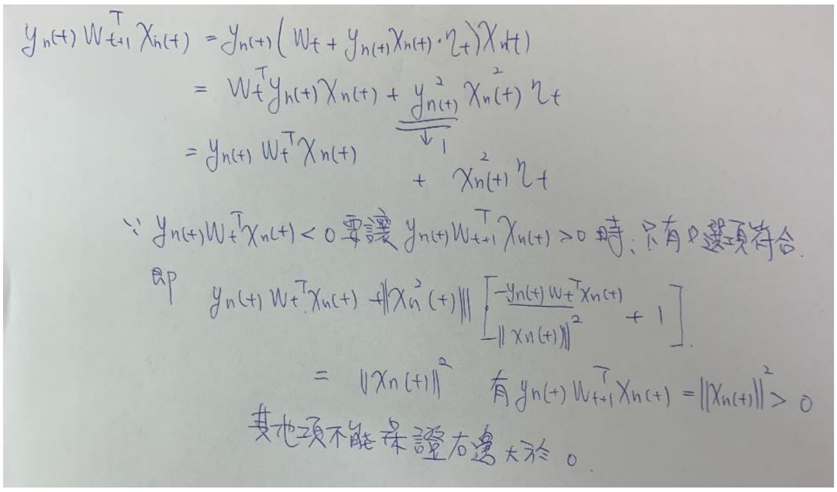
HTML _hw1　　　　　　　　　　　廖浚評 r10521516 土木所交通組碩一

1. (D)

A 用地圖可以直接測量,B 機率問題用算的,C 數學直接計算故選 D

2. (E)



$y_{n(t)} W_{t+1}^T X_{n(t)} = y_{n(t)} (W_t + y_{n(t)} X_{n(t)} \cdot \eta_t) X_{n(t)}$

$= W_t^T y_{n(t)} X_{n(t)} + \underbrace{y_{n(t)}^2}_{1} X_{n(t)}^2 \eta_t$

$= y_{n(t)} W_t^T X_{n(t)} + X_{n(t)}^2 \eta_t$

∵ $y_{n(t)} W_t^T X_{n(t)} < 0$ 要讓 $y_{n(t)} W_{t+1}^T X_{n(t)} > 0$ 時，只有D選項符合.

即 $y_{n(t)} W_t^T X_{n(t)} + \|X_n^2(t)\| \left[\frac{-y_{n(t)} W_t^T X_{n(t)}}{\|X_{n(t)}\|^2} + 1 \right]$.

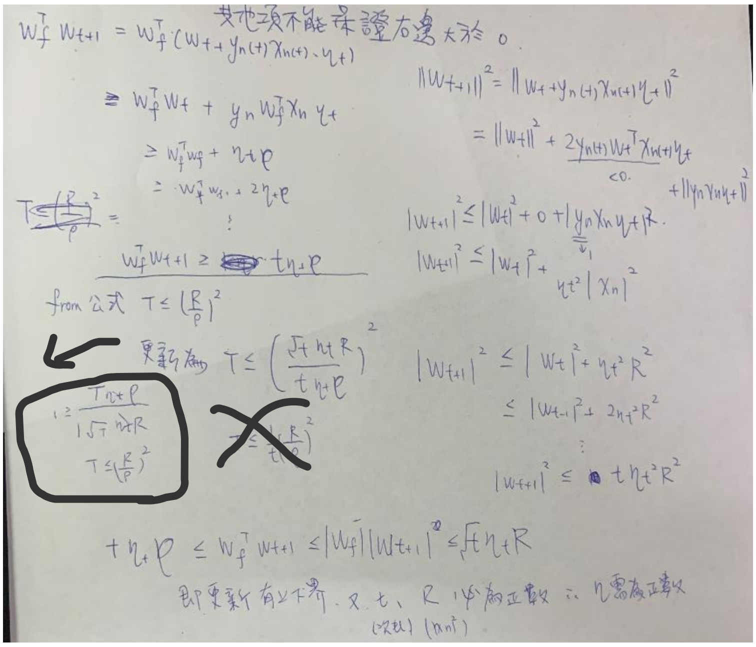
$= \|X_{n(t)}\|^2$ 有 $y_{n(t)} W_{t+1}^T X_{n(t)} = \|X_{n(t)}\|^2 > 0$

其他項不能保證右邊大於 0.

3. (D)

(C)從前一題得知，右邊等於 0，等於不會更新 Wt，那麼就不會停止

其餘選項符合運算，即更新率不影響 T，故皆會 halting，選 4



其他項不能保證右邊大於 0.

$W_f^T W_{t+1} = W_f^T (W_t + y_{n(t)} X_{n(t)} \cdot \eta_t)$

$\geq W_f^T W_t + y_n W_f^T X_n \eta_t$

$\geq W_f^T W_f + \eta_t \rho$

$= W_f^T W_{t-1} + 2\eta_t \rho$

$\vdots$

$W_f^T W_{t+1} \geq t\eta_t \rho$

~~$T \leq \left(\frac{R}{\rho}\right)^2 =$~~

from 公式 $T \leq \left(\frac{R}{\rho}\right)^2$

更新為 $T \leq \left(\frac{\sqrt{t}\,\eta_t R}{t\eta_t \rho}\right)^2$

$\boxed{1 \geq \frac{T\eta_t \rho}{\sqrt{T}\,\eta_t R} \quad T \leq \left(\frac{R}{\rho}\right)^2}$

~~$T \leq \left(\frac{R}{t\rho}\right)^2$~~

$\|W_{t+1}\|^2 = \|W_t + y_{n(t)} X_{n(t)} \eta_t\|^2$

$= \|W_t\|^2 + \underbrace{2y_{n(t)} W_t^T X_{n(t)} \eta_t}_{<0.} + \|y_n X_n \eta_t\|^2$

$|W_{t+1}|^2 \leq |W_t|^2 + 0 + |\underbrace{y_n}_{1} X_n \eta_t|^2$.

$|W_{t+1}|^2 \leq |W_t|^2 + \eta_t^2 |X_n|^2$

$|W_{t+1}|^2 \leq |W_t|^2 + \eta_t^2 R^2$

$\leq |W_{t-1}|^2 + 2\eta_t^2 R^2$

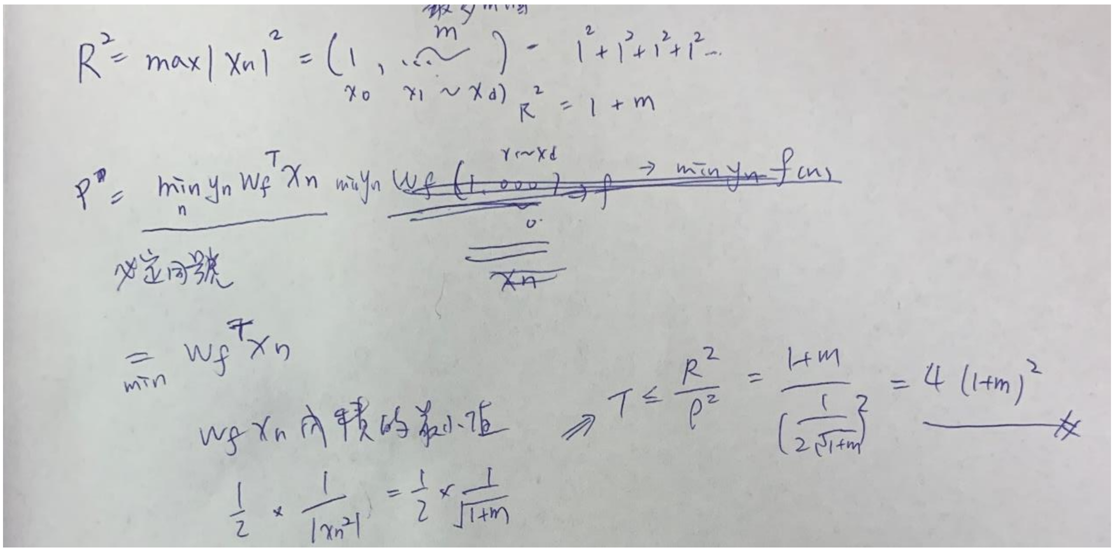
$\vdots$

$|W_{t+1}|^2 \leq t\eta_t^2 R^2$

$t\eta_t \rho \leq W_f^T W_{t+1} \leq |W_f| |W_{t+1}|^2 \leq \sqrt{t}\,\eta_t R$

即更新有上下界. 又 t、R 必為正數 ∴ η需為正數
(次數) ($\max \|x_n\|$)

4. (A)



數 m 個

$R^2 = \max|x_n|^2 = (\underset{x_0}{1}, \underset{x_1 \sim x_d)}{\overset{m}{\cdots}} - 1^2+1^2+1^2+1^2\cdots$

$R^2 = 1 + m$

$\rho^* = \underline{\min_n y_n w_f^T x_n}$ $\min y_n$ ~~$w_f(\overset{x_1 \sim x_d}{1, 0 0 0}) \rightarrow f \rightarrow \min y_n f(x_n)$~~ ~~$x_n$~~

必定同號

$= \min w_f^T x_n$

$w_f x_n$ 內積的最小值

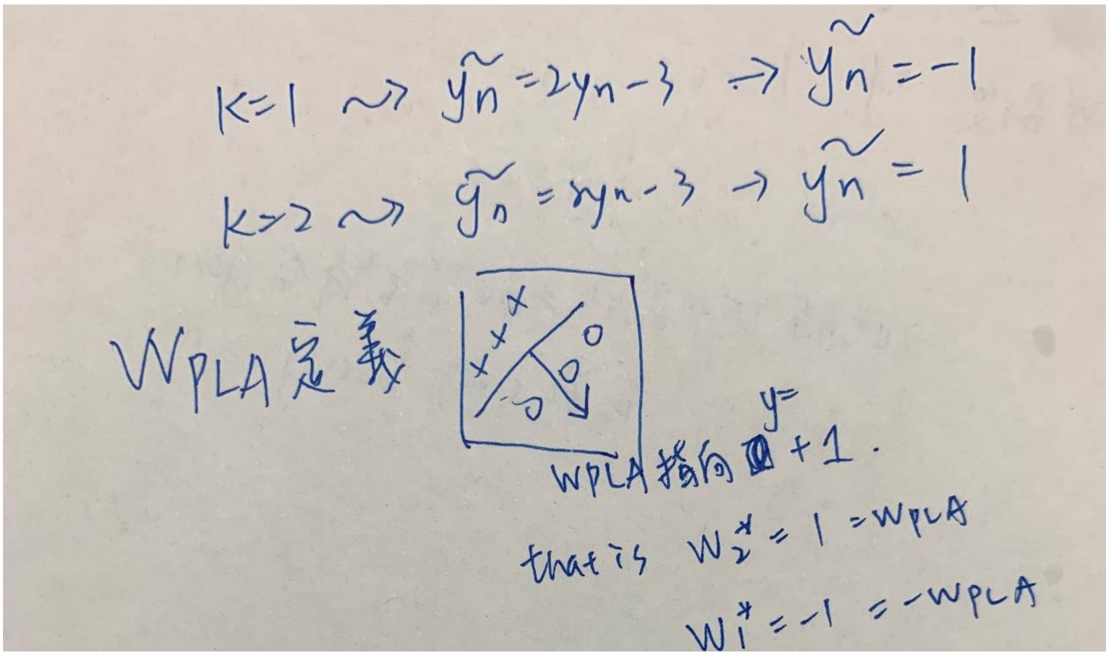
$\frac{1}{2} \times \frac{1}{|x_n^2|} = \frac{1}{2} \times \frac{1}{\sqrt{1+m}}$

$\Rightarrow T \le \frac{R^2}{\rho^2} = \frac{1+m}{(\frac{1}{2\sqrt{1+m}})^2} = 4(1+m)^2$ #

5. (B)

因為這裡的 multiPLA 只有兩項，基本跟 binaryPLA 一樣步驟(題意)，所以做出來的結果要跟 binaryPLA 一樣，因此得出的 W1 指向 yn=-1

W2 指向 yn=1



$k=1 \rightsquigarrow \tilde{y_n}=2y_n-3 \rightarrow \tilde{y_n}=-1$

$k=2 \rightsquigarrow \tilde{y_n}=2y_n-3 \rightarrow \tilde{y_n}=1$

WPLA 定義

WPLA 指向 y=+1.

that is $W_2^*=1=W_{PLA}$

$W_1^*=-1=-W_{PLA}$

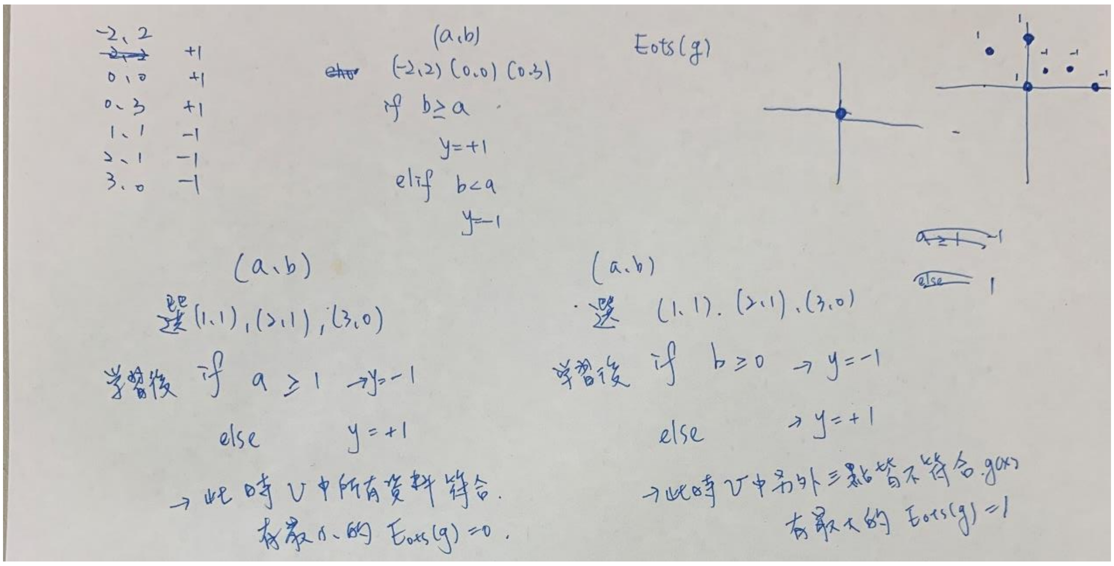
6.　(D)

(A)沒有回饋、互動(B)沒有對正確、錯誤的預測給予反饋(C)只有輸出一種 Yn(vector)(D)沒有給 label 由程式自行定義(E)數據沒有一直更新也沒有動態回饋

7.　(C)

分類到訂定好的 tag 中(multilabel)又一次訓練完數據沒有一直更新也沒有動態回饋(batch learning)

8. (B)



-2、2	
~~2、2~~	+1
0、0	+1
0、3	+1
1、1	-1
2、1	-1
3、0	-1

(a,b)

~~else~~ (-2,2) (0,0) (0,3)

```
if b≥a
    y=+1
elif b<a
    y=-1
```

Eots(g)

(a,b)

選 (1,1),(2,1),(3,0)

學習後 if a ≥ 1 → y=-1

else y=+1

→ 此時 V 中所有資料符合.

有最小的 Eots(g) = 0 .

(a,b)

選 (1,1).(2,1).(3,0)

學習後 if b ≥ 0 → y=-1

else → y=+1

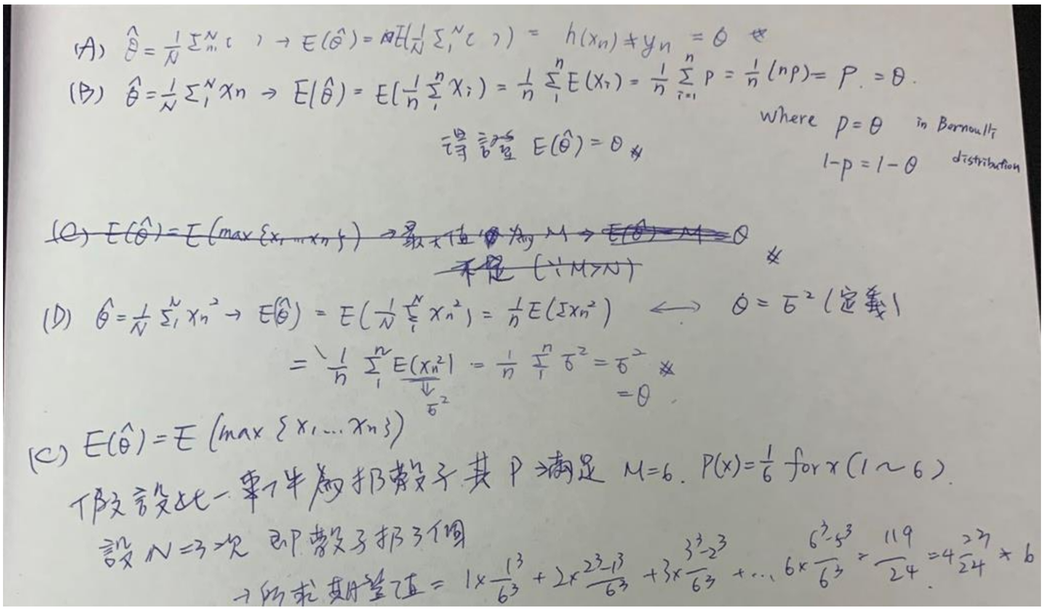
→ 此時 V 中另外三點皆不符合 g(x)

有最大的 Eots(g) = 1

~~a≥1~~ -1

~~else~~ 1

9. (C)



(A) $\hat{\theta} = \frac{1}{N}\sum_{n=1}^{N}(\quad) \to E(\hat{\theta}) = E(\frac{1}{N}\sum_{1}^{N}(\quad)) = h(x_n) \neq y_n = \theta$ せ

(B) $\hat{\theta} = \frac{1}{N}\sum_{1}^{N} x_n \to E(\hat{\theta}) = E(\frac{1}{n}\sum_{1}^{n} x_i) = \frac{1}{n}\sum_{1}^{n} E(x_i) = \frac{1}{n}\sum_{i=1}^{n} p = \frac{1}{n}(np) = p. = \theta.$

where $p = \theta$ in Bernoulli
$1-p = 1-\theta$ distribution

得證 $E(\hat{\theta}) = \theta$ ※

~~(C) $E(\hat{\theta}) = E(\max\{x_1 \dots x_n\})$ → 最大值為 M → $E(\hat{\theta}) = M \neq \theta$~~ ※
~~不是 (∵M>N)~~

(D) $\hat{\theta} = \frac{1}{N}\sum_{1}^{N} x_n^2 \to E(\hat{\theta}) = E(\frac{1}{N}\sum_{1}^{N} x_n^2) = \frac{1}{n}E(\sum x_n^2) \longleftrightarrow \theta = \sigma^2$ (定義)

$$= \frac{1}{n}\sum_{1}^{n} \underbrace{E(x_n^2)}_{\sigma^2} = \frac{1}{n}\sum_{1}^{n}\sigma^2 = \sigma^2 = \theta.$$ ※

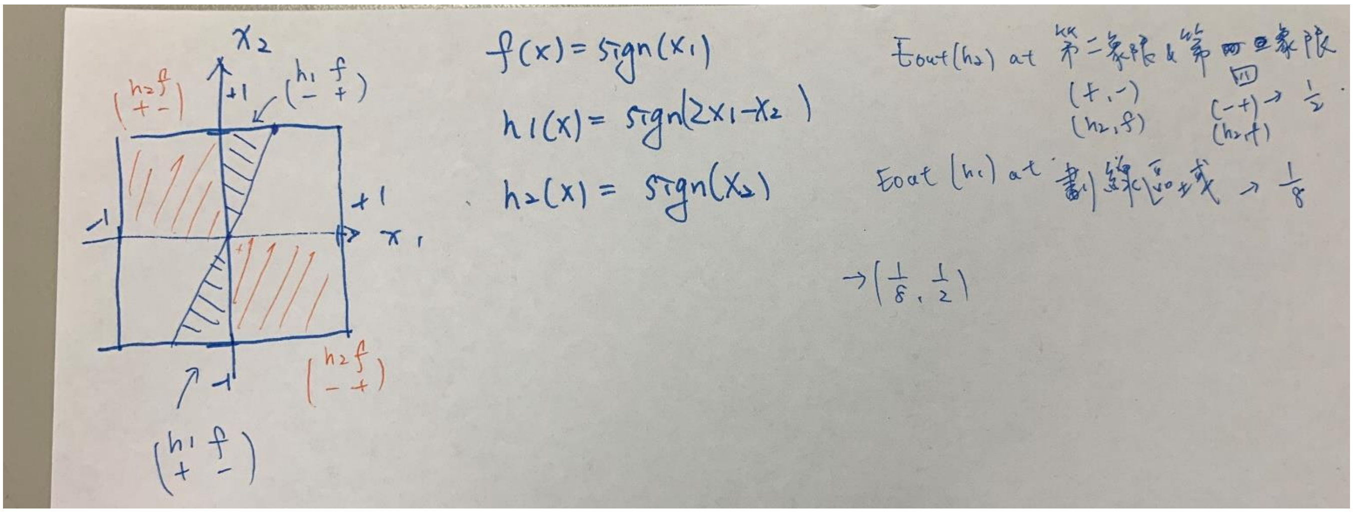
(C) $E(\hat{\theta}) = E(\max\{x_1 \dots x_n\})$

假設此一事件為擲骰子 其 P 滿足 M=6. $P(x) = \frac{1}{6}$ for $x(1 \sim 6)$.

設 N=3次 即骰子擲3個

→所求期望值 $= 1\times\frac{1^3}{6^3} + 2\times\frac{2^3-1^3}{6^3} + 3\times\frac{3^3-2^3}{6^3} + \dots 6\times\frac{6^3-5^3}{6^3} = \frac{119}{24} = 4\frac{23}{24} \neq 6$

10. (A)



$f(x) = \text{sign}(x_1)$

$h_1(x) = \text{sign}(2x_1 - x_2)$

$h_2(x) = \text{sign}(x_2)$

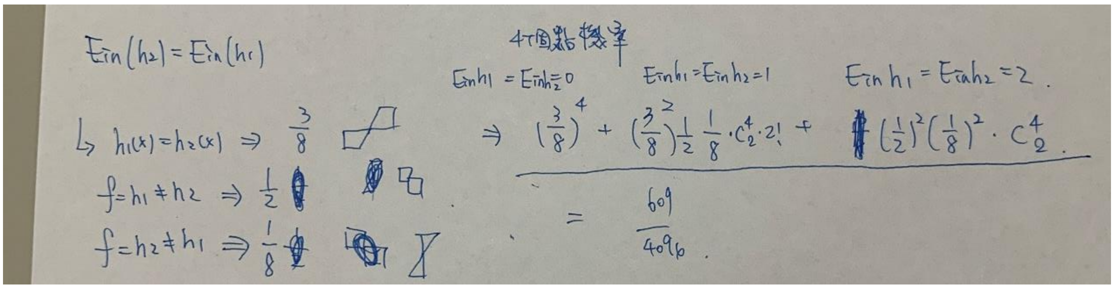
Eout(h_2) at 第二象限&第~~四~~三象限
四

(+,−) (−+) → $\frac{1}{2}$.
(h_2,f) (h_2,f)

Eout(h_1) at 劃線區域 → $\frac{1}{8}$

→ $(\frac{1}{8}, \frac{1}{2})$

11. (B)



Ein(h_2) = Ein(h_1)

↳ $h_1(x) = h_2(x) \Rightarrow \frac{3}{8}$

$f = h_1 \neq h_2 \Rightarrow \frac{1}{2}$.

$f = h_2 \neq h_1 \Rightarrow \frac{1}{8}$

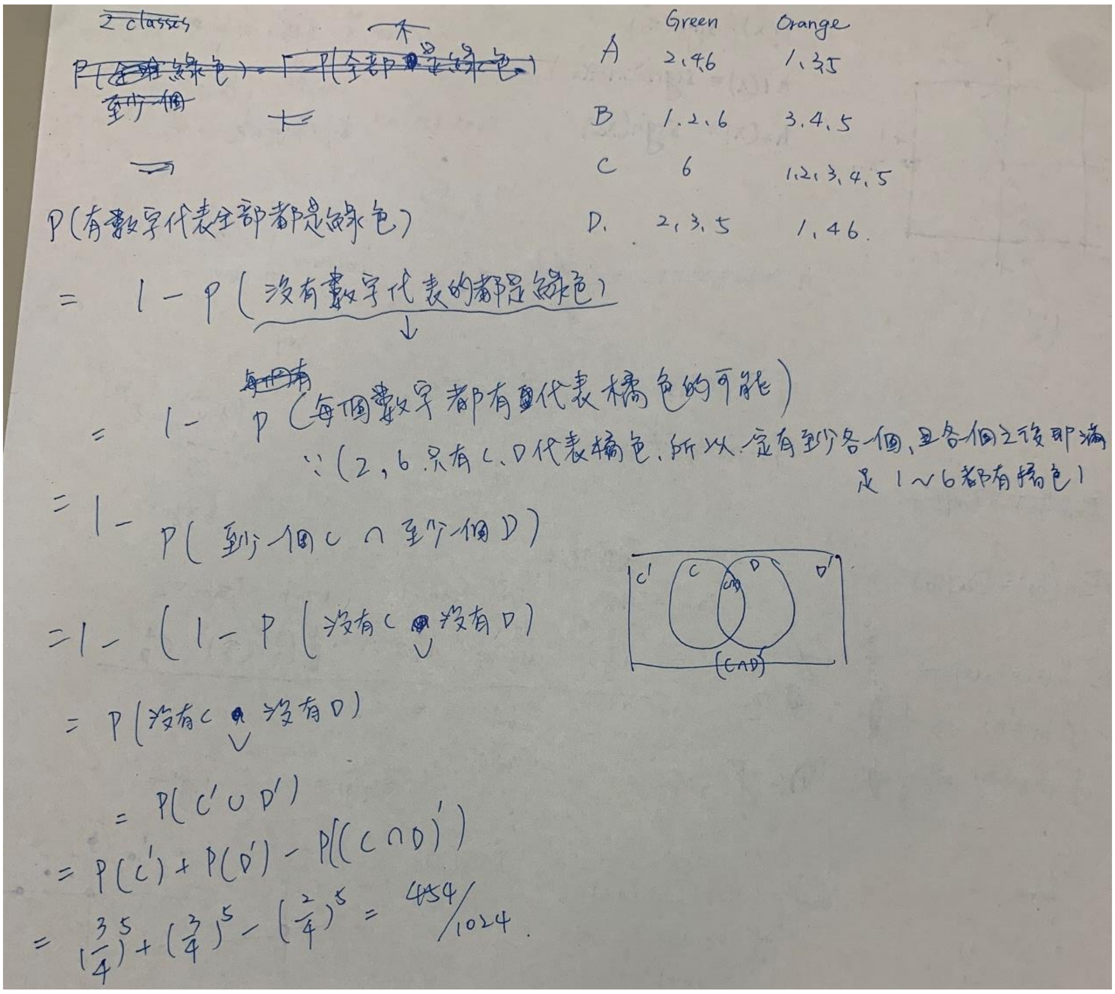
4個點機率

Einh_1 = Einh_2 = 0　Einh_1 = Einh_2 = 1　Einh_1 = Einh_2 = 2.

$$\Rightarrow \left(\frac{3}{8}\right)^4 + \left(\frac{3}{8}\right)^2 \frac{1}{2} \frac{1}{8} \cdot C^4_2 \cdot 2! + \left(\frac{1}{2}\right)^2 \left(\frac{1}{8}\right)^2 \cdot C^4_2.$$

$$= \frac{609}{4096}.$$

12. (B)



~~2 classes~~

~~P(全部綠色) = 1 − P(全部都是綠色)~~
~~至少一個~~

	Green	Orange
A	2, 4, 6	1, 3, 5
B	1, 2, 6	3, 4, 5
C	6	1, 2, 3, 4, 5
D.	2, 3, 5	1, 4, 6.

P(有數字代表全部都是綠色)

$= 1 - P(\underbrace{\text{沒有數字代表的都是綠色}}_{\downarrow})$

~~每個有~~

$= 1 - P(\text{每個數字都有代表橘色的可能})$

∵(2, 6 只有C、D代表橘色，所以一定有至少各一個，且各一個之後即滿足 1～6 都有橘色)

$= 1 - P(\text{至少一個}C \cap \text{至少一個}D)$

$= 1 - (1 - P(\text{沒有}C \cup \text{沒有}D)$

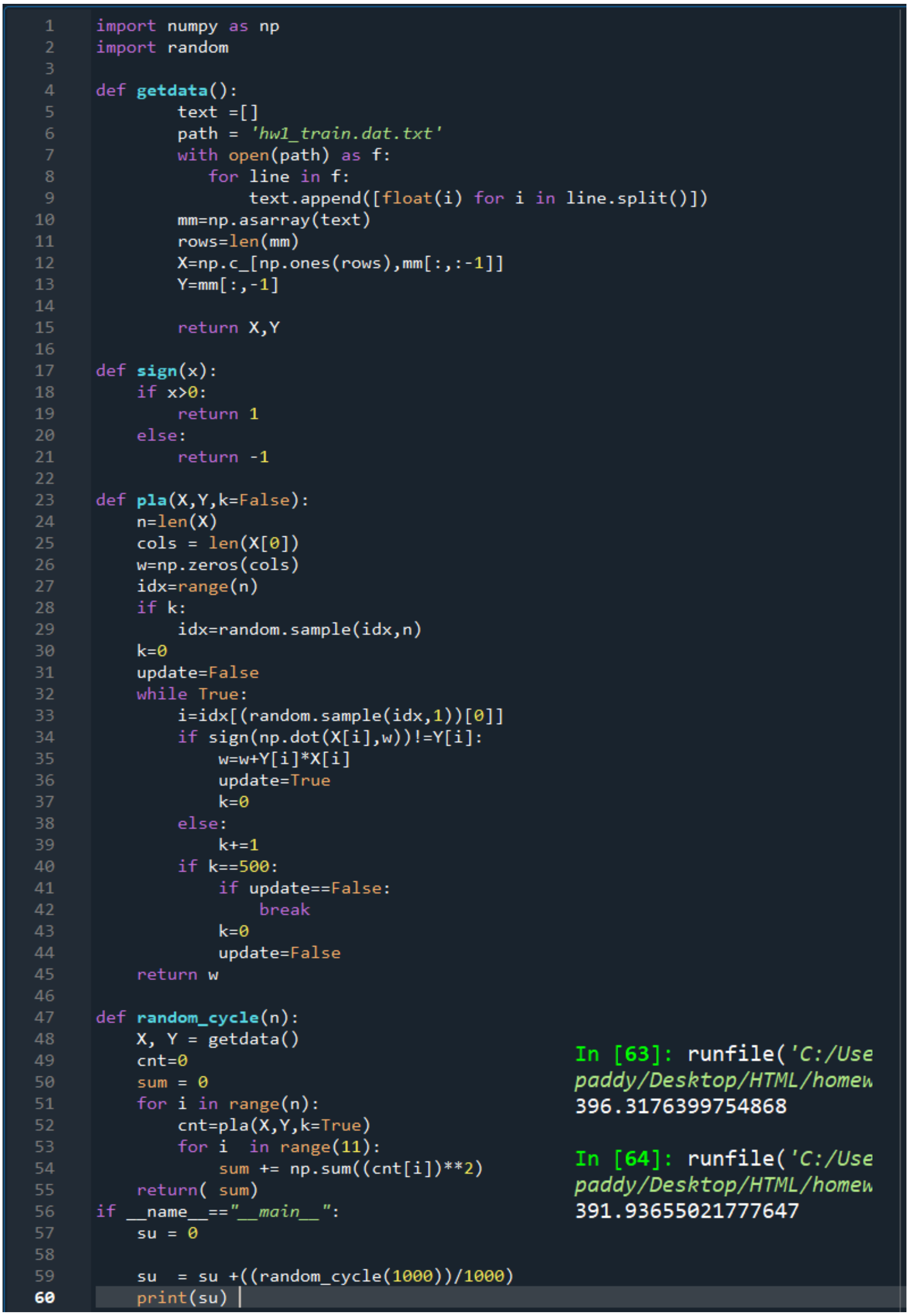
$= P(\text{沒有}C \cup \text{沒有}D)$

$= P(C' \cup D')$

$= P(C') + P(D') - P((C \cap D)')$

$= (\frac{3}{4})^5 + (\frac{3}{4})^5 - (\frac{2}{4})^5 = 454/1024.$

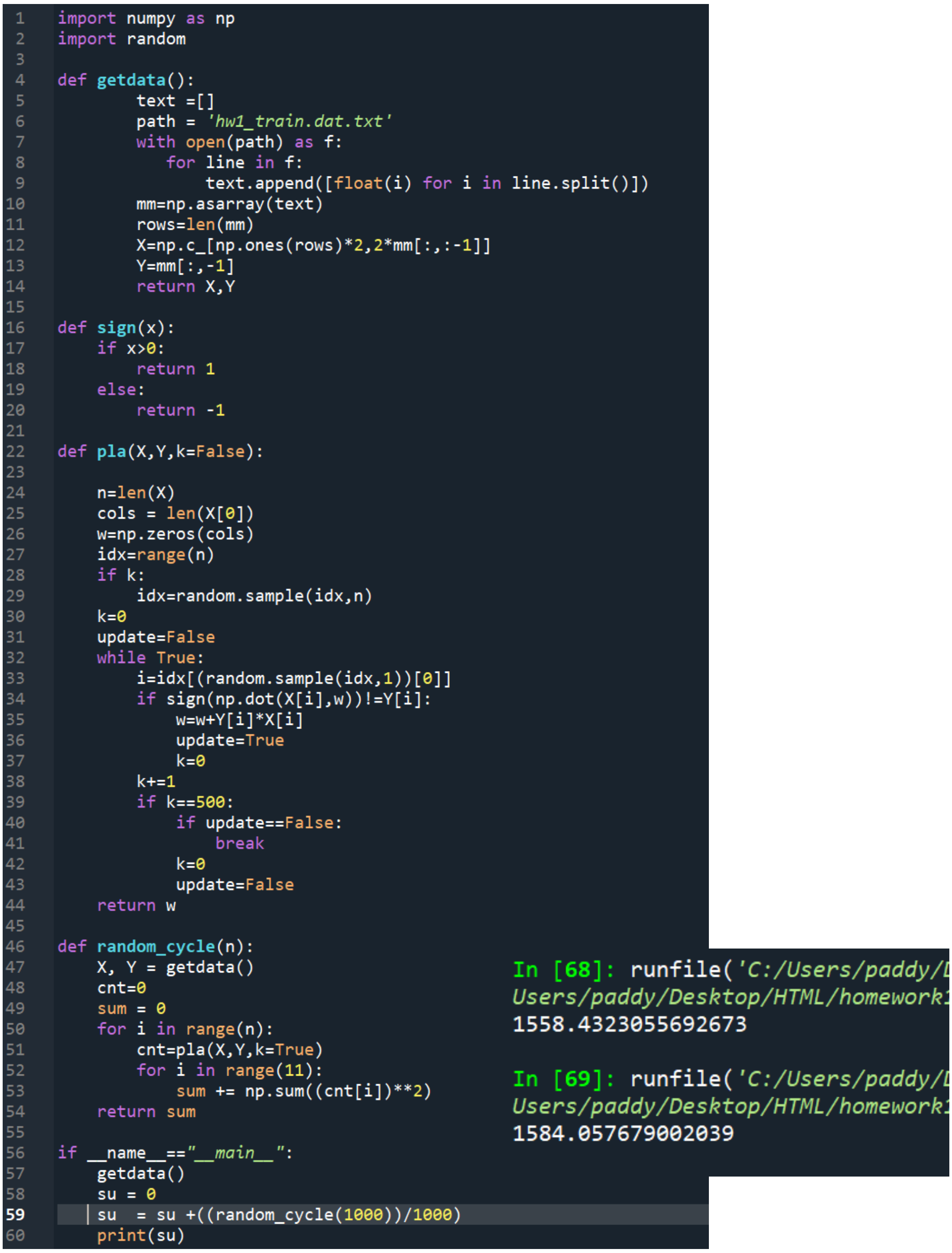
13.　(B)390



```
import numpy as np
import random

def getdata():
        text =[]
        path = 'hw1_train.dat.txt'
        with open(path) as f:
           for line in f:
               text.append([float(i) for i in line.split()])
        mm=np.asarray(text)
        rows=len(mm)
        X=np.c_[np.ones(rows),mm[:,:-1]]
        Y=mm[:,-1]

        return X,Y

def sign(x):
    if x>0:
        return 1
    else:
        return -1

def pla(X,Y,k=False):
    n=len(X)
    cols = len(X[0])
    w=np.zeros(cols)
    idx=range(n)
    if k:
        idx=random.sample(idx,n)
    k=0
    update=False
    while True:
        i=idx[(random.sample(idx,1))[0]]
        if sign(np.dot(X[i],w))!=Y[i]:
            w=w+Y[i]*X[i]
            update=True
            k=0
        else:
            k+=1
        if k==500:
            if update==False:
                break
            k=0
            update=False
    return w

def random_cycle(n):
    X, Y = getdata()
    cnt=0
    sum = 0
    for i in range(n):
        cnt=pla(X,Y,k=True)
        for i  in range(11):
            sum += np.sum((cnt[i])**2)
    return( sum)
if __name__=="__main__":
    su = 0

    su  = su +((random_cycle(1000))/1000)
    print(su)
```

```
In [63]: runfile('C:/Use
paddy/Desktop/HTML/homew
396.3176399754868

In [64]: runfile('C:/Use
paddy/Desktop/HTML/homew
391.93655021777647
```

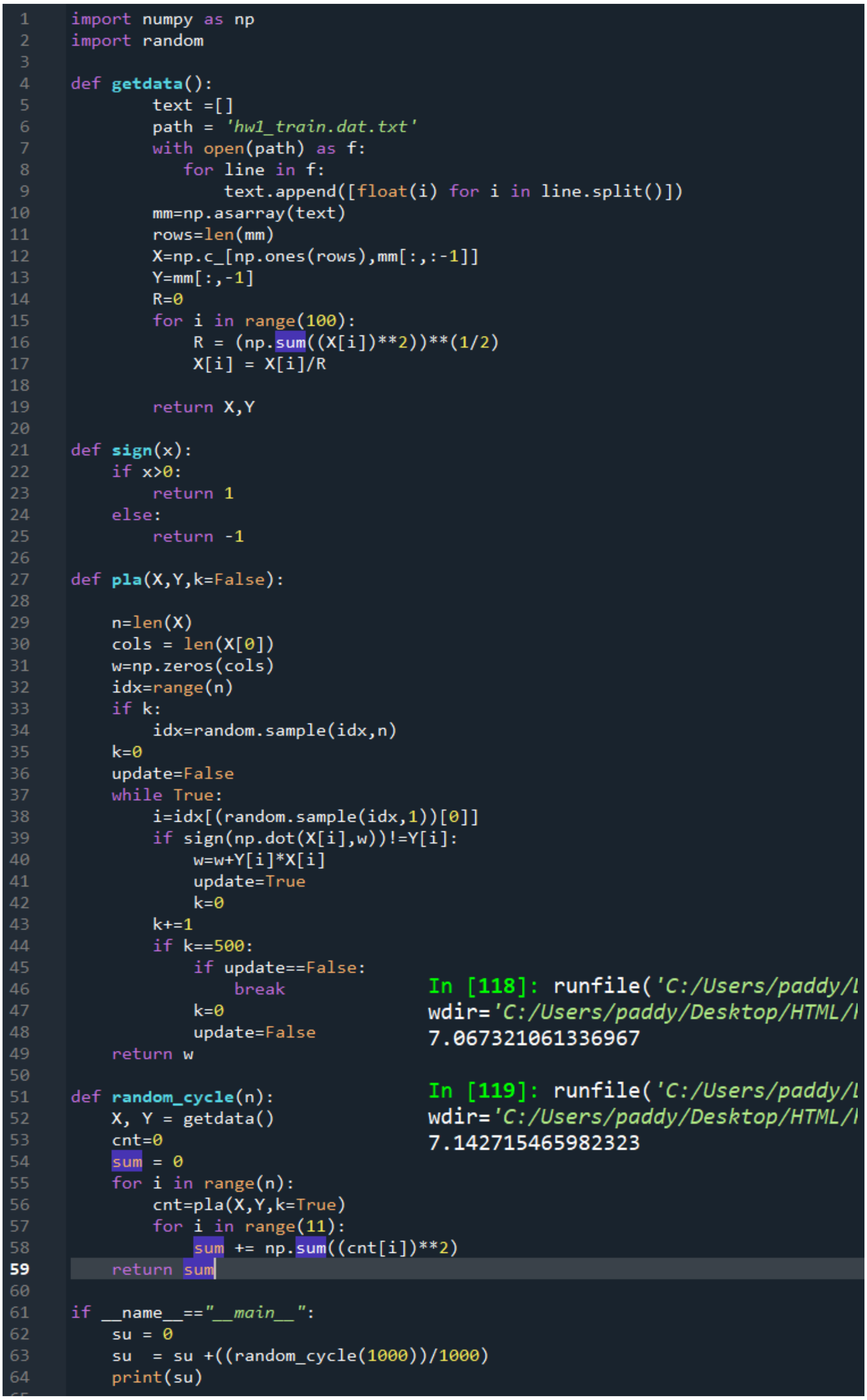
14.　(C)1560



```python
import numpy as np
import random

def getdata():
        text =[]
        path = 'hw1_train.dat.txt'
        with open(path) as f:
           for line in f:
               text.append([float(i) for i in line.split()])
        mm=np.asarray(text)
        rows=len(mm)
        X=np.c_[np.ones(rows)*2,2*mm[:,:-1]]
        Y=mm[:,-1]
        return X,Y

def sign(x):
    if x>0:
        return 1
    else:
        return -1

def pla(X,Y,k=False):

    n=len(X)
    cols = len(X[0])
    w=np.zeros(cols)
    idx=range(n)
    if k:
        idx=random.sample(idx,n)
    k=0
    update=False
    while True:
        i=idx[(random.sample(idx,1))[0]]
        if sign(np.dot(X[i],w))!=Y[i]:
            w=w+Y[i]*X[i]
            update=True
            k=0
        k+=1
        if k==500:
            if update==False:
                break
            k=0
            update=False
    return w

def random_cycle(n):
    X, Y = getdata()
    cnt=0
    sum = 0
    for i in range(n):
        cnt=pla(X,Y,k=True)
        for i in range(11):
            sum += np.sum((cnt[i])**2)
    return sum

if __name__=="__main__":
    getdata()
    su = 0
    su  = su +((random_cycle(1000))/1000)
    print(su)
```

```
In [68]: runfile('C:/Users/paddy/
Users/paddy/Desktop/HTML/homework
1558.4323055692673

In [69]: runfile('C:/Users/paddy/
Users/paddy/Desktop/HTML/homework
1584.057679002039
```

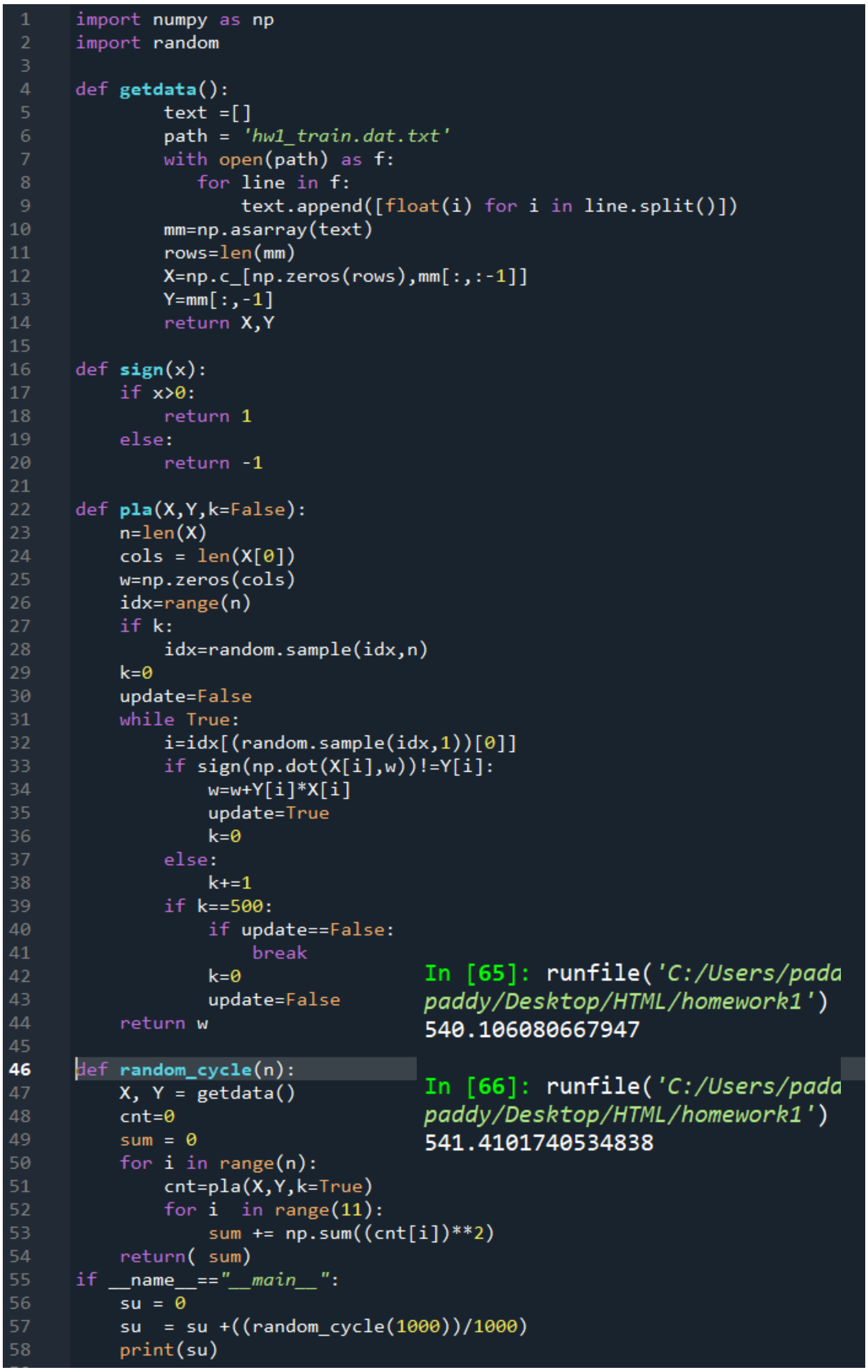
15. (E)7.1



```
import numpy as np
import random

def getdata():
        text =[]
        path = 'hw1_train.dat.txt'
        with open(path) as f:
           for line in f:
               text.append([float(i) for i in line.split()])
        mm=np.asarray(text)
        rows=len(mm)
        X=np.c_[np.ones(rows),mm[:,:-1]]
        Y=mm[:,-1]
        R=0
        for i in range(100):
            R = (np.sum((X[i])**2))**(1/2)
            X[i] = X[i]/R

        return X,Y

def sign(x):
    if x>0:
        return 1
    else:
        return -1

def pla(X,Y,k=False):

    n=len(X)
    cols = len(X[0])
    w=np.zeros(cols)
    idx=range(n)
    if k:
        idx=random.sample(idx,n)
    k=0
    update=False
    while True:
        i=idx[(random.sample(idx,1))[0]]
        if sign(np.dot(X[i],w))!=Y[i]:
            w=w+Y[i]*X[i]
            update=True
            k=0
        k+=1
        if k==500:
            if update==False:
                break
            k=0
            update=False
    return w

def random_cycle(n):
    X, Y = getdata()
    cnt=0
    sum = 0
    for i in range(n):
        cnt=pla(X,Y,k=True)
        for i in range(11):
            sum += np.sum((cnt[i])**2)
    return sum

if __name__=="__main__":
    su = 0
    su  = su +((random_cycle(1000))/1000)
    print(su)
```

```
In [118]: runfile('C:/Users/paddy/l
wdir='C:/Users/paddy/Desktop/HTML/l
7.067321061336967

In [119]: runfile('C:/Users/paddy/l
wdir='C:/Users/paddy/Desktop/HTML/l
7.142715465982323
```

16. (A)530



```
import numpy as np
import random

def getdata():
        text =[]
        path = 'hw1_train.dat.txt'
        with open(path) as f:
           for line in f:
               text.append([float(i) for i in line.split()])
        mm=np.asarray(text)
        rows=len(mm)
        X=np.c_[np.zeros(rows),mm[:,:-1]]
        Y=mm[:,-1]
        return X,Y

def sign(x):
    if x>0:
        return 1
    else:
        return -1

def pla(X,Y,k=False):
    n=len(X)
    cols = len(X[0])
    w=np.zeros(cols)
    idx=range(n)
    if k:
        idx=random.sample(idx,n)
    k=0
    update=False
    while True:
        i=idx[(random.sample(idx,1))[0]]
        if sign(np.dot(X[i],w))!=Y[i]:
            w=w+Y[i]*X[i]
            update=True
            k=0
        else:
            k+=1
        if k==500:
            if update==False:
                break
            k=0
            update=False
    return w

def random_cycle(n):
    X, Y = getdata()
    cnt=0
    sum = 0
    for i in range(n):
        cnt=pla(X,Y,k=True)
        for i  in range(11):
            sum += np.sum((cnt[i])**2)
    return( sum)
if __name__=="__main__":
    su = 0
    su  = su +((random_cycle(1000))/1000)
    print(su)
```

```
In [65]: runfile('C:/Users/pada
paddy/Desktop/HTML/homework1')
540.106080667947

In [66]: runfile('C:/Users/pada
paddy/Desktop/HTML/homework1')
541.4101740534838
```