1.

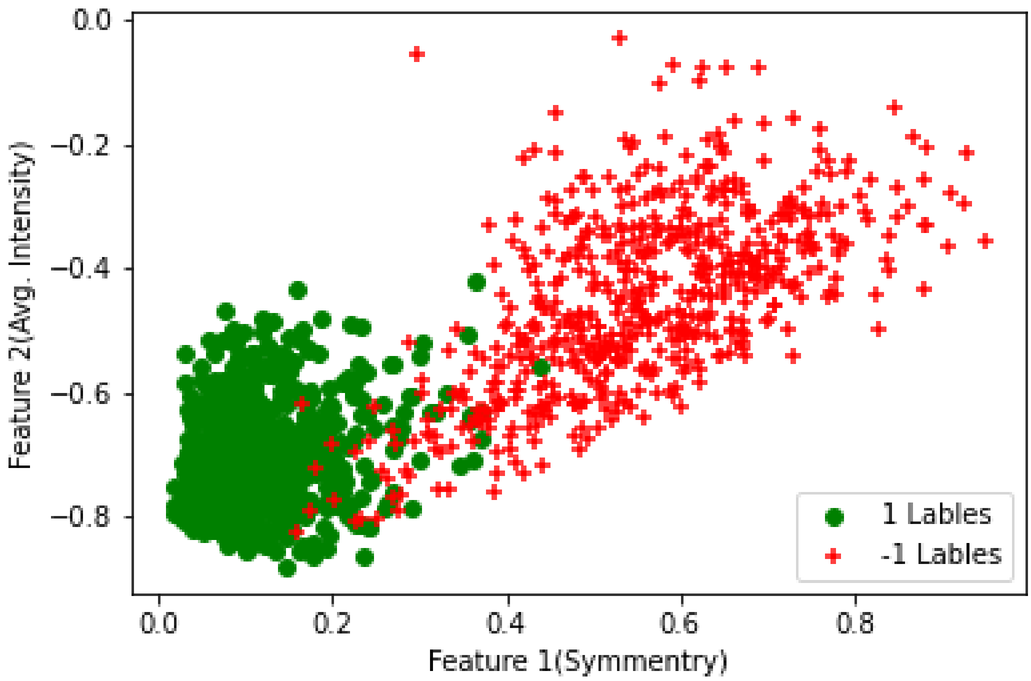
a) Trainvalidsplit splits our training into new training and validation set. We do this to evaluate the performance of the model before actually directly testing on the test set.

b) Yes, we would be able to learn the new features that might be dominant and present in our validation set. Which increases the overall performance of the model.

c) Code.

d) To train and learn the bias weights. This bias would allow us to shift the target learning function to left or right to better fit the data.

f) Scatter Plot of 2 features of the train data.



---

2)

a) (x, y) $y \in (-1, 1)$

$$E(\omega) = \ln\left(1 + e^{-y\omega^T x}\right)$$

b) $\nabla E(w)$

$$= \frac{1}{1+e^{-y w^T x}} \cdot \left(e^{-y w^T x}\right) \cdot \frac{\partial}{\partial w}\left(-y w^T x\right)$$

$$\boxed{\frac{\partial}{\partial w}\left(w^T b\right) = b}$$

$$\frac{\partial}{\partial x}\left(e^{x}\right) = e^{x} \cdot \frac{\partial(x)}{\partial x} \qquad \frac{\partial}{\partial x}\left(\ln(x)\right) = \frac{1}{x} \cdot \frac{\partial(x)}{\partial x} = \frac{1}{x}$$

$$= \frac{-y x \left(e^{-y w^T x}\right)}{1+e^{-y w^T x}} = \frac{-y x}{1+e^{y w^T x}}$$

c)

Since sigmoid is monotonically increasing function. For each of $w^T x$ there will a unique sigmoid value in the same order and hence with the threshold set for $\theta(w^T x)$ the $w^T x$ would be linearly separable.

Some times $w^T x$ values might explode and become too large or diminish and become zero which might cause issues for converging hence we apply sigmoid to keep them within boundary [0,1] which also helps in reading direct probabilities.

d)

Yes, decision boundary would be still linear. But we will see a shifted boundary because of change in threshold values. Reason is same as 2 c) we will have unique values of out with unique inputs with order preserved.

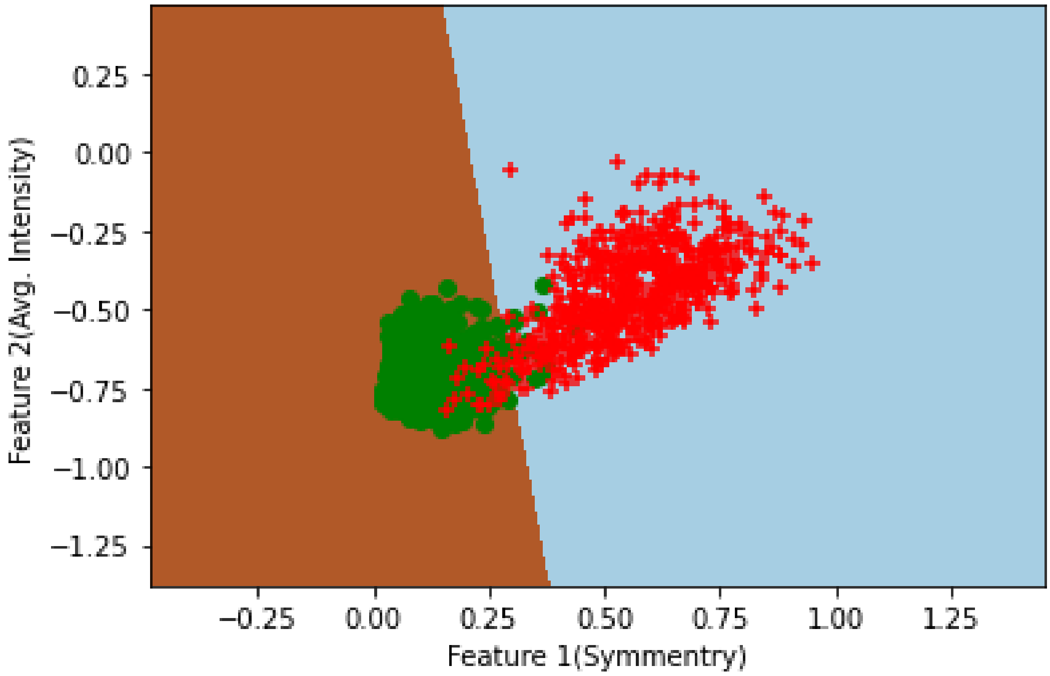
e)

Sigmoid function being monotonically increasing.

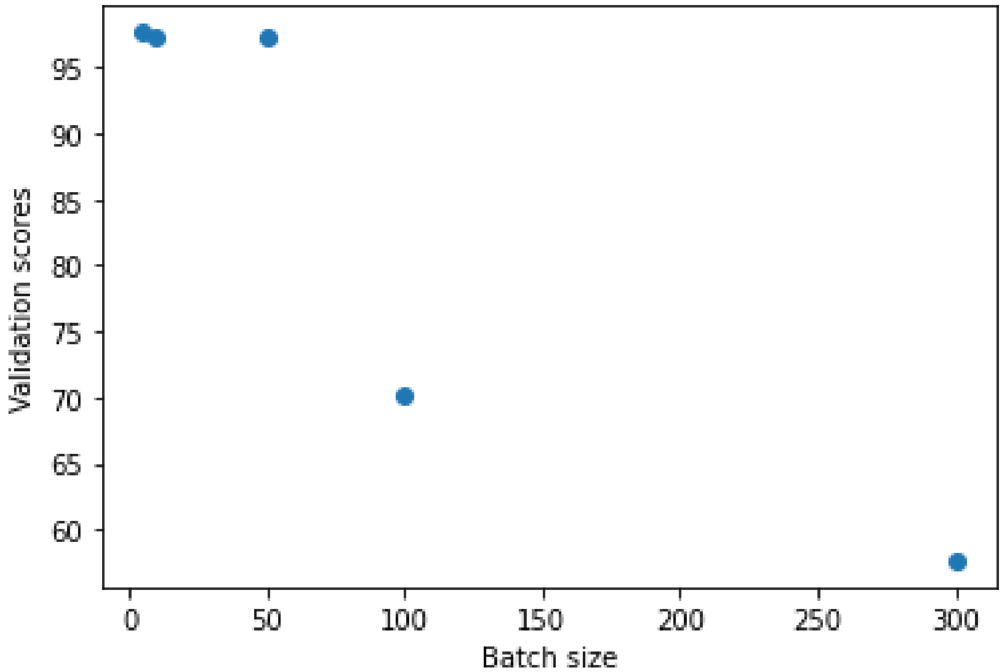
3)

a) b) c) coding

d) **Best sigmoid model Decision boundary visualization**

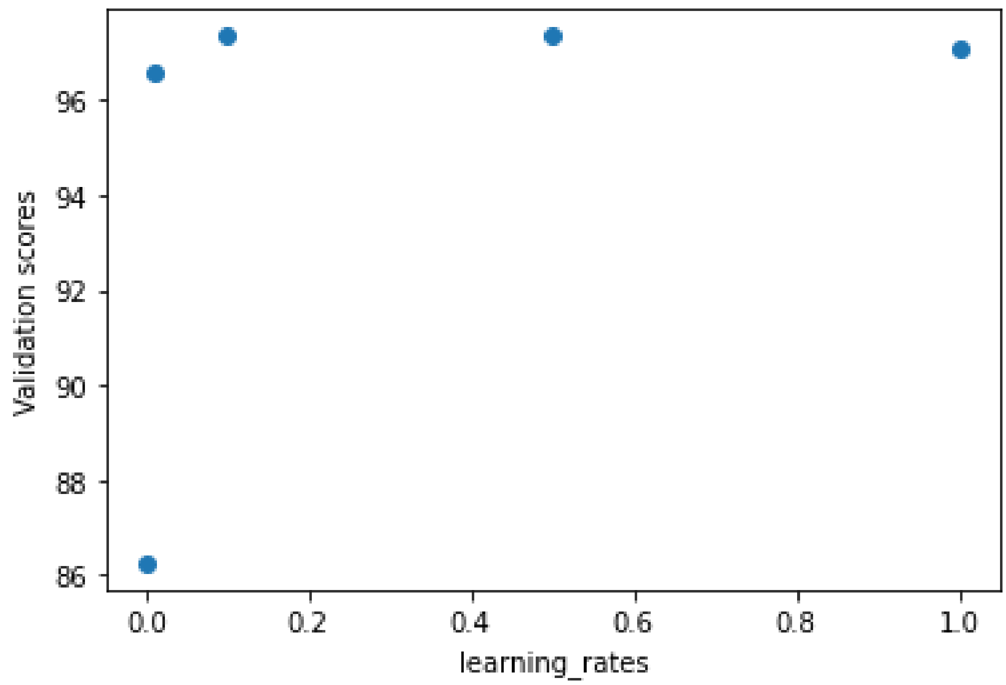


Best hyper parameter searching



Best batch size 5

- 5 batch size has slightly high validation set accuracy score than 10



Best LR 0.5

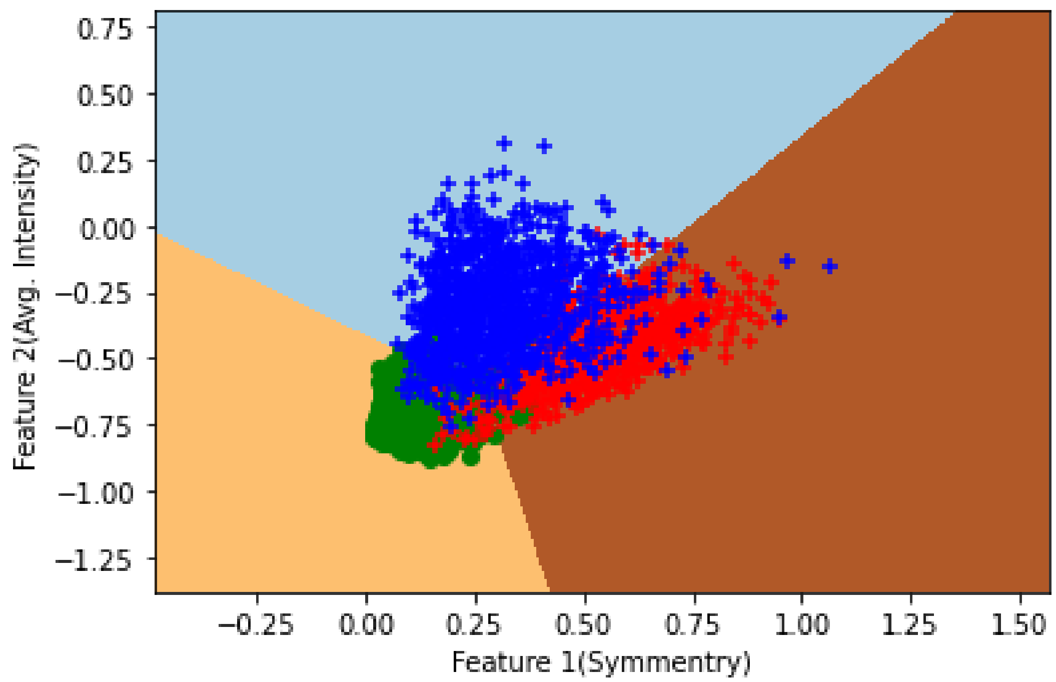
e) Test accuracy score on 5 Batch size and 0.5LR sigmoid model = **93.5064935064935**

---

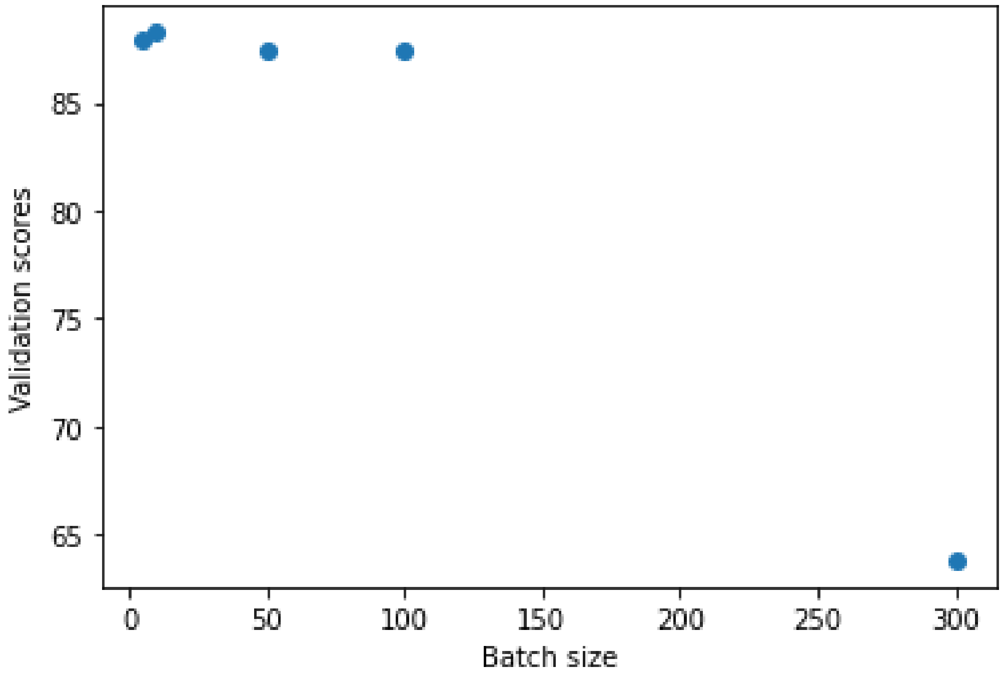
4)

a) b) c) coding part

d) **Best SoftMax model Decision boundary visualization**

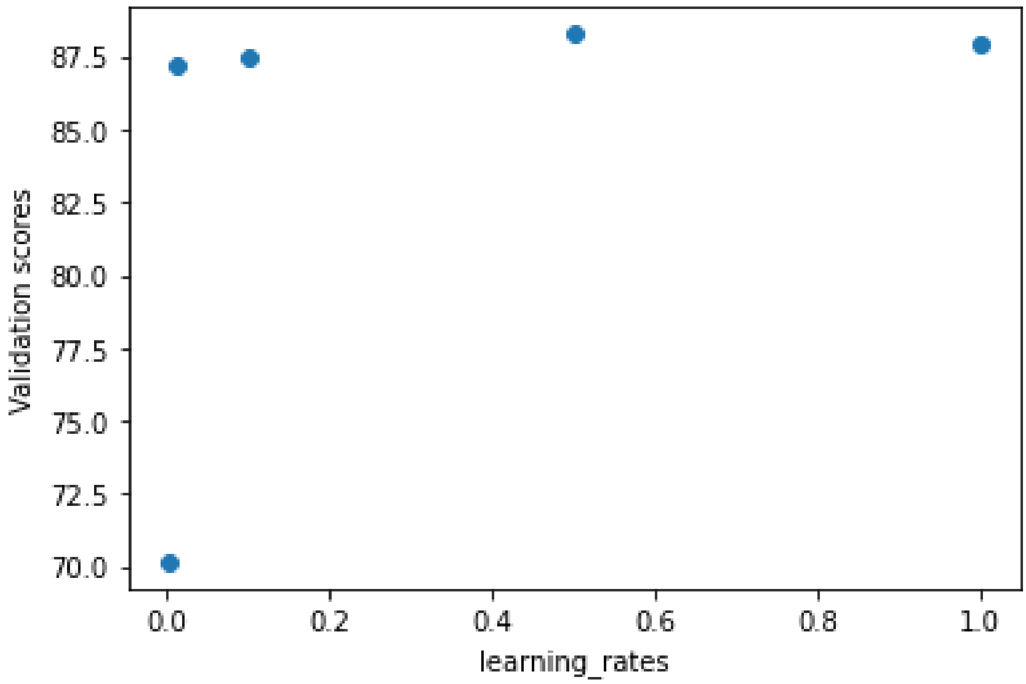


Best hyper parameter searching



Best batch size 10

- 10 batch size has high validation set accuracy score.



Best LR 0.5

e) Test score on 10 Batch size and 0.5LR SoftMax model **86.35809987819732**

5

a)

2 Class

	Accuracy on validation set for 2 Class	Final weights
Sigmoid LR	97.35449735449735	[8.23307634 -27.76268238 0.48470943]
SoftMax LR	97.35449735449735	[[-4.79291425 4.79291425] [14.86824347 -14.86824347] [-0.89698737 0.89698737]]

For 2 class problem the final Sigmoid LR = SoftMax LR model. Final weights are the same so the prediction for each and every sample would be same.

b)

```
logisticR_classifier = logistic_regression(learning_rate=0.5, max_iter=100)
logistic_regression_multiclass(learning_rate=0.5, max_iter=1000,  k= 2)
```

The weight vectors are as follows

And the final weights are

	1^{st} batch 1^{st} Epoch
Sigmoid LR	[0.02148619 -4.35539402 -2.33857023]
SoftMax LR	[[0.00369303 -0.00369303] [3.23010029 -3.23010029] [1.62306612 -1.62306612]]

W seems to be approximately same the values for W1 and W2 for the same Learning rate of 0.5

W = W1 – W2 doesn't hold here for the same LR

To set W = W1-W2 for each iteration the W needs to be doubled or W1 / W2 needs to be halved for the we can decrease the learning late of SoftMax by half

```
logisticR_classifier_multiclass = logistic_regression_multiclass(learning_rate
=0.25, max_iter=1000,  k= 2)
```

So the new Learning rates are 0.25 for SoftMax and 0.5 for Sigmoid

	1st batch 1st Epoch (Decreased LR)
Sigmoid LR	[0.02148619 -4.35539402 -2.33857023]
SoftMax LR	[[-0.0107431 0.0107431] [2.17769701 -2.17769701] [1.16928512 -1.16928512]]

Now W = W1- W2 condition satisfies for each epoch.

[0.02148619 -4.35539402 -2.33857023] = [0.0107431 -2.17769701 -1.16928512] - [-0.0107431 2.17769701 1.16928512]