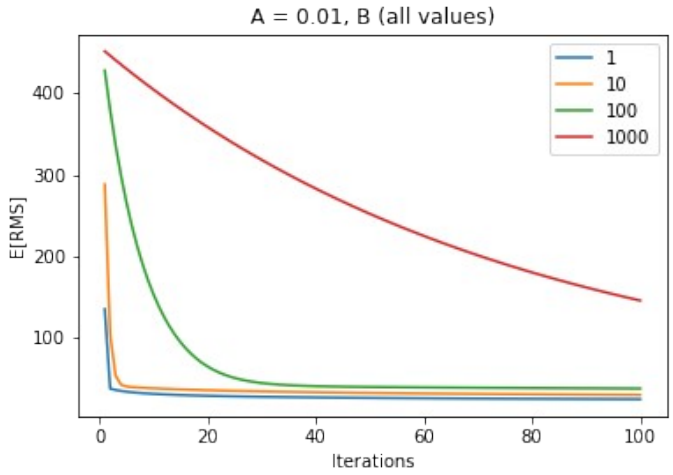
HOMEWORK 5

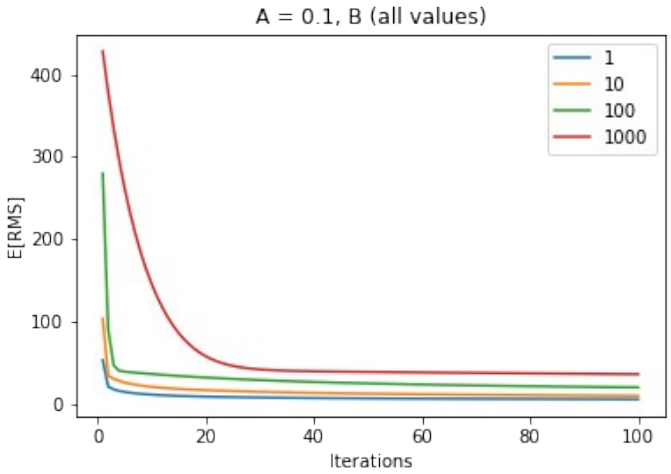
1.

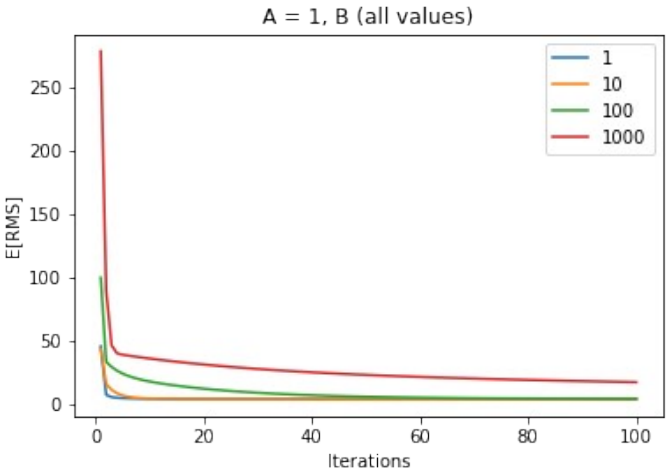
(a) **No**, the convergence criterion on the learning rate for MSE classification cannot be used for MSE regression. We should make sure that the learning rate (eta) is greater than 0 and that the sum of the learning rates for all the epochs should tend towards 0, as the number of epochs increase. Eta decides the step that has to be taken during gradient descent.The reason is as follows : Large values of eta can result in unstable learning (during the gradient descent) and the objective of finding the global minima might be lost. However, with a low value of eta (>0), convergence is slower and finding a global minima is achieved gradually, but certainly. A large learning rate can cause the model to converge too quickly to a suboptimal solution. And so, a smaller learning rate, with a learning rate schedular that changes the learning rate during training is often preffered even though there's always an issue with epochs (as we see later in the problem).

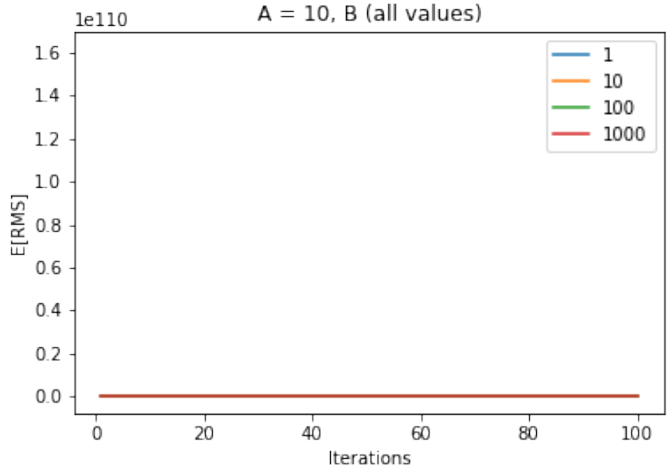
In the classification case, we would just obtain the value of a cost function and so, the weights, and check if the discriminant function's value is greater or lesser than 0, and based on that assign classes. The value of the learning rate would matter but not to a large extent. However, in a regression problem, our objective is to find the most optimal weights that would help in obtaining numerical predictions and so the learning rate plays a very crucial rule in obtaining the global minima.

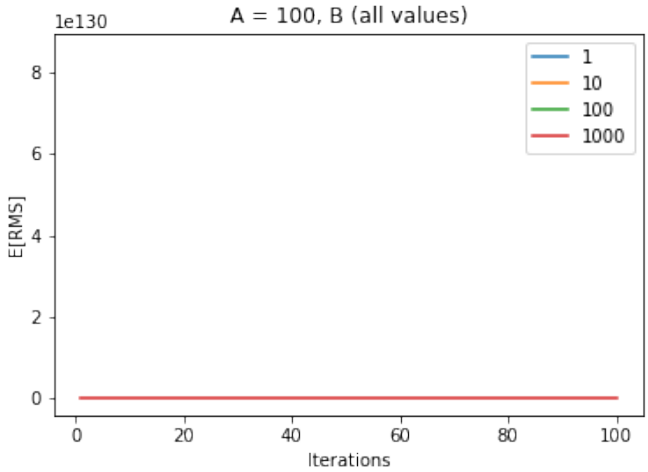
(b) The linear MSE regressor uses an iterative gradient descent for optimisation of the cost function. Following are the learning plots ($E_{RMS}^{(m)}$ *vs* m), where m is the epoch number.











(c) The learning curves are very much self-explanatory. We observe that for A between 0.01 to 1, the learning curves begin to smooth out even more. For A = 0.01, we observe a very sharp learning curve when B = 1000, which translates to the fact that there's not much learning. For smaller values of B, the curve is smooth and convergence is reached as the iterations begin to increase. However, we notice that when A = 1, the $E_{RMS}^{(m)}$ values are even lower in the beginning and in particular, when B = 10, the $E_{RMS}^{(m)}$ starts out with

the smallest value and reaches convergence. We expect this pair of A and B to perform the best on the test data. When A = 10 and 100, the value of the cost function J_w is extremely high because of which the weight updates and $E_{RMS}^{(m)}$ are extremely high (and hence, not easy to plot). Our intuition is that, as long as the ratio of A/B is close to 0.1 or 0.01, the model is going to perform well on the unseen test data. This is due to the fact that, finding an optimal weight vector is achieved gradually as gradient descent progresses.

(d) From the execution of the code and the explanation presented above, **A = 1, and B = 10**, as a pair, give the lowest E_{RMS} on the test data i.e **4.6516**. As mentioned earlier, while training, $E_{RMS}^{(m)}$ was the lowest in the beginning, when A was equal to 1, and B was 10 (evident from the learning curve attached above).

(e) The trivial regressor outputs the mean of the training labels. And so, the $E_{RMS}^{(m)}$ of this regressor on the test data was **18.867**.

Yes, the error from (d) is definitely lower than that of this trivial regressor. This is due to the fact that the trivial regressor involves no training and so finding an optimal weight, that corresponds to a global minima of the cost function is out of the picture.

2. We're given,

$$J(\underline{w}) = \| \underline{\underline{X}}^{(t)} \underline{w}^{(t)} - \underline{y} \|_2^2 + \lambda \| \underline{w}^{(0)} \|_2^2 \quad - (1)$$

where $w^{(0)} \rightarrow$ bias term.

We perform this in non-augmented space.

~~Taking the gradient of (1)~~:

$$J(\underline{w}) = (\underline{\underline{X}}^{(t)} \underline{w}^{(t)} - \underline{y})^T (\underline{\underline{X}}^{(t)} \underline{w}^{(t)} - \underline{y}) + \lambda \underline{w}^{(0)T} \underline{w}^{(0)}$$

$$= \underline{w}^{(t)T} \underline{\underline{X}}^{tT} \underline{\underline{X}}^{t} \underline{w}^{(t)} - \underline{y}^T \underline{\underline{X}}^{(t)} \underline{w}^{(t)} - \underline{y} (\underline{\underline{X}}^{t} \underline{w}^{t})^T + \underline{y}^T \underline{y} + \lambda \underline{w}^{(0)T} \underline{w}^{(0)}$$

$$= \underline{w}^{(t)} \underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} \underline{w}^{(t)} - 2 \underline{y}^T \underline{\underline{X}}^{(t)} \underline{w}^{(t)} + \underline{y}^T \underline{y} + \lambda \underline{w}^{(0)T} \underline{w}^{(0)}.$$

$$\nabla_{\underline{w}} J(\underline{w}) = 2 \underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} \underline{w}^{(t)} - 2 \underline{y}^T \underline{\underline{X}}^{(t)} + 2\lambda \underline{w}^{(0)}.$$

Now, $I' = \text{diag}\{0, 1, 1, \ldots, 1\}$. and so, $w^{(0)} = I' \underline{w}$.

Optimal $\underline{w}$ can be found by equating $\nabla_{\underline{w}} J(\underline{w}) = 0$

$$\therefore \quad 2 \underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} \underline{w}^{(t)} - 2 \underline{y}^T \underline{\underline{X}}^{(t)} + 2\lambda I' \underline{w}^{(t)} = 0$$

By $w^{(t)} \rightarrow \hat{w}^{(t)}$

$$\therefore \quad \underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} \hat{\underline{w}}^{(t)} + \lambda I' \hat{\underline{w}}^{(t)} = \underline{y}^T \underline{\underline{X}}^{(t)}$$

or, $$\hat{\underline{w}}^{(t)} = \frac{\underline{y}^T \underline{\underline{X}}^{(t)}}{(\underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} + \lambda I')} = (\underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} + \lambda I')^{-1} \underline{y}^T \underline{\underline{X}}^{(t)}$$

if the inverse exists.

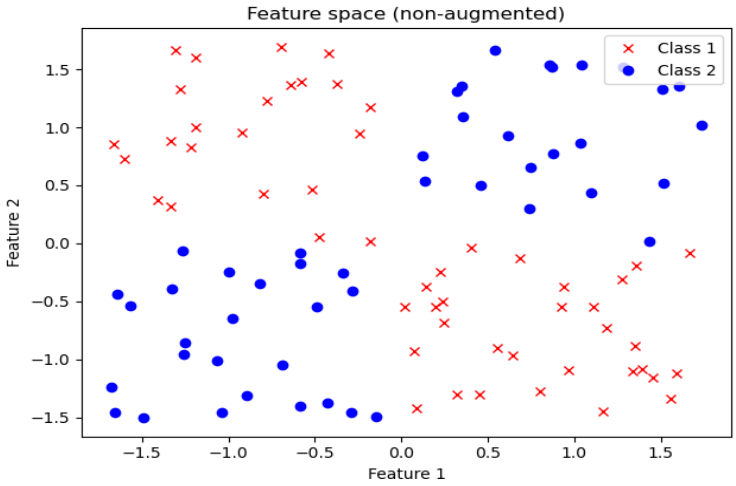
$$\therefore \quad \hat{\underline{w}} = (\underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} + \lambda I')^{-1} \underline{y}^T \underline{\underline{X}}^{(t)} \quad \text{or} \quad (\underline{\underline{X}}^{(t)T} \underline{\underline{X}}^{(t)} + \lambda I')^{-1} \underline{\underline{X}}^{(t)T} \underline{y}$$

if the inverse exists. //.

where $I' = \text{diag}\{0, 1, 1, \ldots\}$

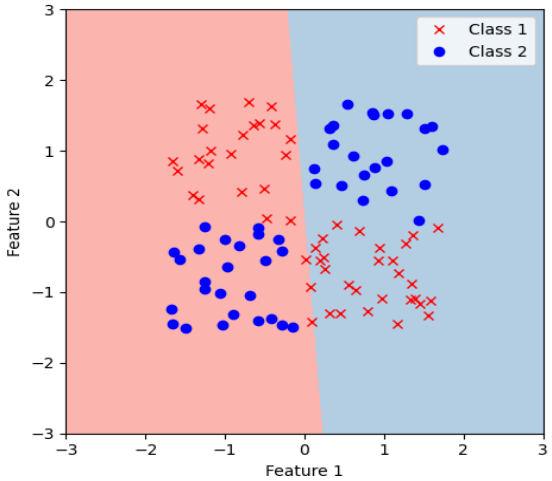
3. The following problem involves implementation of a nonlinear mapping from a 2D feature space to an expanded feature space.

(a) **No**, this data set is clearly not linearly seperable in this non-augmented feature space.



(b) By using sklearn.linear_model's perceptron on this dataset, the classification accuracy is **0.53**.

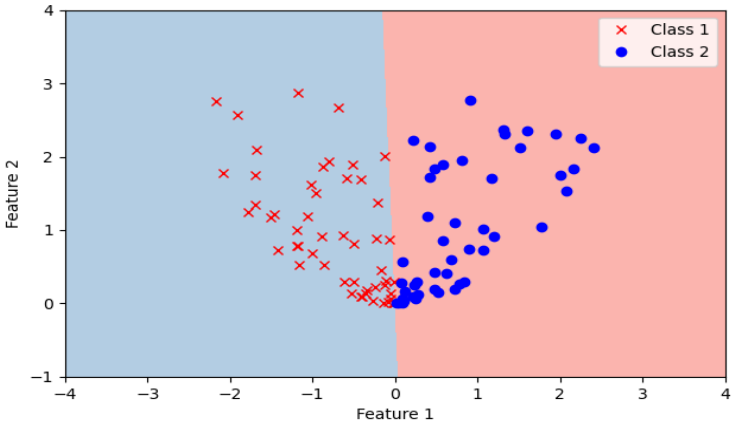
(c) The learned decision boundary of the perceptron on this data set is illustrated below:



(d) The perceptron model (from sklearn), gives a classification accuracy of **1.0** on the expanded feature space. **No**, the data is not linearly seperable in the expanded feature. We obtain five features in the expanded space and so achieving linear seperability is not possible unless we choose the most relevant features or perform PCA or t-SNE.

(e)

(i) The weights of the expanded feature space is as follows: **[-0.10562188, -0.14390327, 7.616127, 0.05743614, -0.28634851]**.
(ii) In absolute value, $\mathbf{x_1x_2}$ **and** $\mathbf{x_2^2}$ received the highest weights.
(iv) **Yes, the dataset is linearly seperable** in this feature space i.e by considering x_1x_2 and x_2^2 as the most relevant features.



(f) In the original feature space, the non-linear decision boundary has been plotted and is shown below.