Machine Learning Hw 3　吳偉樂 R09946023

1) (b) 30

$$E_D[E_{in}(w_{lin})] = \sigma^2(1 - \frac{d+1}{N}) > 0.006 \quad , \sigma = 0.1 , d = 11$$

$$0.1^2(1 - \frac{11+1}{N}) > 0.006 \;\rightarrow\; \frac{12}{N} < 0.4 \;\rightarrow\; N > 30$$

2) (a) There exists at least one solution for the normal equation.

For this normal equation, $\mathbf{w} = (\mathbf{X}^\mathbf{T}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^\mathbf{T}\mathbf{y}$

By Linear Algebra,

If $\boldsymbol{X^T X}$ is invertible, then **w** has unique one solution

If $\boldsymbol{X^T X}$ is not invertible (i.e Singular), then **w** has many optimal solutions

So there exists at least one solution for the normal equation .

3) (c)

3. (c) multiplying each of the n-th row by $\frac{1}{n}$

Multiplying each of the n-th row by $\frac{1}{n}$

$\Leftrightarrow$ $\exists$ square matrix $\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & & \vdots \\ \vdots & & \ddots & \\ 0 & & & \frac{1}{n} \end{bmatrix}_{n \times n} = X_i$ s.t. $\underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & & \vdots \\ \vdots & & \ddots & \\ 0 & & & \frac{1}{n} \end{bmatrix}_{n \times n}}_{X_i} \underbrace{\begin{bmatrix} - x_1 - \\ \vdots \\ - x_n - \end{bmatrix}_{n \times d}}_{X_A} = \begin{bmatrix} - x_1 - \\ - \frac{1}{2}x_2 - \\ \vdots \\ - \frac{1}{n}x_n - \end{bmatrix}$

Note : X_i is invertible since it only has values on diagonal .

Given $H = X_A(X_A^T X_A)^{-1}X_A^T$, where $X_A^T X_A$ invertible

After scaling , does $(X_i X_A)[(X_i X_A)^T (X_i X_A)]^{-1}(X_i X_A)^T = X_A(X_A^T X_A)X_A^T$?

$\Rightarrow$ $X_i X_A[(X_i X_A)^T(X_i X_A)]^{-1}(X_i X_A)^T = X_i X_A[X_A^T X_i^T X_i X_A]^{-1} X_A^T X_i^T \leftarrow X_i^T = X_i$

$= X_i X_A [X_A^T X_i^2 X_A]^{-1} X_A^T X_i^T$

Case ①: If X_A invertible, then by Linear Algebra

$X_i X_A[X_A^T X_i X_i X_A]^{-1} X_A^T X_i^T = X_i X_A X_A^{-1} X_i^{-1} X_i^{-1} (X_A^T)^{-1} X_A^T X_i^T$

$= X_i X_i^{-1} X_i^{-1} X_i = I = X_A(X_A^T X_A)^{-1} X_A^T$

$= X_A X_A^{-1}(X_A^T)^{-1}X_A^T$

Case ②: If X_A not invertible, then we have

$X_i X_A[X_A^T X_i X_i X_A]^{-1} X_A^T X_i \neq X_A[X_A^T X_A]^{-1} X_A^T$

$\Rightarrow$. By ②, the answer in c .

※ .

4) (e) 4

4. (e) 4

① Hoeffding ineq : $P(|\bar{X}-E(\bar{X})|>\varepsilon) \le 2e^{-2\varepsilon^2 N}$

Note : $\nu = \frac{1}{N}\sum_{n=1}^{N} y_n = \bar{y}$ $\quad \because y_1,\dots,y_N \overset{iid}{\sim} \text{Bernoulli}(N,\theta)$

$E(\nu) = E(\frac{1}{N}\sum_{n=1}^{N} y_n) = \frac{1}{N}(E(y_1)+\dots+E(y_N)) = \frac{1}{N}\cdot N E(y_1) = E(y_1) = \theta$

$\therefore P(|\bar{X}-E(\bar{X})|>\varepsilon) = P(|\nu - E(\nu)|>\varepsilon) = \underline{P(|\nu-\theta|>\varepsilon) \le 2e^{-2\varepsilon^2 N}}$

② Let $L(y|\theta) = \prod_{i=1}^{N} P(y_i|\theta) = \prod_{i=1}^{N} \theta^{y_i}(1-\theta)^{1-y_i} = \theta^{\sum y_i}(1-\theta)^{N-\sum y_i}$

$l(y|\theta) = \log L(y|\theta) = \log[\theta^{\sum y_i}(1-\theta)^{N-\sum y_i}] = \sum_{i=1}^{N} y_i \log\theta + (N-\sum_{i=1}^{N} y_i)\log(1-\theta)$

Maixmize L $\Leftrightarrow$ maximize l

$\therefore \frac{dl}{d\theta} = \frac{\sum y_i}{\theta} - \frac{N-\sum y_i}{1-\theta} = 0 \Rightarrow \sum_{i=1}^{N} y_i - \theta\sum_{i=1}^{N} y_i = N\theta - \sum_{i=1}^{N} y_i\,\theta \Rightarrow \underline{\hat\theta = \sum_{i=1}^{N} y_i \frac{1}{N} = \nu}$

$\frac{d^2 l}{d\theta^2} = -\frac{\sum y_i}{\theta^2} - \frac{N-\sum y_i}{(1-\theta)^2} < 0 \Rightarrow l$ has max. value at $\frac{dl}{d\theta}=0$

$\therefore \hat\theta = \nu$ means that ν maximizes likelihood$(\hat\theta)$ over $\hat\theta \in [0,1]$

③ $E_{in}(\hat y) = \frac{1}{N}\sum_{n=1}^{N}(\hat y - y_n)^2$

$\frac{dE}{d\hat y} = \frac{2}{N}\sum_{n=1}^{N}(\hat y - y_n) = 0 \Rightarrow N\hat y = N\nu \Rightarrow \underline{\hat y = \nu}$

$\frac{d^2E}{d\hat y^2} = 2 > 0 \Rightarrow E$ has min. vale at $\frac{dE}{d\hat y}=0$

$\therefore \nu$ minimized $E_{in}(\hat y) = \frac{1}{N}\sum_{n=1}^{N}(\hat y - y_n)^2$ over all $\hat y \in \mathbb{R}$

④ $\nabla E_{in}(\hat y) = \frac{dE}{d\hat y} = \frac{2}{N}\sum_{n=1}^{N}(\hat y - y_n)$

$-\nabla E_{in}(\hat y)\big|_{\hat y=0} = -\frac{2}{N}\sum_{n=1}^{N}(-y_n) = \frac{2}{N}\sum_{n=1}^{N}(y_n) = 2\nu$

$\therefore 2\nu$ is the negative gradient direction $-\nabla E_{in}(\hat y)$ at $\hat y = 0$

5) (a) $(\frac{1}{\theta})^N$

5. (a) $(\frac{1}{\theta})^N$

$I(0\le y_i \le\theta) = \begin{cases}1, & 0\le y_i\le\theta \\ 0, & o.w\end{cases}$

$y_1,\dots,y_N \overset{iid}{\sim} U(0,\theta) \Rightarrow f(y_i) = \begin{cases}\frac{1}{\theta}, & 0\le y_i\le\theta \\ 0, & o.w\end{cases}$

$L(y|\theta) = \prod_{i=1}^{N} f(y_i|\theta) = (\frac{1}{\theta})^N \prod_{i=1}^{N} I(0\le y_i\le\theta)$, where $0\le y_1,y_2,\dots,y_N \le \theta$ $\Leftrightarrow \max(y_1,\dots,y_N)\le\theta$

$\therefore$ For any $\hat\theta \ge \max(y_1,\dots,y_N)$, its likelihood is $(\frac{1}{\hat\theta})^N$

Additional : MLE.

Since $(\frac{1}{\theta})$ is decreasing function, $\therefore \max(f(y_i)) = \max\{y_1,\dots,y_N\}$

$\Rightarrow$ MLE is $\max\{y_1,\dots,y_N\}$

6) (b) $err(w,x,y) = max(0,-yw^Tx)$

6. (b) $err(W,X,y) = max(0,-yw^TX)$

$W_{t+1} = W_t + \frac{1}{N}\sum_{n=1}^{N} [\![sign(W_t^TX_n) \neq y_n]\!] y_nX_n$

Set $y_n = 1$, if $sign(W_t^TX_n) = 1$, then $W_t^TX_n > 0 \Rightarrow y_nW_t^TX_n > 0 \Rightarrow error = 0$

$y_n = -1$, if $sign(W_t^TX_n) = -1$, then $W_t^TX_n < 0 \Rightarrow y_nW_t^TX_n > 0 \Rightarrow Error = 0$

$y_n = 1$, if $sign(W_t^TX_n) = -1$, then $W_t^TX_n < 0 \Rightarrow y_nW_n^TX_n < 0 \Rightarrow error = y_nW_n^TX_n$

$y_n = -1$, if $sign(W_t^TX_n) = 1$, then $W_t^TX_n > 0 \Rightarrow y_nW_n^TX_n < 0 \Rightarrow error = y_nW_n^TX_n$

Combine the statements above, we have $error(W,X,y) = max(0, -y_nW_n^TX_n)$

7) (a) $+y_nx_nexp(-y_nw^Tx_n)$

7. (a) $+y_nx_n e^{-y_nw^Tx_n}$

$err_{exp}(w,x,y) = e^{-yw^Tx} = e^{-y(w_0x_0+\dots+w_nx_n)}$

$$\nabla err_{exp}(w,x,y) = \begin{pmatrix} \frac{derr}{dw_0} \\ \frac{derr}{dw_1} \\ \vdots \\ \frac{derr}{dw_n} \end{pmatrix} = \begin{pmatrix} -ye^{-yw^Tx}x_0 \\ -ye^{-yw^Tx}x_1 \\ \vdots \\ -ye^{-yw^Tx}x_n \end{pmatrix} = -ye^{-yw^Tx}\begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix} = -ye^{-yw^Tx}X$$

$\Rightarrow -\nabla err_{exp}(w,x,y) = yxe^{-(yw^Tx)}$

8) (b) $-(A_E(u))^{-1}b_E(u)$

8. (b) $-(A_E(u))^{-1}b_E(u)$

$w \leftarrow u+v \Rightarrow v = w-u$

$\Rightarrow E(w) \simeq E(u) + b_E(u)^T \cdot v + \frac{1}{2}v^TA_E(u)v$

$\nabla_v E_{in}(w) = b_E(u)^T + \frac{1}{2}\cdot 2A_E(u)v = b_E(u)^T + vA_E(u) = 0$

$\Rightarrow vA_E(u) = -b_E(u)^T \Rightarrow v = -\underset{1\times d}{b_E(u)^T}\cdot \underset{d\times d}{(A_E(u))^{-1}}$

$A_E(u)$ is symmetric, $\therefore A_E(u)^{-1}$ is also symmetric (By Linear Algebra)

$\Rightarrow V = -\underbrace{b_E(u)^T(A_E(u))^{-1}}_{1\times d} = -[A_E(u)^{-1}]^T b_E(u)^{TT} = -\underbrace{(A_E(u))^{-1}b_E(u)}_{d\times 1}$

same elements but different interpretation method : $(\cdots)$ or $\begin{pmatrix} \vdots \end{pmatrix}$

9) (b)

9. (b) $\frac{2}{N}X^TX$

$E = E_{in} = \frac{1}{N}\sum_{i=1}^{N}(w^Tx_n - y_n)^2$, where $x_n = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1d}\end{pmatrix}$, $X = \begin{bmatrix} - x_1^T - \\ \vdots \\ - x_N^T - \end{bmatrix}_{N\times d}$

$\Rightarrow E_{in} = \frac{1}{N}\Big[(w_1x_{11} + w_2x_{12} + \ldots + w_d x_{1d} - y_1)^2 + (w_1x_{21} + w_2x_{22} + \ldots + w_d x_{2d} - y_2)^2 + \ldots$
$+ (w_1x_{N1} + w_2x_{N2} + \ldots + w_d x_{Nd} - y_n)^2\Big]$

$\Rightarrow \frac{dE_{in}}{dw_1} = \frac{2}{N}\Big[(w_1x_{11} + \ldots + w_d x_{1d} - y_1)x_{11} + \ldots + (w_1x_{N1} + w_2x_{N2} + \ldots + w_d x_{Nd} - y_n)x_{N1}\Big]$

$\Rightarrow \frac{dE_{in}}{dw_k} = \frac{2}{N}\Big[(w_1x_{11} + \ldots + w_d x_{1d} - y_1)x_{1k} + \ldots + (w_1x_{N1} + w_2x_{N2} + \ldots + w_d x_{Nd} - y_n)x_{Nk}\Big]$ for $k = 1, 2, \ldots, d$

$\frac{d^2E_{in}}{dw_k^2} = \frac{2}{N}\Big[x_{1k}^2 + x_{2k}^2 + \ldots + x_{Nk}^2\Big]$ for $k = 1, \ldots, d$

$\frac{d^2E_{in}}{dw_k w_j} = \frac{2}{N}\Big[x_{1j}x_{1k} + x_{2j}x_{2k} + \ldots + x_{Nj}x_{Nk}\Big]$ for $\begin{matrix} k = 1, \ldots, d \\ j = 1, \ldots, d\end{matrix}$, $j \neq k$

$$\therefore A_E(w) = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2}(w) & \frac{\partial^2 E}{\partial w_1 \partial w_2}(w) & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_d}(w) \\ \vdots & & & \\ \frac{\partial^2 E}{\partial w_d \partial w_1}(w) & \frac{\partial^2 E}{\partial w_d \partial w_2}(w) & \cdots & \frac{\partial^2 E}{\partial w_d^2}(w)\end{bmatrix}_{d\times d}$$

$$= \frac{2}{N}\begin{bmatrix} x_{11}^2 + x_{21}^2 + \ldots + x_{N1}^2 & x_{12}x_{11} + x_{22}x_{21} + \ldots + x_{N2}x_{N1} & \cdots & x_{1d}x_{11} + x_{2d}x_{21} + \ldots + x_{Nd}x_{N1} \\ \vdots & & \ddots & \\ x_{1d}x_{11} + \ldots + x_{Nd}x_{N1} & x_{1d}x_{12} + x_{2d}x_{22} + \ldots + x_{Nd}x_{N2} & \cdots & x_{1d}^2 + x_{2d}^2 + \ldots + x_{Nd}^2\end{bmatrix}$$

$$= \frac{2}{N}\begin{bmatrix} x_{11} & x_{21} & \cdots & x_{N1} \\ x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & & & \vdots \\ x_{1d} & x_{2d} & \cdots & x_{Nd}\end{bmatrix}_{d\times N}\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & & \\ x_{N1} & x_{N2} & \cdots & x_{Nd}\end{bmatrix}_{N\times d} = \underline{\underline{\frac{2}{N}X^TX}}$$

10) (b)

10. (b) $(h_k(x) - [\![y=k]\!])x_i$

$W = \begin{bmatrix} | & | & & | & & | \\ w_1 & w_2 & \cdots & w_k & \cdots & w_K \\ | & | & & | & & | \end{bmatrix}_{(d+1)\times K} = \begin{bmatrix} w_{11} & & w_{1k} & & w_{1K} \\ w_{21} & \cdots & w_{2k} & \cdots & w_{2K} \\ w_{i1} & & \boxed{w_{ik}} & & w_{iK} \\ w_{d+1,1} & & w_{d+1,k} & & w_{d+1,K}\end{bmatrix}$, $h_y(x) = \frac{e^{w_y^Tx}}{\sum_{i=1}^{K} e^{w_i^Tx}}$

$err(W, x, y) = -\ln h_y(x) = -\sum_{j=1}^{K}[\![y=j]\!]\ln h_j(x)$

given x & y

$= -\sum_{j=1}^{K}[\![y=j]\!]\ln \frac{e^{w_j^Tx}}{\sum_{i=1}^{K} e^{w_i^Tx}}$

$= -[\![y=1]\!]\ln\frac{e^{w_1^Tx}}{\sum_{i=1}^{K} e^{w_i^Tx}} - \ldots - [\![y=k]\!]\ln\frac{e^{w_k^Tx}}{\sum_{i=1}^{K} e^{w_i^Tx}} - \ldots - [\![y=K]\!]\ln\frac{e^{w_K^Tx}}{\sum_{i=1}^{K} e^{w_i^Tx}}$

$\frac{\partial\, err(W,x,y)}{\partial W_{ik}} = -[\![y=1]\!]\cdot\frac{1}{h_1(x)}\cdot\frac{-e^{w_1^Tx}\cdot e^{w_k^Tx}x_i}{(\sum_{i=1}^{K} e^{w_i^Tx})^2} - \ldots - [\![y=k]\!]\cdot\frac{1}{h_k(x)}\cdot\frac{\sum_{i=1}^{K} e^{w_i^Tx}\cdot e^{w_k^Tx}x_i - e^{w_k^Tx}\cdot e^{w_k^Tx}x_i}{(\sum_{i=1}^{K} e^{w_i^Tx})^2}$
$- \ldots - [\![y=K]\!]\cdot\frac{1}{h_K(x)}\frac{-e^{w_K^Tx}\cdot e^{w_k^Tx}x_i}{(\sum_{i=1}^{K} e^{w_i^Tx})^2}$

$= [\![y=1]\!]\frac{1}{h_1(x)}\cdot h_1(x)h_k(x)x_i + \ldots + [\![y=k]\!]\frac{1}{h_k(x)}\Big[-h_k(x) + h_k(x)h_k(x)\Big]x_i + \ldots$
$+ [\![y=K]\!]\frac{1}{h_K(x)}\cdot h_K(x)h_k(x)x_i$

$= \sum_{j=1}^{K}[\![y=j]\!]h_k(x)x_i - [\![y=k]\!]x_i$

$= \Big[\sum_{j=1}^{K}[\![y=j]\!]h_k(x) - [\![y=k]\!]\Big]x_i$

$j=k$ $= \Big[h_k(x) - [\![y=k]\!]\Big]x_i$ ※.

11) (e)

11. (e) $w_2^* - w_1^*$

when K=2, error in MLR is equivalent to error in logistic reg.

$\therefore \ err(w,x,y)|_{MLR} = -\ln h_y(x) = err(w,x,y)|_{LR} = -\ln \theta(yw^T x)$

$\Rightarrow h_y(x) = \theta(yw^T x) = \frac{1}{1+e^{-yw^T x}} \Leftrightarrow h_{y_n}(x_n) = \theta(y_n w^T x_n)$

relable :

$k = y_n = 1 \Rightarrow y_n' = -1$

$\therefore h_1(x_n) = \frac{e^{w_1^{*T} x_n}}{e^{w_1^{*T} x_n} + e^{w_2^{*T} x_n}} = \frac{1}{1+e^{(w_2^{*T} - w_1^{*T}) x_n}} = \theta(y_n' w^T x_n) = \frac{1}{1+e^{w^T x}}$

$\Rightarrow w^T = w_2^{*T} - w_1^{*T} \Rightarrow w^T = (w_2^* - w_1^*)^T \Rightarrow \underline{\underline{w = w_2^* - w_1^*}}$

$k = 2 = y_n \Rightarrow y_n' = 1$

$\therefore h_2(x_n) = \frac{e^{w_2^{*T} x_n}}{e^{w_2^{*T} x_n} + e^{w_1^{*T} x_n}} = \frac{1}{1+e^{(w_1^{*} - w_2^{*})^T x_n}} = \theta(y_n' w^T x_n) = \frac{1}{1+e^{-w^T x_n}}$

$\Rightarrow -w^T = (w_1^* - w_2^*)^T \Leftrightarrow \underline{\underline{w = w_2^* - w_1^*}}$

12) (e) [-7,0,0,2,-2,3]

Draw each 5 curve on the graph , and mark those points, obviously is (e)

13) (b)

13. $(x_1, .., x_d) \in \mathbb{R}^d \xrightarrow[\Phi_{(k)}]{} (1, x_k) \in \mathbb{R}^2$ ← Each $\Phi_{(k)}$ is a decision stump.

$\Phi_{(1)} : (1, x_1) \Rightarrow w_0 + w_1 x_1 \Rightarrow m_{H_1}(N) = 2N$

$\Phi_{(2)} : (1, x_2) \Rightarrow w_0 + w_2 x_2 \Rightarrow m_{H_2}(N) = 2N$

$\vdots$

$\Phi_{(d)} : (1, x_d) \Rightarrow w_0 + w_d x_d \Rightarrow m_{H_d}(N) = 2N$

$\Rightarrow m_{\cup_{k=1}^{d} H_k}(N) = 2Nd$

$d_{vc}(\cup_{k=1}^{d} H_k) = \{\text{largest } N \text{ for which } m_H(N) = 2^N\} = N$

$2^N \leq m_H(N) = 2Nd$ ← we cannot shatter X if $2^N > m_H(N)$

$2^{N-1} \leq Nd$

$N-1 \leq \log_2 d + \log_2 N \leq \log_2 d + \frac{N}{2} \Rightarrow N \leq 2(\log_2 d + 1)$

(b) $2[\log_2 d + 1]$

14) (d) 0.60

```
import numpy as np
import random
import math
```

```
dat_file = r'C://Users//USER//Desktop//hw3_train.dat.txt'
with open(dat_file, 'r') as f:
    text = f.read()

data = text.split()  #split string into a list

D = []
for i in range(1000):
    K = list(data[i*11:(i+1)*11])
    D.append(K)

for i in range(len(D)):
    for j in range(len(D[0])):
        D[i][j] = float(D[i][j])
```

```
# 14 (d)
X = []
Y = []
for i in range(len(D)):
    X.append([1.0] + D[i][0:10])  # xi = [x0, x1, ..., x10]
    Y.append(D[i][-1])

XTX = np.transpose(X).dot(X)
W_LIN = np.linalg.inv(XTX).dot(np.transpose(X)).dot(Y)
E_in = np.square(np.subtract(np.array(X).dot(W_LIN),Y)).mean()
print(E_in)
```

```
0.6053223804672918
```

15) (c) 1800

```python
# 15 (c)
iteration = 0
random.seed(40)
SUM = 0
while iteration != 1000:
    E_wt = 10  # initialize
    W = [0]*11  # initialize
    i=0

    while E_wt>1.01*E_in:
        r = random.randint(0, 999)
        neg_gra_err = 2*(Y[r] - np.array(X[r]).dot(W))*np.array(X[r])
        W = W + 0.001* neg_gra_err
        E_wt = np.square(np.subtract(np.array(X).dot(W),Y)).mean()
        i+=1
    SUM = SUM+i
    iteration +=1

SUM = SUM/1000
print(SUM)
## 20 mins runtime
```

```
1772.577
```

16) (c) 0.56

```python
# 16 (c) 0.56
iteration = 0
SUM = 0
random.seed(50)
while iteration != 1000:
    E_in_c = 0
    W = [0]*11
    i = 0

    while i != 500:
        r = random.randint(0, 999)
        Exp = math.exp(-Y[r]*np.array(W).dot(X[r]))
        neg_gra_err = Y[r]* np.array(X[r]) *Exp/(1+Exp)
        W = W + 0.001* neg_gra_err
        for j in range(1000):
            Ep = math.exp(-Y[j]*np.array(W).dot(X[j]))
            E_in_c += math.log(1+Ep)
        E_in_c = E_in_c/1000
        i+=1
    SUM = SUM + E_in_c
    iteration += 1
SUM = SUM/1000
print(SUM)
# Almost 30 mins runtime
```

```
0.5694796431407935
```

17) (b) 0.50

```
# 17 (b) 0.50
iteration = 0
SUM = 0
random.seed(50)
while iteration != 1000:
    E_in_c = 0
    W = W_LIN
    i = 0

    while i != 500:
        r = random.randint(0, 999)
        Exp = math.exp(-Y[r]*np.array(W).dot(X[r]))
        neg_gra_err = Y[r]* np.array(X[r]) *Exp/(1+Exp)
        W = W + 0.001* neg_gra_err
        for j in range(1000):
            Ep = math.exp(-Y[j]*np.array(W).dot(X[j]))
            E_in_c += math.log(1+Ep)
        E_in_c = E_in_c/1000
        i+=1
    SUM = SUM + E_in_c
    iteration += 1
SUM = SUM/1000
print(SUM)
# Almost 30 mins runtime
```

```
0.5033221372400456
```

18) (a) 0.32

```
# 18 (a) 0.32
test_file = r'C://Users//USER//Desktop//hw3_test.dat.txt'
with open(test_file, 'r') as f:
    text = f.read()

tdata = text.split()  #split string into a list

T = []
for i in range(int(len(tdata)/11)):
    K = list(tdata[i*11:(i+1)*11])
    T.append(K)

for i in range(len(T)):
    for j in range(len(T[0])):
        T[i][j] = float(T[i][j])

X_test = []
Y_test = []
for i in range(len(T)):
    X_test.append([1.0] + T[i][0:10])  # xi = [x0, x1, ..., x10]
    Y_test.append(T[i][-1])
```

```
def sign(y):
    h = y
    if h>0:   h = 1
    else:     h = -1
    return h
```

```
E_in_binary = 0
Y_temporary = np.array(X).dot(W_LIN)
Y_hat = [sign(Y_temporary[j]) for j in range(len(Y_temporary))]
for j in range(len(Y)):
    if Y_hat[j] != Y[j]:
        E_in_binary+=1
E_in_binary = E_in_binary/len(Y)

E_out_binary = 0
Y_test_temp = np.array(X_test).dot(W_LIN)
Y_test_hat = [sign(Y_test_temp[j]) for j in range(len(Y_test_temp))]
for j in range(len(Y_test)):
    if Y_test_hat[j] != Y_test[j]:
        E_out_binary+=1
E_out_binary = E_out_binary/len(Y_test)

Diff_E = abs( E_out_binary - E_in_binary )
print(Diff_E)
```

```
0.32266666666666666
```

19) (b) 0.36

```
# 19  (b) 0.36
X_new = X.copy()
for i in range(2):
    for j in range(len(X_new)):
        X_new[j] = X_new[j]+[np.power(X[j][k+1],i+2) for k in range(len(X[0])-1)]

XTX_t = np.transpose(X_new).dot(X_new)
W_LIN_t = np.linalg.inv(XTX_t).dot(np.transpose(X_new)).dot(Y)

E_in_bi = 0
Y_train_hat_temp = np.array(X_new).dot(W_LIN_t)
Y_hatrain = [sign(Y_train_hat_temp[j]) for j in range(len(Y_train_hat_temp))]
for j in range(len(Y)):
    if Y_hatrain[j] != Y[j]:
        E_in_bi +=1
E_in_bi = E_in_bi/len(Y)


X_test_new = X_test.copy()
for i in range(2):
    for j in range(len(X_test_new)):
        X_test_new[j] = X_test_new[j]+[np.power(X_test[j][k+1],i+2) for k in range(len(X_test[0])-1)]

E_out_bi = 0
Y_test_hat_temp = np.array(X_test_new).dot(W_LIN_t)
Y_hatest = [sign(Y_test_hat_temp[j]) for j in range(len(Y_test_hat_temp))]
for j in range(len(Y_test)):
    if Y_hatest[j] != Y_test[j]:
        E_out_bi +=1
E_out_bi = E_out_bi/len(Y_test)

Diff_E_new = abs(E_out_bi - E_in_bi)
print(Diff_E_new)
```

```
0.37366666666666665
```

20) (d) 0.44

```python
# 20 (d) 0.44
X_new = X.copy()
for i in range(9):
    for j in range(len(X_new)):
        X_new[j] = X_new[j]+[np.power(X[j][k+1],i+2) for k in range(len(X[0])-1)]
XTX_t = np.transpose(X_new).dot(X_new)
W_LIN_t = np.linalg.inv(XTX_t).dot(np.transpose(X_new)).dot(Y)

E_in_bi = 0
Y_train_hat_temp = np.array(X_new).dot(W_LIN_t)
Y_hatrain = [sign(Y_train_hat_temp[j]) for j in range(len(Y_train_hat_temp))]
for j in range(len(Y)):
    if Y_hatrain[j] != Y[j]:
        E_in_bi +=1
E_in_bi = E_in_bi/len(Y)


X_test_new = X_test.copy()
for i in range(9):
    for j in range(len(X_test_new)):
        X_test_new[j] = X_test_new[j]+[np.power(X_test[j][k+1],i+2) for k in range(len(X_test[0])-1)]

E_out_bi = 0
Y_test_hat_temp = np.array(X_test_new).dot(W_LIN_t)
Y_hatest = [sign(Y_test_hat_temp[j]) for j in range(len(Y_test_hat_temp))]
for j in range(len(Y_test)):
    if Y_hatest[j] != Y_test[j]:
        E_out_bi +=1
E_out_bi = E_out_bi/len(Y_test)

Diff_E_new = abs(E_out_bi - E_in_bi)
print(Diff_E_new)
```

```
0.44666666666666666
```