2021-2학기 AI+X 딥러닝 (AIX0003)

Midterm TakeHome Exam

한양대학교ERICA 소프트웨어학부 2016003254 고동현

In [ ]:
```
import pandas as pd
import random
```

Task-1

Write a statistical analysis script to display the most frequently appeared number to the least.

In [ ]:
```
data = pd.read_csv('./lottery.csv')
```

In [ ]:
```
num_count_list = [[i,0] for i in range(1,46)]
for i in range(2,9):

        temp = data.iloc[:,i].value_counts().to_dict()

        for j in temp:
                num_count_list[j-1][1] += temp[j]
```

In [ ]:
```
for n in sorted(num_count_list,key=lambda x:x[1],reverse=True):
        print(f'{n[0]}\t-> {n[1]} times')
```

```
43      -> 179 times
27      -> 170 times
34      -> 170 times
1       -> 168 times
13      -> 167 times
17      -> 167 times
33      -> 164 times
4       -> 163 times
12      -> 163 times
39      -> 162 times
2       -> 161 times
10      -> 161 times
20      -> 160 times
40      -> 160 times
18      -> 159 times
14      -> 158 times
26      -> 158 times
37      -> 158 times
38      -> 157 times
21      -> 156 times
24      -> 156 times
31      -> 156 times
3       -> 155 times
7       -> 155 times
11      -> 154 times
16      -> 154 times
36      -> 152 times
6       -> 151 times
45      -> 151 times
5       -> 150 times
15      -> 150 times
19      -> 150 times
35      -> 150 times
```

```
42      -> 150 times
8       -> 149 times
30      -> 148 times
44      -> 148 times
25      -> 143 times
23      -> 138 times
28      -> 138 times
32      -> 138 times
41      -> 137 times
29      -> 134 times
9       -> 128 times
22      -> 127 times
```

Task-2:

Create a modified lottery data format by adding a new column.

In [ ]:

```
# 이미 있는 숫자 저장
history = []
for i in data.iloc:
    temp = ""
    for j in range(1, 46):
        if j in [i['first'], i['second'], i['third'], i['fourth'],
                 i['fifth'], i['sixth'], i['bonus']]:
            temp += "1"
        else:
            temp += "0"
    history.append(int(temp, 2))  # 정수로 저장하기 위해 2진수로 변환
```

In [ ]:

```
new_data = pd.DataFrame(columns=['round', 'date', 'first',
                         'second', 'third', 'fourth', 'fifth',
                                                'sixth', 'bonus', 'win'])
lottery_num = [i for i in range(1, 46)]
for i in data.iloc:
    origin_data = dict(i)
    origin_data['win'] = 1
    new_data = new_data.append(origin_data, ignore_index=True)
    while True:
        random.shuffle(lottery_num)
        # 랜덤 숫자 생성 후 2진수로 변환
        temp = int(''.join(['1' if n in lottery_num[:7]
                   else '0' for n in range(1, 46)]), 2)
        if temp not in history:  # 이미 있는지 검사 없다면 아래 수행
            origin_data['first'] = lottery_num[0]
            origin_data['second'] = lottery_num[1]
            origin_data['third'] = lottery_num[2]
            origin_data['fourth'] = lottery_num[3]
            origin_data['fifth'] = lottery_num[4]
            origin_data['sixth'] = lottery_num[5]
            origin_data['bonus'] = lottery_num[6]
            origin_data['win'] = 0
            new_data = new_data.append(origin_data, ignore_index=True)
            break


# data.sort_values(axis=0,by='round',ascending=False)
```

In [ ]:

```
print(new_data.head(10))
print(new_data.tail(10))
```

```
  round        date first second third fourth fifth sixth bonus win
0   989  2021.11.13    17     18    21     27    29    33    26   1
1   989  2021.11.13    45      3    28     38     8    23     9   0
2   988  2021.11.06     2     13    20     30    31    41    27   1
3   988  2021.11.06    30     25     4     40    13    10     9   0
```

```
4    987   2021.10.30      2       4     15      23     29     38      7    1
5    987   2021.10.30     15      22     32       8     41     45     25    0
6    986   2021.10.23      7      10     16      28     41     42     40    1
7    986   2021.10.23      2      21     44       4      7     18      1    0
8    985   2021.10.16     17      21     23      30     34     44     19    1
9    985   2021.10.16     43      27     17      32     31     13     22    0
      round        date first second third fourth fifth sixth bonus win
1968      5  2003.01.04    16      24    29      40     41     42      3    1
1969      5  2003.01.04     4      19    35       3     12     16     36    0
1970      4  2002.12.28    14      27    30      31     40     42      2    1
1971      4  2002.12.28    13      38    45      26     10     19     22    0
1972      3  2002.12.21    11      16    19      21     27     31     30    1
1973      3  2002.12.21    17      30    22       4     11     21     42    0
1974      2  2002.12.14     9      13    21      25     32     42      2    1
1975      2  2002.12.14    22      36    41       6     19      8      2    0
1976      1  2002.12.07    10      23    29      33     37     40     16    1
1977      1  2002.12.07    33      35    17      10      4     22      5    0
```

Task-3

Feature engineering: Create a new feature and add it to the column list (to the dataset from Task-2)

In [ ]:

```
# 각 자리수를 비트로 생각하고, 나온 자리를 1 나오지 않은 자리를 0으로 표현한다음,
# 나올수있는 가장 큰 경우의 수인 45,44,43,42,41,40,39,38을
# 비트로 나타낸 수로 나눈 값을 새로운 Feature로 사용한다.

new_data_with_feature = pd.DataFrame(columns=[
                                      'round', 'date', 'first', 'second',
                                      'third', 'fourth', 'fifth', 'sixth',
                                      'bonus', 'win', 'feature'])
most_biggest_case = int('1'*7+'0'*38, 2)
for i in new_data.iloc:
    temp = dict(i)
    temp['feature'] = int(''.join(['1' if n in [i['first'], i['second'],
                                                i['third'], i['fourth'],
                                                i['fifth'], i['sixth'],
                                                i['bonus']]
                                   else '0' for n in
                                   range(1, 46)])[::-1], 2) / most_biggest_case
    new_data_with_feature = new_data_with_feature.append(
        temp, ignore_index=True)


# data.sort_values(axis=0,by='round',ascending=False)
```

In [ ]:

```
new_data_with_feature.head()
```

Out[ ]:

	round	date	first	second	third	fourth	fifth	sixth	bonus	win	feature
0	989	2021.11.13	17	18	21	27	29	33	26	1	0.000134
1	989	2021.11.13	45	3	28	38	8	23	9	0	0.507878
2	988	2021.11.06	2	13	20	30	31	41	27	1	0.031544
3	988	2021.11.06	30	25	4	40	13	10	9	0	0.015764
4	987	2021.10.30	2	4	15	23	29	38	7	1	0.003945

Task-4

Explain your plan how you use the data file from Task 2 or 3 to create the smart lottery prediction agent.

랜덤포레스트에 Date와 Round를 제외한 7개의 숫자와 feature를 기준으로 win이 0인지 1인지 학습시킨다.
train set과 test set은 기존 데이터를 랜덤으로 추출해 80%를 train set, 20%를 test set으로 사용한다.

In [ ]:
```
# 랜덤 포레스트로 학습시킨다

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [ ]:
```
# 7개의 숫자와 feature로 학습
data = new_data_with_feature[['first', 'second', 'third', 'fourth',
                              'fifth', 'sixth', 'bonus', 'feature']]
target = new_data_with_feature['win']
```

In [ ]:
```
# 데이터 셋 중 20%를 테스트셋으로 활용
x_train, x_test, y_train, y_test = train_test_split(data,
                                                    target,
                                                test_size=0.2,
                                                stratify=target)
```

In [ ]:
```
x_train
```

Out[ ]:

	first	second	third	fourth	fifth	sixth	bonus	feature
875	18	30	39	5	44	21	24	0.259858
621	27	26	38	33	43	34	19	0.130293
110	1	3	30	33	36	39	12	0.008997
747	12	42	3	36	20	31	38	0.067944
867	22	18	34	13	32	6	27	0.000310
...	...	...	...	...	...	...	...	...
366	14	20	23	31	37	38	27	0.005938
1952	22	23	25	37	38	42	26	0.068899
489	4	42	27	25	33	34	37	0.065332
271	32	27	37	5	39	3	10	0.009906
1773	25	10	8	22	36	34	26	0.001232

1582 rows × 8 columns

In [ ]:
```
y_train = y_train.astype('int')
y_test = y_test.astype('int')
```

In [ ]:
```
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
pred = rf.predict(x_test)
accuracy = accuracy_score(y_test,pred)
```

In [ ]:
```
print(f'정확도 : {accuracy*100}%')
```

정확도 : 99.24242424242425%

In [ ]:

```
# 로또 번호 20개 추출
count = 0
total_count = 0
buy_this = pd.DataFrame(
    columns=['first', 'second', 'third', 'fourth', 'fifth', 'sixth'])
while True:
    total_count += 1
    my_lottery = pd.DataFrame(columns=['first', 'second', 'third',
                                       'fourth', 'fifth', 'sixth', 'bonus'])
    random.shuffle(lottery_num)
    temp = int(''.join(['1' if n in lottery_num[:7]
                else '0' for n in range(1, 46)]), 2)
    temp_dict = {}
    if temp not in history:
        temp_dict['first'] = int(lottery_num[0])
        temp_dict['second'] = int(lottery_num[1])
        temp_dict['third'] = int(lottery_num[2])
        temp_dict['fourth'] = int(lottery_num[3])
        temp_dict['fifth'] = int(lottery_num[4])
        temp_dict['sixth'] = int(lottery_num[5])
        temp_dict['bonus'] = int(lottery_num[6])
        temp_dict['feature'] = int(''.join(['1' if n in [temp_dict['first'],
                                                         temp_dict['second'],
                                                         temp_dict['third'],
                                                         temp_dict['fourth'],
                                                         temp_dict['fifth'],
                                                         temp_dict['sixth'],
                                                         temp_dict['bonus']]
                                            else '0' for n in range(1, 46)])[::-1],
                                   2) / most_biggest_case
        my_lottery = my_lottery.append(temp_dict, ignore_index=True)

    if rf.predict(my_lottery)[0]:
        buy_this = buy_this.append(
            my_lottery[['first', 'second', 'third', 'fourth',
                        'fifth', 'sixth']], ignore_index=True)
        count += 1

    if count >= 20:
        break
```

In [ ]:

```
buy_this #1등 예상번호
```

Out[ ]:

	first	second	third	fourth	fifth	sixth
0	1.0	2.0	7.0	3.0	6.0	39.0
1	4.0	17.0	15.0	24.0	41.0	35.0
2	7.0	1.0	23.0	21.0	32.0	30.0
3	6.0	20.0	3.0	22.0	35.0	42.0
4	7.0	1.0	5.0	25.0	33.0	44.0
5	2.0	19.0	22.0	12.0	31.0	32.0
6	10.0	19.0	6.0	36.0	39.0	33.0
7	11.0	13.0	2.0	23.0	19.0	45.0
8	6.0	4.0	7.0	37.0	41.0	42.0
9	5.0	22.0	36.0	42.0	43.0	41.0
10	2.0	13.0	34.0	35.0	33.0	36.0
11	4.0	1.0	34.0	16.0	40.0	33.0

	first	second	third	fourth	fifth	sixth
12	2.0	21.0	13.0	29.0	34.0	42.0
13	5.0	26.0	6.0	43.0	34.0	45.0
14	3.0	17.0	28.0	24.0	32.0	39.0
15	2.0	20.0	1.0	14.0	29.0	35.0
16	3.0	11.0	27.0	24.0	45.0	37.0
17	16.0	18.0	30.0	20.0	32.0	39.0
18	4.0	13.0	31.0	23.0	41.0	42.0
19	11.0	9.0	12.0	36.0	27.0	33.0
20	1.0	7.0	24.0	16.0	25.0	29.0

Task-5: Write one paragraph explaining your tasks and any difficulties you had.

내가 만든 알고리즘을 통해 생성한 로또번호 40개를 추출하고, 나온대로 4만원 어치의 복권을 샀다.
로또 번호는 완전 독립 시행이라 예측이 불가능할 것이라고 생각했지만, 정확도가 97이상길래, 나도 모르게 기대를 했다.
로또 1등에 당첨될 것이라는 희망에 부풀었지만, 결과는 5000원짜리도 당첨되지 않았다.

분명 정확도는 97프로 이상이였는데, 왜 하나도 맞지 않았나 생각을 해봤는다.

20개의 로또 번호를 생성하기 위해서 약 1200개의 번호를 생성하는데, 그렇다는 말은 로또 당첨되지 않는 번호를 생성확률은 약 98%이다.
따라서 예측 결과가 실패라고하면 높은 확률로 예측에 성공했다고 처리하기 때문이지 않을까? 라고 생각한다.