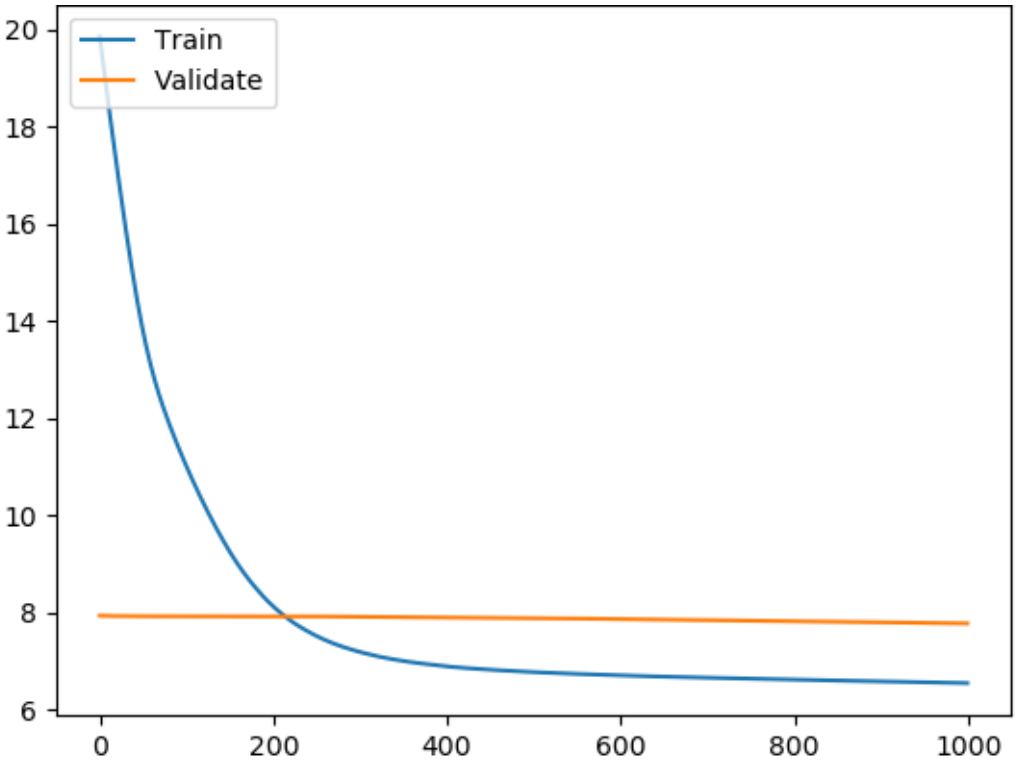
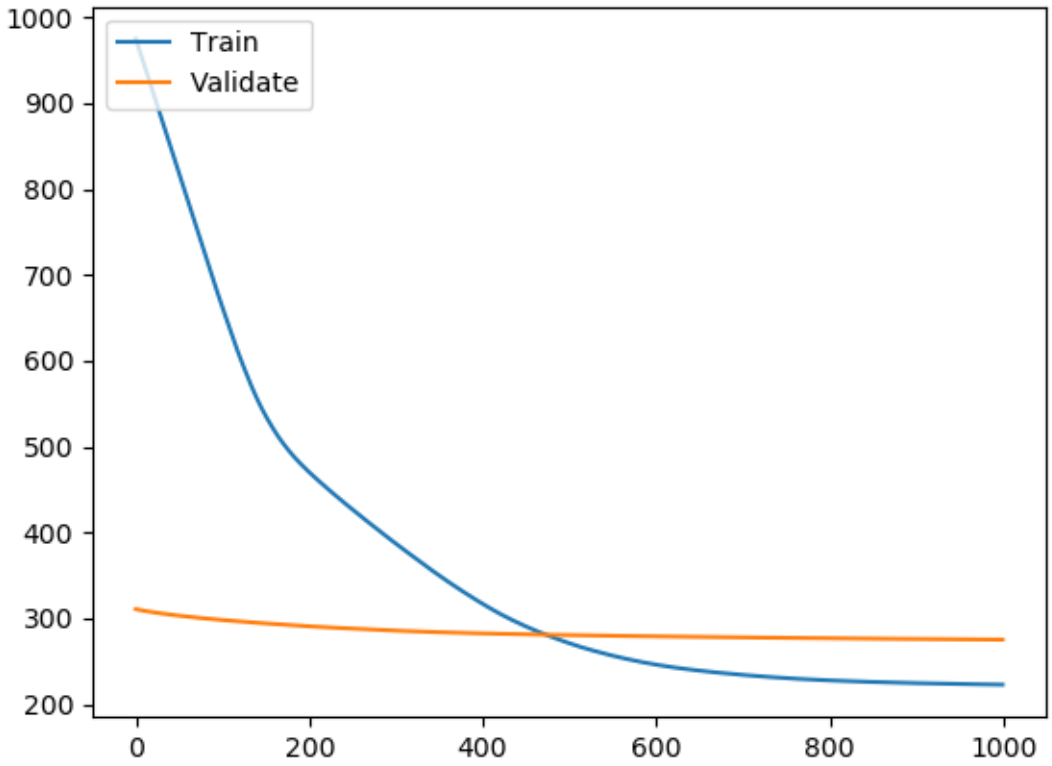
Q1. A, B) Dataset 1 RMSE



Best Test Loss – 5.48433 Best Validation Loss – 5.0594

K=10, Fold = 4

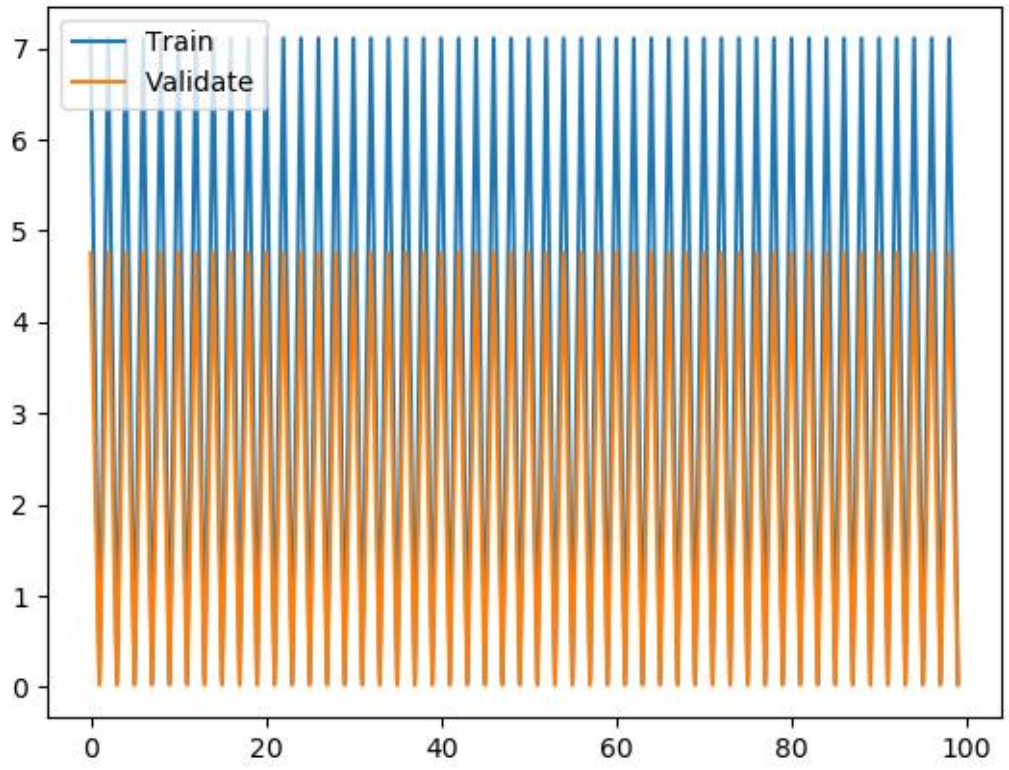
Dataset 1 MAE



Best Test Loss – 270.7969 Best Validation Loss – 113,577

K=10, Fold = 7

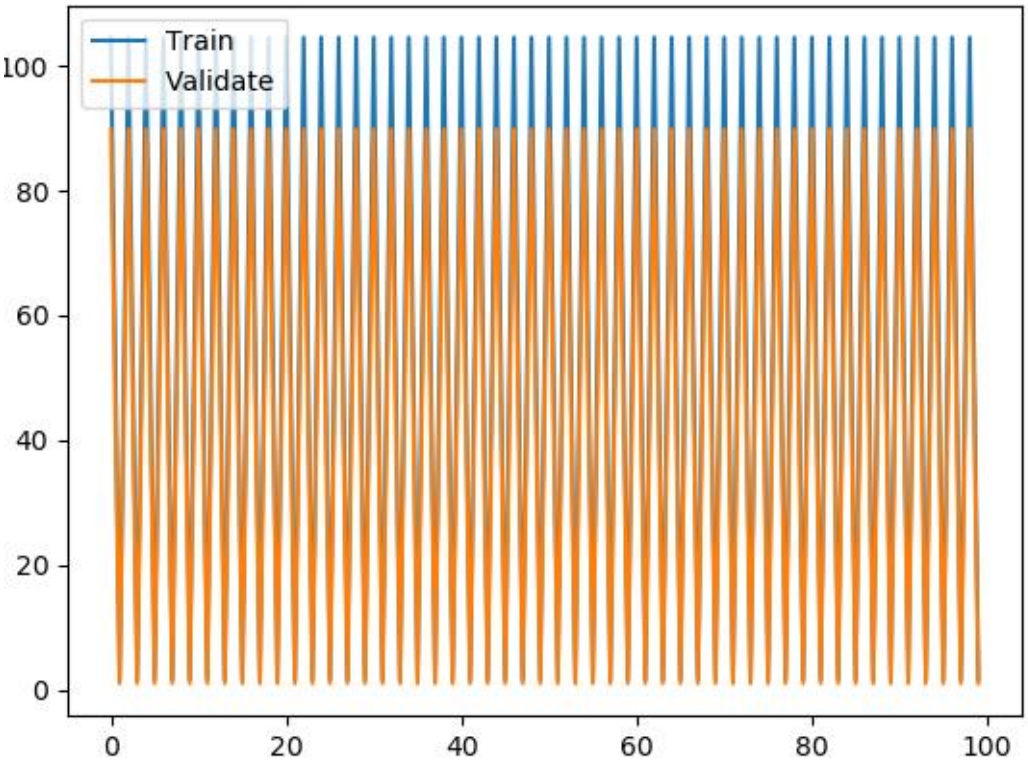
Dataset 2 RMSE



Best Test Loss – 0.03138

Best Validation Loss – 0.03774

K=10, Fold = 3



Dataset 2 MAE

Best Test Loss – 1.507 Best Validation Loss – 1

K=10, Fold = 4

C, D) RMSE and MAE are expected to give similar values when the predicted value is in +1/-1 range of the true value. In general MAE <= RMSE. MAE should be used when there is no difference between a small error and a large error but in all other cases RMSE is preferred.

For Dataset 1, RMSE converged faster and had a smaller loss than MAE. In the end MAE did not converge even after 1000 epochs and so for better speed, I would prefer RMSE.

For Dataset 2, both RMSE and MAE gave somewhat similar results because the data itself is faulty. In our case, we only use Critic and User score as our data samples which turn out to be not so good features hence the loss turns out to be similar to an oscillating function. RMSE is preferred because it gave better results.

Below is an example from the dataset 1 and it shows there is no pattern which can be formed from only the 2 features we are using.

Sales	1	2
82.53	76	51
35.52	82	73
32.77	80	73
29.8	89	65
28.92	58	41

I chose K =10 because it is a general standard and gives a model with low bias and moderate variance.

For dataset 1, the gender was converted to 0/1/3 and part from that I scaled y by 100 for better plots×.

For dataset 2, I converted user score to the scale of 100 because the critic score was from 1-100. Also, ceiled the y values for better prediction as integers rather than float.

D)

Theta = $(X'X)^{-1}X'Y$

Test Loss – 3.7469 Validation Loss – 8.178255

This normal equation form is a higher bias fit than the RMSE form.

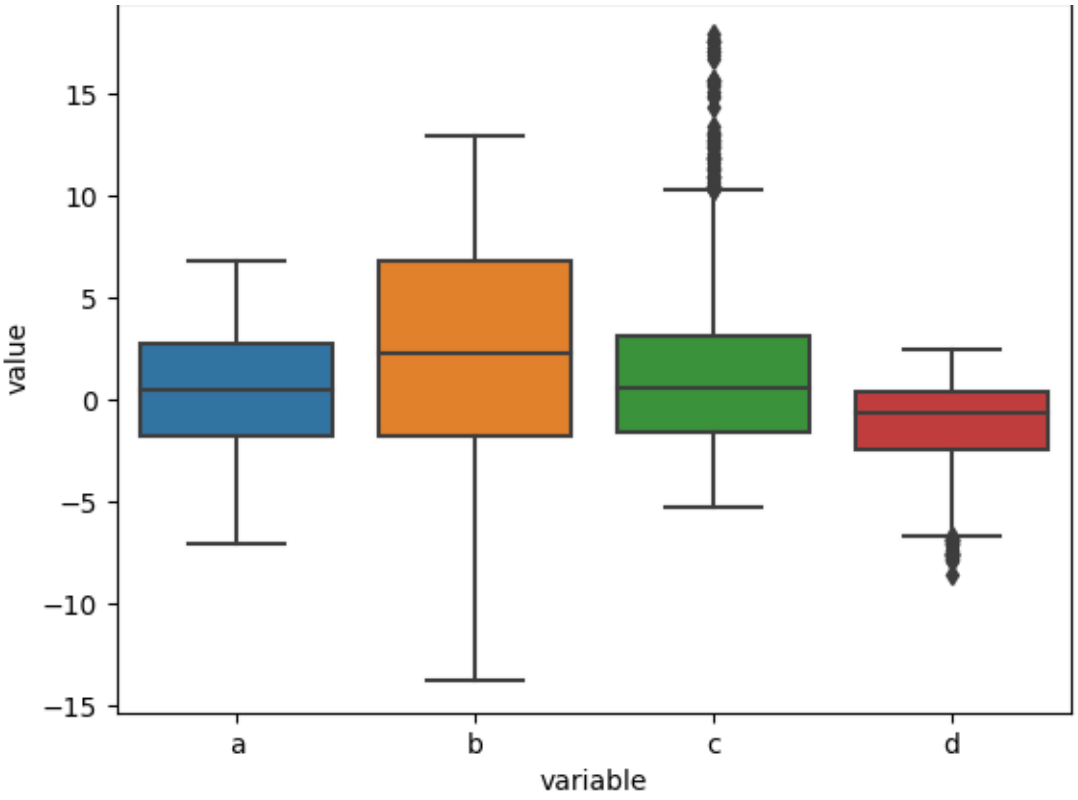
Q2. Below are the max and min values of the 4 features in this dataset

6.8248 -7.0421 Mean – 0.4337 Median – 0.496
12.9516 -13.7731 Mean – 1.9223 Median – 2.319
17.9274 -5.2861 Mean – 1.3976 Median – 0.616
2.4495 -8.5482 Mean – -1.191 Median - -0.58

Class Distribution – 0 - 762

1 - 610

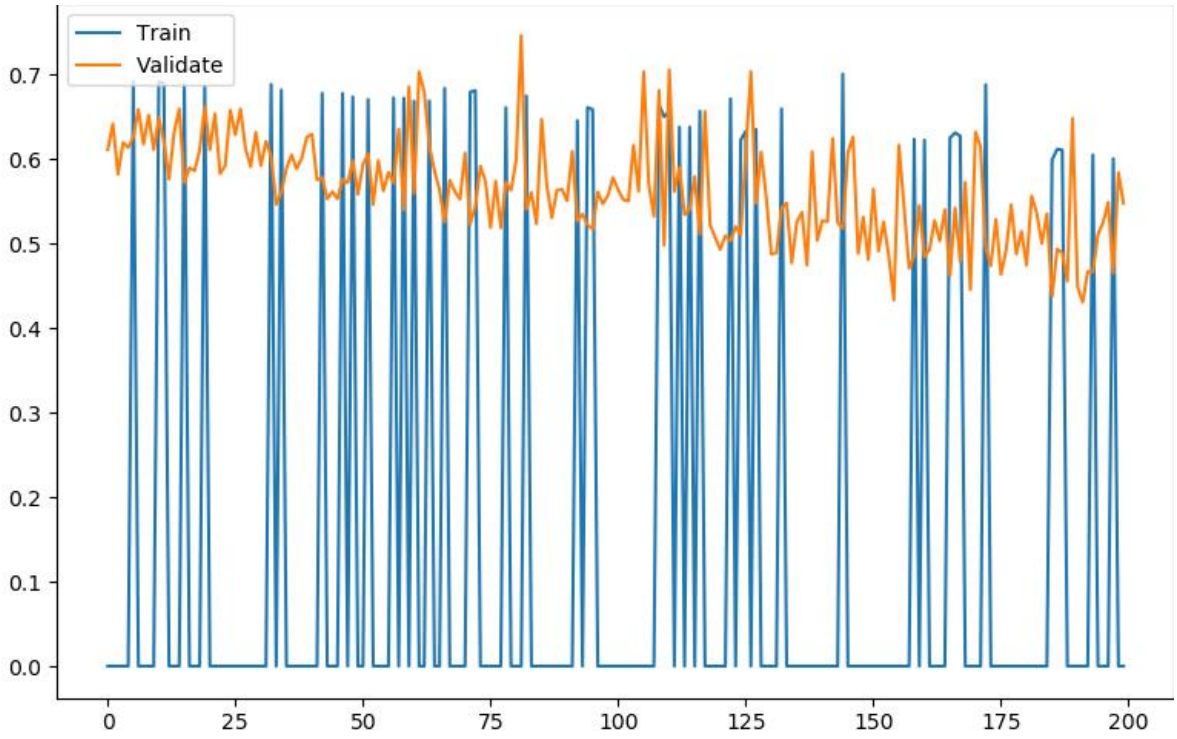
Class



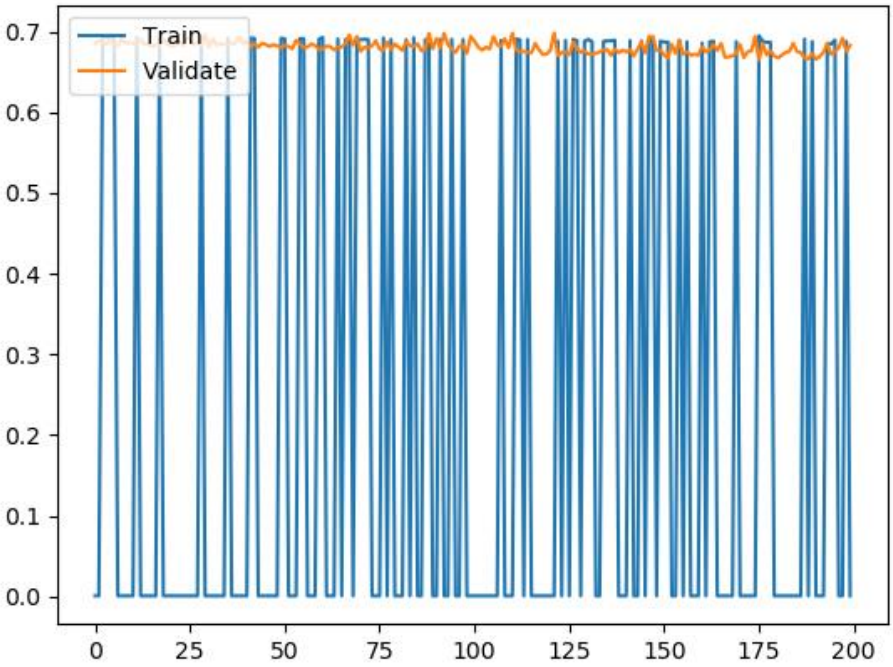
Distribution is almost 1:1 ratio which gives ample data points of each category and improves model performance.

This boxplot shows that the 3rd and 4th features contain a significant amount of outliers but other than that all the features are on the same scale and similar in value.

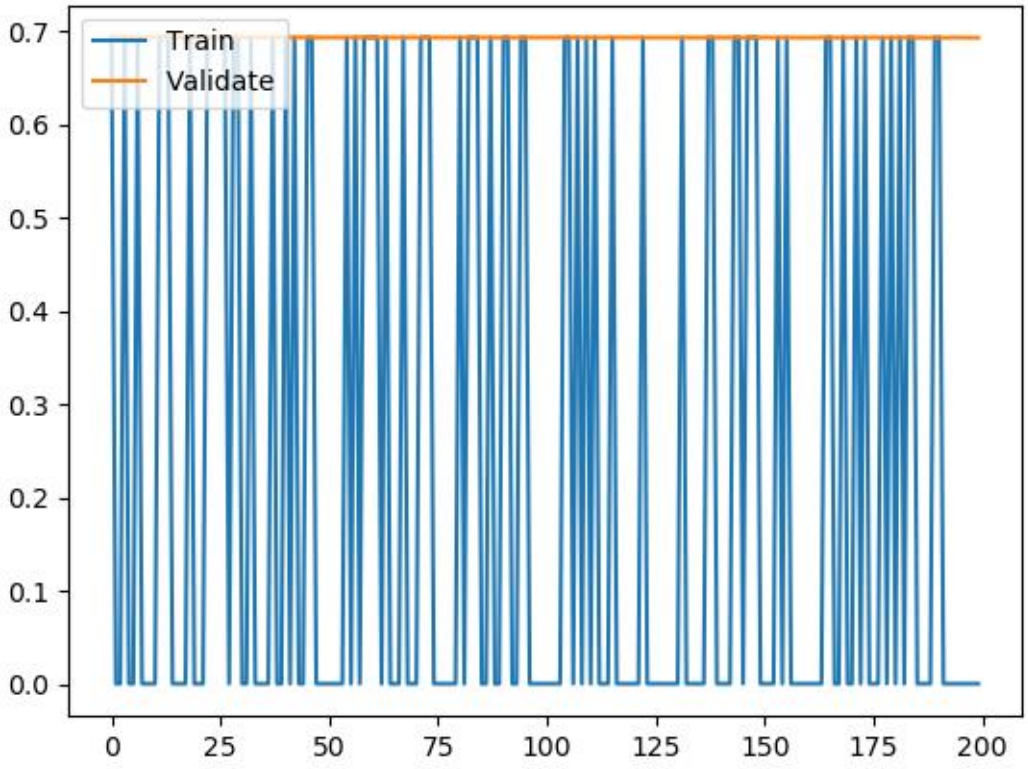
Using learning rate=0.1 and number of epochs=200 for SGD.



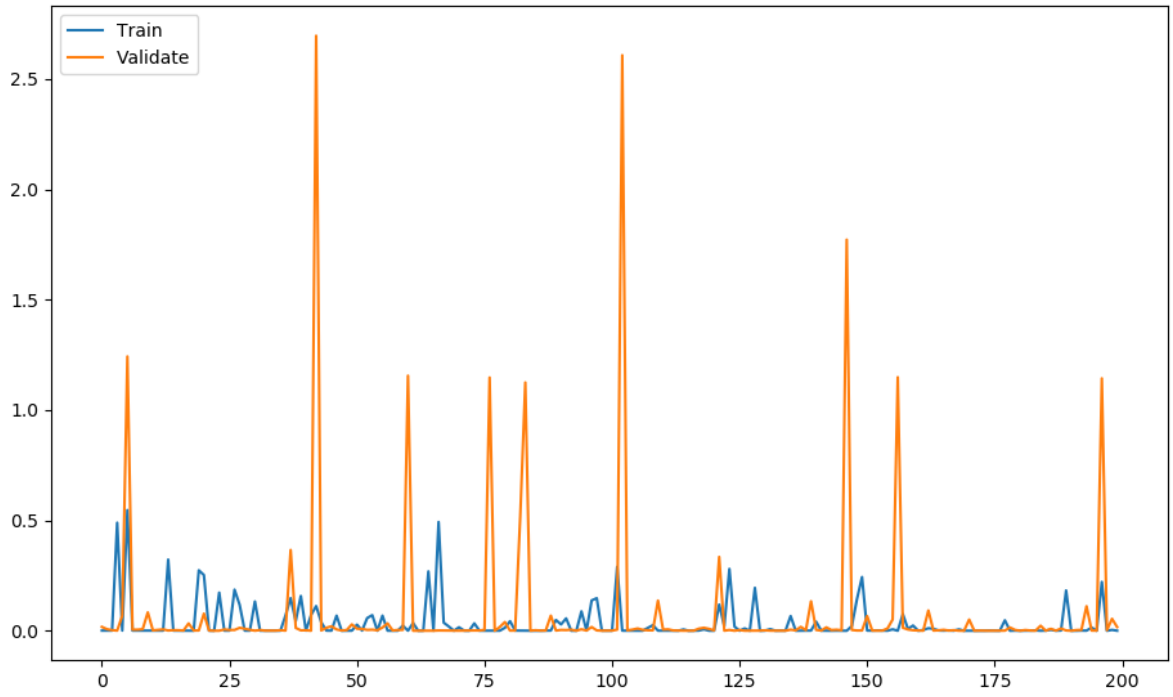
With learning rate changed to 0.01



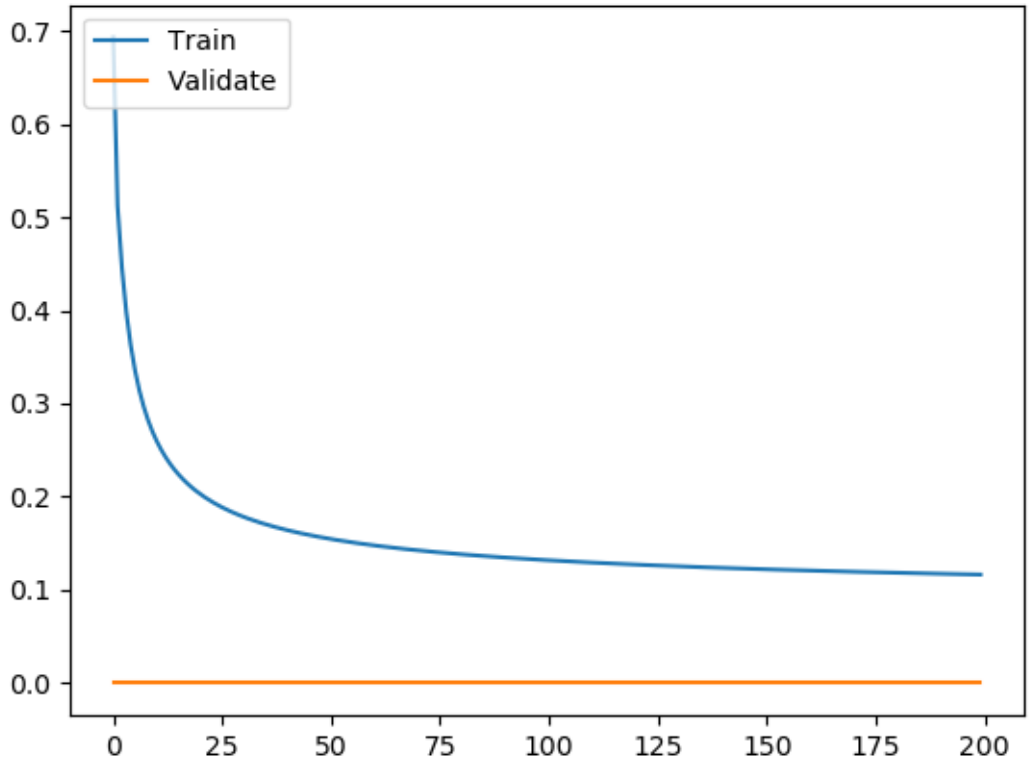
With learning rate changed to 0.0001



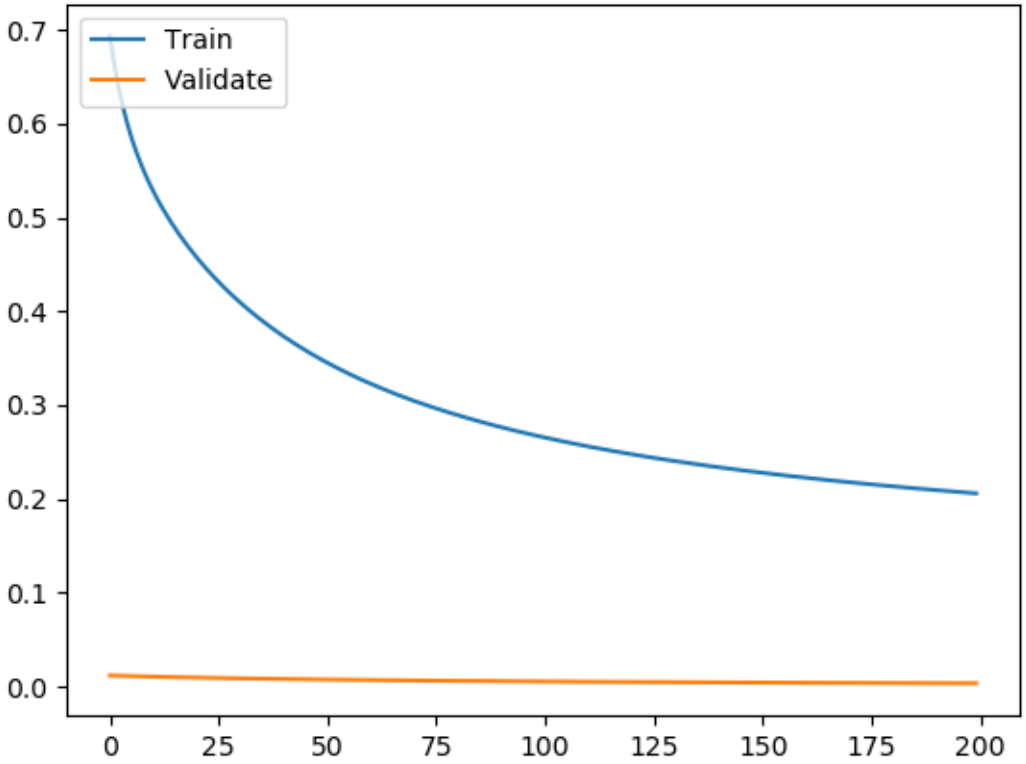
With learning rate changed to 10



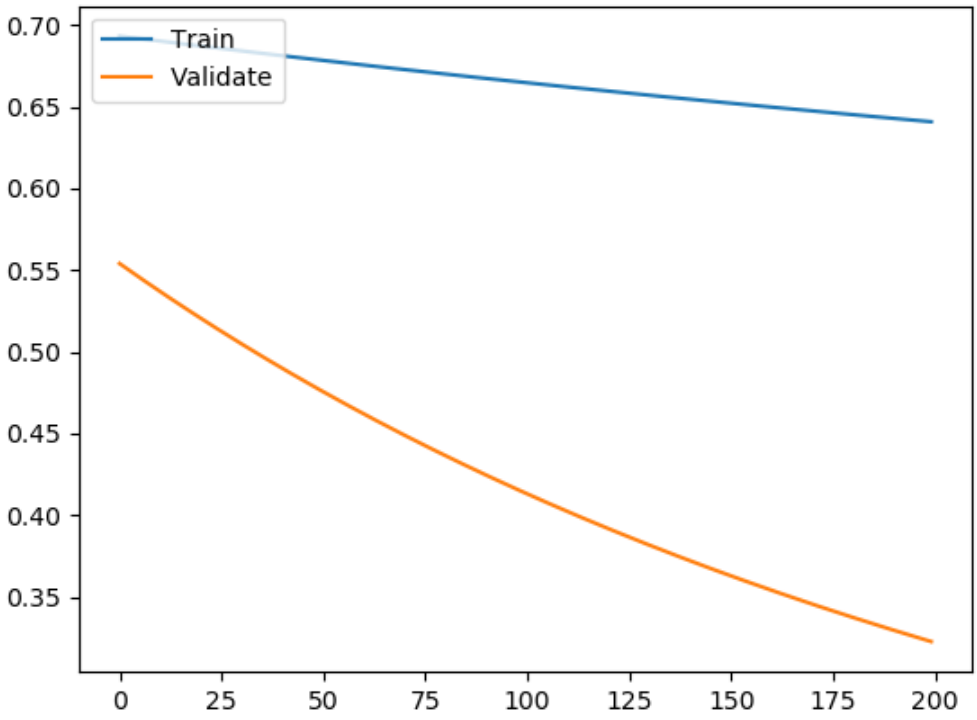
Similarly, for Batched Gradient Descent



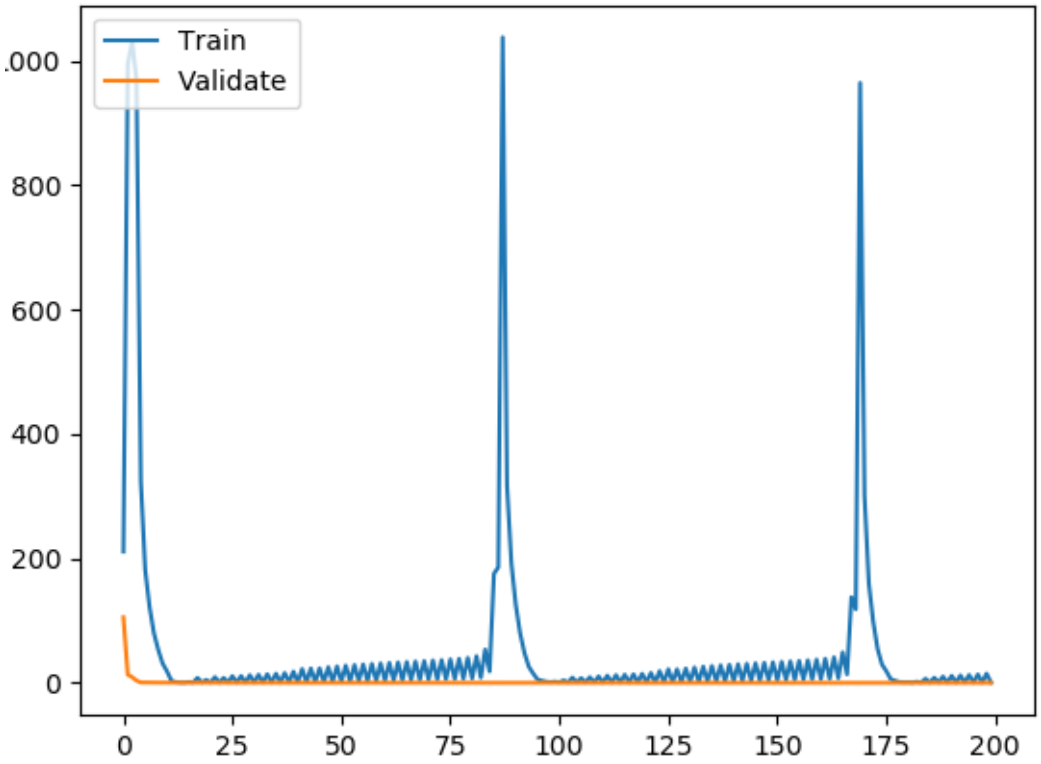
With learning rate changed to 0.01



With learning rate changed to 0.0001



With learning rate changed to 10



In SGD, for all learning rates, minima is reached early at around 10-20 epochs but the function does not stay there and keeps moving around.

In BGD, LR=0.1, convergence is reached around 100. As we lower the LR, number of epochs required, increase and for LR=0.0001, many more epochs are needed for convergence.

For LR=10, we reach the minima very quickly but then because the learning rate is too high, we overshoot it but eventually come back at it again.

BGD Train Accuracy = 0.97 Validation Accuracy = 0.93

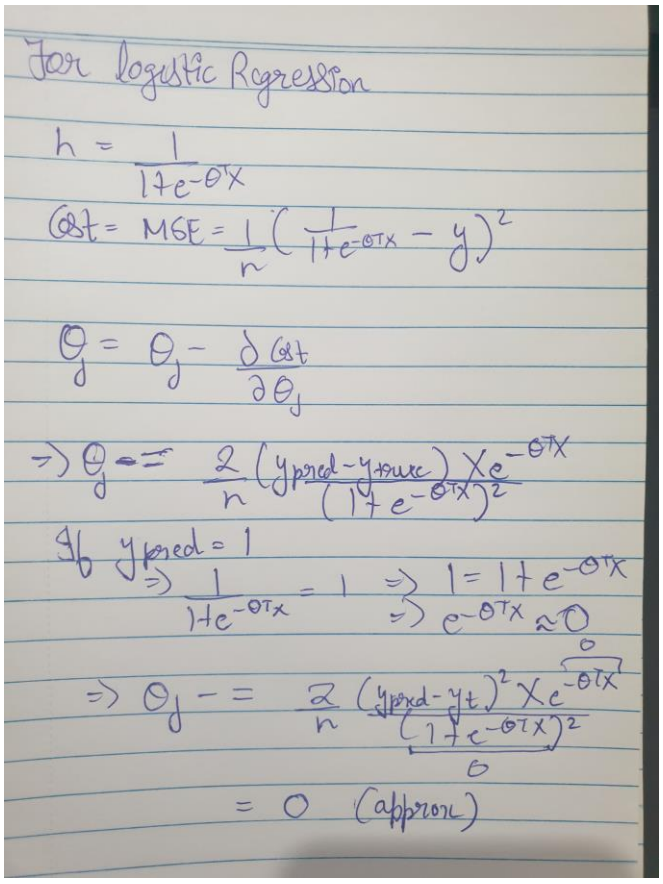
SGD Train Accuracy = 0.82 Validation Accuracy = 0.85

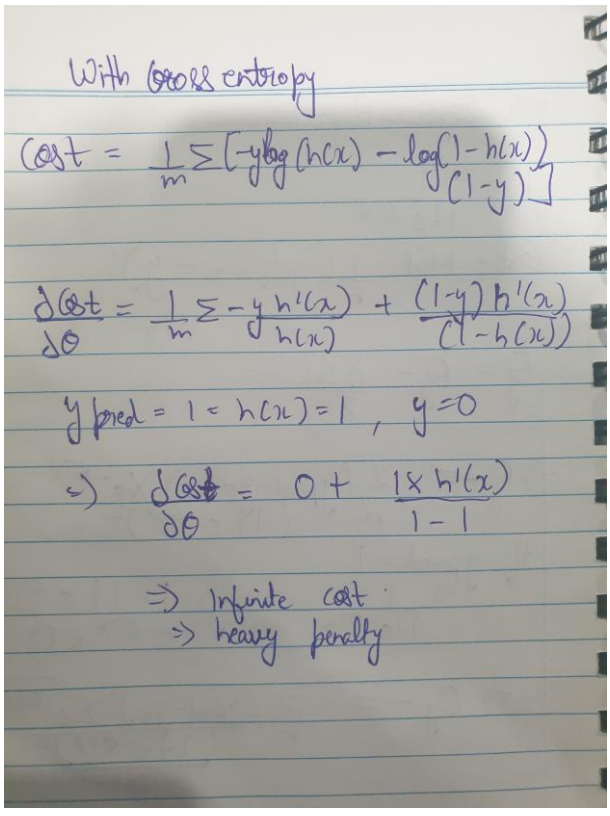
SGD is able to converge faster but the solution it gives is worse than that given by BGD, which gives the optimal solution.

Sklearn Accuracy = 0.99

Sklearn applies regularization which might be one of the reasons for better accuracy

Q3.





$(1+e^{-theta.T.X})$ = 1 and not 0 in above image.

With MSE, the gradient comes out to be 0 and hence no penalty is being applied even though the output labels are different. With no penalty given for wrong prediction, our model wouldn't learn properly and there would be bias in the output.

For cross entropy, the gradient comes out to be very high and hence a high penalty is being imposed which means our model will learn when it predicts the wrong output.

Q4.

$$\log f(x) = \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(y - h(x,\beta))^2}{2\sigma^2} = L$$

To maximise β

$$h(x,\beta) = X.\beta$$

$$\frac{dL}{d\beta} = -2\frac{(y - X.\beta)}{2\sigma^2}\frac{dX\beta}{d\beta}$$

$$0 = -2X'y + 2X'X\beta$$

$$\Rightarrow \beta = (X'X)^{-1}X'y$$

If we add a column of 1s to X

$$X = \begin{matrix} [a^{(x_2)} & b^{(x_1)} & c^{(x_0)}] \\ \beta_0 & \beta_1 & \beta_2 \end{matrix}$$

For this case-

$$\beta = \begin{bmatrix} 0.013026\,12 \\ 0.111\,44569 \\ -0.408\,24507 \end{bmatrix} \begin{matrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{matrix}$$

Fitted Response function-

$$= e^{0.013X_1 + 0.111X_2 - 0.4082}$$

$\exp(\beta_0) = 1.01311$, means that for increase in age by 1, odds ratio of disease recurring ($y=1$) to not recurring ($y=0$), increases by 1.01311 times.

$\exp(\beta_1) = 1.1178$, similarly for increase in spread of disease by 1, increases odds ratio by 1.117

$X_2 = 2 \qquad X_1 = 75$

$$\Rightarrow \frac{e^{0.026 + 8.325 - 0.4082}}{1 + e^{0.026 + 8.325 - 0.4082}}$$

$$\Rightarrow \frac{e^{7.869}}{1 + e^{7.869}} = 0.9996$$

0.9996 probability of recurring of disease

Q5. In least squares, we want to minimize the sum of squares of an affine function, which is a linear function + a constant term which includes the equation we use for regression.

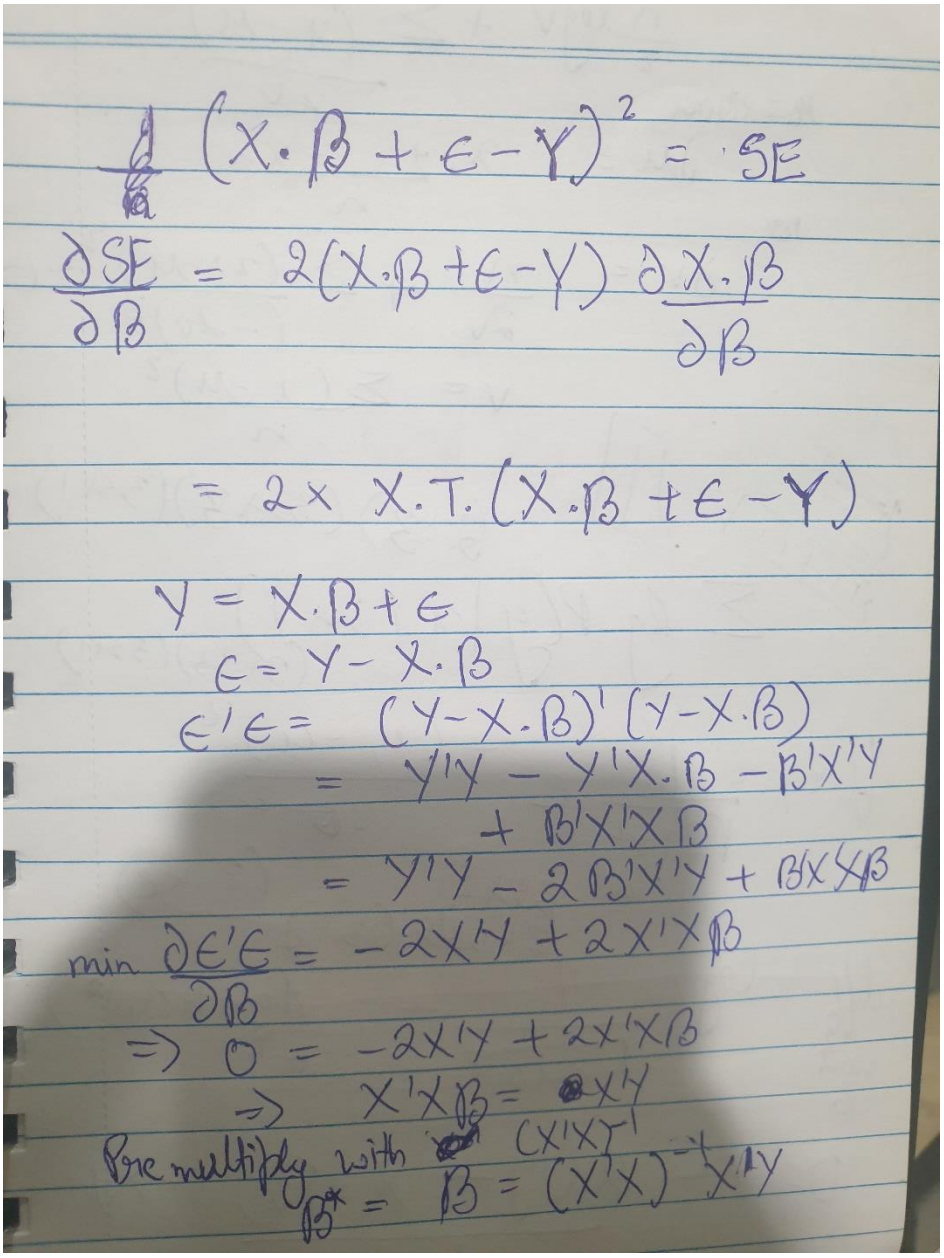
$Y = \phi.T.X + c$

$C = Y - \phi.T.X$

Sum of squares $= \sum c^2 = || Y - \phi.T.X ||^2$

$= c'c = (Y - \phi.T.X)'(Y - \phi.T.X)$

Now if we minimize c'c, we will get the required ϕ



Conditions – This solution will always exist if both X and Y are in a linear combination.