

MACHINE LEARNING

REPORT OF HOMEWORK 2

```
import numpy as np
import pandas as pd
from pandas import ExcelWriter
from random import randint, random
import matplotlib.pyplot as plt
```

Figure 1 Libraries

```
train_data = pd.read_excel("DataForPerceptron.xlsx", sheet_name="TRAINData")
x_train = train_data.iloc[:, 1:10].values
y_train = train_data.iloc[:, 10:11].values
x_train_list = x_train.tolist()
y_train_list = y_train.tolist()

test_data = pd.read_excel("DataForPerceptron.xlsx", sheet_name="TESTData")
x_test = test_data.iloc[:, 1:10].values
x_test_list = x_test.tolist()
```

Figure 2 Reading File

```
for i in range (len(x_test_list)):
    x_test_list[i].append(1)

for i in range (len(x_train_list)):
    x_train_list[i].append(1)

print("Appending 1 every matrix's last item for x_test_list : ", x_test_list[0])
print("Appending 1 every matrix's last item x_train_list : ", x_train_list[0])
✓ 0.7s

Appending 1 every matrix's last item for x_test_list : [5, 7, 10, 10, 5, 10, 10, 10, 1, 1]
Appending 1 every matrix's last item x_train_list : [5, 1, 1, 1, 2, 1, 3, 1, 1, 1]
```

Figure 3 Appending 1 Every Matrix's Last Item

```
w = []
✓ for i in range (0,10):
    w.append(randint(1,10))
    print("Creating Random Numbers Array: ",w)
✓ 0.5s

Creating Random Numbers Array: [3, 9, 8, 10, 10, 8, 2, 6, 4, 10]
```

Figure 5 Creating Random Numbers Array

```
learning_rate = 1
✓ 0.4s
```

Figure 4 Learning Rate

PERCEPTRON LEARNING ALGORITHM IMPLEMENTATION

```

while True:
    m = 0
    w_arr = []
    for i in range(len(x_train_list)):
        wt = []
        for j in range(len(x_train_list[i])):
            wt.append(w[j] * x_train_list[i][j])

        if sum(wt) <= 0 and y_train_list[i][0] == 4:

            for k in range(len(x_train_list[i])):
                w[k] = w[k] + (x_train_list[i][k] * learning_rate)
                w_arr.append(w[k])
            m += 1

        elif sum(wt) >= 0 and y_train_list[i][0] == 2:
            for l in range(len(x_train_list[i])):
                w[l] = w[l] - (x_train_list[i][l] * learning_rate)
                w_arr.append(w[l])
            m -= 1

        else:
            continue

    if m == 0:
        break
    for i in range(len(w_arr)):
        print("Values of w which created with Perceptron Learning Algorithm: ",w_arr[i])

```

✓ 0.1s

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```

Values of w which created with Perceptron Learning Algorithm: 12
Values of w which created with Perceptron Learning Algorithm: 21
Values of w which created with Perceptron Learning Algorithm: 6
Values of w which created with Perceptron Learning Algorithm: 0
Values of w which created with Perceptron Learning Algorithm: -21
Values of w which created with Perceptron Learning Algorithm: 10
Values of w which created with Perceptron Learning Algorithm: 4
Values of w which created with Perceptron Learning Algorithm: -4
Values of w which created with Perceptron Learning Algorithm: 5
Values of w which created with Perceptron Learning Algorithm: -319
Values of w which created with Perceptron Learning Algorithm: 17

```

Figure 6 Values of w Which Created with PLA

CLASSIFICATION OF TESTData USING PLA

```

class_value = []
wt_test_arr = []
for i in range (len(x_test_list)):
    wt_test = []
    for j in range (len(x_test_list[i])):
        wt_test.append(w_arr[j] * x_test_list[i][j])

    if sum(wt_test)<=0:
        wt_test_arr.append(sum(wt_test))
        class_value.append(2)

    elif sum(wt_test)>=0:
        wt_test_arr.append(sum(wt_test))
        class_value.append(4)

for i in range (len(wt_test_arr)):
    print("TESTData Classification with Perceptron Learning Algorithm: ",wt_test_arr[i]," = ",class_value[i]," (Class Value)")
print("Length of TESTData: ", len(wt_test_arr))

test_data['Class'] = class_value

with pd.ExcelWriter("DataForPerceptron.xlsx") as writer:
    train_data.to_excel(writer,sheet_name="TRAINData" , index=False)
    test_data.to_excel(writer,sheet_name="TESTData" , index=False)

```

✓ 0.3s

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```

TESTData Classification with Perceptron Learning Algorithm: -52 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -269 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -247 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -143 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: 16 = 4 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -28 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -6 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -279 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -303 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -27 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -245 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -255 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -303 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -303 = 2 (Class Value)
TESTData Classification with Perceptron Learning Algorithm: -299 = 2 (Class Value)

```

Figure 7 TESTData Classification with PLA

WRITE CLASSIFICATION VALUES OF TESTData TO EXCEL

DataForPerceptron.xlsx											
	A	B	C	D	E	F	G	H	I	J	K
1	SubjectID	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	Class
2	1	5	1	1	1	2	1	3	1	1	2
3	2	5	4	4	5	7	10	3	2	1	2
4	3	3	1	1	1	2	2	3	1	1	2
5	4	6	8	8	1	3	4	3	7	1	2
6	5	4	1	1	3	2	1	3	1	1	2
7	6	8	10	10	8	7	10	9	7	1	4
8	7	1	1	1	1	2	10	3	1	1	2
9	8	2	1	2	1	2	1	3	1	1	2
10	9	2	1	1	1	2	1	1	1	5	2
11	10	4	2	1	1	2	1	2	1	1	2
12	11	1	1	1	1	1	1	3	1	1	2
13	12	2	1	1	1	2	1	2	1	1	2
14	13	5	3	3	3	2	3	4	4	1	4
15	14	1	1	1	1	2	3	3	1	1	2
16	15	8	7	5	10	7	9	5	5	4	4
17	16	7	4	6	4	6	1	4	3	1	4
18	17	4	1	1	1	2	1	2	1	1	2
19	18	4	1	1	1	2	1	3	1	1	2
20	19	10	7	7	6	4	10	4	1	2	4
21	20	6	1	1	1	2	1	3	1	1	2
22	21	7	3	2	10	5	10	5	4	4	4
23	22	10	5	5	3	6	7	7	10	1	4
24	23	3	1	1	1	2	1	2	1	1	2
25	24	1	1	1	1	2	1	3	1	1	2
26	25	5	2	3	4	2	7	3	6	1	4
27	26	3	2	1	1	1	1	2	1	1	2
28	27	5	1	1	1	2	1	2	1	1	2
29	28	2	1	1	1	2	1	2	1	1	2
30	29	1	1	3	1	2	1	1	1	1	2
31	30	3	1	1	1	1	1	2	1	1	2
32	31	2	1	1	1	2	1	3	1	1	2
33	32	10	7	7	3	8	5	7	4	3	4
34	33	2	1	1	2	2	1	3	1	1	2
35	34	3	1	2	1	2	1	2	1	1	2

Figure 8 TRAINData Classification

DataForPerceptron.xlsx											
	A	B	C	D	E	F	G	H	I	J	K
1	SubjectID	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	Class
2	551	5	7	10	10	5	10	10	10	1	2
3	552	3	1	2	1	2	1	3	1	1	2
4	553	4	1	1	1	2	3	2	1	1	2
5	554	8	4	4	1	6	10	2	5	2	2
6	555	10	10	8	10	6	5	10	3	1	4
7	556	8	10	4	4	8	10	8	2	1	2
8	557	7	6	10	5	3	10	9	10	2	2
9	558	3	1	1	1	2	1	2	1	1	2
10	559	1	1	1	1	2	1	2	1	1	2
11	560	10	9	7	3	4	2	7	7	1	4
12	561	5	1	2	1	2	1	3	1	1	2
13	562	5	1	1	1	2	1	2	1	1	2
14	563	1	1	1	1	2	1	2	1	1	2
15	564	1	1	1	1	2	1	2	1	1	2
16	565	1	1	1	1	2	1	3	1	1	2
17	566	5	1	2	1	2	1	2	1	1	2
18	567	5	7	10	6	5	10	7	5	1	2
19	568	6	10	5	5	4	10	6	10	1	2
20	569	3	1	1	1	2	1	1	1	1	2
21	570	5	1	1	6	3	1	1	1	1	2
22	571	1	1	1	1	2	1	1	1	1	2
23	572	8	10	10	10	6	10	10	10	1	2
24	573	5	1	1	1	2	1	2	2	1	2
25	574	9	8	8	9	6	3	4	1	1	2
26	575	5	1	1	1	2	1	1	1	1	2
27	576	4	10	8	5	4	1	10	1	1	2
28	577	2	5	7	6	4	10	7	6	1	2
29	578	10	3	4	5	3	10	4	1	1	2
30	579	5	1	2	1	2	1	1	1	1	2
31	580	4	8	6	3	4	10	7	1	1	2
32	581	5	1	1	1	2	1	2	1	1	2
33	582	4	1	2	1	2	1	2	1	1	2
34	583	5	1	3	1	2	1	2	1	1	2

Figure 9 TESTData Classification

RESULT

The Perceptron Learning Algorithm creates a new and meaningful array by including the Attributes of the TRAINData data we have in the formula " $w * x_{train_list}$ ". If the sum of the values of this array is greater than 0 and the class number is 2, a new w value is obtained by including it in the " $w = w - (x_{train_list} * learning_rate)$ " formula. If the sum of the values of this array is less than 0 and the class number is 4, a new w value is obtained by including it in the " $w = w + (x_{train_list} * learning_rate)$ " formula. By including these values in the formula " $wt_test = w_arr * x_test_list$ " along with the attributes of TESTData, the logic of the Class values in TRAINData is implemented into the Class values of TESTData.

Finally, according to the newly created array, if the sum of the values of this array in each matrix is less than 0, the Class Value of this array is updated as 2, else if the total is greater than 0, the Class Value is updated to 4. Thus, missing Class values in TESTData are completed.