<Machine Learning Midterm Take-Home Exam>

영어영문학과 2016610010 김유민

Q1. 위의 그래프를 그리는데 사용된 단기적 사업 중요도 문항과 중장기적 사업 중요도 문항 18개 문항에 대한 응답을 사용하여 응답자를 분류하여 보시오.

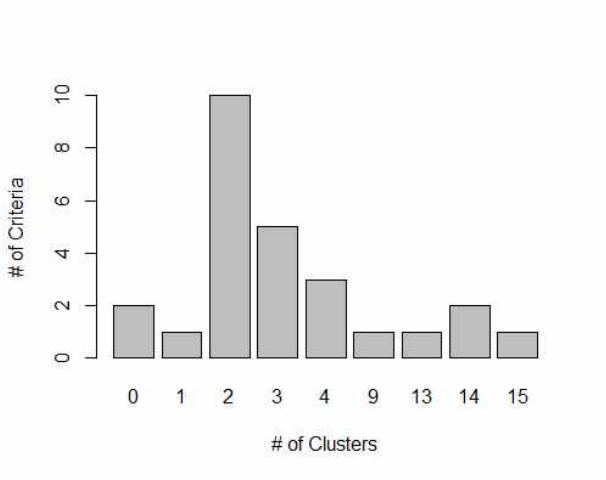
A.

Step1. 데이터를 불러온 후 a1~a9와 b1~b9 columns만을 포함하는 데이터셋을 추출하였다.

```
> ####Q1####
> df = read.csv("C:\\Users\\Kim Yuum\\Downloads\\NPO2019.csv",header = T)
> head(df)
  ID outsider a1 a2 a3 a4 a5 a6 a7 a8 a9 b1 b2 b3 b4 b5 b6 b7 b8 b9 recommend future_recommend lookfor future_lookfor
1  1        1  5  9  5  7  9  9  9  7  7  7  9  5  7  9  7  9  5  7         1               9       7              7
2  2        1 10 10  7  8  7  7  9 10  9 10 10  9 10  8  8 10  8  8         1               8       9              8
3  3        1 10 10 10 10 10  8  9 10  8  8 10 10 10 10 10 10 10 10         1              10      10             10
4  4        1  9 10  9  9  9  9 10 10  9  9  9  9  9  9  9 10  9  9         1              10      10             10
5  5        1 10 10 10  5  5  7  6 10  6 10 10 10  6  6  6  8 10  7         0              10       7             10
6  6        1 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10         1               8       5              5
  trust firstknow knowyear experience times
1     7      2014         5       10.0     3
2    10      2016         3        6.0     2
3    10      2016         3        2.0     1
4    10      2017         2        3.5     1
5     9      2016         3        3.5     2
6    10      2014         5       10.0     2
>
> df.ab = cbind(df[3:20])
> head(df.ab)
  a1 a2 a3 a4 a5 a6 a7 a8 a9 b1 b2 b3 b4 b5 b6 b7 b8 b9
1  5  9  5  7  9  9  9  7  7  7  9  5  7  9  7  9  5  7
2 10 10  7  8  7  7  9 10  9 10 10  9 10  8  8 10  8  8
3 10 10 10 10 10  8  9 10  8  8 10 10 10 10 10 10 10 10
4  9 10  9  9  9  9 10 10  9  9  9  9  9  9  9 10  9  9
5 10 10 10  5  5  7  6 10  6 10 10 10  6  6  6  8 10  7
6 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
```

Step2. NbClust 패키지를 통해 Step1에서 추출한 데이터를 가지고 적절한 군집의 수를 판단하였다.

```
> library(NbClust)
> nc = NbClust(df.ab, min.nc=2, max.nc=15, method="kmeans") #2개~15개 군집 #3개가 가장 많이 voting
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a
                significant increase of the value of the measure i.e the significant peak in Hubert
                index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significant peak in Dindex
                second differences plot) that corresponds to a significant increase of the value of
                the measure.

*******************************************************************
* Among all indices:
* 10 proposed 2 as the best number of clusters
* 5 proposed 3 as the best number of clusters
* 3 proposed 4 as the best number of clusters
* 1 proposed 9 as the best number of clusters
* 1 proposed 13 as the best number of clusters
* 2 proposed 14 as the best number of clusters
* 1 proposed 15 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2


*******************************************************************
> table(nc$Best.n[1,])  #그래프에서 어디서 꺾어지는지 눈대중으로 보는 것을 수치화

 0  1  2  3  4  9 13 14 15
 2  1 10  5  3  1  1  2  1
> par(mfrow=c(1,1))
> barplot(table(nc$Best.n[1,]), xlab="# of Clusters", ylab="# of Criteria") #table내용을 barplot으로
```

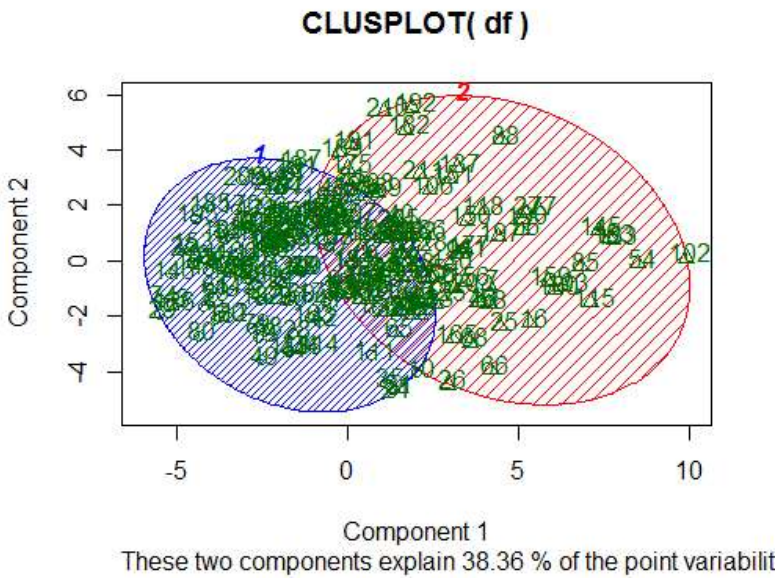



```
> table(nc$Best.n[1,])  #그라

 0  1  2  3  4  9 13 14 15
 2  1 10  5  3  1  1  2  1
```

위의 barplot과 table을 확인했을 때 k=2일 경우가 가장 적합하다고 voting되었으므로 군집의 수를 2개로 결정하였다.

Step3. Step1의 데이터셋으로 K-means clustering을 시행해 응답자를 2가지로 분류하고, 분류된 cluster를 새로운 column으로 추가하였다.

```
> df.km = kmeans(df.ab, 2) #kmeans
> df$cluster = as.factor(df.km$cluster)
> library(cluster)
> clusplot(df, df$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



Q2. 1)에서 응답자들을 응답 패턴에 따라 응답자들을 분류한 프로파일링을 사업 중요도 문항이 아닌 다른 설문 문항에 대한 응답에 적용하여, 어떤 특징을 가진 응답자들이 해당 분류에 속하는지 응답자들 분류에 따른 응답자들의 성격을 특정하여 보시오.

A.

/Step1. 결측치 처리/

column별로 결측치의 개수를 확인하였을 때 a1~a9, b1~b9, outsider ID를 제외한 나머지 변수에서 모두 결측치가 5개 이상 발견되었다. 이에 각 변수 특성에 따라 결측값을 대체하였다.

① recommend: Yes 또는 No로 구분되는 더미 변수이므로 평균값이나 최빈값을 사용할 경우 응답자의 특성이 정반대로 분류될 수 있으므로 로지스틱 회귀를 통한 추정치로 결측값을 대체하였다. 이 때 단계적 회귀를 통해 recommend에 유의한 영향을 미치는 변수만 회귀 변수로 적합 되도록 하였고, 문제가 될 수 있는 다중공산성 여부도 확인한 후 진행하였다.

```
> df.nona = na.omit(df)  #결측치 제거한 df
> # recommend 결측치: 로지스틱 회귀를 통해 추정값으로 대체 #
> df.ab.nona = cbind(df.nona[3:20],df.nona["recommend"]) #a b 설문과 recommend (결측치 x)
> df.ab.rec = cbind(df.ab, df["recommend"]) #a b 설문과 recommend(결측치 포함)
> logit1 = step(glm(recommend ~ ., family = binomial, data=df.ab.nona)) #AIC 값을 이용하여 단계적 회귀 수행
> rec.predict = predict(logit1, newdata=df.ab.rec[is.na(df.ab.rec$recommend),], type="response") #결측치였던 recommend 변수 예측
> pred = ifelse(rec.predict < 0.5, "no", "yes") #0.5이하는 NO로 예측
> df[is.na(df$recommend),]$recommend = factor(pred) #결측치 대체
```

```
Step:  AIC=155.17
recommend ~ a2 + a5 + a6 + b6

       Df Deviance    AIC
<none>      145.17 155.17
- a5    1   147.61 155.61
- a2    1   147.92 155.92
- a6    1   148.58 156.58
- b6    1   150.81 158.81
```

```
> vif(logit1) #10이상 값이 없으므로 다중공산성 존재 X
      a2       a5       a6       b6
1.060296 1.310626 2.966517 3.069299
```

② future_recommend: 향후에 추천할 가능성은 이미 지원센터를 다른 이에게 추천했는지의 여부인 'recommend'와 연관성이 높을 것이라고 판단해 상관계수를 살펴보았다. 이 때, 0.4784789의 값을 가지므로 생각보다 강한 상관성을 가지지는 않는다고 판단하여 평균값으로 대체하였다. (상관계수가 0.7~1.0일 경우에 높은 상관성을 가진다고 판단하였다.)

```
> df$future_recommend = ifelse(is.na(df$future_recommend),
mean(df$future_recommend, na.rm = T), df$future_recommend)
```

```
> cor(df$recommend, df$future_recommend,use="complete.obs")  #0.4784789
[1] 0.4775896
```

③ lookfor: 정보 우선성은 평균값으로 대체하였다.

```
> # lookfor 결측치: 평균값 대체
> df$lookfor = ifelse(is.na(df$lookfor), mean(df$lookfor, na.rm = T), df$lookfor)
```

④ future_lookfor: 향후 정보 우선성은 현재의 정보 우선성 정도와 연관성이 높을 것이라고 판단해 상관계수를 살펴보았다. 이 때, 0.8097908의 값을 가지므로 강한 상관성을 가짐을 알 수 있었고, 이에 a1~a9, b1~b9 문항과 lookfor, future_lookfor 변수만 가지는 데이터셋을 따로 생성해 future_lookfor에 대한 선형회귀분석을 진행하였다. 마찬가지로 단계적 회귀를 통해 future_lookfor에 유의한 영향을 미치는 변수만 회귀 변수로 적합 되도록 하였고, 당연히 lookfor 변수는 포함되었다. 또한 다중공산성 여부도 문제가 없음을 확인하였다. 이를 통해 추정된 값으로 결측치를 대체하였다.

```
> df.ab.lf = cbind(df.ab, df[c("future_lookfor","lookfor")]) #a b설문과 future_lookfor, lookfor(결측치 포함)
> df.ab.lf_nona = cbind(df.nona[3:20], df.nona[c("future_lookfor","lookfor")]) #a b설문과 future_lookfor, lookfor(결측치 x)
> logit2 = step(lm(future_lookfor ~ ., data=df.ab.lf_nona)) #단계적 회귀를 통해 변수선택
> pred = predict(logit2, df.ab.lf)
> df$future_lookfor = ifelse(is.na(df$future_lookfor), pred, df$lookfor) #회귀 추정치로 결측값 대체
```

```
Step:  AIC=97.18
future_lookfor ~ a1 + a5 + a7 + b6 + lookfor

           Df Sum of Sq    RSS     AIC
<none>                  305.31  97.178
- a7        1      3.18 308.49  97.238
- a5        1      4.25 309.56  97.926
- a1        1      4.28 309.59  97.948
- b6        1     20.34 325.65 108.014
- lookfor   1    462.69 768.00 278.748
```

⑤ trust: 신뢰도는 평균값으로 대체하였다.

```
> # trust 결측치: 평균값 대체
> df$trust = ifelse(is.na(df$trust), mean(df$trust, na.rm = T), df$trust)
```

⑥ knowyear(편의상 firstknow 이전에 배치하였다): 지원센터를 알고 지낸 시간은 평균값으로 대체하였다.

```
> # knowyear 결측치: 평균값 대체
> df$knowyear = ifelse(is.na(df$knowyear),mean(df$knowyear, na.rm = T),
df$knowyear)
```

⑦ firstknow: (2019 - knowyear)의 값으로 대체하였다.

```
> # firstknow 결측치: knowyear과의 계산을 통해 대체
> df$firstknow = ifelse(is.na(df$firstknow), 2019-df$knowyear , df$firstknow)
```

⑧ experience: NPO 영역 경력은 평균값으로 대체하였다.

```
> # experience 결측치: 평균값 대체
> df$experience = ifelse(is.na(df$experience), mean(df$experience, na.rm = T),
df$experience)
```

⑨ times: 설문 응답한 횟수 역시 평균값으로 대체하였다.

```
> # times 결측치: 평균값 대체
> df$times = ifelse(is.na(df$times), mean(df$times, na.rm = T), df$times)
```

/Step2. RandomForest를 통한 변수 중요도 파악/

① test set과 train set을 분리해 RandomForest의 정확도를 살펴보았다.

```
> ##(2) 랜덤포레스트를 통해 중요한 변수 파악
> library(randomForest)
> df2 = df[-c(1,3:20)]
> head(df2)

> i = sample(1:nrow(df2), round(nrow(df2)*0.7))
> df2.train = df2[i,]
> df2.test = df2[-i,]
> rfmodel = randomForest(cluster ~ ., data=df2.train, importance=TRUE, ntree=500,
mtry = 2, do.trace=100)

> pred.rf = predict(rfmodel, newdata=df2.test)
> tab=table(df2.test$cluster, pred.rf, dnn=c("Actual", "Predicted"))
> print(tab)

> error_rate = 1-sum(diag(tab)/sum(tab))
> error_rate #오분류 0.2
```

```
> i = sample(1:nrow(df2), round(nrow(df2)*0.7))
> df2.train = df2[i,]
> df2.test = df2[-i,]
> rfmodel = randomForest(cluster ~ ., data=df2.train, importance=TRUE, ntree=500, mtry = 2, do.trace=100)
ntree      OOB      1      2
  100:  25.00% 12.15% 58.54%
  200:  26.35% 11.21% 65.85%
  300:  24.32%  9.35% 63.41%
  400:  24.32%  9.35% 63.41%
  500:  25.00% 10.28% 63.41%
>
> pred.rf = predict(rfmodel, newdata=df2.test)
> tab=table(df2.test$cluster, pred.rf, dnn=c("Actual", "Predicted"))
> print(tab)
      Predicted
Actual  1  2
     1 43  0
     2 14  6
>
> error_rate = 1-sum(diag(tab)/sum(tab))
> error_rate #오분류 0.2
[1] 0.2222222
```

② 정확도가 0.8에 달해 어느 정도의 신뢰성을 가지므로 변수 중요도를 분석해보았고, 그 결과 future_recommend > trust> future_lookfor > lookfor > times 순으로 중요한 변수임을 알 수 있었다. 즉, 위의 순서가 cluster를 분류하는 데 중요한 지표라는 것이다.

```
> importance(rfmodel, type=1)
                 MeanDecreaseAccuracy
outsider                    0.8416066
recommend                   2.2738110
future_recommend           14.1053473
lookfor                     4.5884611
future_lookfor             10.8429622
trust                      12.7762501
firstknow                   1.6105668
knowyear                    1.8187412
experience                 -0.3288786
times                       2.7889468
```

/Step3. Decision Tree를 통한 변수 중요도 파악/

① pruning 마지노선 설정 후 tree size 결정

```
> set.seed(1234)
> fit.df2 = rpart(cluster ~., data=df2, method="anova")
> print(fit.df2)
> printcp(fit.df2) #가지치기를 어디서 멈출 것인지 확인
```

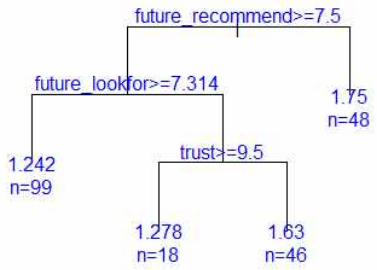
② 가지치기를 수행한 결과 출력

```
>fit.prun.df2=prune(fit.df2,
cp=fit.df2$cptable[which.min(fit.df2$cptable[,"xerror"]),"CP"]) #에러율이 가장 낮을
때의 tree size
> print(fit.prun.df2)
```

③ 완성된 Decision Tree

DT를 통해 파악한 변수의 중요도는 future_recommend > future_lookfor > lookfor > trust > recommend 순으로 높았다.

```
> plot(fit.prun.df2, uniform=T, compress=T, margin=0.1) #managable한 size로!
> text(fit.prun.df2, use.n = T, col ="blue")
> summary(fit.prun.df2)
```




```
Variable importance
future_recommend   future_lookfor          lookfor            trust        recommend
              35               21               20               16                8
```

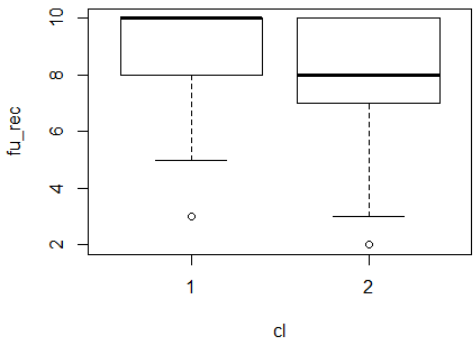
/Step4. 응답자들의 성격 특정/

Step2와 Step3의 결과로 볼 때 응답자들을 분류하는 데 중요한 변수는 공통적으로 future_recommend, future_lookfor, trust, lookfor이 있었다. 이에 대해 각 응답자들은 어떤 성격을 보였는지 확인한다. 각 문항의 점수 크기에 대한 기준은 Decision Tree의 변수별 결과를 반영하였다.

```
> ##(3)중요 변수를 통해 응답자들의 성격 특정하기
> df$fu_rec = ifelse(df$future_recommend > 8, 1, 0)
> table(df$fu_rec,df$cluster)

> df$fu_lf = ifelse(df$future_lookfor> 7,1,0)
> table(df$fu_lf,df$cluster)

> df$tru = ifelse(df$trust > 8,1,0)
> table(df$tru, df$cluster)

> df$lf = ifelse(df$lookfor >7, 1, 0)
> table(df$lf, df$cluster)
```

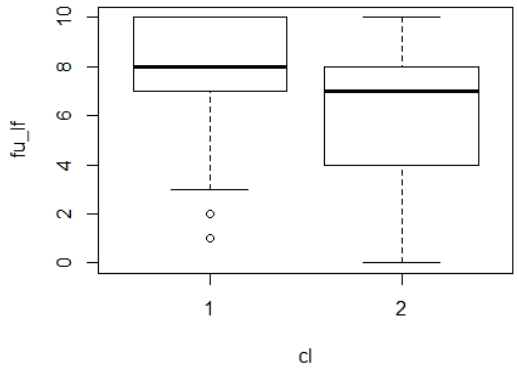
```
> ##(3)중요 변수를 통해 응답자들의 성격 특정하기
> df$fu_rec = ifelse(df$future_recommend > 8, 1, 0)
> table(df$fu_rec,df$cluster)

    1  2
  0 36 57
  1 81 37
> df$fu_lf = ifelse(df$future_lookfor> 7,1,0)
> table(df$fu_lf,df$cluster)

    1  2
  0 40 63
  1 77 31
> df$tru = ifelse(df$trust > 8,1,0)
> table(df$tru, df$cluster)

    1  2
  0 32 54
  1 85 40
> df$lf = ifelse(df$lookfor >7, 1, 0)
> table(df$lf, df$cluster)

    1  2
  0 40 61
  1 77 33
>
```

① future_recommend(향후에 지원센터를 추천할 가능성)



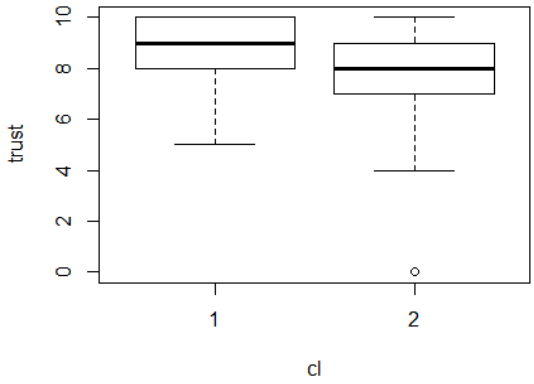
먼저, future_recommend(향후에 추천할 가능성)을 cluster에 따라 보면 cluster1은 cluster2에 비해 분포가 좁고 상위에 몰려있으므로 대부분 향후에 추천할 가능성에 높은 점수를 주었다.

② future_lookfor(향후 정보 우선성)



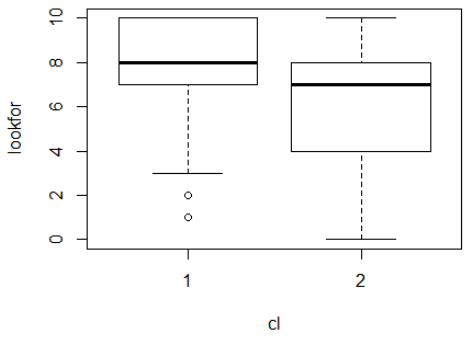
마찬가지로 cluster1이 cluster2에 비해 높은 점수를 주었고, future_recommend 점수보다 확연한 차이가 드러난다. cluster2는 점수 폭이 훨씬 높은 경향을 보였다.

③ trust(신뢰도)



trust에서는 극명한 차이를 보이지는 않지만 마찬가지로 cluster1이 cluster2에 비해 높은 점수를 주었다. median이 box의 거의 중앙에 위치한 것으로 보아 고르게 신뢰도 점수는 고르게 분포함을 알 수 있다.

④ lookfor(정보 우선성)



lookfor에 있어서는 cluster에 따라 비교적 극명한 차이를 보인다. cluster1이 cluster2에 비

해 높은 점수를 주었고, 특히 cluster2는 lookfor의 점수가 고르게 분포하는 것을 알 수 있다.

▶결론
응답자들을 분류한 기준을 결정하는 데에는 향후에 지원센터를 추천할 가능성, NPO 관련 정보가 필요할 때 센터 정보를 우선적으로 찾아보는 정도, 또는 향후 우선적으로 찾아볼 가능성, 그리고 지원센터에 대한 신뢰도가 중요한 역할을 하였다. cluster1의 응답자들은 이러한 기준 변수에 대해 모두 높은 점수(대부분 7점 또는 8점 이상)를 집중적으로 준 집단이었고, cluster2는 이에 비해 상대적으로 낮게 주거나 눈에 띄게 높은 점수를 주지는 않은 집단이었다.

Q3. 해당 결과를 가지고 인사이트를 도출하시오.

A.

먼저, 사업 중요도에 대한 평가를 통한 응답자들의 분류가 '향후에 추천할 가능성', '정보 우선성', '향후 정보 우선성', '지원센터에 대한 신뢰도' 문항에서도 유의미한 분류가 이루어졌다는 점에서 살펴볼 필요가 있다. 이는 곧 4가지 항목이 응답자들이 사업 중요도에 대해 가지는 생각과 연관이 있음을 뜻한다. cluster1의 경우 4개 문항에 대한 점수가 매우 높은 쪽에 집중되었다. 즉 cluster1에 속하는 응답자들은 향후에 지원센터를 타인에게 추천할 가능성이 높고, 정보가 필요할 때 주로 지원센터를 우선적으로 찾아보며, 지원센터에 대한 신뢰도가 높은 편에 속한다. 따라서 이들은 기본적으로 지원센터에 대한 관심도가 높고 긍정적인 반응을 보이는 응답자들임을 알 수 있다. 이러한 점에서 2017년과 2018년을 비교한 데이터에서 전반적인 중장기적 중요도가 단기적 중요도보다 높은 사업들의 중요도가 하락했다는 결과는 비교적 점수를 많이 주고 긍정적이었던 cluster1의 영향이 클 수 있으므로 해당 응답자들의 왜 그러한 결론에 도달했는지를 추가적으로 살펴보면 좋을 것이다.

한편 cluster1에 비해 4가지 항목에 대해 저조한 점수를 부여했던 cluster2는 지원센터에 대한 신뢰도와 긍정 반응이 상대적으로 낮은 집단이다. 해당 집단으로 분류된 응답자들은 지원센터가 특정 부분에서 부족하다고 느껴 이러한 점수를 주었을 것이므로, '역량 강화'와 관련된 설문에 어떤 응답을 나타냈는지 파악해 해당 역량을 집중적으로 강화하는 방향으로 나아가면 cluster2가 부정적으로 바라보았던 부분을 어느 정도 보완할 수 있을 것이다. 이 같은 추가 분석을 통해 2017년과 2018년을 비교한 데이터에서 감지된 역량 강화와 관련된 항목들의 변동도 해석할 수 있을 것이다.

다만 이 데이터는 '설문 결과'라는 점에서 설문 응답자 개개인마다 점수가 반영하는 중요도를 다른 크기로 여길 수 있으므로, cluster1은 그저 모든 설문에 높은 점수를 주는 성향을 가진 개인들일 수도 있다는 점을 간과해서는 안 된다. 예를 들어, cluster1의 한 응답자는 기본이 8점이고 그 이하일 때 6점 정도를 부여할 수 있지만 cluster2의 한 응답자는 기본이 5점이고 그 이상일 때 6점을 부여할 수도 있기 때문이다.