P1. C

$$E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N} w^T w$$

微分

$$\Rightarrow \underline{\nabla E_{aug}(w) = \nabla E_{in}(w) + \frac{2\lambda}{N} w}$$ 代入 update rule 得

$$w(t+1) \leftarrow w(t) - \eta \left[\nabla E_{in}(w(t)) + \frac{2\lambda}{N} \cdot w(t) \right]$$

$$\Rightarrow w(t+1) \leftarrow \underbrace{\left[1 - \frac{2\lambda\eta}{N}\right]}_{\text{故选 C}}^{=S \#} w(t) - \eta \nabla E(w(t))$$

$\boxed{P2.B}$ By Silde 27. P29

$$\nabla E_{in}(W_{REG}) + \frac{2\lambda}{N}(W_{REG}) = 0$$

$$\Rightarrow \frac{2}{N}\sum_{n=1}^{N}(W - y_n) + \frac{2\lambda}{N}W = 0$$

$$\Rightarrow \frac{2}{N}\left(NW - \sum_{n=1}^{N}y_n\right) + \frac{2\lambda}{N}W = 0$$

$$\Rightarrow NW - \sum_{n=1}^{N}y_n + \lambda W = 0$$

$$\Rightarrow W(N+\lambda) = \sum_{n=1}^{N}y_n$$

$$\Rightarrow W = \frac{\sum_{n=1}^{N}y_n}{N+\lambda}$$

$$C = (W^*)^2$$

$$= \underline{\left(\frac{\sum_{n=1}^{N}y_n}{N+\lambda}\right)^2 \#} \text{ 选B}$$

$\boxed{P3.\ D}$ 若 w 与 w^T 存在線性关係，則2个式子仍能維持等價关係、

令 $w = V \cdot \tilde{w}$，代入下式

$$\min_{w \in R^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (w^T x_n - y_n)^2 + \frac{\lambda}{N} (w^T U w)$$

$$\Rightarrow \min_{w \in R^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (\tilde{w}^T V^T x_n - y_n)^2 + \frac{\lambda}{N} (\tilde{w}^T V^T U V \tilde{w}) \cdots\cdots ①$$

($V^T = V$，$\because$ V是对角矩陣)

$$\min_{w \in R^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (\tilde{w}^T V x_n - y_n)^2 + \frac{\lambda}{N} (\tilde{w}^T \tilde{w}) \cdots\cdots ②$$

比較①、②2式可知，當 $V^T U V = I$ 時，2式相等

因此 $\underline{U = (V^{-1})^2}$ #

P4.C

$$\min_{w\in\mathbb{R}^{d+1}} E\left[\frac{1}{N}\sum_{n=1}^{N}(w^T\Phi(x_n)-y_n)^2\right] \cdots \text{上式}$$

用matrix 代换上式 $\tilde{X}=\begin{bmatrix}-\phi(x_1)^T-\\-\phi(x_2)^T-\\ \vdots \\-\phi(x_n)^T-\end{bmatrix}$, $Y=\begin{bmatrix}y_1\\y_2\\ \vdots \\y_n\end{bmatrix}$

$$\min_{w\in\mathbb{R}^{d+1}} E\left[\frac{1}{N}(\tilde{X}w-Y)^2\right]$$

对w偏微

$$E\left[\frac{2}{N}\tilde{X}^T(\tilde{X}w-Y)\right]=0$$

$$\Rightarrow E[\tilde{X}^T\tilde{X}w]=E[\tilde{X}^TY]$$

代入: $\tilde{X}=\begin{bmatrix}-\phi(x_1)^T-\\-\phi(x_2)^T-\\ \vdots \\-\phi(x_n)^T-\end{bmatrix}=\begin{bmatrix}x_1^T\\x_2^T\\ \vdots \\x_n^T\end{bmatrix}+\begin{bmatrix}\varepsilon_1\\\varepsilon_2\\ \vdots \\\varepsilon_n\end{bmatrix}=X+\mathcal{E}$

$$\Rightarrow E[(X+\mathcal{E})^T(X+\mathcal{E})w]=E[(X+\mathcal{E})^TY]$$

$E(\mathcal{E})=0$

$$\Rightarrow E[(X^TX+2\mathcal{E}X+\mathcal{E}^T\mathcal{E})w]=E[X^TY+\mathcal{E}^TY]$$

$E(\mathcal{E}^T\mathcal{E})=\sigma^2I$

$$\Rightarrow X^TXw+0+\sigma^2w=X^TY+0$$

$$\Rightarrow (X^TX+\sigma^2I)w=X^TY$$

$$\Rightarrow \underline{w=(X^TX+\sigma^2I)^{-1}X^TY}$$

下式：$\min_{w \in \mathbb{R}^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (w^T x_n - y_n)^2 + \frac{\lambda}{N} \|w\|_2^2$

用前頁一樣的方式換成matrix form

$$\min_{w \in \mathbb{R}^{d+1}} \frac{1}{N} (Xw - Y)^2 + \frac{\lambda}{N} (w^T w)$$

对w偏微分

$$\frac{2}{N} X^T (Xw - Y) + \frac{2\lambda}{N} w = 0$$

$$\Rightarrow \frac{2}{N} X^T X w - \frac{2}{N} X^T Y + \frac{2\lambda}{N} w = 0$$

$$\Rightarrow \left(X^T X + \frac{2\lambda}{N} I\right) w = X^T Y$$

代入w

$$\Rightarrow \left(X^T X + \frac{2\lambda}{N} I\right) (X^T X + \sigma^2 I)^{-1} X^T Y = X^T Y$$

上述等式要成立必須滿足

$$\sigma^2 = \frac{2\lambda}{N}$$

$$\Rightarrow \underline{\lambda = \frac{N\sigma^2}{2}} \#$$ 故选C

$\boxed{\text{P5. D}}$

$$\min_{y\in\mathbb{R}} \frac{1}{N}\sum_{n=1}^{N}(y-y_n)^2+\frac{\alpha K}{N}\Omega(y)$$

$\downarrow$ 对y微分 =0

$$\frac{1}{N}\sum_{n=1}^{N}2(y-y_n)+\frac{\alpha K}{N}\cdot\frac{d}{dy}\Omega(y)=0$$

$$\Rightarrow \frac{2}{N}\left(Ny-\sum_{n=1}^{N}y_n\right)+\frac{\alpha K}{N}\cdot\frac{d}{dy}\Omega(y)=0$$

$$\Rightarrow 2Ny-2\sum_{n=1}^{N}y_n+\alpha K\frac{d}{dy}\Omega(y)=0$$

代入 $y^*=\dfrac{\sum_{n=1}^{N}y_n+\alpha}{N+\alpha K}$，進入上式 等號会成立

$$\Rightarrow 2N\frac{\sum_{n=1}^{N}y_n+\alpha}{N+\alpha K}-2\sum_{n=1}^{N}y_n+\alpha K\frac{d}{dy}\Omega(y^*)=0$$

$$\Rightarrow \frac{d}{dy}\Omega(y^*)=\frac{1}{\alpha K}\cdot\frac{-2N\left(\sum_{n=1}^{N}y_n+\alpha\right)+2(N+\alpha K)\sum_{n=1}^{N}y_n}{N+\alpha K}$$

$$=\frac{\cancel{-2N\sum_{n=1}^{N}y_n}-2N\alpha+\cancel{2N\sum_{n=1}^{N}y_n}+2\alpha K\sum_{n=1}^{N}y_n}{(N+\alpha K)(\alpha K)}$$

$$=\frac{-\frac{2N}{K}+2\sum_{n=1}^{N}y_n}{N+\alpha K}$$

$$=\frac{2\cdot\sum_{n=1}^{N}y_n}{N+\alpha K}+\frac{-2N}{K(N+\alpha K)}+\frac{2\alpha}{N+\alpha K}-\frac{2\alpha}{N+\alpha K}$$

$$=2\cdot\underbrace{\left(\frac{\sum_{n=1}^{N}y_n+\alpha}{N+\alpha K}\right)}_{\text{“}y^*\text{”}}-2\cdot\frac{\cancel{N+\alpha K}}{\cancel{(N+\alpha K)}K}$$

$$=2\cdot y^*-2\cdot\frac{1}{K}$$

$$\Rightarrow \frac{d}{dy}\Omega(y^*)=2\left(y^*-\frac{1}{K}\right)\xrightarrow{\text{積分}}\Omega(y)=\underline{\left(y-\frac{1}{K}\right)^2}\ \#\ \text{選 D}$$

P6.B

$$\nabla \tilde{E}_{aug}(w) \Rightarrow \nabla\left(\tilde{E}_{in}(w) + \frac{\lambda}{N}\|w\|^2\right) = 0$$

$$\Rightarrow \nabla \tilde{E}_{in}(w) + \frac{\lambda}{N}\frac{d}{dw}\|w\|^2 = 0$$

$$\Rightarrow \underbrace{\frac{d}{dw}\Big(E_{in}(w^*)}_{=0} + \frac{1}{2}(w-w^*)^T H (w-w^*)\Big) + \frac{\lambda}{N}\frac{d}{dw}\|w\|^2 = 0$$

$$\Rightarrow \frac{1}{2}\left[H(w-w^*) + H(w-w^*)\right] + \frac{\lambda}{N}\frac{d}{dw}(w^T w) = 0$$

$$\Rightarrow H(w-w^*) + \frac{2\lambda}{N}w = 0$$

$$\Rightarrow Hw - Hw^* + \frac{2\lambda}{N}w = 0$$

$$\Rightarrow \left(H + \frac{2\lambda}{N}I\right)w = Hw^*$$

$$\Rightarrow \underline{w = \left(H + \frac{2\lambda}{N}I\right)^{-1} H w^* \quad \#}$$

$\boxed{\text{P7. A}}$ 每次都会predict minority，而每次validation set刚好都是minority.

$$E_{loocv}(A_{algorithm}) = \frac{1}{2N}[N \cdot 0 + N \cdot 0] = \underline{0} \ \#$$

P8.c

train: $(x_1, y_1) = (2,0)$ $(x_2, y_2) = (\rho, 2)$	test $(x_3, y_3) = (-2,0)$	constant hypothesis $h(x) = 1$ $E_{squ} = (1-0)^2 = 1$	linear hypthesis $h(x) = \frac{2}{\rho-2}x + \frac{-4}{\rho-2}$ $E_{squ} = (\frac{-8}{\rho-2} - 0)^2$
train: $(x_1, y_1) = (2,0)$ $(x_3, y_3) = (-2,0)$	test $(x_2, y_2) = (\rho, 2)$	$h(x) = 0$ $E_{squ} = (2-0)^2 = 4$	$h(x) = 0$ $E_{squ} = (2-0)^2 = 4$
train: $(x_2, y_2) = (\rho, 2)$ $(x_3, y_3) = (-2,0)$	test $(x_1, y_1) = (2,0)$	$h(x) = 1$ $E_{squ} = (1-0)^2 = 1$	$h(x) = \frac{2}{\rho+2}x + \frac{4}{\rho+2}$ $E_{squ} = (\frac{8}{\rho+2} - 0)^2$

$$\frac{64}{(\rho-2)^2} + \frac{64}{(\rho+2)^2} = 2$$

$$\Rightarrow \rho \pm 8.6713 \#$$

P9. B

$$E\left(\frac{1}{K}\sum_{n=N-K+1}^{N}(y_n-\bar{y})^2\right)$$

$$\Rightarrow E\left(\frac{1}{K}\sum_{n=N-K+1}^{N}(y_n^2-2y_n\bar{y}+\bar{y}^2)\right)$$

$\because$ $y_1 \sim y_n$ 是 i.i.d sample 出来

$$\Rightarrow \frac{1}{K}\sum_{n=N-K+1}^{N}\left(E(y_n^2)-2E(y_n\bar{y})+E(\bar{y}^2)\right)$$

$$\Rightarrow \frac{1}{K}\sum_{n=N-K+1}^{N}\left(\sigma^2-2E\left(y_n\cdot\frac{\sum_{n=1}^{N-K}y_n}{N-K}\right)+E\left(\frac{\sum_{n=1}^{N-K}y_n}{N-K}\cdot\frac{\sum_{n=1}^{N-K}y_n}{N-K}\right)\right)$$

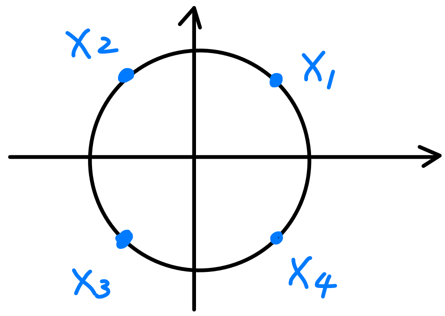
$$\Rightarrow \frac{1}{K}\sum_{n=N-K+1}^{N}\left[\sigma^2-\frac{2}{N-K}\underbrace{E\left(y_n\cdot\sum_{n=1}^{N-K}y_n\right)}_{\downarrow\text{ i.i.d.}}+\frac{1}{(N-K)^2}\cdot \underline{E\left(\left(\sum_{n=1}^{N-K}y_n\right)^2\right)}\right]$$

$$\Rightarrow \frac{1}{K}\sum_{n=N-K+1}^{N}\left[\sigma^2-\frac{2}{N-K}\underbrace{E(y_n)\cdot E\left(\sum_{n=1}^{N-K}y_n\right)}_{=0}+\frac{1}{(N-K)^2}\cdot\left[\underbrace{Var\left(\sum_{n=1}^{N-K}y_n\right)}_{=(N-K)\sigma^2}+\underbrace{E\left(\sum_{n=1}^{N-K}y_n\right)^2}_{=0}\right]\right]$$

$$\Rightarrow \frac{1}{K}\sum_{N-K+1}^{N}\left[\sigma^2+\frac{(N-K)\sigma^2}{(N-K)^2}\right]$$

$$\Rightarrow \underline{\sigma^2+\frac{\sigma^2}{(N-K)}}\ \#$$

$\boxed{\text{P10\_A}}$



y_1	y_2	y_3	y_4	0/1 $E_{in}(w^*)$
+	+	+	+	0
+	+	+	−	0
+	+	−	+	0
+	−	+	+	0
−	+	+	+	0
+	+	−	−	0
+	−	+	−	1/4
−	+	+	−	0
+	−	−	+	0
−	+	−	+	1/4
−	−	+	+	0
−	−	−	+	0
−	−	+	−	0
−	+	−	−	0
+	−	−	−	0
−	−	−	−	0

$$\Rightarrow \frac{1}{16}\left(\frac{1}{4}+\frac{1}{4}\right)=\underline{\frac{1}{32}}\ \#$$

P11, B

$$E_{out}(g) = P \cdot \varepsilon_+ + (1-P)\varepsilon_- = E_{out}(g_-) = P$$

$$\Rightarrow P\varepsilon_+ + \varepsilon_- - P\varepsilon_- = P$$

$$\Rightarrow \varepsilon_+ + \frac{\varepsilon_-}{P} - \varepsilon_- = 1$$

$$\Rightarrow \frac{1}{P} = \frac{1 - \varepsilon_+ + \varepsilon_-}{\varepsilon_-}$$

$$\Rightarrow \underline{P = \frac{\varepsilon_-}{\varepsilon_- - \varepsilon_+ + 1}} \quad \#$$

P12.B

```
TRAIN='hw4_test_trans.dat'
TEST='hw4_test_trans.dat'

./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
```

```python
import numpy as np
import random

# Output
TRAIN_OUTPUT_FILE = "hw4_train_trans.dat"
TEST_OUTPUT_FILE = "hw4_test_trans.dat"

def data_load(file_name):
    l = []
    with open(file_name, 'r') as f:
        for i in f.readlines():
            x = [float(x_n) for x_n in i.split('\n')[0].split(' ')]
            l.append(x)
    return l

def feature_transfor_full_order_3(data_list):
    ans = []
    D = len(data_list[0])
    print(f"Number of feature: {D}")

    for raw_features in data_list: # One row of data
        trans_f = []
        # 0 order polynominal
        trans_f.append(1) # x_0
        # 1 order polynominal
        for i in raw_features:
            trans_f.append(i)

        # 2 order polynominal
        for b_1 in range(D): # Where is the first ball
            for b_2 in range(D): # where is the second ball
                if b_2 < b_1:
                    continue
                else:
                    trans_f.append(raw_features[b_1] * raw_features[b_2])

        # 3 order polynominal
        for b_1 in range(D): # Where is the first ball
            for b_2 in range(D): # where is the second ball
                for b_3 in range(D): # where is the third ball
                    if b_2 < b_1 or b_3 < b_2:
                        continue
                    else:
                        trans_f.append(raw_features[b_1] * raw_features[b_2] * raw_features[b_3])
        ans.append(trans_f)

    # print(ans[0])
    # print(len(ans[0]))
    return ans

# Get Training data
train_data = data_load('hw4_train.dat')
print(f"Total {len(train_data)} training data loaded.")
print(f"First row of training data is {train_data[0]}")
test_data = data_load('hw4_test.dat')
print(f"Total {len(test_data)} testing data loaded.")
print(f"First row of testing data is {test_data[0]}")

# Feature Transformation
train_data_trans = feature_transfor_full_order_3( [i[:-1] for i in train_data] ) # Get rid of
test_data_trans  = feature_transfor_full_order_3( [i[:-1] for i in test_data] ) # Get rid of label

# Convert train and test data into LIBLINEAR format
s = ""
for i, features in enumerate(train_data_trans) :
    # First index is label:
    row_s = str(int(train_data[i][-1])) + " "
    # Reset of the index is feature transformed
    for j, feature in enumerate(features):
        row_s += f"{j+1}:{feature}"
        row_s += " "
    s += row_s + "\n"
with open(TRAIN_OUTPUT_FILE, 'w') as f:
    f.write(s)
print(f"Output .dat to {TRAIN_OUTPUT_FILE}")
# print(s.split('\n')[0])


s = ""
for i, features in enumerate(test_data_trans) :
    # First index is label:
    row_s = str(int(test_data[i][-1])) + " "
    # Reset of the index is feature transformed
    for j, feature in enumerate(features):
        row_s += f"{j+1}:{feature}"
        row_s += " "
    s += row_s + "\n"
with open(TEST_OUTPUT_FILE, 'w') as f:
    f.write(s)
print(f"Output .dat to {TEST_OUTPUT_FILE}")
```

P.13.B

```
TRAIN='hw4_train_trans.dat'
TEST='hw4_train_trans.dat'

./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
```

P14. C

```
TRAIN='hw4_p14_train.dat'
TEST='hw4_p14_val.dat'

./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv

# Test on real testing set
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
./predict ../hw4_test_trans.dat $TRAIN.model 123.csv
```

P15, D

```
TRAIN='hw4_train_trans.dat'
TEST='hw4_test_trans.dat'

# Test on real testing set
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
```

P16_B

```python
### P16
n_fold = 5
with open('hw4_train_trans.dat', 'r') as f:
    lines = f.readlines()
    for fold in range(n_fold):
        s_train = ""
        s_val = ""
        for i, line in enumerate(lines):
            if (i >= fold*40 and i < (fold+1)*40):
                s_val += line
            else:
                s_train += line
        print(len(s_train.split('\n')))
        print(len(s_val.split('\n')))

        with open(f'hw4_p16_train_{fold}_fold.dat', 'w') as f:
            f.write(s_train)
        with open(f'hw4_p16_val_{fold}_fold.dat', 'w') as f:
            f.write(s_val)
```

```
TRAIN='hw4_p16_train_0_fold.dat'
TEST='hw4_p16_val_0_fold.dat'
./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_1_fold.dat'
TEST='hw4_p16_val_1_fold.dat'
./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_2_fold.dat'
TEST='hw4_p16_val_2_fold.dat'
./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_3_fold.dat'
TEST='hw4_p16_val_3_fold.dat'
./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_4_fold.dat'
TEST='hw4_p16_val_4_fold.dat'
./train -q -s 0 -c 5000 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
echo "======="
TRAIN='hw4_p16_train_0_fold.dat'
TEST='hw4_p16_val_0_fold.dat'
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_1_fold.dat'
TEST='hw4_p16_val_1_fold.dat'
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_2_fold.dat'
TEST='hw4_p16_val_2_fold.dat'
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_3_fold.dat'
TEST='hw4_p16_val_3_fold.dat'
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_4_fold.dat'
TEST='hw4_p16_val_4_fold.dat'
./train -q -s 0 -c 50 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
echo "======="
TRAIN='hw4_p16_train_0_fold.dat'
TEST='hw4_p16_val_0_fold.dat'
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_1_fold.dat'
TEST='hw4_p16_val_1_fold.dat'
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_2_fold.dat'
TEST='hw4_p16_val_2_fold.dat'
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_3_fold.dat'
TEST='hw4_p16_val_3_fold.dat'
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_4_fold.dat'
TEST='hw4_p16_val_4_fold.dat'
./train -q -s 0 -c 0.5 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
echo "======="
TRAIN='hw4_p16_train_0_fold.dat'
TEST='hw4_p16_val_0_fold.dat'
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_1_fold.dat'
TEST='hw4_p16_val_1_fold.dat'
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_2_fold.dat'
TEST='hw4_p16_val_2_fold.dat'
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_3_fold.dat'
TEST='hw4_p16_val_3_fold.dat'
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
TRAIN='hw4_p16_train_4_fold.dat'
TEST='hw4_p16_val_4_fold.dat'
./train -q -s 0 -c 0.005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model 123.csv
echo "======="
TRAIN='hw4_p16_train_0_fold.dat'
TEST='hw4_p16_val_0_fold.dat'
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model
TRAIN='hw4_p16_train_1_fold.dat'
TEST='hw4_p16_val_1_fold.dat'
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model
TRAIN='hw4_p16_train_2_fold.dat'
TEST='hw4_p16_val_2_fold.dat'
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model
TRAIN='hw4_p16_train_3_fold.dat'
TEST='hw4_p16_val_3_fold.dat'
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model
TRAIN='hw4_p16_train_4_fold.dat'
TEST='hw4_p16_val_4_fold.dat'
./train -q -s 0 -c 0.00005 -e 0.000001 ../$TRAIN; ./predict ../$TEST $TRAIN.model
echo "======="
```