In [ ]:
```
import pandas as pd
import numpy as np
import copy
```

In [ ]:
```
data_url="https://www.csie.ntu.edu.tw/~htlin/course/ml21fall/hw1/hw1_train.dat"
col_names = [i for i in range(1,12)]
data = pd.read_csv(data_url, header=None, names=col_names, delimiter='\t')
data
```

Out[ ]:

	1	2	3	4	5	6	7	8	9	10	1
0	1.56186	-2.54905	-1.98638	-0.30684	-1.00062	3.52667	2.62601	-0.30951	1.32496	-2.26376	-1
1	3.89045	-0.02852	2.20650	1.20511	0.12892	1.16363	1.41855	-1.30547	-2.31854	-1.40395	-1
2	-1.85626	-1.38071	-0.07550	-2.90992	-2.75206	-0.23195	-1.10457	-1.11643	-2.35446	-1.43411	1
3	-1.58778	-2.07548	0.00738	-4.24154	0.91851	-5.61822	0.97835	0.06143	4.33760	-1.50129	1
4	2.15052	5.26269	-1.26788	-2.68478	-1.15078	-0.36360	0.74234	1.31526	-2.77029	-0.14857	1
...	...	...	...	...	...	...	...	...	...	...	
95	-3.08999	-3.59724	5.58267	-0.73572	5.16949	3.41667	-0.97299	-0.67884	0.98183	-0.39283	-1
96	-0.17985	1.50163	0.66528	1.89992	0.79647	-1.60727	0.31752	-0.06467	-0.51961	3.71141	-1
97	-3.81450	0.89167	-2.15984	-3.80682	-4.75878	-0.78957	-0.28329	0.45259	-1.57172	0.15997	1
98	-2.11276	-1.91391	-0.63889	-3.53088	-2.24357	-1.22243	0.65278	2.75600	-3.45234	-1.29036	1
99	-2.50787	-1.02966	0.52740	3.15535	-3.28735	1.44250	1.93997	-0.31516	1.14198	0.64107	-1

100 rows × 11 columns

In [ ]:
```
target = data[11]
target
```

Out[ ]:
```
0    -1.0
1    -1.0
2     1.0
3     1.0
4     1.0
     ...
95   -1.0
96   -1.0
97    1.0
98    1.0
99   -1.0
Name: 11, Length: 100, dtype: float64
```

In [ ]:
```
data = data.drop(11, axis=1)
data
```

Out[ ]:

	1	2	3	4	5	6	7	8	9	10
0	1.56186	-2.54905	-1.98638	-0.30684	-1.00062	3.52667	2.62601	-0.30951	1.32496	-2.26376
1	3.89045	-0.02852	2.20650	1.20511	0.12892	1.16363	1.41855	-1.30547	-2.31854	-1.40395

	1	2	3	4	5	6	7	8	9	10
2	-1.85626	-1.38071	-0.07550	-2.90992	-2.75206	-0.23195	-1.10457	-1.11643	-2.35446	-1.43411
3	-1.58778	-2.07548	0.00738	-4.24154	0.91851	-5.61822	0.97835	0.06143	4.33760	-1.50129
4	2.15052	5.26269	-1.26788	-2.68478	-1.15078	-0.36360	0.74234	1.31526	-2.77029	-0.14857
...	...	...	...	...	...	...	...	...	...	...
95	-3.08999	-3.59724	5.58267	-0.73572	5.16949	3.41667	-0.97299	-0.67884	0.98183	-0.39283
96	-0.17985	1.50163	0.66528	1.89992	0.79647	-1.60727	0.31752	-0.06467	-0.51961	3.71141
97	-3.81450	0.89167	-2.15984	-3.80682	-4.75878	-0.78957	-0.28329	0.45259	-1.57172	0.15997
98	-2.11276	-1.91391	-0.63889	-3.53088	-2.24357	-1.22243	0.65278	2.75600	-3.45234	-1.29036
99	-2.50787	-1.02966	0.52740	3.15535	-3.28735	1.44250	1.93997	-0.31516	1.14198	0.64107

100 rows × 10 columns

Q13 ~ 16

PLA with different data preprocessing

In [ ]:
```
def sign(val):
    if (val <=0): return -1.0
    else: return 1.0

def PLA(data, target, random_seed, preprocess):
    data_in = copy.deepcopy(data)
    data_in = preprocess(data_in)

    rng = np.random.RandomState(random_seed)

    N = data_in.shape[0] # Number of Samples
    w = np.zeros(data_in.shape[1]) # Init to zeros

    while(True):
        success=True
        for i in range(5*N):
            idx = rng.randint(0, N)
            tmp = np.dot(w, data_in[idx])
            if (sign(tmp) != target[idx]):
                w = w + target[idx] * data_in[idx]
                success = False
                break
        if (success):
            break

    return w
```

Q13

In [ ]:
```
def preprocess_0(data_in):
    data_in.insert(0, 0, np.ones(100))
    data_in = data_in.to_numpy()
```

```
    return data_in

w_pla_length=[]
for i in range(1000):
    wt = PLA(data, target, i, preprocess_0)
    w_pla_length.append(np.linalg.norm(wt)**2)
print(np.average(w_pla_length))
```

```
388.0550458239121
```

Q14

In [ ]:
```
def preprocess_1(data_in):
    data_in.insert(0, 0, np.ones(100))
    data_in = data_in.to_numpy()
    return data_in * 2

w_pla_length=[]
for i in range(1000):
    wt = PLA(data, target, i, preprocess_1)
    w_pla_length.append(np.linalg.norm(wt)**2)
print(np.average(w_pla_length))
```

```
1552.2201832956484
```

Q15

In [ ]:
```
def preprocess_2(data_in):
    data_in.insert(0, 0, np.ones(100))
    data_in = data_in.to_numpy()
    for i in data_in:
        i /= np.linalg.norm(i)
    return data_in

w_pla_length=[]
for i in range(1000):
    wt = PLA(data, target, i, preprocess_2)
    w_pla_length.append(np.linalg.norm(wt)**2)
print(np.average(w_pla_length))
```

```
7.0702934116262535
```

Q16

In [ ]:
```
def preprocess_3(data_in):
    data_in.insert(0, 0, np.zeros(100))
    data_in = data_in.to_numpy()
    return data_in

w_pla_length=[]
for i in range(1000):
    wt = PLA(data, target, i, preprocess_3)
    w_pla_length.append(np.linalg.norm(wt)**2)
print(np.average(w_pla_length))
```

```
541.4407857585554
```