CMPE 257 (ML) ASSIGNMENT 1

RISHIKA MACHINA

SID: 012525227

rishika.machina@sjsu.edu

EXERCISE PROBLEMS

1.3 **The weight update rule, which has the nice interpretation that it moves in the direction of classifying x(t) correctly.**

(a) Show that y(t)wT(t)x(t) < 0. [Hint: x(t) is misclassified by w(t).]

(b) Show that y(t)wT(t +l)x(t) > y(t)wT(t)x(t). [Hint: Use (1.3).]

(c) As far as classifying x(t) is concerned, argue that the move from w(t) to w(t + 1) is a move 'in the right direction'.

ANS

(a) Show that y(t)wT(t)x(t) < 0. [Hint: x(t) is misclassified by w(t).]

According to the perceptron algorithm, $Y(t) = \text{sign}(w^T(t)\, x(t))$

Where, sign(s) = +1 for s>=0 and -1 for s<0

If x(t) is misclassified, y(t) is not equal to sign($w^T(t)$ x(t)), which means both are of opposite signs since two possible outputs are +1 and -1.

(i.e) y(t) = 1 if sign($w^T(t)$ x(t)) = -1 and y(t) = -1 if sign($w^T(t)$ x(t)) = 1

Therefore, for a misclassified example, $y(t)\, w^T(t)\, x(t)<0$.

(b) Show that y(t) wT(t+l) x(t) > y(t) wT(t) x(t). [Hint: Use (1.3).]

Weight is updated based on the given rule,

$$w(t + 1) = w(t) + y(t)x(t)$$

Using the above equation in y(t) wT(t+l) x(t)= y(t) x(t) [w(t) + y(t) x(t)] = y(t) x(t) w(t) + $y^2(t)\, x^2(t)$, which is evidently greater than y(t) w(t) x(t).

(c) As far as classifying x(t) is concerned, argue that the move from w(t) to w(t + 1) is a move 'in the right direction'.

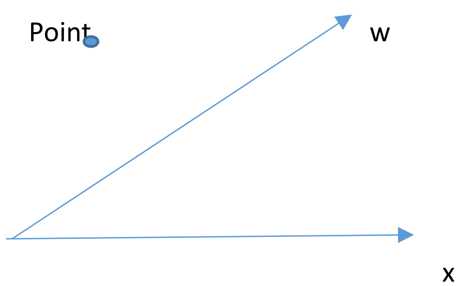
Consider a misclassified example, with input x(t) and initial weight w(t). They are said to be misclassified only if y(t)= -1 where it has to be +1 or vice versa.

Two cases are possible

First case:

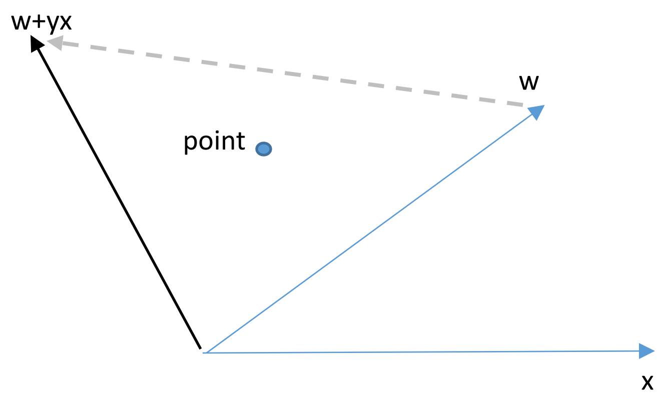
Let ‘P’ be the angle between w and x

For 0 < P < (pi/2) and (3*pi)/2 < P < 2*pi



Here y should be +1 but as the point is misclassified. Y is -1.

In order to change y as +1, If we attempt to add yx to the w available. The following change will occur.

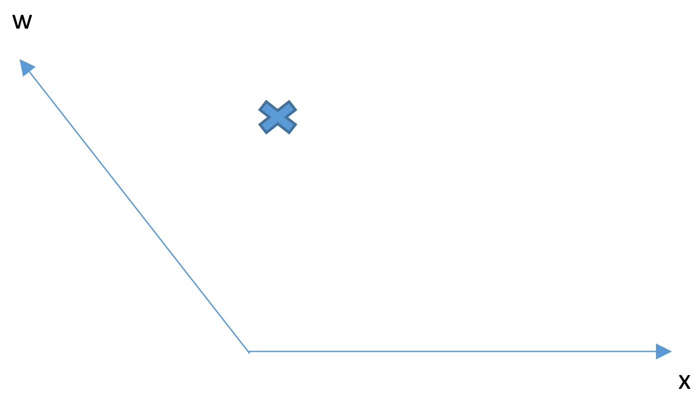


After changing the weight to w+yx, y for the point will be +1.

Hence for this case, changing the weight according the rule is correct.

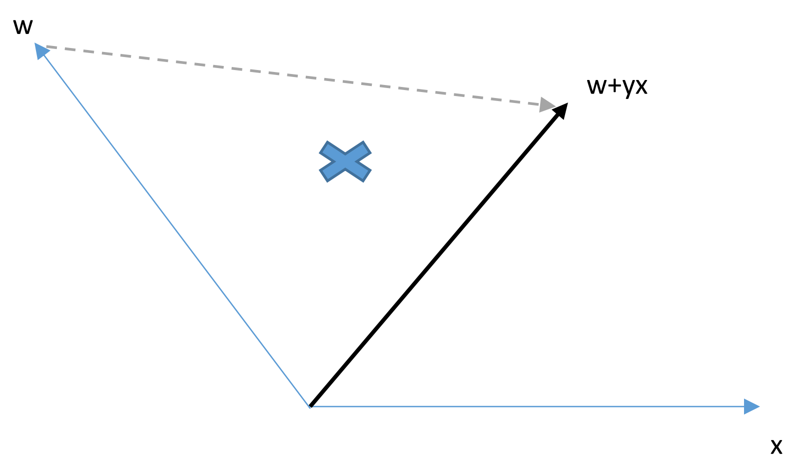
Second case:

(pi/2)< P < (3*pi)/2



For this case y should be -1 but it is +1, since it is a misclassified example.

After changing weight as w+yx, the figure will change as shown.



After changing the weight to w+yx, y for the point will be -1.

Hence for this case, changing the weight according the rule is correct.

Finally, changing w(t) to w(t+1) will move in the right direction.

<u>1.6</u> For each of the following tasks, identify which type of learning is involved (supervised, reinforcement, or unsupervised) and the training data to be used. If a task can fit more than one type, explain how and describe the training data for each type.

(a) Recommending a book to a user in an online bookstore

(b) Playing tic tac toe

(c) Categorizing movies into different types

(d) Learning to play music

(e) Credit limit: Deciding the maximum allowed debt for each bank customer

<u>ANS</u>

(a) Recommending a book to a user in an online bookstore is purely '**supervised learning'** since the model will be based on the previous customers and choices made by them.

(b) Playing tic tac toe could be **'Reinforced or Unsupervised learning'** since the machine should either be fed with the game rules or it should learn by its own.

(c) Categorizing movies into different types is purely **'Supervised learning'** since it is based on past movies and their categorization.

(d) Learning to play music can be **'Reinforced or Unsupervised learning'** since the machine should be fed with basics of music or should learn on its own.

(e) Credit limit is purely **'Supervised learning'** as the required factors change from bank to bank and the machine either approves or declines based on the previous data.

1.8 If μ = 0.9, what is the probability that a sample of 10 marbles will have v <= 0.1?

[Hints: 1. Use binomial distribution. 2. The answer is a very small number.]

ANS

We should find the probability of v<=0.1, which means the number of red marbles in the sample should be either 0 or 1. So we should find the sum of probabilities for v=0 and v=1.

Binomial distribution, $P[x==r] = \binom{n}{r} * p^r * q^{n-r}$

Let 'X' ne the number of red marbles picked whereas 'v' is the probability of red marbles in the sample

$$P[v<=0.1] = P[X==0] + P[X==1] = \binom{10}{0} * (0.9)^0 * (0.1)^{10} + \binom{10}{1} * (0.9)^1 * (0.1)^9$$

$$= 1*1*0.1^{10} + 10*0.9*0.1^9$$

$$= 0.1^{10} + 10*0.9*0.1^9$$

$$= 0.1^{10} + 90*0.1^{10}$$

$$= 91 * 10^{-10}$$

1.9 If μ = 0.9, use the Hoeffding Inequality to bound the probability that a sample of 10 marbles will have v <= 0.1 and compare the answer to the previous exercise.

ANS

We know that Hoeffding inequality is **$P[|v-u|>E] <= 2*\exp(-2E^2N)$**

We know that if |A|<1 implies -1 < A < +1. Using this for the left part of the Hoeffding inequality

$|v-u|>E \longrightarrow -E < v-u < E \longrightarrow (-E+u) < v < (E+u) \longrightarrow (-E+0.9) < v < (E+0.9)$

To find P(v<=0.1), $E+0.9=0.1 \longrightarrow E= 0.1-0.9 = -0.8$

Finally, $P[v<=0.1] <= 2*\exp(-2*-0.8^2*10) <= 2*\exp(-12.8) <= 5.52*10^{-6}$

Therefore, using Hoeffding inequality, $P[v<=0.1] <= 5.52*10^{-6}$

The probability obtained using **binomial** distribution is very **less than** that obtained using **Hoeffding** inequality.

PROBLEMS

1.2 Consider the perceptron in two dimensions: $h(x) = \text{sign}(w^T x)$ where $w = [wo, w1, w2]^T$ and $x = [1, x1, x2]^T$. Technically, x has three coordinates, but we call this perceptron two-dimensional because the first coordinate is fixed at 1.

(a) Show that the regions on the plane where h(x) = +1 and h(x) = -1 are separated by a line. If we express this line by the equation x2 = ax1 + b, what are the slope a and intercept b in terms of w0, w1, w2?

(b) Draw a picture for the cases $w = [1, 2, 3]^T$ and $w = -[1, 2, 3]^T$.

In more than two dimensions, the +1 and -1 regions are separated by a hyperplane, the generalization of a line.

ANS

a) $H(x)= \text{sign}(w^T x)$

$$W^T x = \begin{pmatrix} w0 & w1 & w2 \end{pmatrix} \begin{pmatrix} 1 \\ X1 \\ X2 \end{pmatrix} = w0 + (w1* x1) + (w2*x2)$$

We know that sign(x) = -1 for negative x and is +1 for positive x, which means $\text{sign}(w^T x)= 0$ will differentiate both the regions.

$w0 + (w1*x1) + (w2*x2)= 0 \longrightarrow x2 = (-w1/w2)x1 + (-w0/w2)$

Comparing it with x2=a.x1+b, **Slope(a) = -w1/w2 and Intercept(b) = -w0/w2**

b) When $w=[1\ 2\ 3]^T$ line will be 1+ 2*x1 + 3*x2 = 0

plots for both w set is the same but the region only differ

c) When $w=[-1\ -2\ -3]^T$ line will be -1- 2*x1 – 3*x2 = 0 → 1+ 2*x1 + 3*x2 = 0

1.4 **In Exercise 1.4, we use an artificial data set to study the perceptron learning algorithm. This problem leads you to explore the algorithm further with data sets of different sizes and dimensions.**

(a) Generate a linearly separable data set of size 20 as indicated in Exercise 1.4. Plot the examples { (x_n, Y_n) } as well as the target function f on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.

(b) Run the perceptron learning algorithm on the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples { (x_n, Y_n) }, the target function f, and the final hypothesis g in the same figure. Comment on whether f is close to g.

(c) Repeat everything in (b) with another randomly generated data set of size 20. Compare your results with (b).

(d) Repeat everything in (b) with another randomly generated data set of size 100. Compare your results with (b).

(e) Repeat everything in (b) with another randomly generated data set of size 1000. Compare your results with (b).

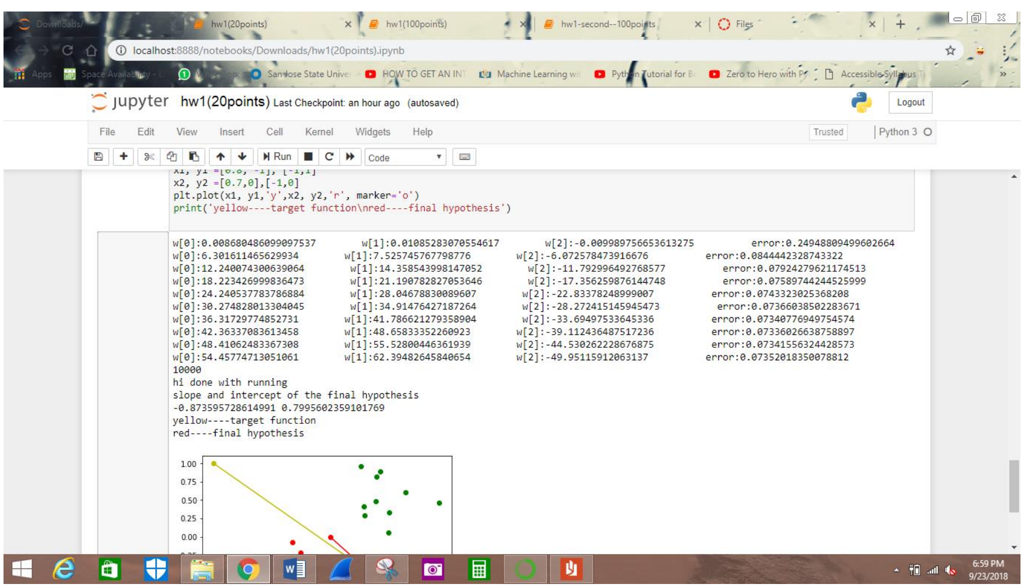
<u>ANS</u>

For the given data set, two major differences were observed,

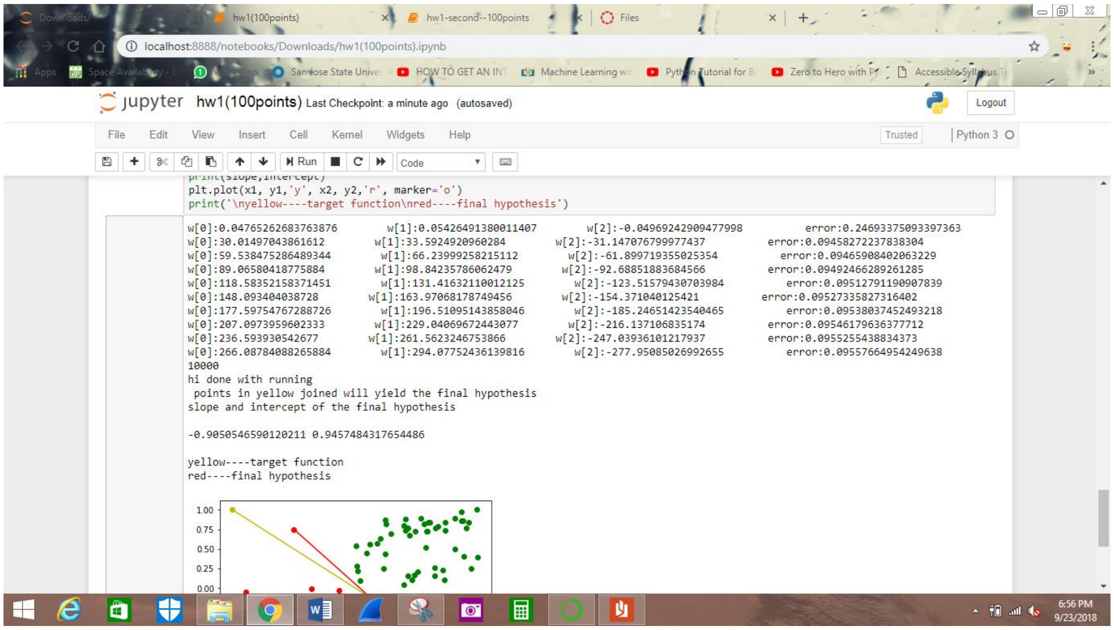
1) slope of the final hypothesis is different even if the data is in first and third quadrant
2) running time is different

'.pynb file' for 3 cases is submitted.

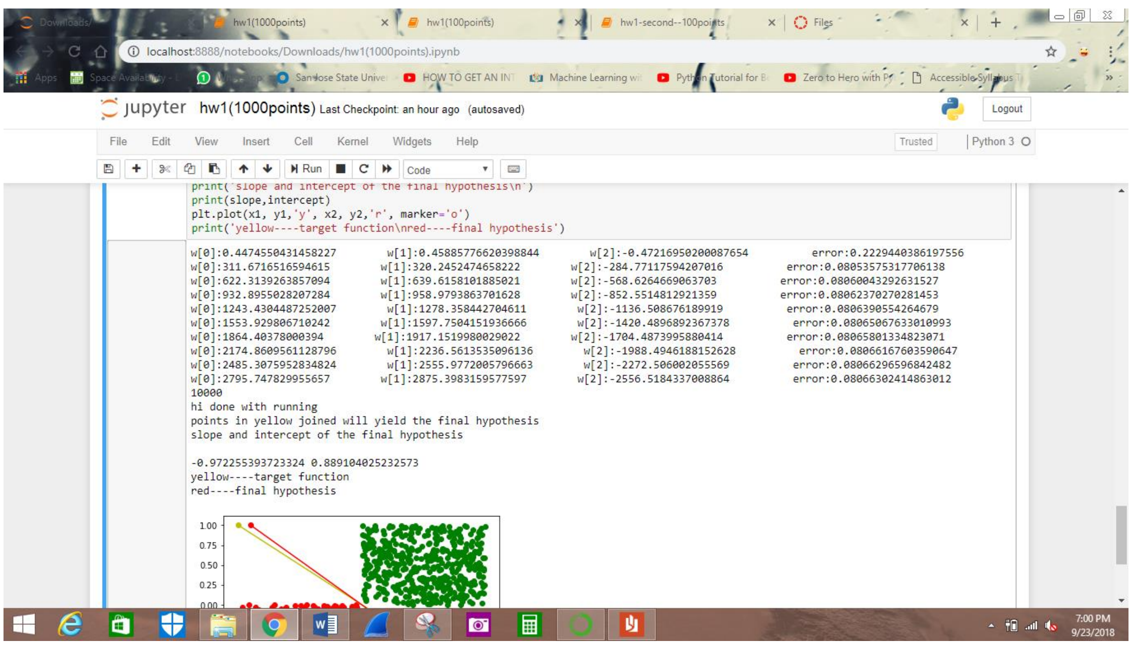
20 points output:



100 points output:



1000 points output:



1.5 **The perceptron learning algorithm works like this: In each iteration t, pick a random (x(t), y(t)) and compute the 'signal' s(t) = w^T(t)x(t). If y(t) · s(t) <= 0, update w by**

$$\mathbf{w(t + 1) +\text{-}w(t) + y(t) \cdot x(t) ;}$$

One may argue that this algorithm does not take the 'closeness' between s(t) and y(t) into consideration. Let's look at another perceptron learning algorithm: In each iteration, pick a random (x(t), y(t)) and compute s(t). If y(t) · s(t) <=1, update w by

$$\mathbf{w(t + 1) +\text{-}w(t) + n \bullet (y(t)\ s(t)) \cdot x(t) ,}$$

where n is a constant. That is, if s(t) agrees with y(t) well (their product is > 1), the algorithm does nothing. On the other hand, if s(t) is further from y(t), the algorithm changes w(t) more. In this problem, you are asked to implement this algorithm and study its performance.

(a) Generate a training data set of size 100 similar to that used in Exercise 1.4. Generate a test data set of size 10, 000 from the same process. To get g, run the algorithm above with n = 100 on the training data set, until a maximum of 1, 000 updates has been reached. Plot the training data set, the target function f, and the final hypothesis g on the same figure. Report the error on the test set.

(b) Use the data set in (a) and redo everything with n= 1.

(c) Use the data set in (a) and redo everything with n = 0.01.

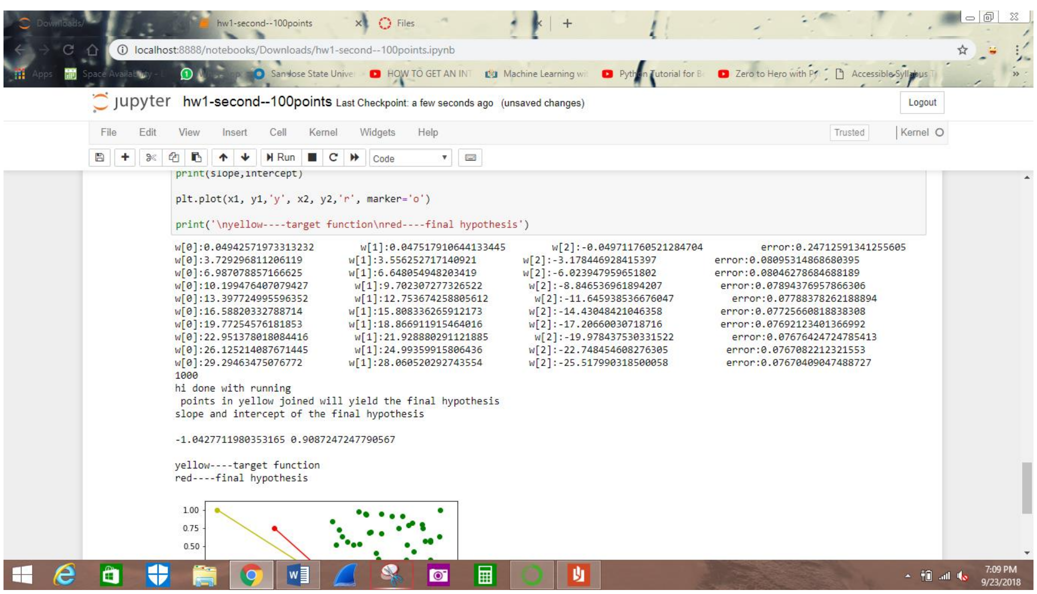
(d) Use the data set in (a) and redo everything with n = 0.0001.

(e) Compare the results that you get from (a) to (d).

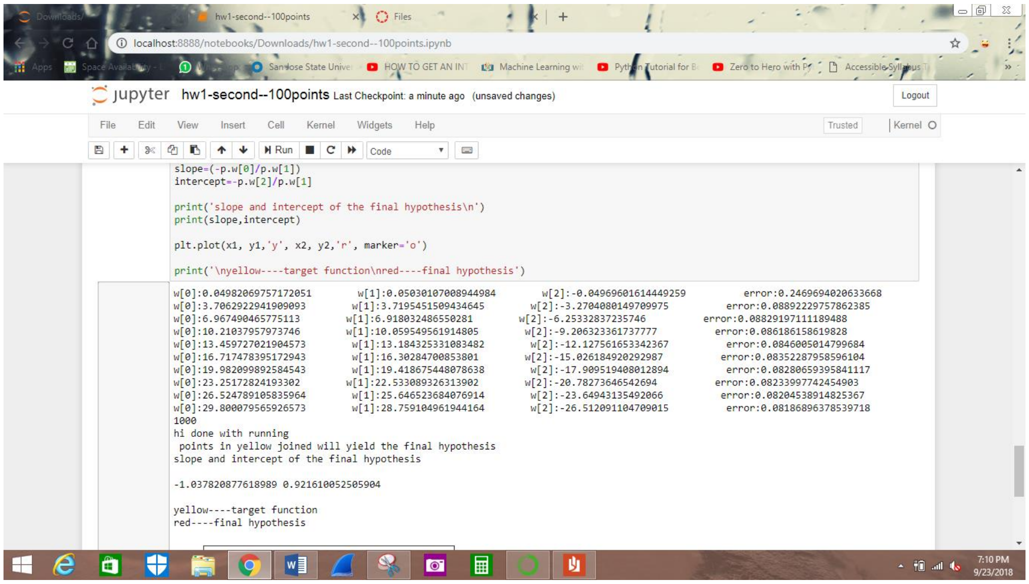
The algorithm above is a variant of the so called Adaline (Adaptive Linear Neuron) algorithm for perceptron learning.

<u>**ANS**</u> Error is decreased a bit when compared to 1.4 problem

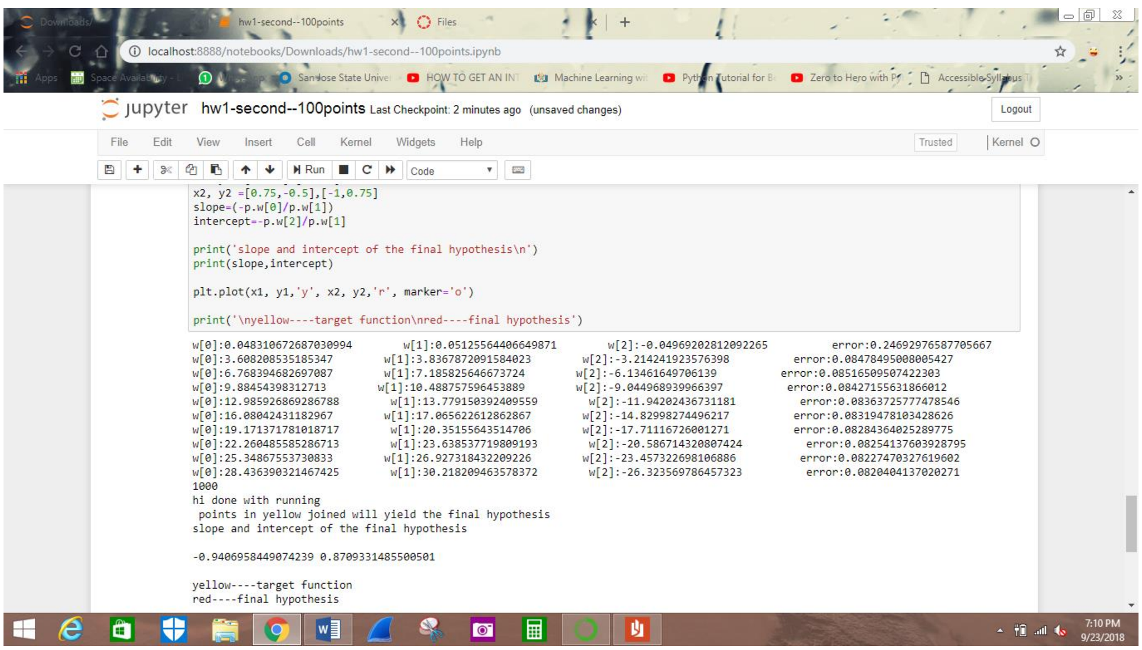
For 100 points—n=100



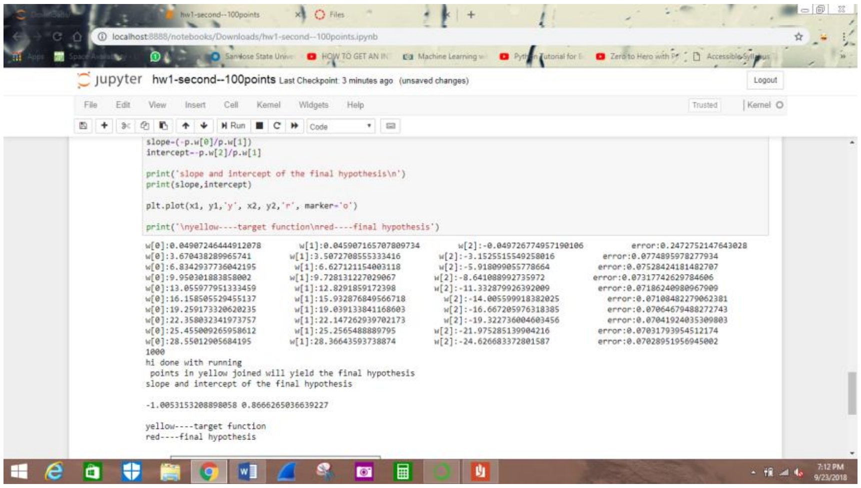
n= 1



n=0.01



n=0.0001



1.11 **The matrix which tabulates the cost of various errors for the CIA and Supermarket applications in Example 1.1 is called a risk or loss matrix.**

For the two risk matrices in Example 1.1, explicitly write down the in sample error E_{in} that one should minimize to obtain g. This in-sample error should weight the different types of errors based on the risk matrix.

[Hint: Consider Y_n = + 1 and Y_n= -1 separately.]

ANS

We know that E_{in} is 'In sample error '

for Supermarket

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}\left[h(x_n) \neq f(x_n)\right]$$

$$= \frac{1}{N}\left[\sum_{y_n=1} e(h(x_n),1) + \sum_{y_n=-1} e(h(x_n),-1)\right]$$

$$= \frac{1}{N}\left[\sum_{y_n=1} 10\,(h(x_n)\neq y_n) + \sum_{y_n=-1}(h(x_n)\neq y_n)\right]$$

Similary, for CIA.

$$E_{in}(h) = \frac{1}{N}\left[\sum_{y_n=1}(h(x_n)\neq y_n) + 1000\sum_{y_n=-1}(h(x_n)\neq y_n)\right]$$