스피드 데이팅의 애프터 확률을 높이는 지표에 대한 분석

201411073 신준녕

201411122 이혁진

Purpose

기존의 스피드 데이팅 서비스(Speed dating service) 제공 어플리케이션은 서로에 대한 사전 정보가 없이 즉석에서 채팅(또는 만남)이 이루어지도록 되어 있다. 이는 다양한 사람들을 만날 수 있다는 장점이 있으나 상대에 대한 사전 정보가 부족하다는 단점이 있다. 예컨대 나는 진지한 관계를 원하는데 상대는 장난삼아 미팅에 임하는 것인지 알 수 없다는 것이다. 이러한 단점은 실제 미팅이 성사되어 커플로 발전할 여부를 불확실하게 만든다. 그렇기 때문에 자신과 마음이 맞는 사람을 만나기 위해서는 운이 좋아야 하거나 오랜 시간을 필요로 한다. 우리는 이러한 단점을 개선하기 위해 나의 정보를 입력하면 이를 바탕으로 나와 커플로 발전할 확률이 높은 이성의 후보군을 추려내어 그들과의 스피드 데이팅을 주선하는 알고리즘을 만들어보고자 한다.

Dataset

사용한 데이터셋의 출처는 Kaggle[1]이다. 2002년 10월 16일부터 2004년 04월 07일까지 21차례에 걸쳐 기록된 데이터이며 총 552명의 개인 정보와 스피드 데이팅 매칭 결과를 담고 있다.

Methods

먼저, 데이터셋의 'match' column을 통해 전체 그룹을 애프터 신청을 받은 그룹(match 값이 1인 데이터)과 그렇지 못한 그룹(match 값이 0인 데이터)으로 나눈다. 각 그룹에 속한 사람들의 개인 정보 column들 중 8가지를 추려 그 값을 비교한다.

개인 정보로 사용할 Column들은 다음과 같다.

1. age: 나이

   전체 데이터셋을 보면 연령대의 범위가 10대 후반에서 50대 중반으로 매우 다양하다. (정확히는 18세부터 55세까지 확인함) 분석은 나이차가 나는 범위를 4가지 경우로 나누어 각각의 분포를 보는 것으로 한다.

나이가 같음
(성별에 관계없이) 나이차가 1-2살
(성별에 관계없이) 나이차가 3-4살
(성별에 관계없이) 나이차가 5살 이상

[1] https://www.kaggle.com/annavictoria/speed-dating-experiment/data

<Figure 01. age 분류>

2. income: 수입 (US 달러 기준)

   애프터 신청을 받은 그룹과 그렇지 못한 그룹 각각의 경우에 남녀 수입의 차이를 평균 내어 비교한다.

3. mn_sat: SAT 점수

   애프터 신청을 받은 그룹과 그렇지 못한 그룹 각각의 경우에 SAT 점수의 차이를 평균 내어 비교한다.

4. field cd: 전공 분야의 코드를 18가지로 나누어 놓은 지표

코드 번호	전공 분야
1	Law
2	Math
3	Social Science, Psychologist
4	Medical Science, Pharmaceuticals, Bio-Tech
5	Engineering
6	English Creative Writing Journalism
7	History Religion, Philosophy
8	Business, Economy, Finance
9	Education, Academia
10	Biological Sciences, Chemistry, Physics
11	Social Work
12	Undergrad, Undecided
13	Political Science, International Affairs
14	Film
15	Fine Arts, Arts Administration
16	Languages
17	Architectures
18	Other

<Figure 02. Field cd 분류표>

18가지의 전공 분야 코드는 RIASEC 코드[2] 분류를 따라 구분한다. 분류 결과는 다음과 같다.

[2] RIASEC 코드: 존 홀랜드(1959)의 직업적 성격유형을 알아보는 검사이며, 우리나라의 진로교육, 진로지도, 진로상담 현장에서 가장 빈번하게 사용되는 심리검사이다. (김희정, 2007; 이제경, 2009)

Realistic (현장형)	5, 17
Investigative (탐구형)	2, 4, 7, 10
Artistic (예술형)	6, 14, 15
Social (사회형)	3, 9, 11, 16
Enterprising (진취형)	1, 8, 13
Conventional (사무형)	-
Other (분류 외)	12, 18

<Figure 03. RIASEC 코드로 재분류한 field cd>

애프터 신청을 받은 그룹과 그렇지 못한 그룹 각각의 경우에 대해 RIASEC 코드의 일치 여부를 조사하여 전체적인 경향성을 알아본다.

5. goal: 스피드 데이팅에 참가하는 이유에 대한 지표

   애프터 신청을 받은 그룹과 그렇지 못한 그룹 각각의 경우에 대해 각 코드 분류가 일치하는지 비교해본다.

코드 번호	이유
1	그냥 재미있어 보여서
2	새로운 사람들을 만나기 위해서
3	이성과 데이트를 하기 위해서
4	진지한 관계(결혼 등)를 맺기 위해서
5	내가 참가했다는 것을 다른 사람들에게 자랑하기 위해서
6	기타

<Figure 04. goal 분류표>

6. date: 데이트를 하러 나가는 빈도에 대한 지표

   애프터 신청을 받은 그룹과 그렇지 못한 그룹 각각의 경우에 대해 각 코드 분류가 일치하는지 비교해본다.

코드 번호	빈도
1	일주일에 서너 번(혹은 그 이상)
2	일주일에 두 번
3	일주일에 한 번
4	한 달에 두 번
5	한 달에 한 번
6	1년 중 서너 번
7	거의 드물게 만남(혹은 외출함)

<Figure 05. date 분류표>

7. go out: (굳이 데이트를 위한 것이 아니더라도) 외출하는 빈도에 대한 지표

이 지표는 Figure 05. 의 분류를 그대로 따른다. 분석 방법 또한 동일하다.

8. 흥미 있는 활동에 대해 1-10점의 점수를 매기는 지표(점수가 높을수록 선호)

지표 이름	activity point
sports (스포츠)	10
tvsports (스포츠 경기 관람)	1
exercise (헬스)	10
dining (외식)	3
museums (박물관, 전시회 관람)	3
art (회화)	2
hiking (등산)	9
gaming (컴퓨터, 비디오 게임)	2
clubbing (춤, 클럽 가기)	8
reading (독서)	1
tv (TV 시청)	1
theater (극장, 연극 관람)	3
movies (집에서 영화 시청)	2
concerts (콘서트 관람)	3
music (음악 듣기)	2
shopping (쇼핑)	3
yoga (요가, 명상)	5

<Figure 06. 흥미 있는 활동 분류표>

각 활동 별로 임의로 activity point를 매긴다. 이 activity point는 외향적인 활동일수록 더 큰 점수를 매기도록 한다. 이를 개인이 각각의 지표에 매긴 점수에 곱한 다음 그 값을 모두 더해 activity score라는 새로운 지표를 만든다. 애프터 신청을 받은 그룹과 그렇지 못한 그룹의 경우에 대해 남녀의 activity score의 차이를 평균 내어 비교해본다.

Results

전체 데이터셋에서 애프터 신청을 받은 그룹(match 값이 1인 데이터)은 남녀 690쌍이며, 그렇지 못한 그룹(match 값이 0인 데이터)은 남녀 3,499쌍이다.

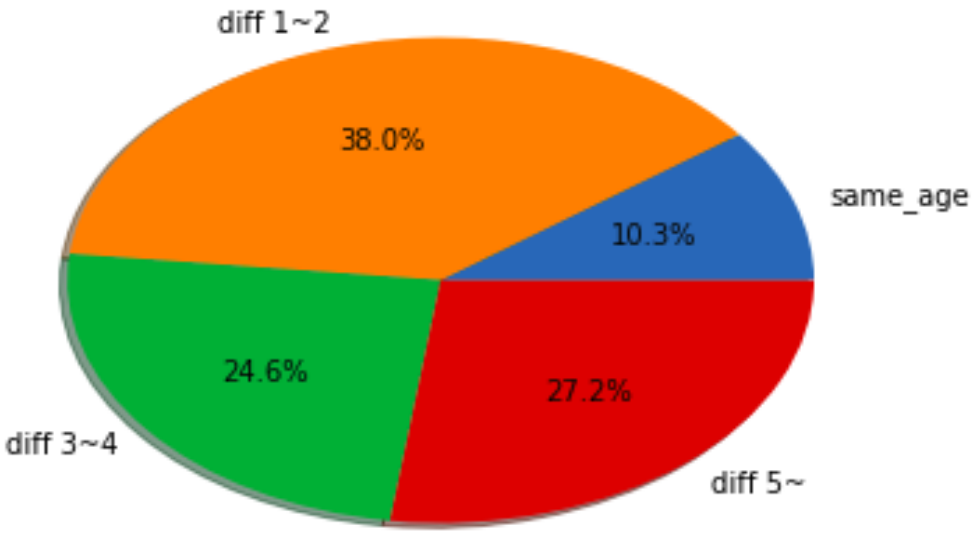
각 지표 별 분석 결과는 다음과 같다.

1. age: 나이

1-1. 애프터 신청을 받은 그룹

```
zero= 0
one = 0
two =0
three =0

for i in range(0,len(n_iid_list)):
    result = get_age_score(n_iid_list[i],n_match)
    zero = zero + result[0]
    one = one + result[1]
    two = two + result[2]
    three = three + result[3]
```

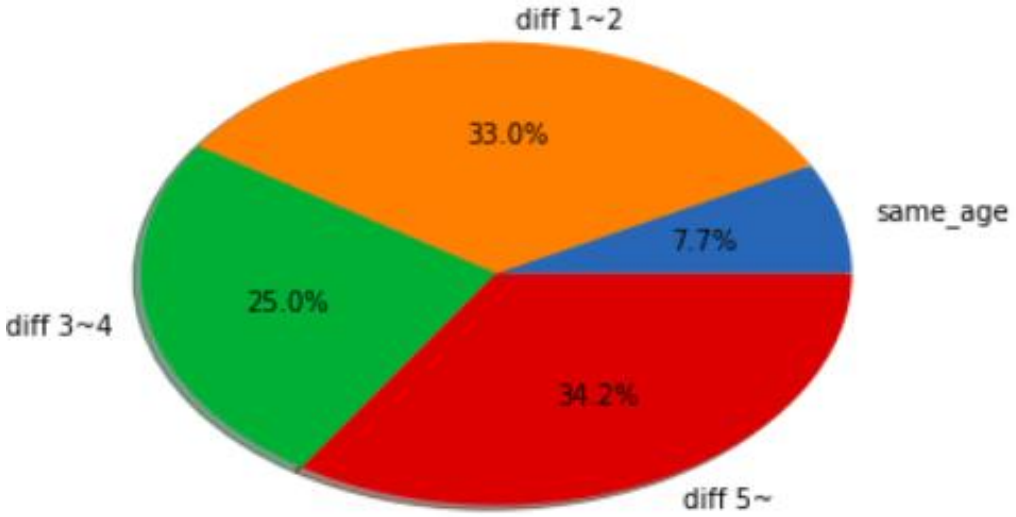
```
labels = ["same_age","diff 1~2","diff 3~4","diff 5~"]
plt.pie([zero,one,two,three], labels=labels, autopct='%1.1f%%', shadow=True)
```



<Figure 07. age 결과 – 애프터 신청을 받은 그룹의 경우>

애프터 신청을 받은 그룹의 경우 나이 차가 1-2살 나는 경우가 전체 690쌍 중 262쌍(38.0%)으로 제일 많았다. 나이 차가 3-4살 나는 경우는 170쌍(24.6%)이고 5살 이상 차이 나는 경우는 187쌍(27.2%)이다. 같은 나이인 경우는 71쌍(10.3%)으로 제일 낮았다.

1-2. 애프터 신청을 받지 못한 그룹



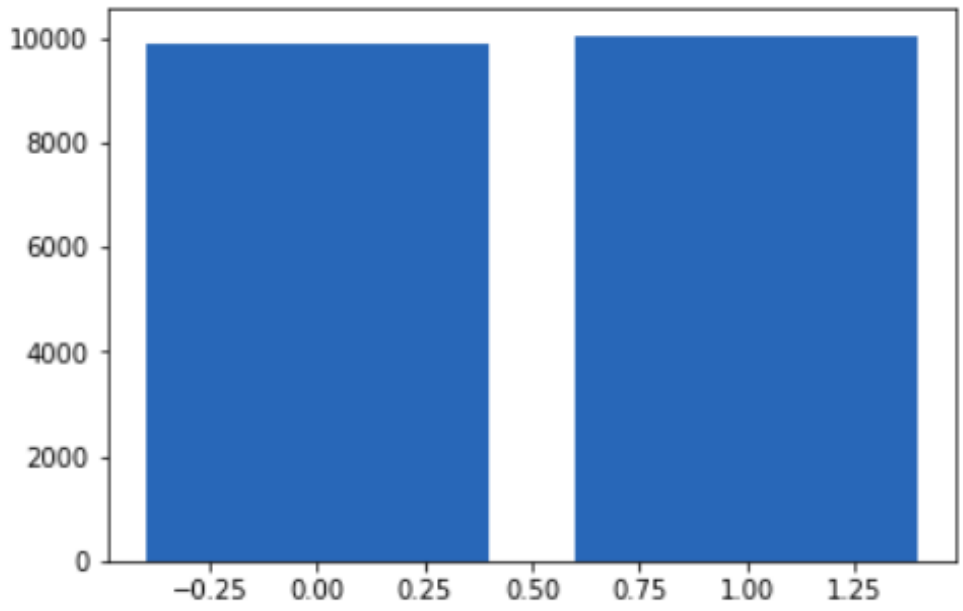
<Figure 08. age 결과 – 애프터 신청을 받지 못한 그룹의 경우>

애프터 신청을 받지 못한 그룹의 경우 나이 차가 1-2살 나는 경우는 전체 3,499쌍 중 1195쌍(33.0%)을 차지했다. 나이 차가 3-4살 나는 경우는 875쌍(25.0%)이고 5살 이상 차이나는 경우는 1197쌍(34.2%)이다. 같은 나이인 경우는 270쌍(7.7%)으로 제일 낮았다.

<Figure 07>과 <Figure 08>을 비교해보면 남녀의 나이 차가 없는 경우나 3~4살 차이 나는 경우가 전체 그룹에서 차지하는 비율은 거의 같다. 나이 차가 1~2살인 경우 애프터 신청을 받은 그룹의 경우 전체 인원의 38.0%를 차지하였고 애프터 신청을 받지 못한 그룹의 경우 전체 인원의 33.0%를 차지하였다. 이는 나이 차가 1~2살 나는 경우, 애프터 신청을 받을 확률이 증가한다고 해석할 수 있으나 그 차이가 크지 않아 영향력은 미미할 것으로 보인다. 반면, 나이 차가 5살 이상인 경우, 애프터 신청을 받은 그룹의 경우 전체 인원의 27.2%를 차지하였고 애프터 신청을 받지 못한 그룹의 경우 전체 인원의 34.2%를 차지한 것으로 보아 남녀의 나이 차가 5살 이상이면 애프터 신청을 받지 못할 확률이 증가한다고 해석할 수 있다.

2. income: 수입 (US 달러 기준)

```
def cal_diff_income(iid,matrix):
    mean_diff_income = 0
    total_diff_income = 0
    me = matrix[matrix['iid']==iid]
    con_me = me.convert_objects(convert_numeric=True)
    my_income = con_me['income'].reset_index(drop=True)[0]
    parter_id_list = me['pid'].reset_index(drop=True)

    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        parter = matrix[matrix['iid']==parter_id]

        if parter.size == 0:
            next
        else:
            con_parter = parter.convert_objects(convert_numeric=True)
            parter_income = con_parter['income'].reset_index(drop=True)[0]

            diff_income = abs(int(my_income) - int(parter_income))
            total_diff_income = total_diff_income + diff_income
            mean_diff_income = total_diff_income/parter_id_list.size

    return mean_diff_income
```

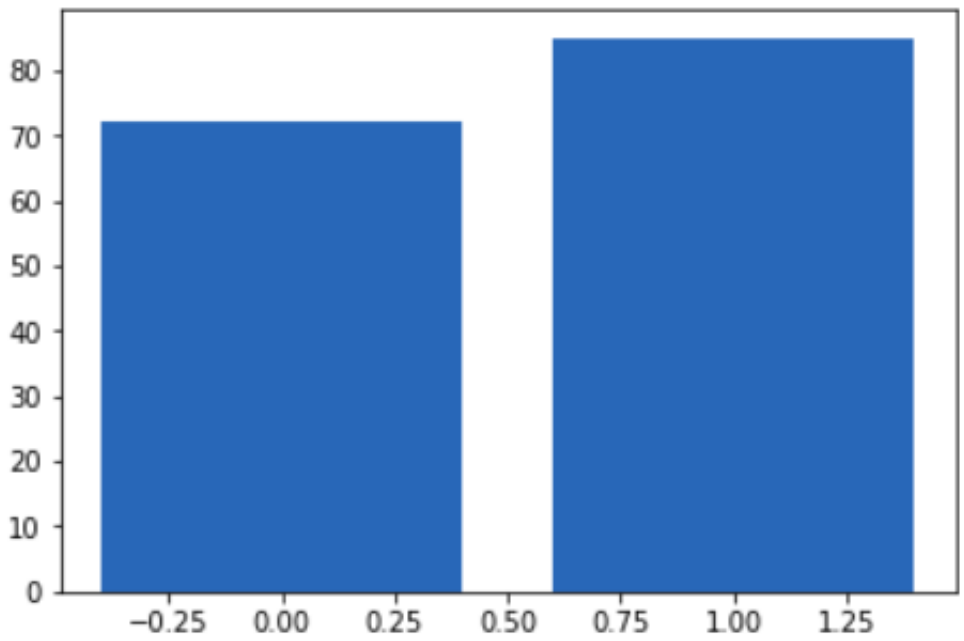


<Figure 09. income 결과 비교>

애프터 신청을 받은 그룹의 경우 남녀 한 쌍의 수입 차이는 9879.446 달러이고 애프터 신청을 받지 못한 그룹의 경우 남녀 한 쌍의 수입 차이는 10042.338 달러이다. 애프터 신청을 받은 그룹의 수입 차이가 162.892 달러만큼 더 낮았다.

3. mn_sat: SAT 점수

```
def cal_diff_sat(iid,matrix):
    mean_diff_sat = 0
    total_diff_sat = 0
    me = matrix[matrix['iid']==iid]
    con_me = me.convert_objects(convert_numeric=True)
    my_sat = con_me['mn_sat'].reset_index(drop=True)[0]
    parter_id_list = me['pid'].reset_index(drop=True)

    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        parter = matrix[matrix['iid']==parter_id]

        if parter.size == 0:
            next
        else:
            con_parter = parter.convert_objects(convert_numeric=True)
            parter_sat = con_parter['mn_sat'].reset_index(drop=True)[0]

            diff_sat = abs(int(my_sat) - int(parter_sat))
            total_diff_sat = total_diff_sat + diff_sat
            mean_diff_sat = total_diff_sat/parter_id_list.size

    return mean_diff_sat
```



<Figure 10. mn_sat 결과 비교>

애프터 신청을 받은 그룹의 경우 남녀 한 쌍의 SAT 점수 차이는 72.213 점이고 애프터 신청을 받지 못한 그룹의 경우 남녀 한 쌍의 SAT 점수 차이는 85.087 점이다. 애프터 신청을 받은 그룹의 점수 차이가 12.874 점만큼 더 낮다.

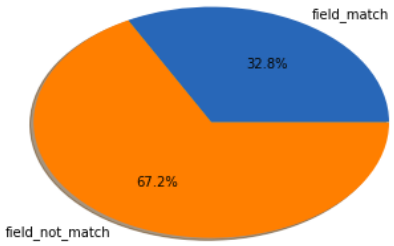
애프터 신청을 받은 그룹의 남녀 점수 차의 평균이 애프터 신청을 받지 못한 그룹의 경우보다 더 낮으므로 남녀의 학력 수준이 비슷할 때 애프터 신청을 받을 확률이 조금 더 높다고 볼 수 있다.

4. field cd: 전공 분야의 코드를 18가지로 나누어 놓은 지표

4-1. 애프터 신청을 받은 그룹

```
def check_partner_field(iid,matrix):

    same = 0
    n_same = 0
    me = matrix[matrix['iid']==iid]
    my_field_cd = me['field_cd'].reset_index(drop=True)[0]
    my_convert_code = field_cd(my_field_cd)
    parter_id_list = me['pid'].reset_index(drop=True)

    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        parter = matrix[matrix['iid']==parter_id]
        parter_field_cd = parter['field_cd'].reset_index(drop=True)[0]
        parter_convert_code = field_cd(parter_field_cd)

        if my_convert_code == parter_convert_code:
            same = same+1
        else:
            n_same = n_same+1

    return [same,n_same]
```

```
In [120]: plt.pie([field_match,field_not], labels=labels, autopct='%1.1f%%', shadow=True)
Out[120]: ([<matplotlib.patches.Wedge at 0x7f7c08dcba90>,
            <matplotlib.patches.Wedge at 0x7f7c08ddb390>],
           [<matplotlib.text.Text at 0x7f7c08dd3860>,
            <matplotlib.text.Text at 0x7f7c08de2160>],
           [<matplotlib.text.Text at 0x7f7c08dd3dd8>,
            <matplotlib.text.Text at 0x7f7c08de26d8>])
```

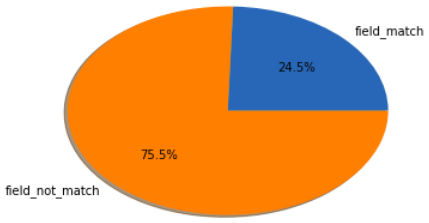


<Figure 11. field cd 결과 – 애프터 신청을 받은 그룹의 경우>

애프터 신청을 받은 그룹의 경우, RIASEC 코드에 따른 전공의 분류가 일치하는 쌍은 전체 690쌍 중 226쌍(32.8%)으로 일치하지 않는 464쌍(67.2%)보다 적었다.

4-2. 애프터 신청을 받지 못한 그룹

```
In [306]: n_iid_list = list(set(n_match['iid']))

In [315]: n_field_match =0
          n_field_not =0
          for i in range(0,len(n_iid_list)):
              result = check_partner_field(n_iid_list[i],n_match)
              n_field_match = n_field_match + result[0]
              n_field_not = n_field_not + result[1]

In [318]: n_field_match

Out[318]: 1710

In [319]: n_field_not

Out[319]: 5278

In [320]: labels = ["field_match","field_not_match"]
          plt.pie([n_field_match,n_field_not], labels=labels, autopct='%1.1f%%', shadow=True)

Out[320]: ([<matplotlib.patches.Wedge at 0x7f7c08a13128>,
            <matplotlib.patches.Wedge at 0x7f7c08a1c9e8>],
           [<matplotlib.text.Text at 0x7f7c08a13eb8>,
            <matplotlib.text.Text at 0x7f7c08a257b8>],
           [<matplotlib.text.Text at 0x7f7c08a1c470>,
            <matplotlib.text.Text at 0x7f7c08a25d30>])
```



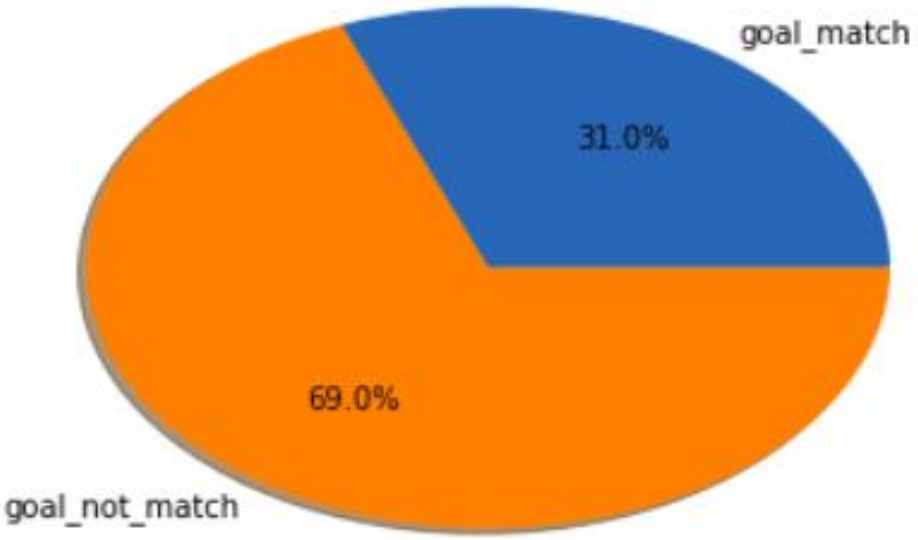
<Figure 12. field cd 결과 – 애프터 신청을 받지 못한 그룹의 경우>

애프터 신청을 받지 못한 그룹의 경우, RIASEC 코드에 따른 전공의 분류가 일치하는 쌍은 전체 3,499쌍 중 856쌍(24.5%)으로 일치하지 않는 2,638쌍(75.5%)보다 적었다. 애프터 신청을 받지 못한 그룹 3,499쌍 중 10명은 pid 값이 Nan으로 데이터 처리 과정에서 제외하였다.

<Figure 11>과 <Figure 12>를 비교해보면 애프터 신청을 받은 그룹에서 남녀의 전공 코드가 일치하는 경우는 전체의 32.8%이고 애프터 신청을 받지 못한 그룹에서는 전체의 24.5%이다. 결과적으로 남녀의 전공이 일치하는 경우(혹은 유사한 경우) 애프터 신청을 받을 확률이 더 높아진다고 해석할 수 있다.

5. goal: 스피드 데이팅에 참가하는 이유에 대한 지표

```
def check_goal(iid,matrix):
    same = 0
    n_same = 0
    me = matrix[matrix['iid']== iid]
    my_goal = me['goal'].reset_index(drop=True)[0]
    parter_id_list = me['pid'].reset_index(drop=True)


    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        if parter_id == None:
            next
        parter = matrix[matrix['iid']==parter_id]
        if parter.size == 0:
            next
        else:
            parter_goal = parter['goal'].reset_index(drop=True)[0]

            if my_goal == parter_goal:
                same = same+1
            else:
                n_same = n_same+1


    return [same,n_same]
```



<Figure 13. goal 결과>

애프터 신청을 받은 그룹의 경우 스피드 데이팅에 참가하는 목적이 같은 경우는 전체 690쌍 중 214쌍(31.0%)이다. 목적이 일치하지 않는 경우는 476쌍(69.0%)이다. 애프터 신

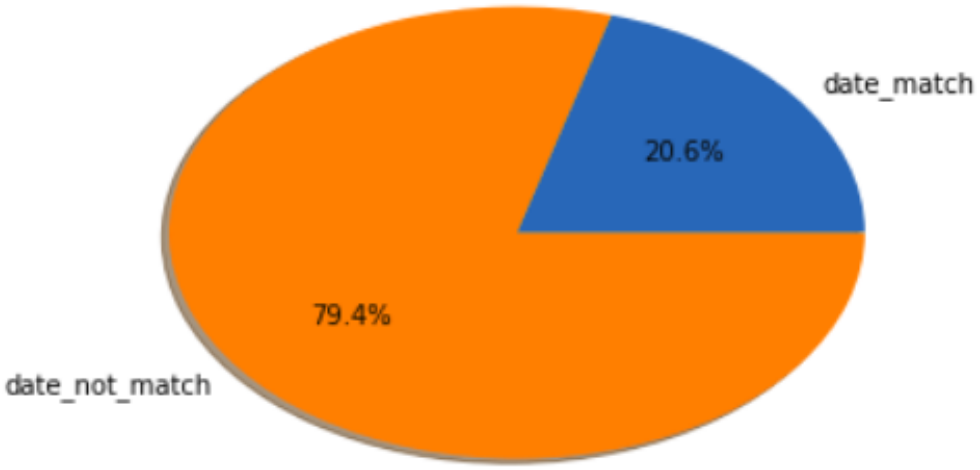
청을 받지 않은 그룹의 경우 또한 비율은 위와 동일하게 측정되었다. 이 경우 목적이 같은 경우는 전체 3,499쌍 중 1,085쌍(31.0%)이다. 목적이 일치하지 않는 경우는 2,414쌍(69.0%)이다.

애프터 신청을 받은 그룹과 그렇지 않은 경우 모두 스피드 데이팅에 참가하는 목적이 일치하는 빈도는 전체의 31.0%를 보여준다. 이는 스피드 데이팅에 참가하는 목적이 애프터 신청 여부에 영향을 주지 않는다는 것으로 해석될 수 있다.

6. date: 데이트를 하러 나가는 빈도에 대한 지표

6-1. 애프터 신청을 받은 그룹

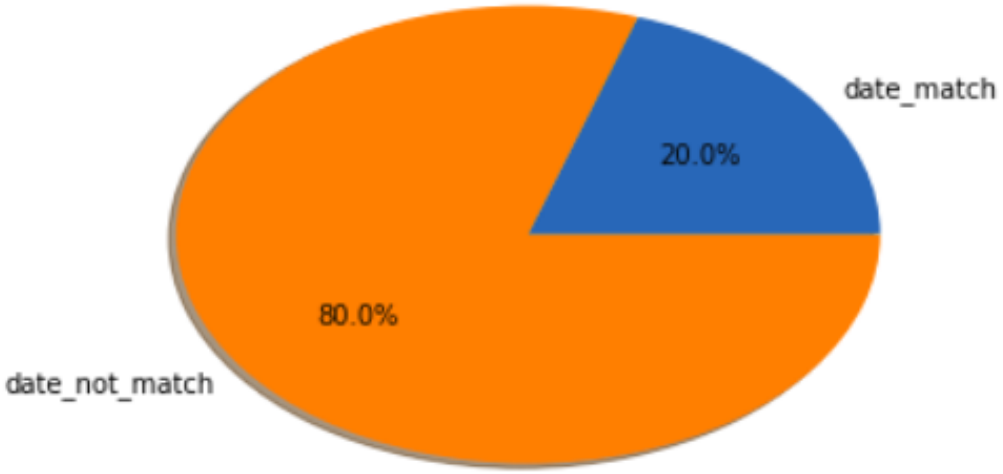
```
def check_date(iid,matrix):
    same = 0
    n_same = 0
    me = matrix[matrix['iid']== iid]
    my_date = me['date'].reset_index(drop=True)[0]
    parter_id_list = me['pid'].reset_index(drop=True)


    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        if parter_id == None:
            next
        parter = matrix[matrix['iid']==parter_id]
        if parter.size == 0:
            next
        else:
            parter_date = parter['date'].reset_index(drop=True)[0]

            if my_date == parter_date:
                same = same+1
            else:
                n_same = n_same+1


    return [same,n_same]
```



<Figure 14. date결과 – 애프터 신청을 받은 그룹의 경우>

애프터 신청을 받은 그룹의 경우 데이트를 하러 나가는 빈도가 같은 경우는 전체 690쌍 중 142쌍(20.6%)이다. 데이트를 하러 나가는 빈도가 다른 경우는 548쌍(79.4%)이다.

6-2. 애프터 신청을 받지 못한 그룹



<Figure 15. date결과 – 애프터 신청을 받지 못한 그룹의 경우>

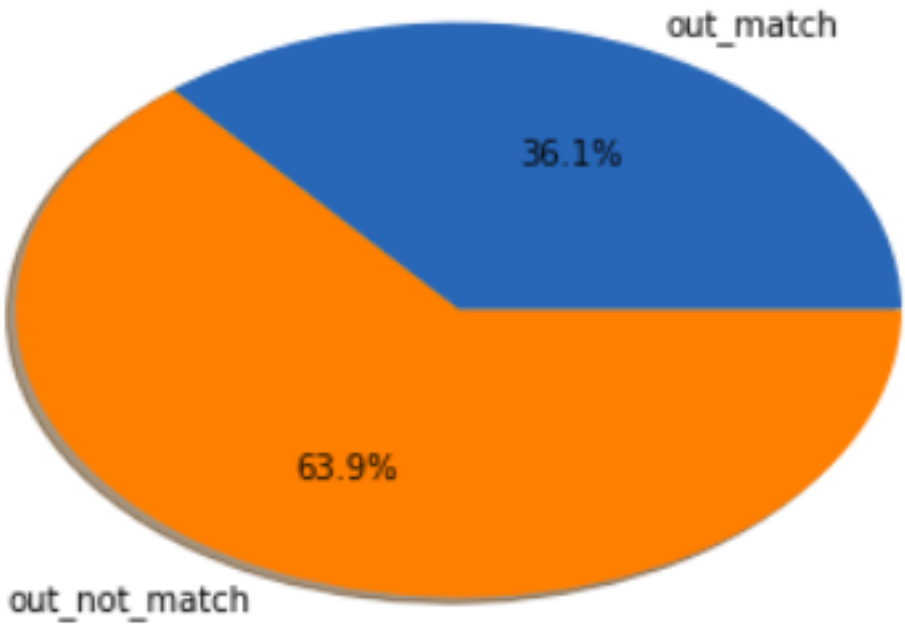
애프터 신청을 받지 못한 그룹의 경우 데이트를 하러 나가는 빈도가 같은 경우는 전체 3,499쌍 중 700쌍(20.0%)이다. 데이트를 하러 나가는 빈도가 다른 경우는 2,799쌍(80.0%)이다.

데이트를 하러 나가는 빈도의 경우, 애프터 신청을 받은 그룹에서는 빈도가 일치하는 경우가 전체의 20.6%를 차지한다. 마찬가지로 애프터 신청을 받지 못한 그룹에서의 경우도 전체의 20.0%를 차지한다. 이는 데이트를 하러 나가는 빈도는 애프터 신청 여부에 큰 영향을 주지 못하는 것으로 해석될 수 있다.

7. go out: (굳이 데이트를 위한 것이 아니더라도) 외출하는 빈도에 대한 지표

7-1. 애프터 신청을 받은 그룹

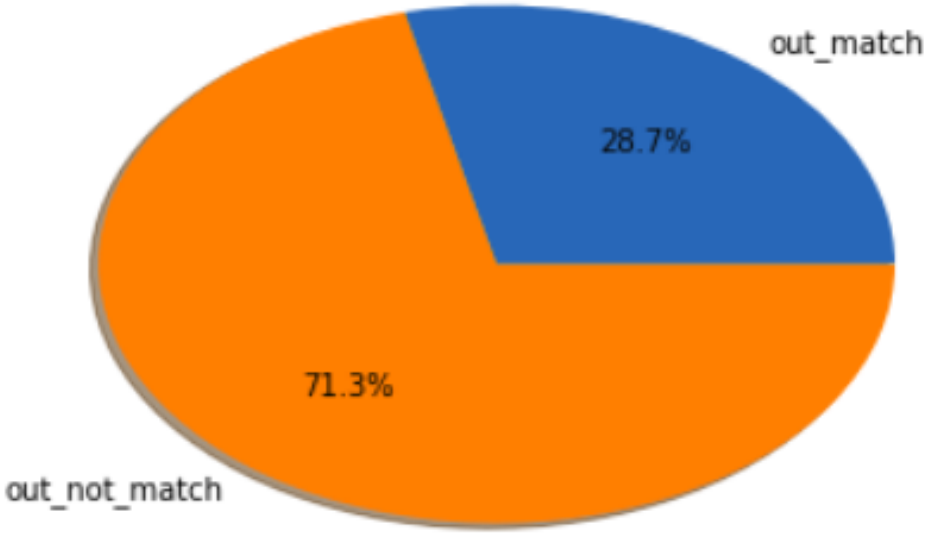
```
def check_out(iid,matrix):
    same = 0
    n_same = 0
    me = matrix[matrix['iid']== iid]
    my_date = me['go_out'].reset_index(drop=True)[0]
    parter_id_list = me['pid'].reset_index(drop=True)


    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        if parter_id == None:
            next
        parter = matrix[matrix['iid']==parter_id]
        if parter.size == 0:
            next
        else:
            parter_date = parter['go_out'].reset_index(drop=True)[0]

            if my_date == parter_date:
                same = same+1
            else:
                n_same = n_same+1


    return [same,n_same]
```



<Figure 16. go out결과 – 애프터 신청을 받은 그룹의 경우>

애프터 신청을 받은 그룹의 경우 외출하는 빈도가 같은 경우는 전체 690쌍 중 249쌍(36.1%)이다. 외출하는 빈도가 다른 경우는 441쌍(63.9%)이다.

7-2. 애프터 신청을 받지 못한 그룹



<Figure 17. go out결과 – 애프터 신청을 받지 못한 그룹의 경우>

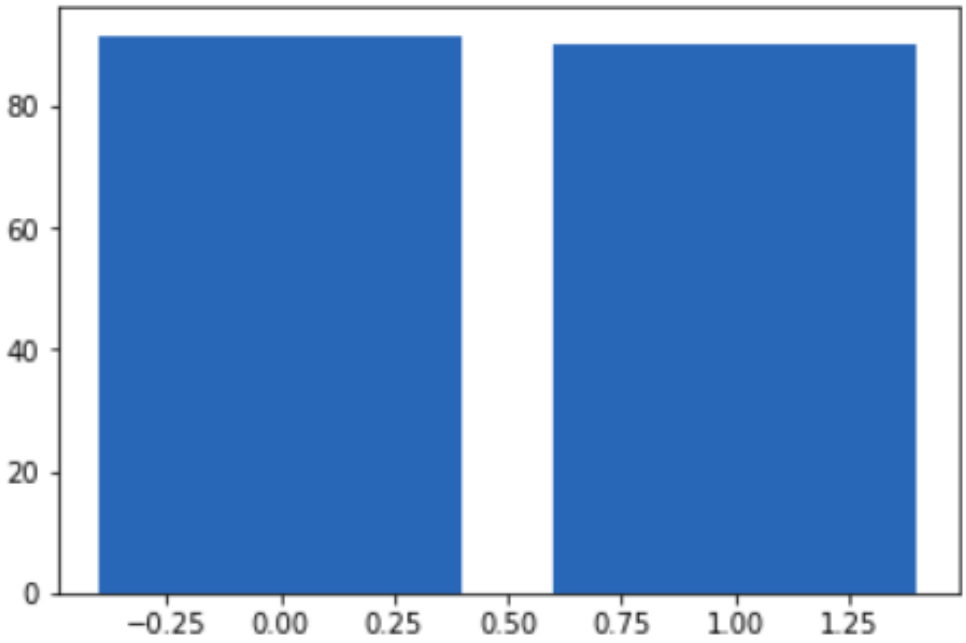
애프터 신청을 받은 그룹의 경우 외출하는 빈도가 같은 경우는 전체 3,499쌍 중 1,004쌍(28.7%)이다. 외출하는 빈도가 다른 경우는 2,495쌍(71.3%)이다.

애프터 신청을 받은 그룹 내에서 외출 빈도가 일치하는 경우는 전체의 36.1%를 차지하는데 비해 애프터 신청을 받지 못한 그룹 내에서 외출 빈도가 일치하는 경우는 전체의 28.7%를 차지한다. 즉, 외출하는 빈도가 같을수록 애프터 신청을 받을 확률이 더 높다고 해석할 수 있다.

8. 흥미 있는 활동에 대해 1-10점의 점수를 매기는 지표(점수가 높을수록 선호)

```
def cal_activity_score(iid,matrix):

    score= 0
    me = matrix[matrix['iid']==iid]
    con_me = me.convert_objects(convert_numeric=True)

    my_sports = con_me['sports'].reset_index(drop=True)[0]
    if ~isnotNaN(my_sports):
        my_sports = 0
    my_tvsports = con_me['tvsports'].reset_index(drop=True)[0]
    if ~isnotNaN(my_tvsports):
        my_tvsports = 0
    my_exercise = con_me['exercise'].reset_index(drop=True)[0]
    if ~isnotNaN(my_exercise):
        my_exercise = 0
    my_dining = con_me['dining'].reset_index(drop=True)[0]
    if ~isnotNaN(my_dining):
        my_dining = 0
    my_museums = con_me['museums'].reset_index(drop=True)[0]
    if ~isnotNaN(my_museums):
        my_museums = 0
    my_art = con_me['art'].reset_index(drop=True)[0]
    if ~isnotNaN(my_art):
        my_art = 0
    my_hiking = con_me['hiking'].reset_index(drop=True)[0]
    if ~isnotNaN(my_hiking ):
        my_hiking  = 0
    my_gaming = con_me['gaming'].reset_index(drop=True)[0]
    if ~isnotNaN(my_gaming):
        my_gaming  = 0
    my_clubbing = con_me['clubbing'].reset_index(drop=True)[0]
    if ~isnotNaN(my_clubbing):
        my_clubbing  = 0
    my_tv = con_me['tv'].reset_index(drop=True)[0]
    if ~isnotNaN(my_tv):
        my_tv  = 0
```

```
def cal_diff_activity_score(iid,matrix):
    mean_diff_as = 0
    total_diff_as = 0
    me = matrix[matrix['iid']==iid]
    my_as = cal_activity_score(iid,matrix)
    parter_id_list = me['pid'].reset_index(drop=True)

    for i in range(0,parter_id_list.size):
        parter_id = parter_id_list[i]
        parter = matrix[matrix['iid']==parter_id]

        if parter.size == 0:
            next
        else:
            parter_as = cal_activity_score(parter_id,matrix)
            diff_as = abs(my_as - parter_as)
            total_diff_as = total_diff_as + diff_as
            mean_diff_as = total_diff_as/parter_id_list.size

    return mean_diff_as
```



<Figure 18. 흥미 있는 활동 결과 비교>

애프터 신청을 받은 그룹의 경우 남녀 한 쌍의 activity score 차이의 평균값은 91.693 점이고 애프터 신청을 받지 못한 그룹의 경우 남녀 한 쌍의 activity score 차이의 평균값은 90.382 점이다. 애프터 신청을 받은 그룹의 activity score 차이의 평균값이 1.311 점만큼 더 높다.

흥미 있는 활동의 경우 애프터 신청을 받은 그룹과 받지 못한 그룹 모두 activity score 차이의 평균값에 큰 차이가 없기 때문에 이 지표는 애프터 신청 여부에 영향을 거의 주지 않는 것으로 해석할 수 있다.

Discussions

각 지표에 대한 결과를 정리해 보면 다음과 같다.

애프터 신청 여부에 유의미한 지표	애프터 신청 여부에 영향이 적은/무의미한 지표
나이, 전공, 외출 빈도	수입, SAT 점수, 좋아하는 활동, 데이트 빈도, 참가 목적

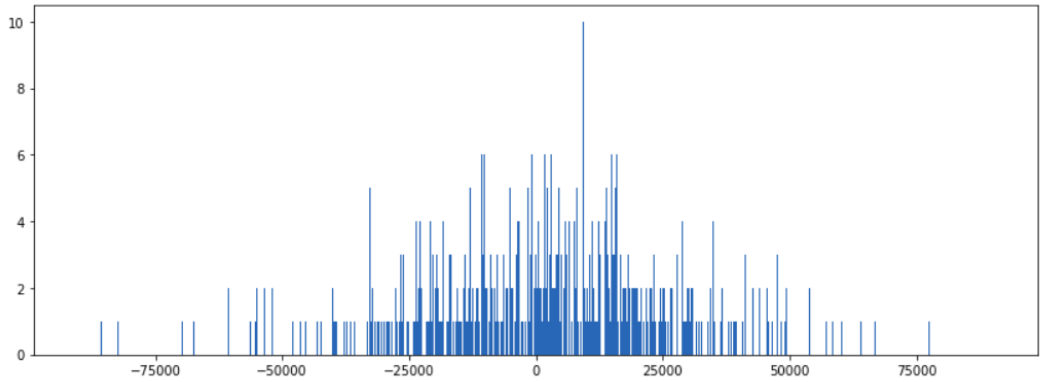
<Figure 19. 지표 별 의미성 여부>

지표를 8가지나 정했음에도 불구하고 유의미한 지표가 적다는 결과가 나온 것이 아쉬운 부분이다. 유의미한 지표의 경우에도 애프터를 받은 그룹과 그렇지 않은 그룹 간에 큰 차이를 보이는 경우는 없었는데 이는 애프터 신청 여부가 우리가 선정한 지표 8가지 외에 외모나 첫인상 같은 주관적인 부분에 대해 크게 의존하기 때문이라고 볼 수 있다.

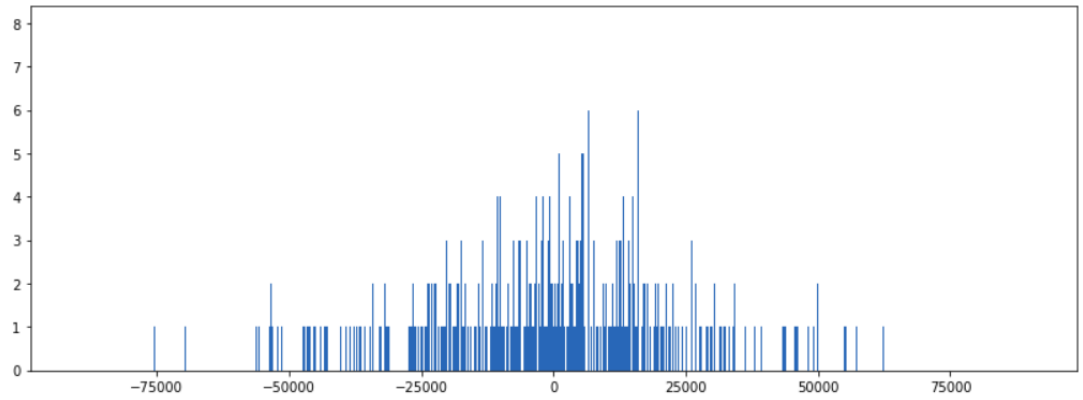
데이터 분석 중 의아했던 부분은 외출 빈도가 애프터 신청 여부에 있어 유의미한 지표임에도 불구하고 데이트 빈도는 애프터 신청 여부에 무의미한 지표라는 것이다. 이것은 아마 외출 빈도의 경우 그 사람의 외향성 혹은 내향성을 간접적으로 판단하는 기준이 되기 때문에 이러한 성향이 일치하거나 유사한 경우에 더 끌린다고 해석해 볼 수 있다. 데이트 빈도의 경우 남녀가 커플이 된 이후의 문제가 될 가능성이 높기 때문에 스피드 데이팅을 하는 시점에서는 큰 영향을 끼치는 지표가 아닐 수 있다.

또, 수입의 차이가 애프터 신청 여부에 큰 영향을 미치지 않는다는 것은 우리가 예상했던 결과와 달라 의외였다. 사회 통념적으로 수입에 대한 정보는 애프터 신청 여부에 상당히 중요한 영향을 가지므로 이러한 결과에 대해 지표에 대한 접근 방법이 잘못되었거나 놓친 부분이 있을 것으로 예상되어 추가적으로 조사를 진행해보았다.

그래서 우리는 첫째로 파트너간 수입 차를 도수분포그래프를 통해 분석해보았다.



<Figure 19. 매칭된 그룹의 수입차 분포>



< Figure 20. 매칭되지 않은 그룹의 수입차 분포>

	Match_Group	Not_Match_Group
평균값	1198.44	1143.22
표준편차	23802.18	23782.79

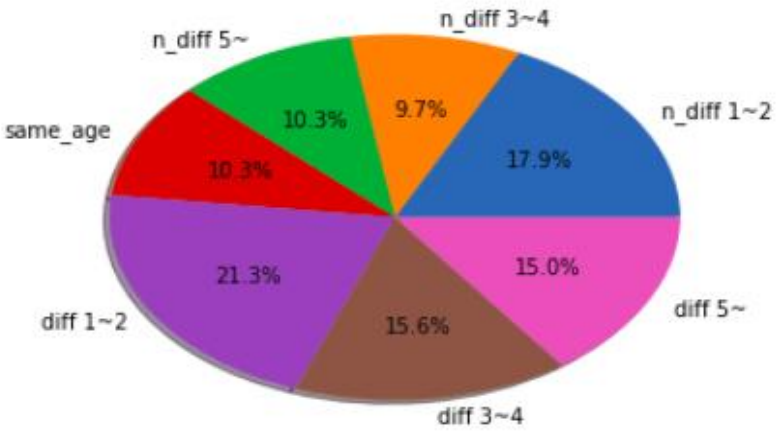
< Figure 21. 파트너간 수입 차 통계 값 결과>

분석결과, 매칭된 그룹의 수입차의 평균값이 +55.2달러(+면 남자 파트너의 수입이 더 큼) 차이가 더 난다.

따라서 파트너에 비해 남자 쪽 수입이 더 높을수록 매칭이 더 잘된다는 것을 알 수 있었다.

표준편차의 경우 매칭된 그룹의 분포가 더 고른 것을 알 수 있었다.

다음으로 파트너간 나이 차와 성별을 통해 나이차이의 분포를 살펴보았다.



< Figure 22. 매칭된 그룹의 파트너간 상대적 나이 차의 분포>

분석결과, 남자 파트너 나이가 여자 파트너 나이보다 1~2살이 많을수록 가장 선호도가 높았고 여자 파트너의 나이가 남자의 파트너 나이보다 3~4살 이상 많으면 선호도가 가장 낮았다.

우리는 다양한 방법을 통한 시도를 통해 더욱 유의미한 결과를 얻을 수 있었다. 따라서 차 후 다양한 지표의 조합을 시도해본다면 더욱 많은 경향성을 관찰 할 수 있을 것으로 보인다.