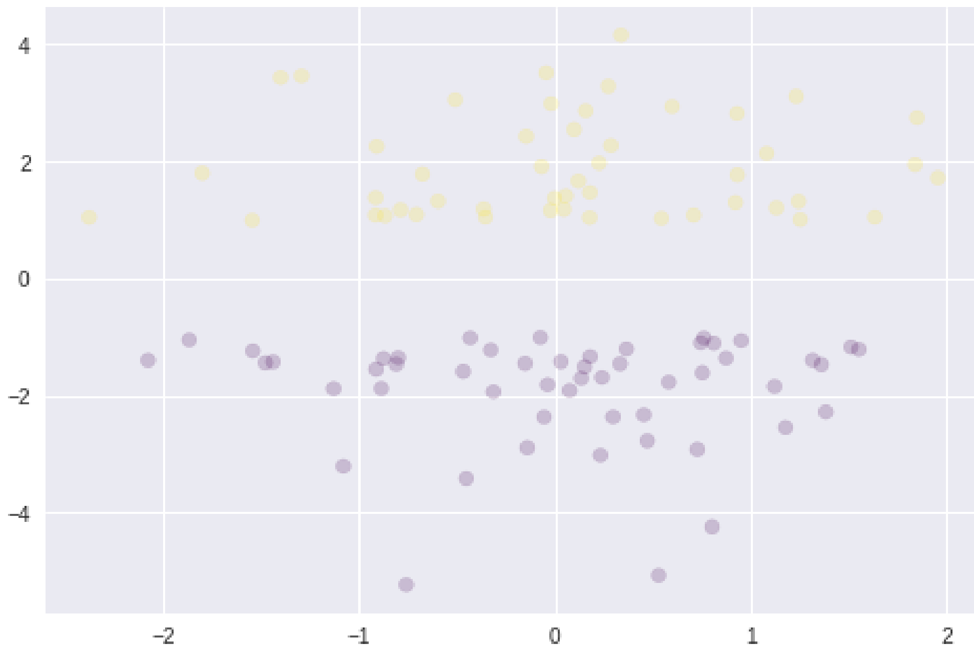
Machine Learning Assignment: Perceptron Learning Algorithm

1. **Show that there is a perceptron that correctly classifies this data. Is this perceptron unique? What is the 'best' perceptron for this data set, theoretically?**



We can see that when the data is plotted for two features, the data is **linearly separable.**

If the data is linearly separable, then definitely there will exist a perceptron that separates the data.

The best perceptron will have the maximum margin. Theoretically since the perceptron's performance is determined by only one variable, the weights of all the others apart from the kth feature will be 0. So, the best perceptron will have all the weights set to zero apart from the kth feature that will have the weight as 1.

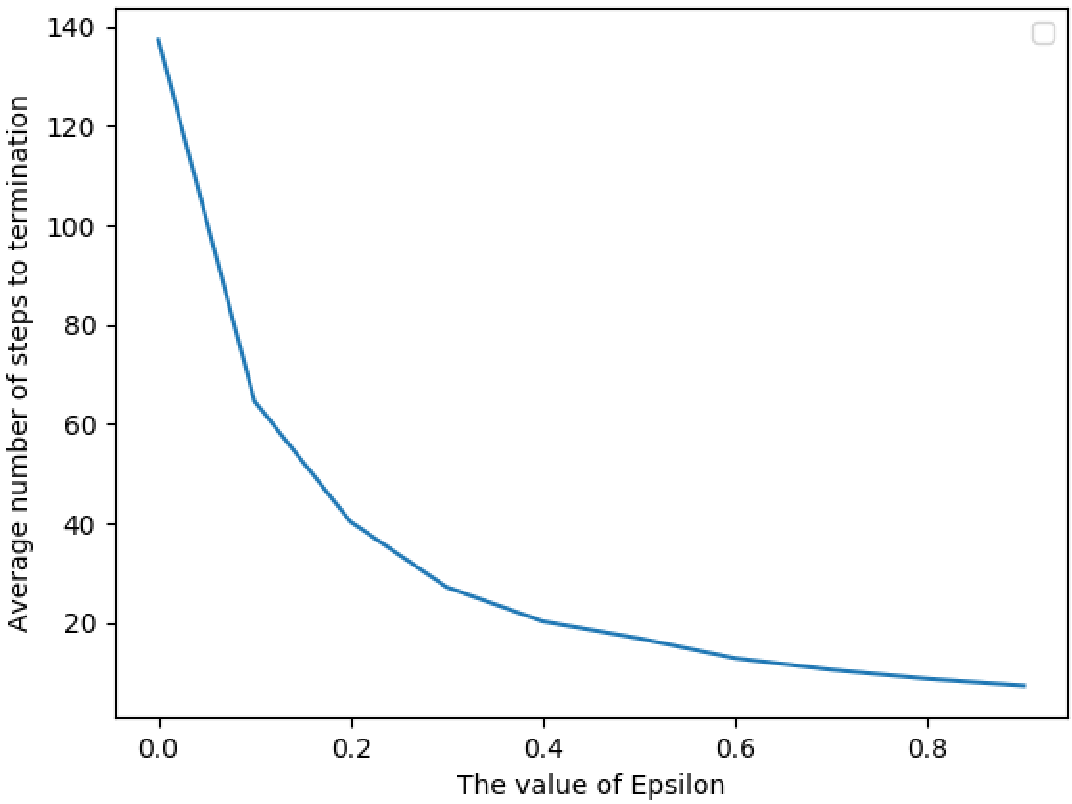
2. We want to consider the problem of learning perceptron from data sets. Generate a set of data of size m= 100 with k= 20 , $\epsilon = 1$. Implement the perceptron learning algorithm. This data is separable, so the algorithm will terminate. How does the output perceptron compare to your theoretical answer in the previous problem?

The weights are as follows:

[0. -0.49544769 -0.02460004 -0.20363605 -0.5436263 0.21761494 -0.52717717 0.71038771 -0.14244272 2.64007698 4.07127086 0.9816189 -1.51826395 1.45534003 0.14938139 -0.52056147 0.36126678 0.13889544 0.73798523 -0.48805224 8.67225233]

We can see that the weight for the last feature is extremely high whereas weights for all the other features are approximately equal to zero or negative indicating that those models are not given importance. This matches our hypothesis that the weight of the last element will be highest. The first term represents bias and all the others represents weights.

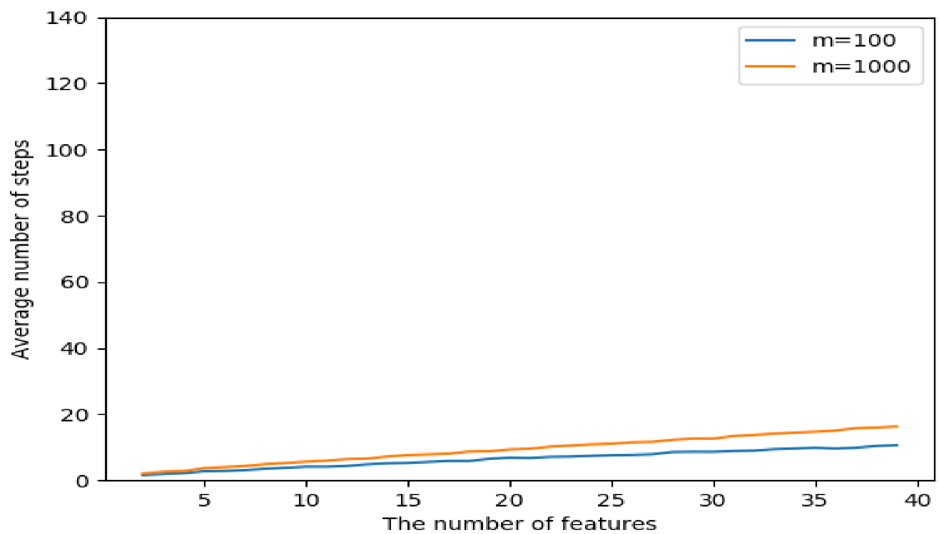
3.) For any given data set, there may be multiple separators with multiple margins - but for our data set, we can effectively control the size of the margin with the parameter epsilon- the bigger this value, the bigger the margin of our separator. For m= 100,k= 20, generate a data set for a given value of and run the learning algorithm to completion. Plot, as a function of epsilon belonging to the interval [0,1], the average or typical number of steps the algorithm needs to terminate. Characterize the dependence



The above graph has been plotted for the various values of epsilon keeping the value of m and k constant. (m=100 and k= 20).

We can see that as the value of epsilon increases, the average number of steps required to convergence decreases. With the increase in epsilon the margin increases and the average number of steps to convergence decreases. The best linear separators have high margins.

4.) One of the nice properties of the perceptron learning algorithm (and perceptron generally) is that learning the weight vector w and bias value b is typically independent of the ambient dimension. To see this, consider the following experiment:–Fixing m= 100, epsilon = 1, consider generating a data set on k features and running the learning algorithm on it. Plot, as a function k (fork= 2,...,40), the typical number of steps to learn a perceptron on a dataset of this size. How does the number of steps vary with k? Repeat form= 1000.



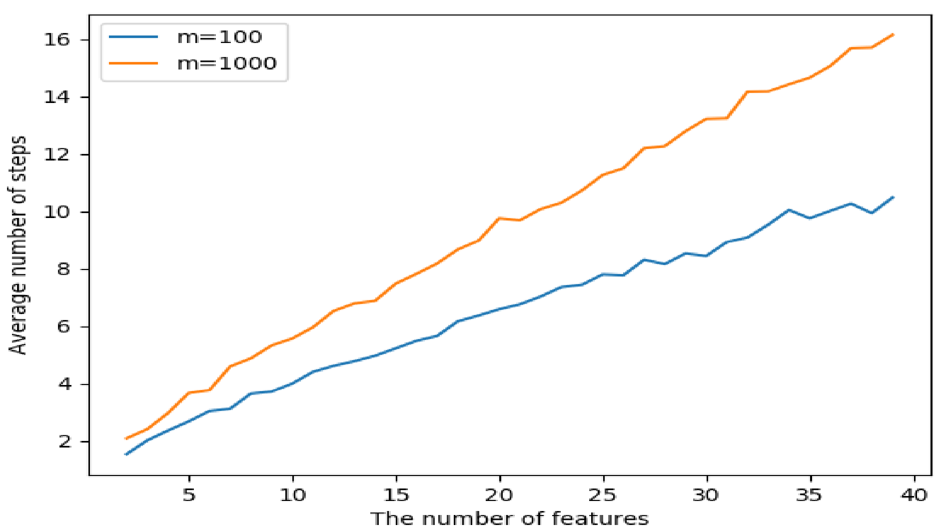
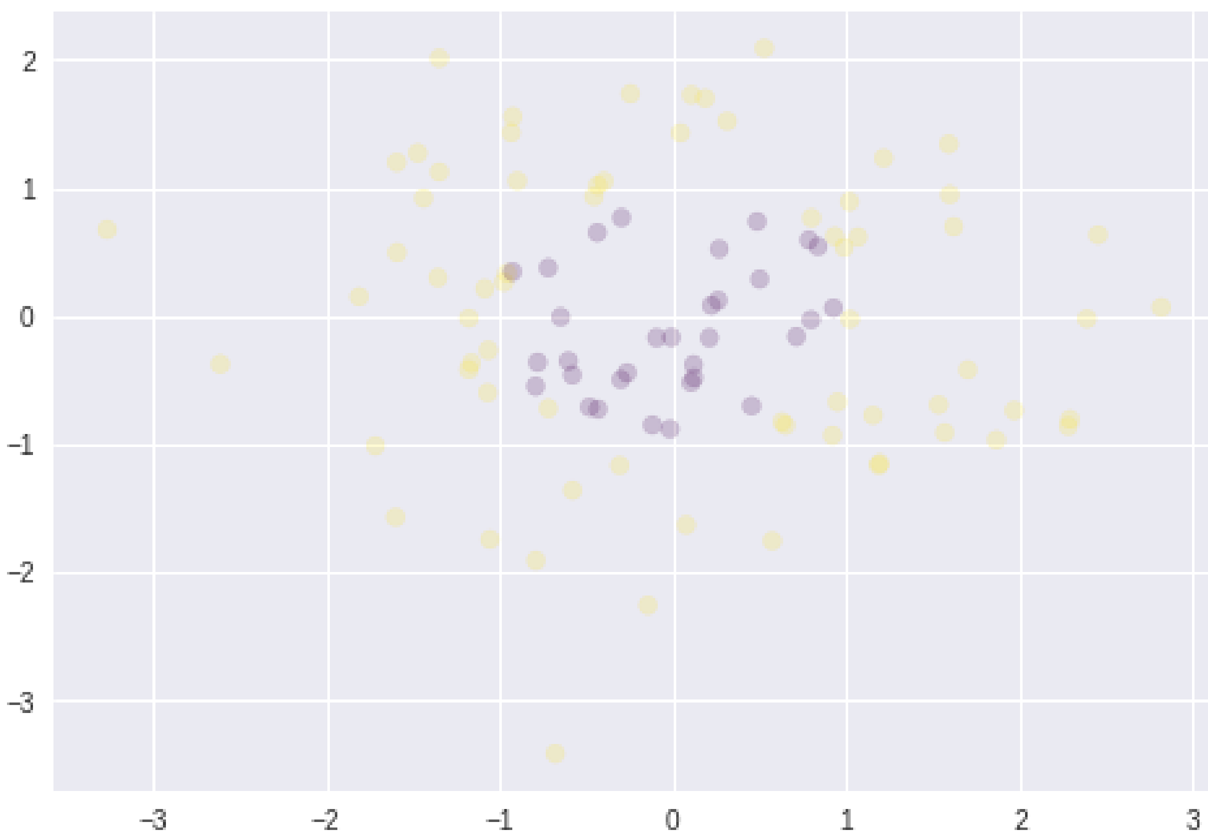


Figure 1: The y axis scaled from 0 to 140

Figure 2: Unscaled y-axis

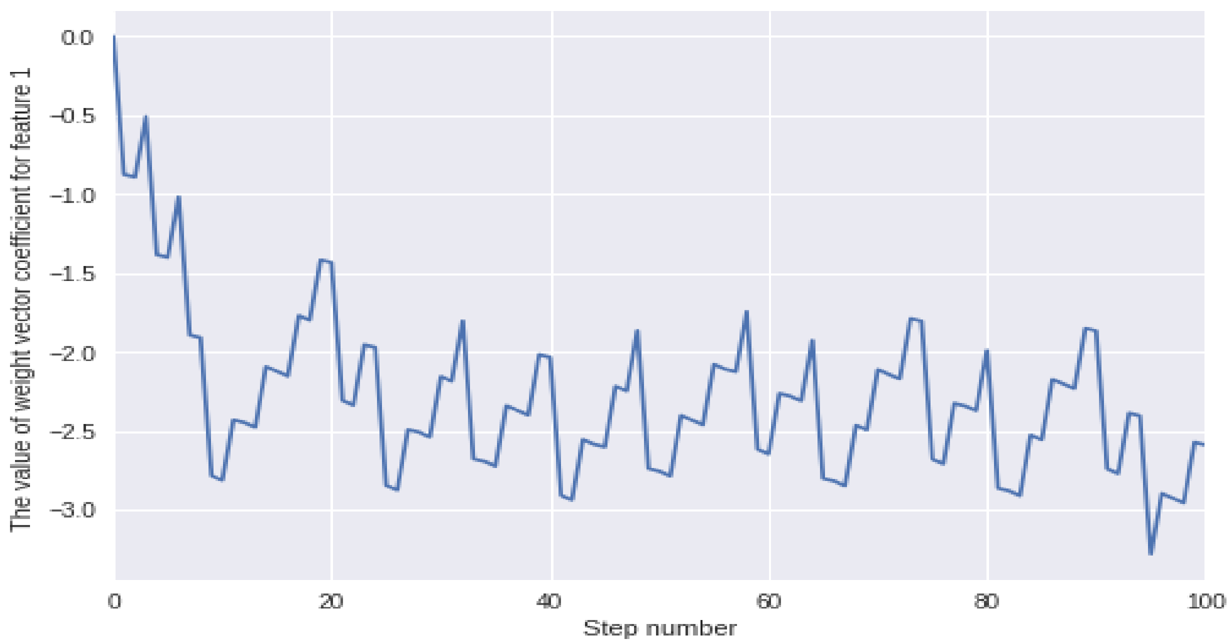
The figure shows that the convergence of an algorithm or the average steps to termination does not depend on the value of ambient dimensions. We can see that for m=100 and m=1000 there is no significant difference in the trend or the number of steps required to termination. This is because we consider feature space rather than raw data space.

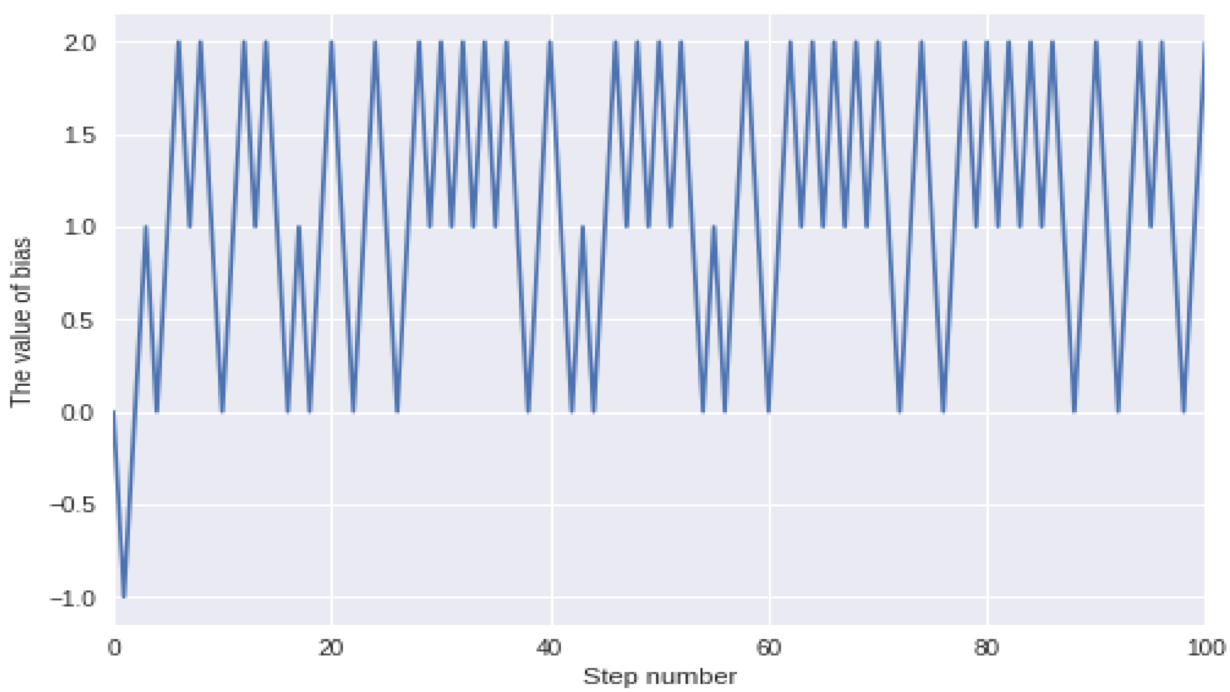
5.) For data defined in this way, there is no universally applicable linear separator. For k= 2,m= 100, generate a data set that is not linearly separable. (How can you verify this?) Then run the perceptron learning algorithm. What does the progression of weight vectors and bias values look like overtime? If there is no separator, this will never terminate - is there any condition or heuristic you could use to determine whether or not to terminate the algorithm and declare no separator found?

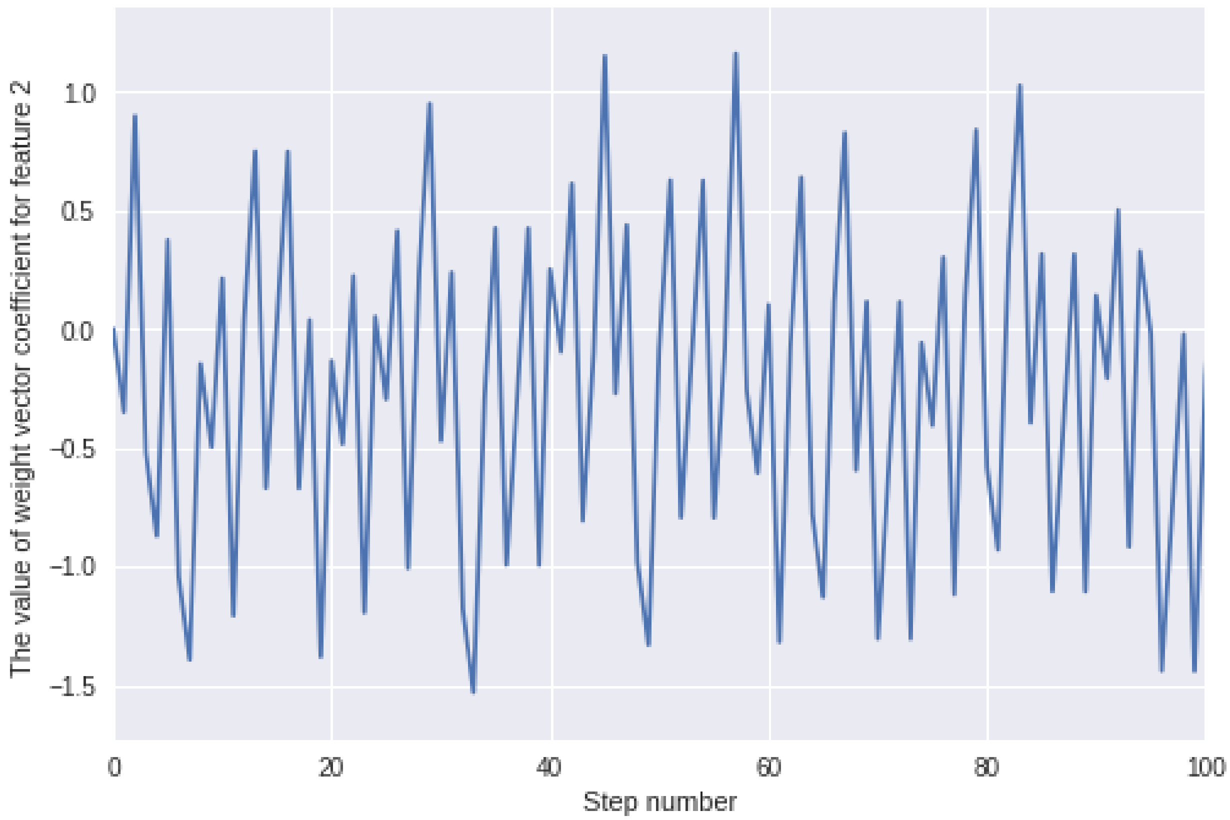


We can see that the data is not linearly separable. So, the perceptron will never terminate.

Variation of bias and coefficients with respect to number of iterations:







Metric for termination/ declaration that no separator found:

We can calculate the number of steps a linearly separable data would need to terminate. If the number of steps exceeds that in case of linearly inseparable data, then we need to stop and declare termination.

We can also keep track of the errors over time and see if the error on training data do not reduce beyond a point after some time. Even in such cases, we can declare the algorithm to stop and terminate.