.

For Videogame dataset, best MAE value =0.5931819212384858. K was taken to be 10, and this value was obtained when 5th fold was used for testing.

c) For Abalone dataset, **RMSE loss should be preferred**. Since data does not have many outliers the few outliers should be punished, so RMSE loss is preferred. The function is differentiable and gives deeper information than MAE.

For Videogame dataset, **MAE loss is more suitable**. Looking at graphs we can clearly see that RMSE losses are more than double than that of MAE loss. This is because Videogame consists of a lot of outliers. RMSE function is more sensitive to outliers than MAE function, so it penalises outliers more. But for this dataset, outliers should be punished less because they are not few in number. Dataset consists of a lot of outliers, so they should be punished less, so MAE should be used.

d)
RMSE >= MAE. RMSE for the same set of values i.e. for the same prediction values and actual values, is always more than MAE error.

RMSE and MAE are expected to give similar values when data consists of few outliers. This is because RMSE is sensitive to outliers and it punished them more harshly than MAE as it has a square function.

In such a case i.e the data has few outliers, **RMSE** should be preferred
RMSE would be better as it does not use the absolute function. MAE uses the absolute function. It is difficult to work with absolute function in mathematical analysis. One major reason for that is that the **absolute function is not differentiable** at all real values. Even though absolute values is easier to interpret, RMSE function is differentiable and gives a bit more information about the data than MAE.

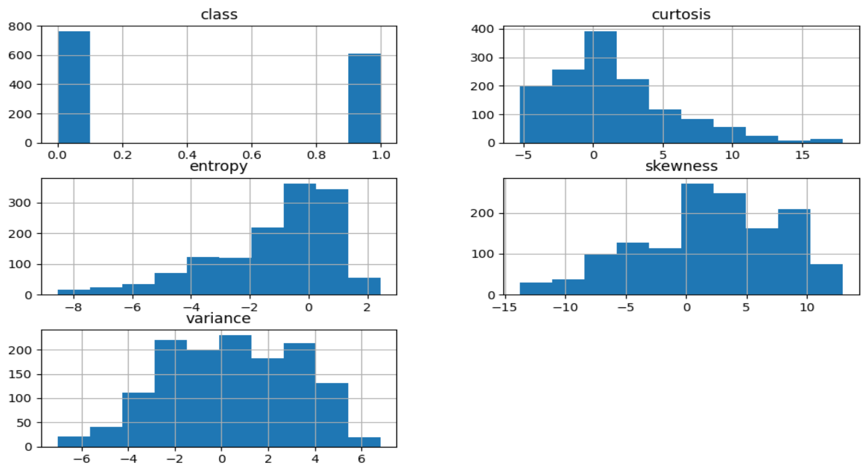
e)
Computing the training and validation loss for model based on Abalone Dataset and with RMSE loss function on the best split which is on split 4(when 4th fold is testing set) when k = 7.

Optimal parameters:
```
[[ 0.22402044]
 [ 0.1395298 ]
 [-0.14668139]
 [-0.05503625]
 [ 1.09896012]
 [ 0.45006952]
 [ 4.40093258]
 [-4.39142556]
 [-1.15978018]
 [ 1.21673703]
 [ 9.93368446]]
```

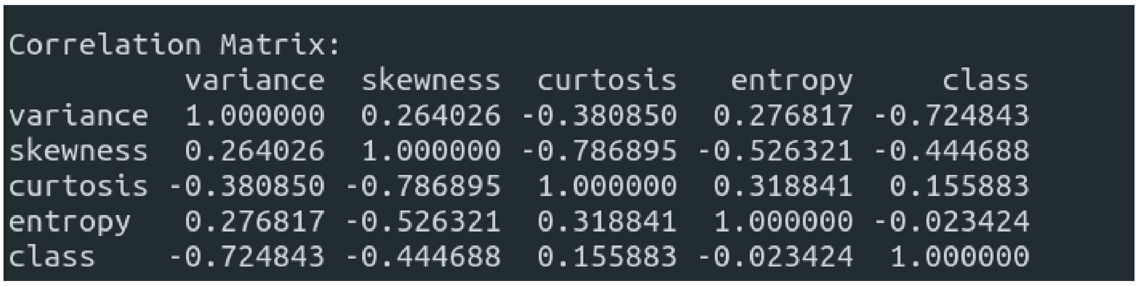
Training loss: 2.2112242988725246
Validation loss: 2.0810340003912815

QUESTION 2

Exploratory Data Analysis:
For EDA, Class variable distribution in the dataset was calculated, and histogram was plotted for the same which gives us information about frequency of the different variables.



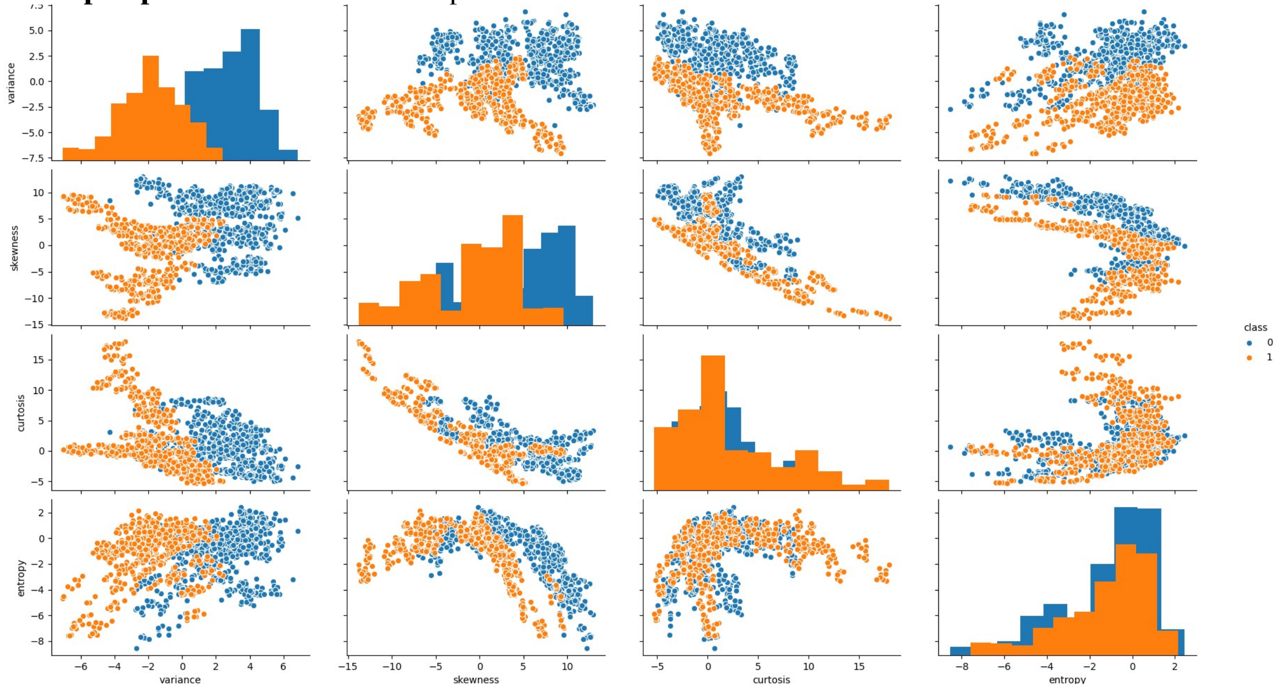
Correlation matrix was computed for the dataset.



```
Correlation Matrix:
          variance  skewness  curtosis   entropy     class
variance  1.000000  0.264026 -0.380850  0.276817 -0.724843
skewness  0.264026  1.000000 -0.786895 -0.526321 -0.444688
curtosis -0.380850 -0.786895  1.000000  0.318841  0.155883
entropy   0.276817 -0.526321  0.318841  1.000000 -0.023424
class    -0.724843 -0.444688  0.155883 -0.023424  1.000000
```

It was observed variance had strong negative correlation with the class variable and skewness had moderately negative correlation with the class variable. Rest of the features did not seem to have significant correlation with the class variable as correlation values were close to zero. Negative correlation in this case means: as the value of variance and skewness increase there is greater probability of the class variable being 0, and when they decrease there are greater chances of class variable being 1.

Also a **pairplot** for the dataset was plotted.



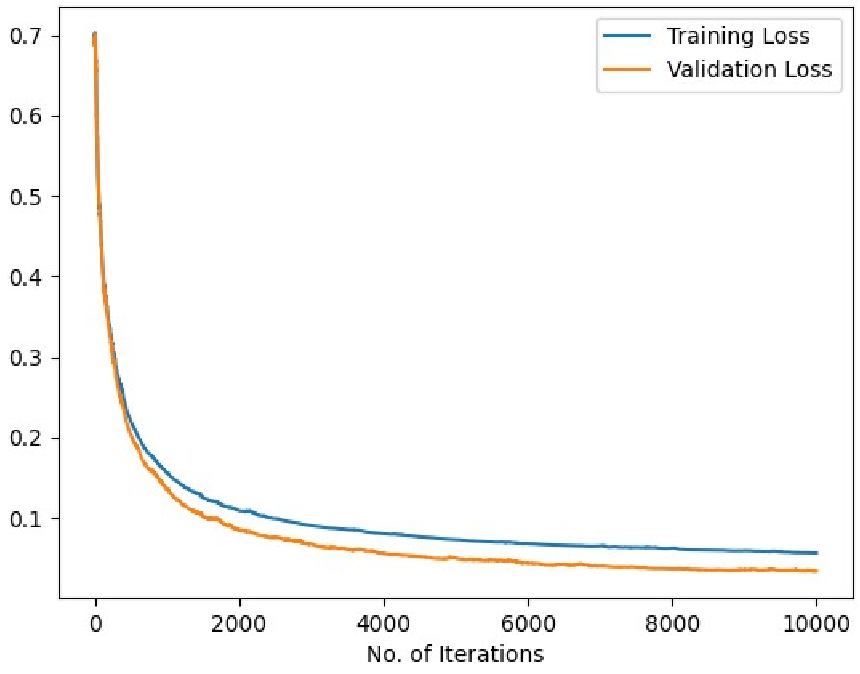
It gives us information about distribution and dependence of class variable on different other features.

RUNING THE MODELS:

SGD

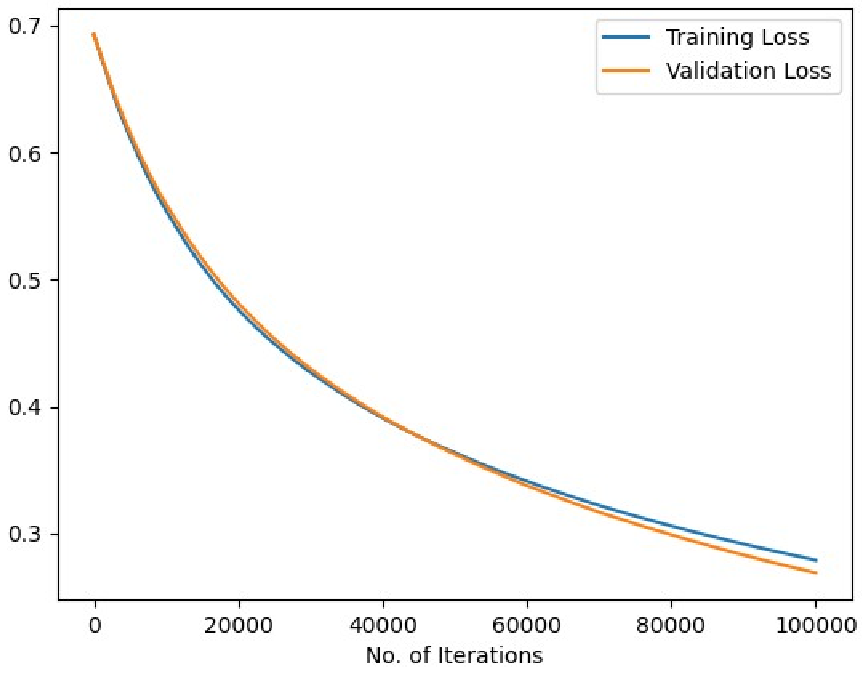
1)Appropriate Paramaters taken- Learning rate: 0.03, No. of iterations: 10,000

Training Accuracy: 97.81477627471385 %
Testing Accuracy: 100.0 %



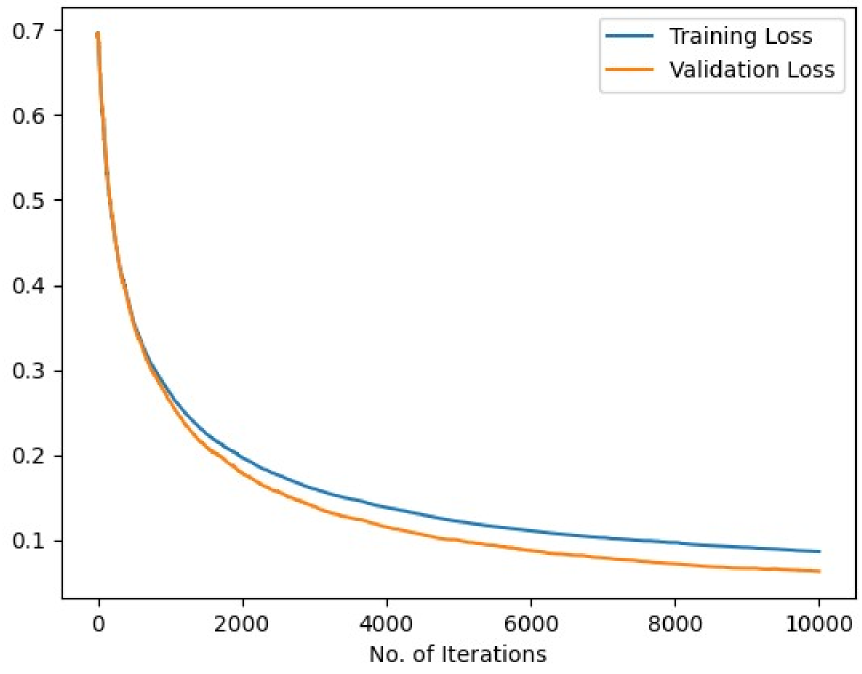
2)For Learning rate: 0.0001, No. of iterations: 100,000

Training Accuracy: 92.92403746097816 %
Testing Accuracy: 92.7007299270073 %



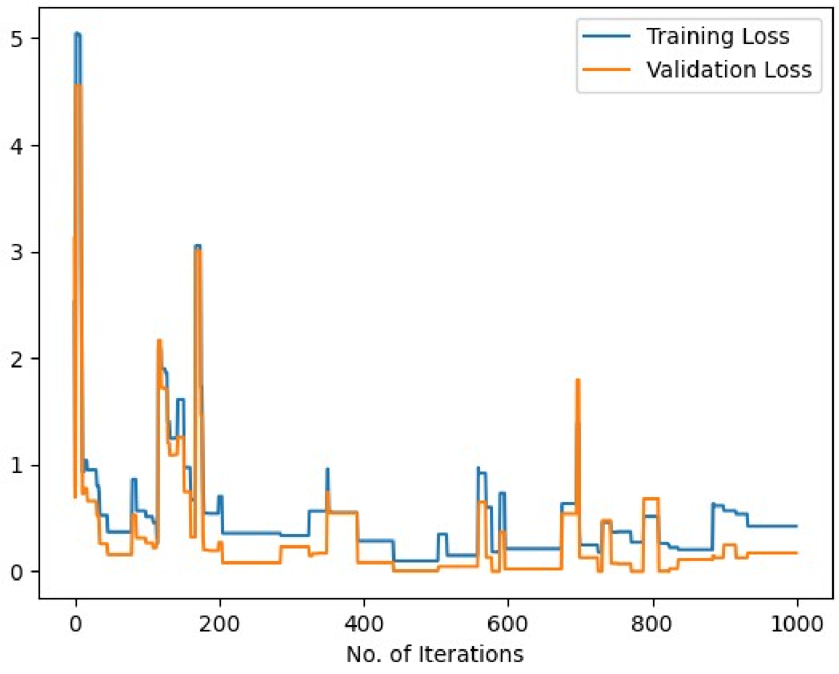
3) For Learning rate: 0.01, No. of iterations: 10,000

Training Accuracy: 97.1904266389178 %
Testing Accuracy: 99.27007299270073 %



4)
For Learning rate: 10, No. of iterations: 1,000

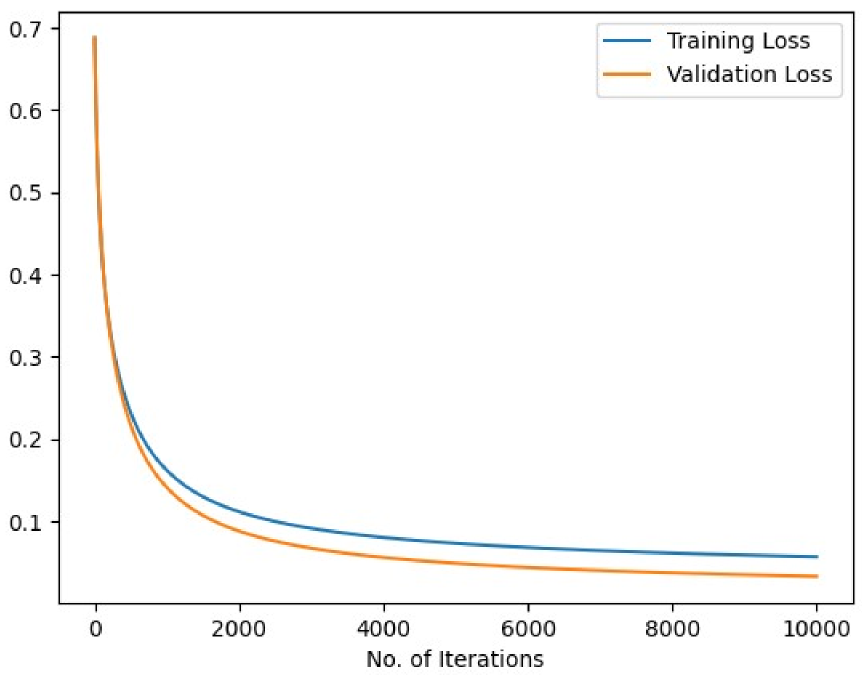
Training Accuracy: 96.56607700312175 %
Testing Accuracy: 98.17518248175182 %



BGD

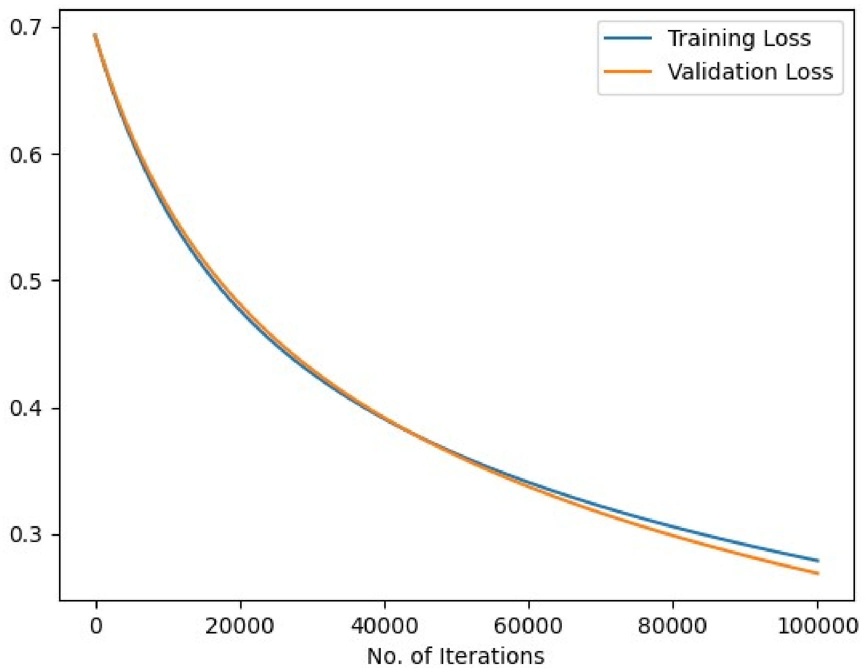
1)Appropriate Paramaters taken- Learning rate: 0.03, No. of iterations: 10,000

Training Accuracy: 97.50260145681582 %
Testing Accuracy: 99.63503649635037 %



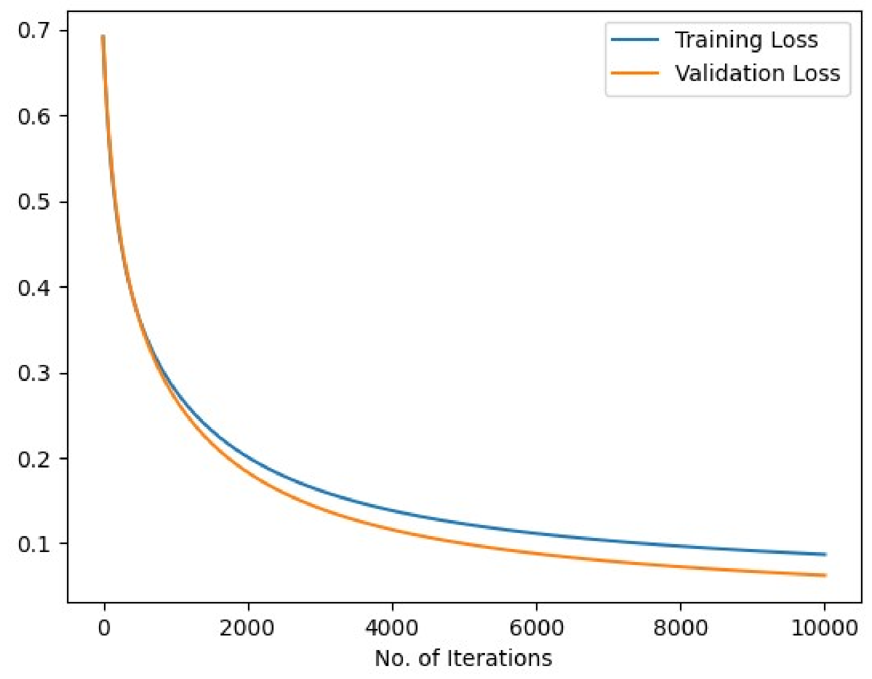
2)For Learning rate: 0.0001, No. of iterations: 100,000

Training Accuracy: 92.7159209157128 %
Testing Accuracy: 92.33576642335767 %



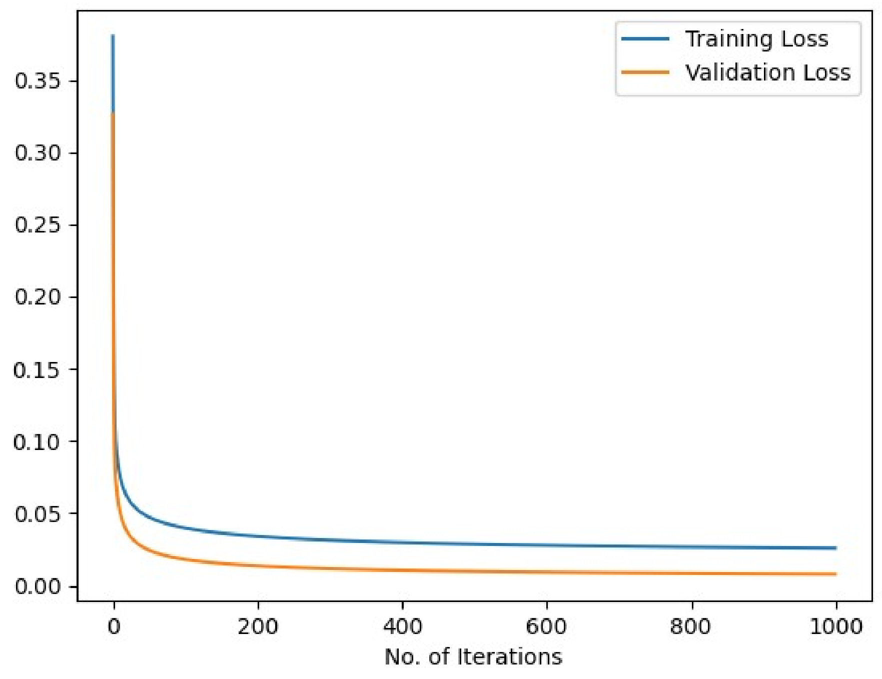
3) For Learning rate: 0.01, No. of iterations: 10,000

Training Accuracy: 97.1904266389178 %
Testing Accuracy: 99.27007299270073 %



4)
For Learning rate: 10, No. of iterations: 1,000

Training Accuracy: 98.43912591050989 %
Testing Accuracy: 99.27007299270073 %



Comparison:
a)
Clearly in all the plots of SGD we can see it is not smooth and the loss plot has some little spike like disturbances. This is more evident when learning rate has a higher value like 0.01 or 10.

All the loss plots of BGD are smooth with little or no spikes compared to SGD. The loss have a downward trend. The SGD loss curves also have a general downward trend but they make go up in little bursts(due to outliers) which is not the case for BGD.

b)
At lower values of learning_rates, it takes approximately same number of iterations for both SGD and BGD to converge.

There is a clear difference for higher learning rates, as we can see in the plots for learning_rate = 10, BGD converges at around 100 epochs, but SGD does not converge even after 100 epochs.

Convergence points:
1) For learning rate = 0.003: Both SGD and BGD converge at around 10,000 epochs.

2) For learning rate = 0.0001: Both SGD and BGD do not converge after 100,000 epochs(they are on the path to do so but number of were less due to computational constraints). Even though both haven’t converge at 100,000 epochs, both SGD and BGD had similar loss values at the end.

3) For learning rate = 0.01: Both SGD and BGD converge at around 8,000 epochs.

4) For learning rate = 10: SGD does not converge after 1000 epochs but BGD converges after 100 epochs. Clear difference in the number of epochs taken by both models to converge is observed.

c) Implemented Logistic Regression and SGD Classifier using sklearn in the file: sklearn_logistic.py .

d) Both were implemented using sklearn.

Logistic Regression:

Training accuracy: 97.81477627471385 %
Test accuracy: 100.0 %

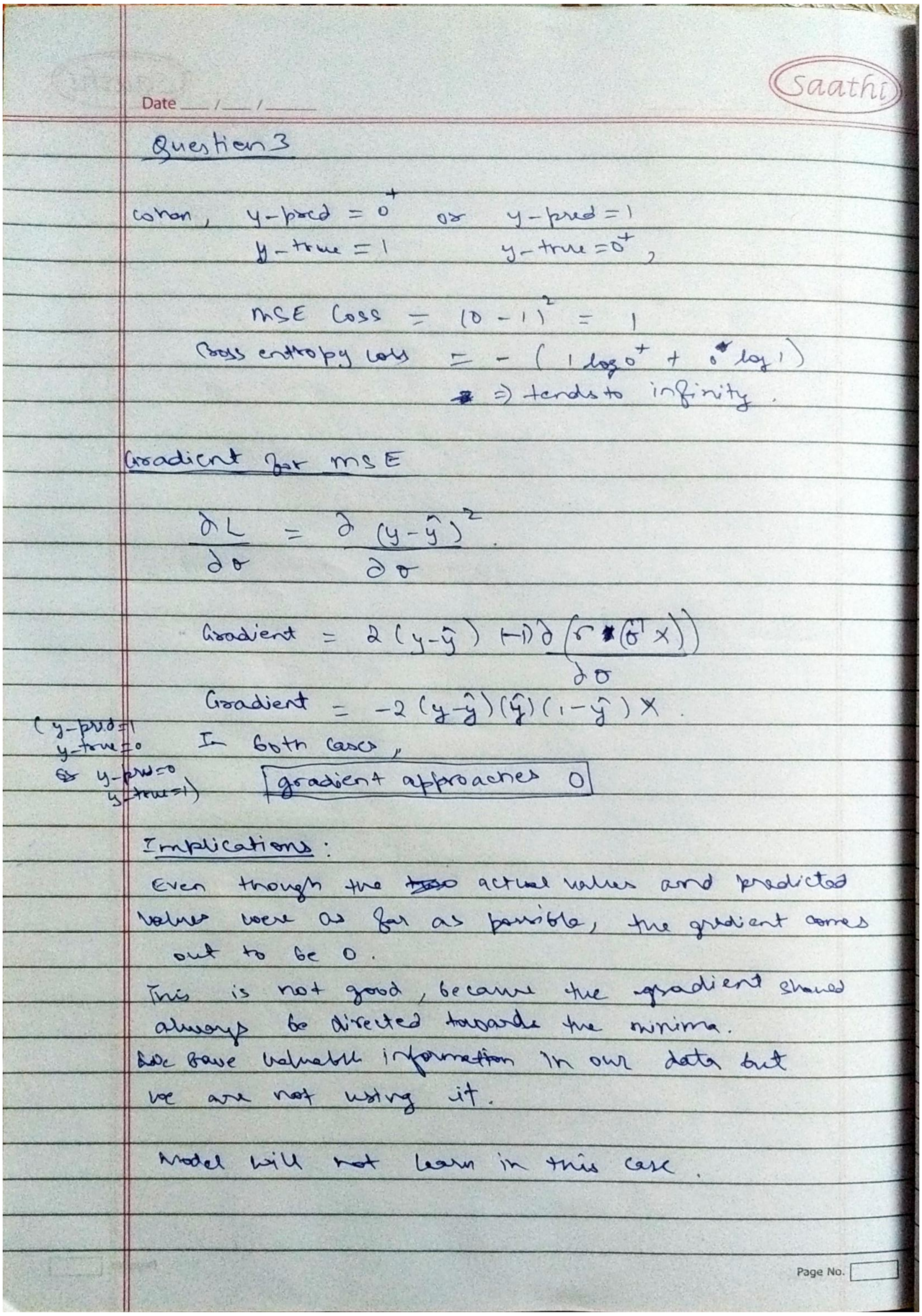
SGD Classifier:

Hyperparameters used were same as the appropriate parameters calculated for my SGD implementation.

Learning Rate = 0.03, Number of Iterations = 10000.

Training accuracy: 97.29448491155047 %
Test accuracy: 99.27007299270073 %

QUESTION 3



Date ___/___/____

Saathi

Question 3

when, $y\text{-pred} = 0^+$ or $y\text{-pred} = 1$

$y\text{-true} = 1$ $\quad y\text{-true} = 0^+$,

$$\text{MSE Loss} = (0-1)^2 = 1$$

Cross entropy loss $= -(1 \log 0^+ + 0 \log 1)$

$\Rightarrow$ tends to infinity.

Gradient for MSE

$$\frac{\partial L}{\partial \theta} = \frac{\partial (y-\hat{y})^2}{\partial \theta}.$$

$$\text{Gradient} = 2(y-\hat{y})(-1)\frac{\partial(\sigma(\theta^T x))}{\partial \theta}$$

$$\text{Gradient} = -2(y-\hat{y})(\hat{y})(1-\hat{y})X.$$

(y-pred=1 y-true=0 or y-pred=0 y-true=1) In both cases,

$\boxed{\text{gradient approaches } 0}$

Implications:

Even though the ~~two~~ actual values and predicted values were as far as possible, the gradient comes out to be 0.

This is not good, because the gradient should always be directed towards the minima.

We have valuable information in our data but we are not using it.

Model will not learn in this case.

Page No.

Date ___ / ___ / ___

If we use cross entropy loss

~~$\sum \sum (y \log \hat{y} + \hat{y} \log y$~~

$$L = -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$$

$$\frac{\partial L}{\partial \theta} \text{ (gradient)} = -\left(\frac{y}{\cancel{\hat{y}}}\cancel{\hat{y}}(1-\hat{y}) + \frac{(1-y)}{\cancel{(1-\hat{y})}}(-1)\hat{y}\cancel{(1-\hat{y})}\right).X$$

$$\text{gradient} = -(y(1-\hat{y}) - \hat{y}(1-y))\,X$$

$$\text{gradient} = -(y - \cancel{y\hat{y}} - \hat{y} + \cancel{\hat{y}y}).X$$

$$\text{gradient} = (\hat{y} - y)\,X$$

~~In both cases gradient comes out~~ to ~~be~~ X.

when y pred = 1 y_true = 0

gradient = X.

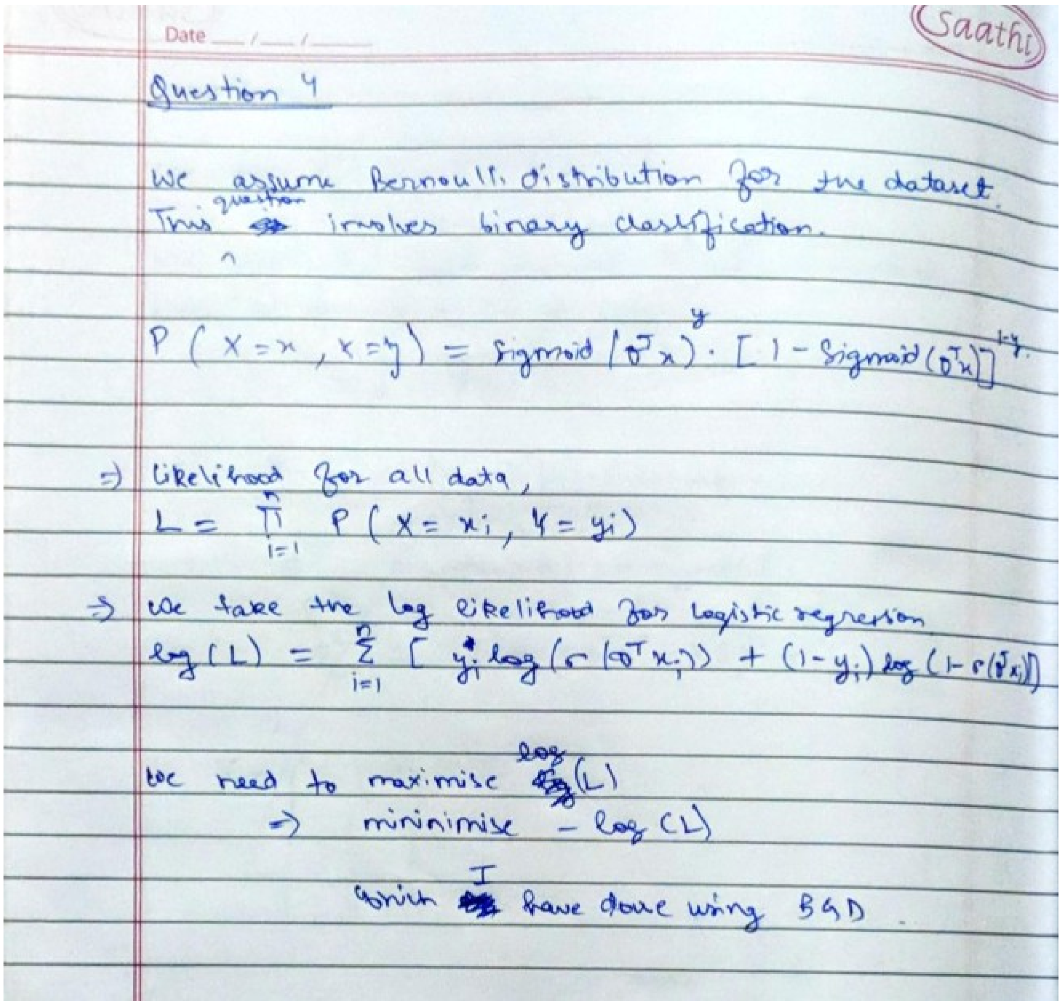
when y_pred = 0 y_true = 1

gradient = −X.

In both the cases, model is learning and the parameters will get updated accordingly.

Page No.

QUESTION 4



Date ___/___/____

Question 4

We assume Bernoulli distribution for the dataset.
This question ~~[illegible]~~ involves binary classification.

$$P(X=x, Y=y) = \text{Sigmoid}(\theta^T x)^{y} \cdot [1 - \text{Sigmoid}(\theta^T x)]^{1-y}.$$

$\Rightarrow$ Likelihood for all data,

$$L = \prod_{i=1}^{n} P(X = x_i, Y = y_i)$$

$\Rightarrow$ We take the log likelihood for logistic regression

$$\log(L) = \sum_{i=1}^{n} [\, y_i \log(\sigma(\theta^T x_i)) + (1-y_i)\log(1-\sigma(\theta^T x_i))]$$

We need to maximise ~~log~~ log (L)

$\Rightarrow$ mininimise $-\log(L)$

which ~~[illegible]~~ I have done using BGD.

Implemented using code in the file Question4.py .
a)
Beta_1 = 0.1615671 (for percentage)
Beta_2 = 0.29575933 (for age)
Beta_0 = -10.91769498 (bias)

b) Response Function

y_hat = 1/(1+exp(-(beta_0 + beta_1*x1 + beta_2*x2)))

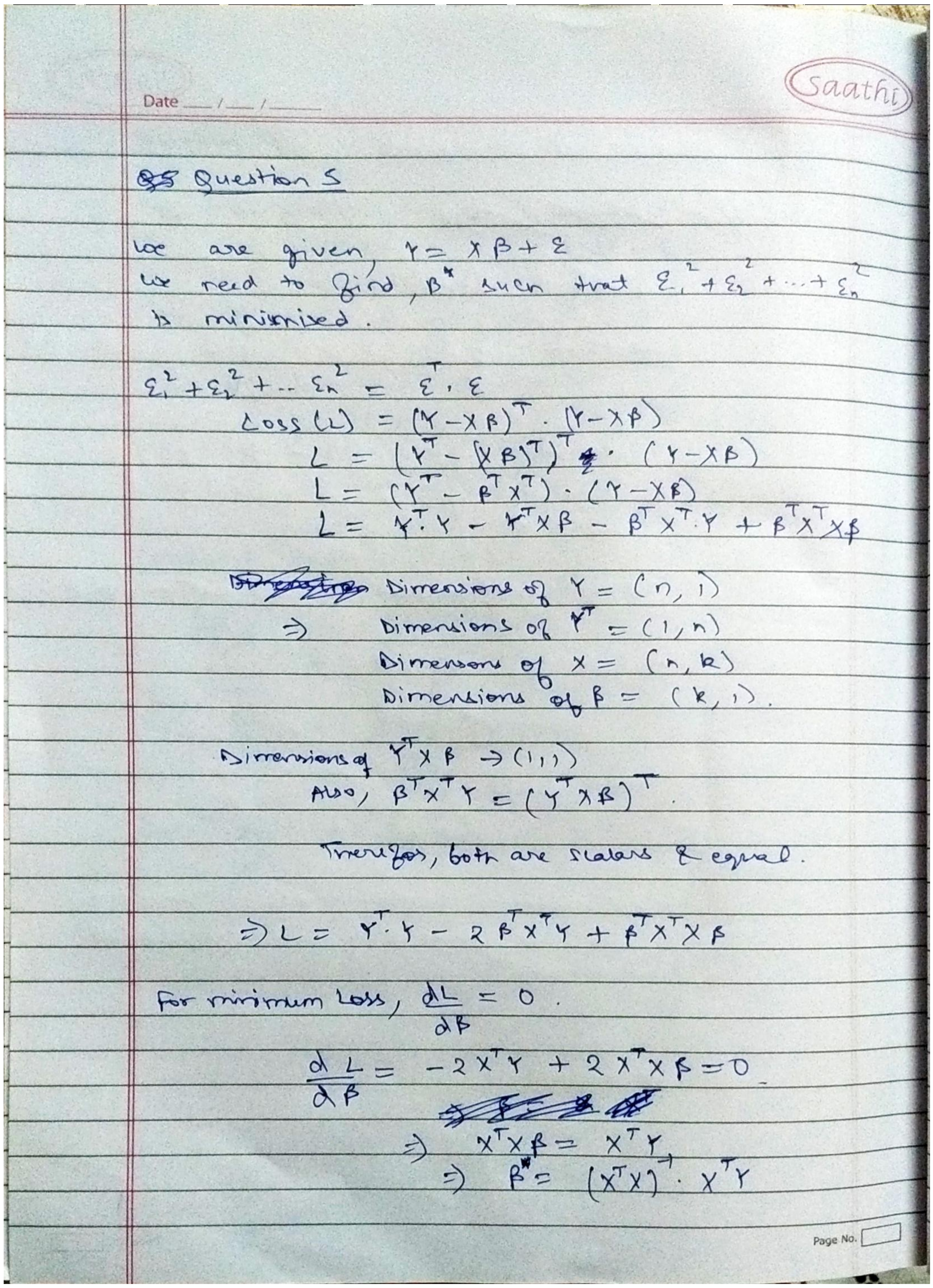
c)
exp(beta_1): 1.1753513252582777
exp(beta_2): 1.3441466238359774

Exp(beta_1) denotes that with with every unit increase in the percentage of spread the odds of disease recurring to not recurring increase by a factor of 1.175.

Exp(beta_2) denotes that with with every unit increase in the age of child the odds of disease recurring to not recurring increase by a factor of 1.344

d) Probability using code: 0.857093497129094

QUESTION 5



Saathi

Date ___ / ___ / ______

~~Q5~~ Question 5

We are given, $Y = X\beta + \varepsilon$
We need to find, β^* such that $\varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_n^2$
is minimised.

$\varepsilon_1^2 + \varepsilon_2^2 + \dots \varepsilon_n^2 = \varepsilon^T \cdot \varepsilon$

$\text{Loss } (L) = (Y - X\beta)^T \cdot (Y - X\beta)$

$L = (Y^T - (X\beta)^T)^T \cdot (Y - X\beta)$

$L = (Y^T - \beta^T X^T) \cdot (Y - X\beta)$

$L = Y^T \cdot Y - Y^T X\beta - \beta^T X^T \cdot Y + \beta^T X^T X\beta$

~~Dimensions~~ Dimensions of $Y = (n, 1)$
$\Rightarrow$ Dimensions of $Y^T = (1, n)$
Dimensions of $X = (n, k)$
Dimensions of $\beta = (k, 1)$.

Dimensions of $Y^T X \beta \rightarrow (1, 1)$
Also, $\beta^T X^T Y = (Y^T X \beta)^T$.

Therefore, both are scalars & equal.

$\Rightarrow L = Y^T \cdot Y - 2\beta^T X^T Y + \beta^T X^T X \beta$

For minimum Loss, $\frac{dL}{d\beta} = 0$.

$\frac{dL}{d\beta} = -2X^T Y + 2X^T X\beta = 0$

$\Rightarrow X^T X \beta = X^T Y$

$\Rightarrow \beta^* = (X^T X)^{-1} \cdot X^T Y$

Page No.

To check, this β^* gives minimum value,
we need to differentiate again

$\Rightarrow$ Double derivative $= X^T X$.

For this to be positive, X
should have a full rank.

Conditions under optimum parameters will exist:

(1) $(X^T X)$ must be invertible (or non singular).
(2) $X^T X$ must have a full rank.