# Machine Learning from Data: Homework #1

Due on September 6, 2021

*Prof. Malik Magdon-Ismail*

**Jared Gridley**

# Problem 1

Exercise 1.3:

a) Show that $y(t)\omega^T(t)x(t) < 0$:

Tinkering:
How to rearrange $\omega(t+1) = y(t)x(t) + \omega(t)$?
Also know that miss-classified points are represented by $y(t) \neq sign(\omega^T(t)x(t)$
In other words, $y(t) \neq \omega^T(t)x(t)$

So from this relationship (and its corresponding sign function), we know that if the point is misclassified, $y(t)$ and $\omega^T(t)x(t)$ will be of different signs.
Therefore the expression $y(t)\omega^T(t)x(t)$ will always be less than 0 for a misclassified point since if $y(t)$ is negative then $\omega^T(t)x(t)$ must be positive (and vice-versa), otherwise the point is not misclassified.

b)Show that $y(t)\omega^T(t+1)x(t) > y(t)\omega^T(t)x(t)$
From part a, we know that the right side of the equation is always going to be negative (for a misclassified point) since $y(t)$ and $\omega^T(t)x(t)$ are of different signs.
For the right side of the equation, we know that from the update rule $\omega^T(t+1)$ will move the hyperplane so that the point $(x(t), y(t))$ is classified correctly, therefore $\omega^T(t+1)x(t)$ and $y(t)$ will be of the same sign and will therefore always be positive.
So the equation above can be reduced to $+1 > -1$ which we know will ways be true.

c)
If the data is linearly seperable, then the algorithm is guaranteed to arrive at a correct solution (though not necessarily the optimal solution)
This is because the weight vector is only allowed to update within a bounded amount in the direction of the negative dot product with the training example. In other words, the move from $\omega(t) \Longrightarrow \omega(t+1)$ is making progress towards a correct hyperplane, with the number of changes being made to $\omega$ shrinking.
Since the signs don't match between the expected and actual classificaiton of the point then it will always move in the opposite direction (towards the right direction), so if its too positive then it will always move negative.

# Problem 2

Exercise 1.5: Determine which examples are more suited to the learning approach and which are suited towards the design approach.

a) Design - specifications are determined beforehand by a doctor and manufacturer and the problem is clearly specified.

b)Design - there are strict rules that determine if a number is prime or not (although the computations can be expensive). The problem specifications are very specific.

c) Learning - no set of specifications for spam mail, need data and labels to differentiate the spam from non-spam.

d) Design - rigid rules of physics that can calculate given input specifications.

e) Learning - There are no rules that control traffic patterns, the function is unknown and needs data to determine patterns.

# Problem 3

Exercise 1.6) Identify what type/types of learning are involved and what the training looks like for each.

a) Unsupervised or Supervised
Supervised: < features of each movie, rating >, an algorithm would predict what movie the user is going to rate highest next based on the given features and previous ratings and recommend that to the user.
Unsupervised: < features of each movie >, This would a type of nearest neighbors algorithm in which when a movie is rated high, it looks at the cluster of other movies near it and recommends one that it closely related to it.

b) Reinforcement: < gameplay, output, rating > the algorithm would take the entire of game, the result of who won, and then a rating of how well it did.

c) Unsupervised: <movie features> this would cluster the movies based on similar features to organize them into different categories.

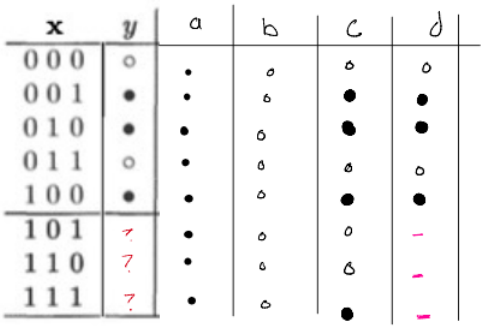
d) Reinforcement or Supervised:
Reinforcement: <music sheet/notes, output, rating> this would have an algorithm that would play some sheet music and then give an output that would be rated on how well it performed the piece.
Supervised:<music sheet/notes, how its supposed to sound> this would have the algorithm learn from the input notes and their corresponding sounds.

e) Supervised
Supervised: <customer features, credit limit> this would train based off of customer features and the credit limit that the bank is willing to give, it would be solved as a regression problem rather than a classification problem.

# Problem 4

Problem 1.7: Evaluate the performance of g on the 3 points in $X$ outside $D$. To measure performance, compute how many of the 8 target functions agree with g.

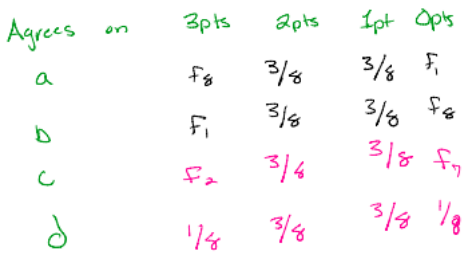
Tinkering: Forming the results for each algorithm for each of the 3 points.



| x | y | a | b | c | d |
|---|---|---|---|---|---|
| 000 | ○ | • | ○ | ○ | ○ |
| 001 | ● | • | ○ | ● | ● |
| 010 | ● | • | ○ | ● | ● |
| 011 | ○ | • | ○ | ○ | ○ |
| 100 | ● | • | ○ | ● | ● |
| 101 | ? | • | ○ | ○ | - |
| 110 | ? | • | ○ | ○ | - |
| 111 | ? | • | ○ | ● | - |

Figure 1: Results for each g.

Tallying up the agreements



| Agrees on | 3pts | 2pts | 1pt | 0pts |
|---|---|---|---|---|
| a | $f_8$ | 3/8 | 3/8 | $f_1$ |
| b | $f_1$ | 3/8 | 3/8 | $f_8$ |
| c | $f_2$ | 3/8 | 3/8 | $f_7$ |
| d | 1/8 | 3/8 | 3/8 | 1/8 |

Figure 2: Agreement with other objective functions.

a) It this case, g fully agrees with only $f_8$ and has no agreement with $f_1$. It partially agrees with 3/8 of the target functions on both 2 and 1 points.

b) It this case it is the opposite of the situation in part a, g fully agrees with only $f_1$ and has no agreement with $f_8$. It partially agrees with 3/8 of the target functions on both 2 and 1 points.

c) In the case in which it is determines by counting the number of ones, g fully agrees with only $f_2$ and has no agreement with $f_7$. It partially agrees with 3/8 of the target functions on both 2 and 1 points.

d) In the case in which it agrees with training data but disagrees with the most test data, g will have fully agrees with one of the objective functions, which means will also completely disagree with another target function. The rest of the agreement would be split equally between the target functions on 1 and 2 points. This is because out of all the target functions, they cover all of the possibilities for outputs based on the test data.

We can see that the performance of each g is the same (when evaluating its agreement with the other possible target functions). While each of the algorithms have agreement with different sets of the possible target function, overall their accuracy would be considered the same.

# Problem 5

Problem 1.1) Two opaque backs, one with 2 black balls, only with 1 black and 1 white. Pick a bag at random, pick the first ball, its black. Pick the second ball from the same bag, what is the probability that it is also black. Tinkering:

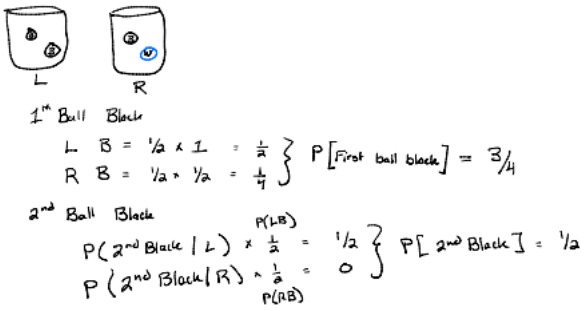



Figure 3: Probability for each bag.

Calculating the probability that the first ball is black:

$$P(L \ \& \ B) = \frac{1}{2} \times 1 = \frac{1}{2}$$
$$P(R \ \& \ B) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

$$P(B) = P(L \ \& \ B) + P(L \ \& \ B) = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$$

Calculating the probability that the second ball is black:

$$P(2B \ |L) \times P(L) = 1 \times \frac{1}{2} = \frac{1}{2}$$
$$P(2B \ |R) \times P(R) = 0 \times \frac{1}{2} = 0$$

$$P(1st \ B \wedge 2nd \ B) = P(2B|L) + P(2B|R) = \frac{1}{2} + 0 = \frac{1}{2}$$

Applyes Bayes' Theorum:

$$P(1st \ B \wedge 2nd \ B) = P(2nd \ B \mid 1st \ B) \times P(1st \ B)$$
$$(\frac{1}{2}) = P(2nd \ B \mid 1st \ B) \times \frac{3}{4}$$
$$P(2nd \ B \mid 1st \ B) = \frac{2}{3}$$

# Problem 6

Problem 1.2: Perceptron in 2 Dimensions $h(x) = sign(\omega^T x)$.

a) Show that regions of the plane where $h(x) = +1$ and $h(x) = -1$ are separated by a line. What are slope and intercept?

$\omega^T x = \omega_0 + \omega_1 x_1 + \omega_2 x_2 = 0$
Since the $\omega$ values will be fixed after training the algorithm, the only variables will be $x_1$ and $x_2$. Since we know that the $\omega$ is a vector constants we can see that this will form a line with:

$$S = [1, S_1, S_2]^T$$
$$h(x) = +1 \; when \; \omega^T S > 0$$
$$h(x) = -1 \; when \; \omega^T S < 0$$

This separation by a line can also be seen by the nature of the sign function, using the $\omega^T x$ equation to separate the data.

Slope a = $\omega_1/(-\omega_2)$
Intercept b = $\omega_0/(-\omega_2)$

b)

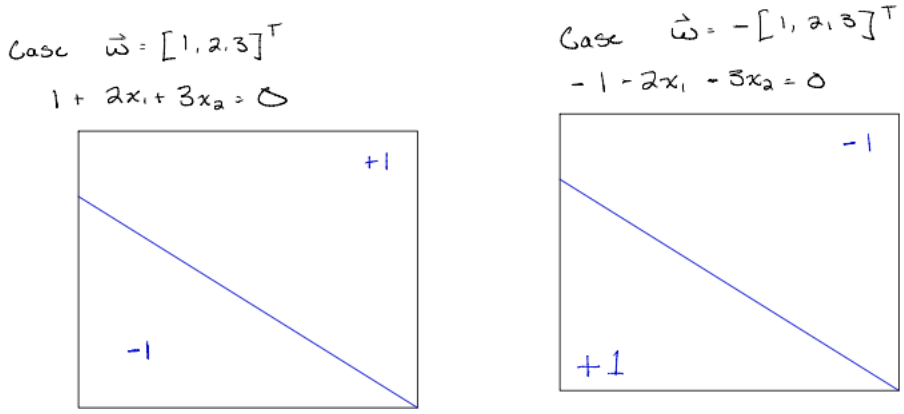



Figure 4: $\omega$ cases.

# Problem 7

Problem 1.4:
a) Chosen Weights: $\omega = [9, -12, -12]$

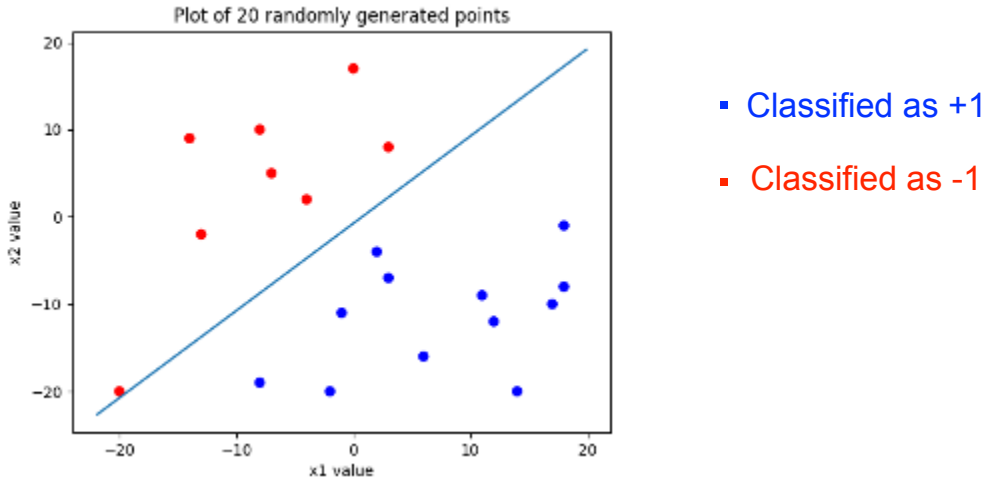



Figure 5: 20 Random points with line from weights [9, -12, -12].

b) After 4 iterations of the perceptron learning algorithms, it found a solution $\omega = [-15.13993, -21.32305, 12.14167]$ that classified all the points correctly.

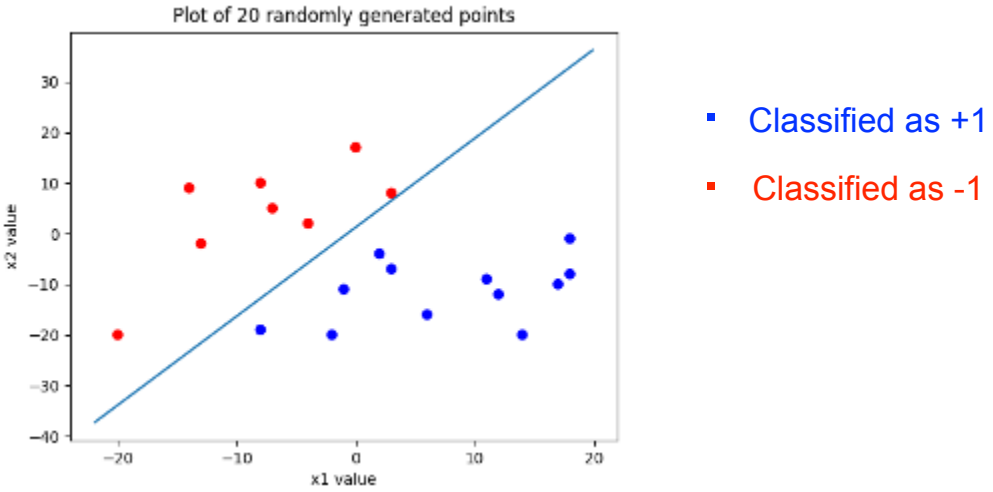



Figure 6: 20 Random points with line from weights [-15.13993,-21.32305,12.14167].

g is realtives close to f, however there is a distinct difference in the slope and intercept that can be seen from the graphs.

c) Chosen weights for 20 random points: $\omega = [5, 3, 5]$

Weights chosen by PLA after 13 iterations: $\omega = [([-19.17920125, 56.89213912, 117.06799108]$

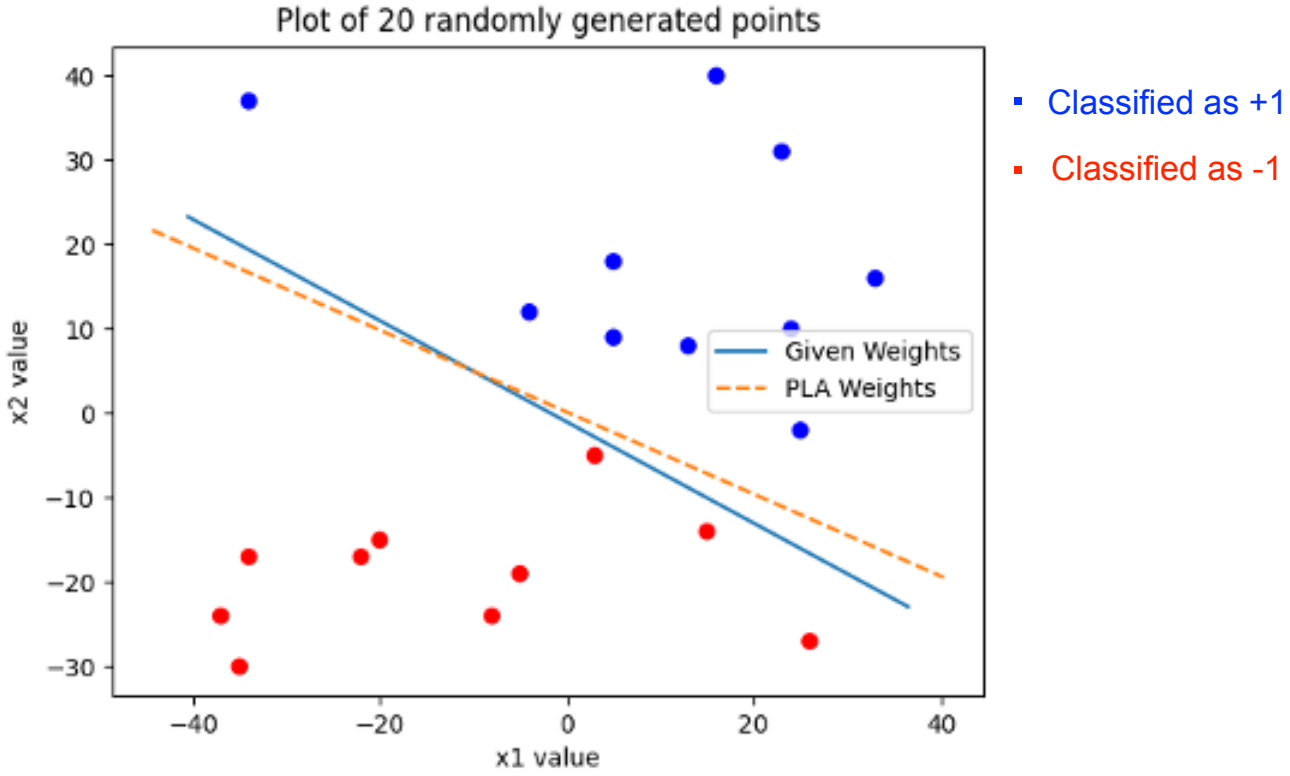



Figure 7: PLA on 20 random points.

Compared to b, this PLA solution reassembles the original weight fitting more closely, however it took more iterations to arrive at this solution. It is notable that even though the weights do not resemble the original derived weights that the line is forms is still similar to the original. The line in this plot is quite distinguishable from the original, which makes sense because there is a good amount of open space separating the points, so may different solutions exist.

Note: Given weights means the weights used to separate the data

d) Chosen weights for 100 points (by me/my algo): $\omega = [-2, -10, 3]$

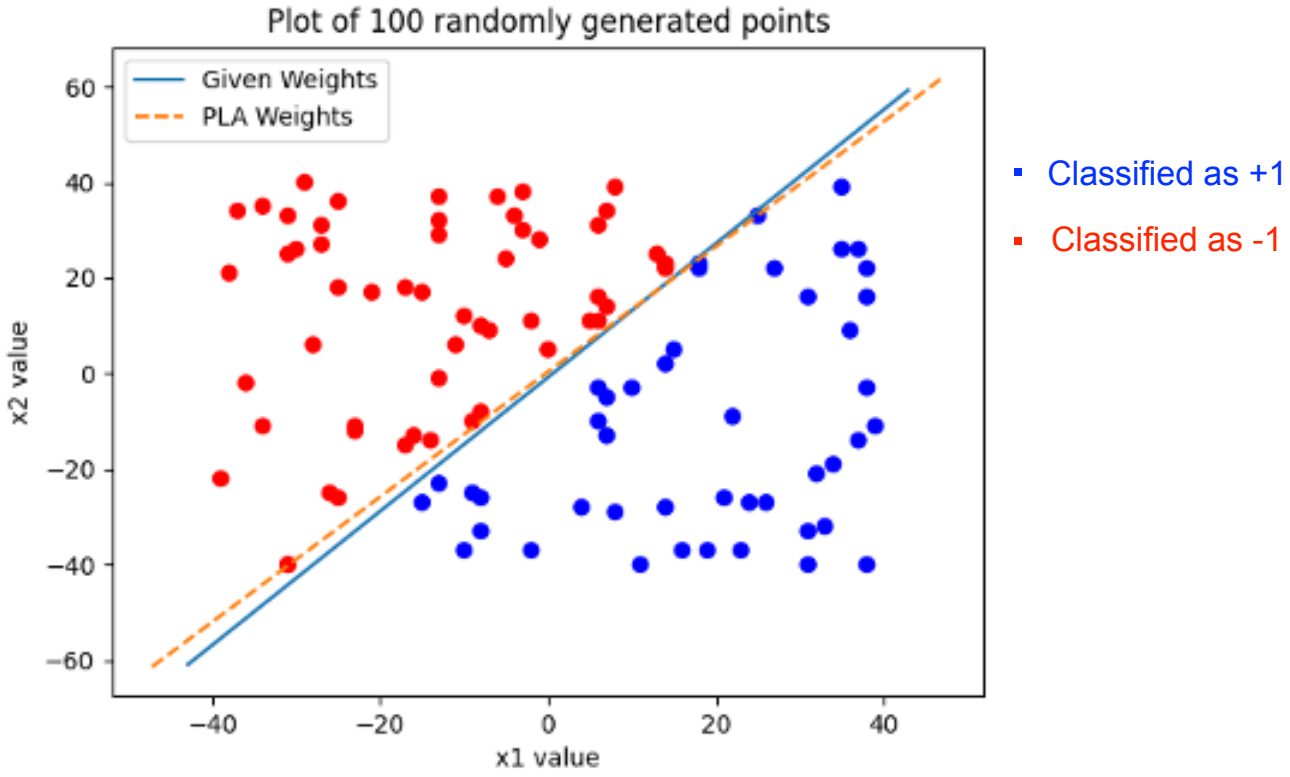



Figure 8: PLA on 100 random points].

Weights chosen by PLA after 42 iterations: $\omega = [34.74859453, 122.3155584, -93.50121477]$

Compared to b, this run of PLA took more iterations to arrive at the solution, however, when running my algorithm on new random data, the number of iterations it took to converge ranged depending on what the randomly generated weights started out as. This lines on this plot are distinguishable but much closer together than in p.

Note: Given weights means the weights used to separate the data

Problem 7 continued on next page...

e) Chosen weights for 1000 points: [-3, -8, -16]

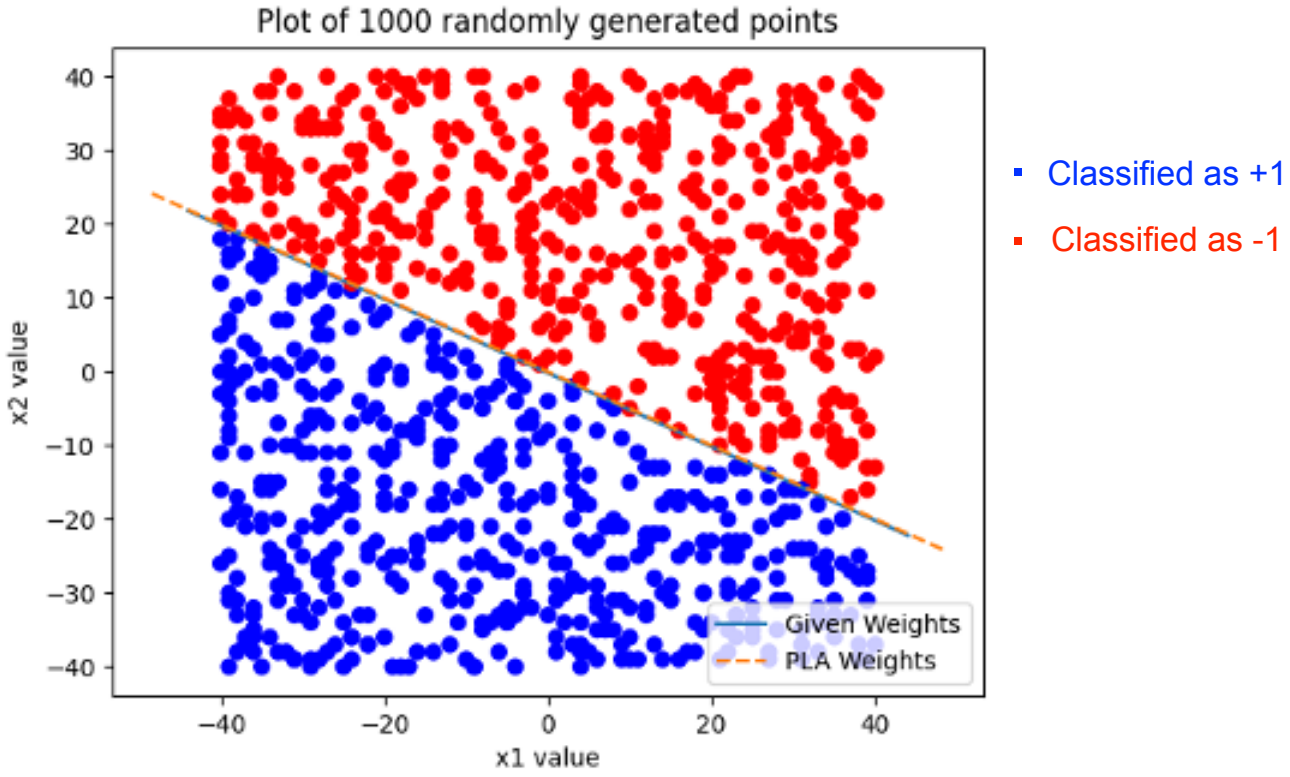



Figure 9: 1000 Random points with PLA.

After 1687 iterations PLA found a solution with weights [-81.2575022 , -2777.62959979, -5565.39792569]

Compared to b this run of PLA also took more iterations to arrive at the solution, however like above, there was a range when I ran the algorithm multiple times. There does appear to be an upper and lower limit on the number of iterations it takes to converge, which when I looked it up could be proved. The lines on this plot look almost indistinguishable, which given that there are so many points makes sense because there is a not a lot of room for variation in the slope and intercept.

Note: Given weights means the weights used to separate the data

As more data was used from parts b through e, it can be seen that the line separating the data become more closely aligned with the one that was used to generate the data. This make sense for the fact that having more data will yield a better approximation to the true target function f.